

# Coverability in VASS Revisited: Improving Rackoff's Bound to Obtain Conditional Optimality

Marvin Künnemann ✉

RPTU Kaiserslautern-Landau, Germany

Filip Mazowiecki ✉

University of Warsaw, Poland

Lia Schütze ✉ 

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

Henry Sinclair-Banks ✉  

Centre for Discrete Mathematics and its Applications (DIMAP) & Department of Computer Science, University of Warwick, Coventry, UK

Karol Węgrzycki ✉ 

Saarland University and Max Planck Institute for Informatics, Saarbrücken, Germany

---

## Abstract

Seminal results establish that the coverability problem for Vector Addition Systems with States (VASS) is in EXPSPACE (Rackoff, '78) and is EXPSPACE-hard already under unary encodings (Lipton, '76). More precisely, Rosier and Yen later utilise Rackoff's bounding technique to show that if coverability holds then there is a run of length at most  $n^{2^{\mathcal{O}(d \log d)}}$ , where  $d$  is the dimension and  $n$  is the size of the given unary VASS. Earlier, Lipton showed that there exist instances of coverability in  $d$ -dimensional unary VASS that are only witnessed by runs of length at least  $n^{2^{\Omega(d)}}$ . Our first result closes this gap. We improve the upper bound by removing the twice-exponentiated  $\log(d)$  factor, thus matching Lipton's lower bound. This closes the corresponding gap for the exact space required to decide coverability. This also yields a deterministic  $n^{2^{\mathcal{O}(d)}}$ -time algorithm for coverability. Our second result is a matching lower bound, that there does not exist a deterministic  $n^{2^{o(d)}}$ -time algorithm, conditioned upon the Exponential Time Hypothesis.

When analysing coverability, a standard proof technique is to consider VASS with bounded counters. Bounded VASS make for an interesting and popular model due to strong connections with timed automata. Withal, we study a natural setting where the counter bound is linear in the size of the VASS. Here the trivial exhaustive search algorithm runs in  $\mathcal{O}(n^{d+1})$ -time. We give evidence to this being near-optimal. We prove that in dimension one this trivial algorithm is conditionally optimal, by showing that  $n^{2-o(1)}$ -time is required under the  $k$ -cycle hypothesis. In general fixed dimension  $d$ , we show that  $n^{d-2-o(1)}$ -time is required under the 3-uniform hyperclique hypothesis.

**2012 ACM Subject Classification** Theory of computation → Models of computation

**Keywords and phrases** Vector Addition System, Coverability, Reachability, Fine-Grained Complexity, Exponential Time Hypothesis,  $k$ -Cycle Hypothesis, Hyperclique Hypothesis

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2023.131

**Category** Track B: Automata, Logic, Semantics, and Theory of Programming

**Related Version** *Full Version:* <https://arxiv.org/abs/2305.01581> [31]

**Funding** *Marvin Künnemann:* Research partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 462679611.

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

*Henry Sinclair-Banks:* Supported by EPSRC Standard Research Studentship (DTP), grant number EP/T5179X/1.

*Karol Węgrzycki:* Supported by the ERC grant TIPEA, agreement no. 850979.



© Marvin Künnemann, Filip Mazowiecki, Lia Schütze, Henry Sinclair-Banks, and Karol Węgrzycki; licensed under Creative Commons License CC-BY 4.0

50th International Colloquium on Automata, Languages, and Programming (ICALP 2023).

Editors: Kousha Etessami, Uriel Feige, and Gabriele Puppis; Article No. 131; pp. 131:1–131:20

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



**Acknowledgements** We would like to thank our anonymous reviewers for their comments, for their time, and especially for highlighting a piece of related work [30].

## 1 Introduction

Vector Addition Systems with States (VASS) are a popular model of concurrency with a number of applications in database theory [9], business processes [49], and more (see the survey [47]). A  $d$ -dimensional VASS ( $d$ -VASS) consists of a finite automaton equipped with  $d$  non-negative valued counters that can be updated by transitions. A configuration in a  $d$ -VASS consists of a state and a  $d$ -dimensional vector over the naturals. One of the central decision problems for VASS is the *coverability problem*, that asks whether there is a run from a given initial configuration to some configuration with at least the counter values of a given target configuration. Coverability finds application in the verification of safety conditions, which often equate to whether or not a particular state can be reached without any precise counter values [13, 24]. Roughly speaking, one can use VASS as a modest model for concurrent systems where the dimension corresponds with the number of locations a process can be in and each counter value corresponds with the number of processes in a particular location [21, 25].

In 1978, Rackoff [45] showed that coverability is in EXPSPACE, by proving that if coverability holds then there exists a run of double-exponential length. Following, Rosier and Yen [46] analysed and discussed Rackoff’s ideas in more detail and argued that if a coverability holds then it is witnessed by a run of length at most  $n^{2^{\mathcal{O}(d \log d)}}$ , where  $n$  is the size of the given unary encoded  $d$ -VASS. Furthermore, this yields a  $2^{\mathcal{O}(d \log d)} \cdot \log(n)$ -space algorithm for coverability. Prior to this in 1976, Lipton [37] proved that coverability is EXPSPACE-hard even when VASS is encoded in unary, by constructing an instance of coverability witnessed only by a run of double-exponential length  $n^{2^{\Omega(d)}}$ . Rosier and Yen [46] also presented a proof that generalises Lipton’s constructions to show that  $2^{\Omega(d)} \cdot \log(n)$ -space is required for coverability. Although this problem is EXPSPACE-complete in terms of classical complexity, a gap was left open for the exact space needed for coverability [46, Section 1]. By using an approach akin to Rackoff’s argument, we close this thirty-eight-year-old gap by improving the upper bound to match Lipton’s lower bound.

**Result 1:** If coverability holds then there exists a run of length at most  $n^{2^{\mathcal{O}(d)}}$  (Theorem 3.3). Accordingly, we obtain an optimal  $2^{\mathcal{O}(d)} \cdot \log(n)$ -space algorithm that decides coverability (Corollary 3.4).

Our bound also implies the existence of a deterministic  $n^{2^{\mathcal{O}(d)}}$ -time algorithm for coverability. We complement this with a matching lower bound on the deterministic running time that is conditioned upon the Exponential Time Hypothesis (ETH).

**Result 2:** Under ETH, there is no deterministic  $n^{2^{\mathcal{O}(d)}}$ -time algorithm deciding coverability in unary  $d$ -VASS (Theorem 4.2).

While our results establish a fast-increasing, conditionally optimal exponent of  $2^{\Theta(d)}$  in the time complexity of the coverability problem, they rely on careful constructions that enforce the observation of large counter values. In certain settings, however, it is natural to instead consider a restricted version of coverability, where all counter values remain *bounded*. This yields one of the simplest models, fixed-dimension bounded unary VASS, for which we obtain even tighter results. Decision problems for  $B$ -bounded VASS, where  $B$  forms part of the input, have been studied due to their strong connections to timed automata [27, 22, 41].

We consider linearly-bounded unary VASS, that is when the maximum counter value is bounded above by a constant multiple of the size of the VASS. Interestingly, coverability and reachability are equivalent in linearly-bounded unary VASS. The trivial algorithm that employs depth-first search on the space of configurations runs in  $\mathcal{O}(n^{d+1})$ -time for both coverability and reachability. We provide evidence that the trivial algorithm is optimal.

**Result 3:** Reachability in linearly-bounded unary 1-VASS requires  $n^{2-o(1)}$ -time, subject to the  $k$ -cycle hypothesis (Theorem 5.4).

This effectively demonstrates that the trivial algorithm is optimal in the one-dimensional case. For the case of large dimensions, we show that the trivial algorithm only differs from an optimal deterministic-time algorithm by at most an  $n^{3+o(1)}$ -time factor.

**Result 4:** Reachability in linearly-bounded unary  $d$ -VASS requires  $n^{d-2-o(1)}$ -time, subject to the 3-uniform  $k$ -hyperclique hypothesis (Theorem 5.8).

Broadly speaking, these results add a time complexity perspective to the already known space complexity, that is for any fixed dimension  $d$ , coverability in unary  $d$ -VASS is NL-complete [45].

**Organisation and Overview.** Section 3 contains our first main result, the improved upper bound on the space required for coverability. Most notably, in Theorem 3.3 we show that if coverability holds then there exists a run of length at most  $n^{2^{\mathcal{O}(d)}}$ . Then, in Corollary 3.4 we are able to obtain a non-deterministic  $2^{\mathcal{O}(d)} \cdot \log(n)$ -space algorithm and a deterministic  $n^{2^{\mathcal{O}(d)}}$ -time algorithm for coverability. In much of the same way as Rackoff, we proceed by induction on the dimension. The difference is in the inductive step; Rackoff's inductive hypothesis dealt with a case where all counters are bounded by the same well-chosen value. Intuitively speaking, the configurations are bounded within a  $d$ -hypercube. This turns out to be suboptimal. This is due to the fact that the volume of a  $d$ -hypercube with sides of length  $\ell$  is  $\ell^d$ ; unrolling the induction steps gives a bound of roughly  $n^{d \cdot (d-1) \cdot \dots \cdot 1} = n^{d!} = n^{2^{\mathcal{O}(d \log d)}}$ , hence the twice-exponentiated  $\log(d)$  factor. The key ingredient in our proof is to replace the  $d$ -hypercubes with a collection of objects with greatly reduced volume, thus reducing the number of configurations in a run witnessing coverability.

Section 4 contains our second main result, the matching lower bound on the time required for coverability that is conditioned upon ETH. In Lemma 4.3, we first reduce from finding a  $k$ -clique in a graph to an instance of coverability in bounded unary 2-VASS with zero-tests. Then, via Lemma 4.4, we implement the aforementioned technique of Rosier and Yen to, when there is a counter bound, remove the zero-tests at the cost of increasing to a  $d$ -dimensional unary VASS. Then, in Theorem 4.2 we are able to conclude, by setting  $k = 2^d$ , that if ETH holds, then there is no deterministic  $n^{2^{\mathcal{O}(d)}}$ -time algorithms for coverability in unary  $d$ -VASS. This is because ETH implies that there is no  $f(k) \cdot n^{o(k)}$ -time algorithm for finding a  $k$ -clique in a graph with  $n$  vertices (Theorem 4.1).

Section 5 contains our other results where we study bounded fixed dimension unary VASS. Firstly, Theorem 5.4 states that under the  $k$ -cycle hypothesis (Hypothesis 5.2), there does not exist a deterministic  $n^{2-o(1)}$ -time algorithm deciding reachability in linearly-bounded unary 1-VASS. Further, we conclude in Corollary 5.5, if the  $k$ -cycle hypothesis is assumed then there does not exist a deterministic  $n^{2-o(1)}$ -time algorithm for coverability in (not bounded) unary 2-VASS. Following, we prove Theorem 5.8, that claims there does not exist a deterministic  $n^{d-o(1)}$ -time algorithm reachability in linearly-bounded unary  $(d+2)$ -VASS under the 3-uniform  $k$ -hyperclique hypothesis (Hypothesis 5.7). We achieve this with two

components. First, in Lemma 5.9, we first reduce from finding a  $4d$ -hyperclique to an instance of reachability in a bounded unary  $(d + 1)$ -VASS with a fixed number of zero-tests. Second, via Lemma 5.10, we implement the newly developed controlling counter technique of Czerwiński and Orlikowski [16] to remove the fixed number of zero-tests at the cost of increasing the dimension by one.

**Related Work.** The coverability problem for VASS has plenty of structure that still receives active attention. The set of configurations from which the target can be covered is upwards-closed, meaning that coverability still holds if the initial counter values are increased. An alternative approach, the *backwards algorithm* for coverability, relies on this phenomenon. Starting from the target configuration, one computes the set of configurations from which it can be covered [1]. Thanks to the upwards-closed property, it suffices to maintain the collection of minimal configurations. The backwards algorithm terminates due to Dickson’s lemma, however, using Rackoff’s bound one can show it runs in double-exponential time [10]. This technique has been deeply analysed for coverability in VASS and some extensions [23, 32]. Despite high complexity, there are many implementations of coverability relying on the backwards algorithm that work well in practice. Intuitively, the idea is to prune the set of configurations, using relaxations that can be efficiently implemented in SMT solvers [21, 7, 8].

Another central decision problem for VASS is the *reachability problem*, asking whether there is a run from a given initial configuration to a given target configuration. Reachability is a provably harder problem. In essence, reachability differs from coverability by allowing one zero-test to each counter. Counter machines, well-known to be equivalent to Turing machines [43], can be seen as VASS with the ability to arbitrarily zero-test counters; coverability and reachability are equivalent here and are undecidable. In 1981, Mayr proved that reachability in VASS is decidable [39], making VASS one of the richest decidable variants of counter machines. Only recently, after decades of work, has the complexity of reachability in VASS been determined to be Ackermann-complete [35, 16, 34]. A widespread technique for obtaining lower bounds for coverability and reachability problems in VASS is to simulate counter machines with some restrictions. Our overall approach to obtaining lower bounds follows suit; we first reduce finding cliques in graphs, finding cycles in graphs, and finding hypercliques in hypergraphs to various intermediate instances of coverability in VASS with extra properties such as bounded counters or a fixed number of zero-tests. These VASS, that are counter machines restricted in some way, are then simulated by standard higher-dimensional VASS. Such simulations are brought about by the two previously developed techniques. Rosier and Yen leverage Lipton’s construction to obtain VASS that can simulate counter machines with bounded counters [46]. Czerwiński and Orlikowski have shown that the presence of an additional counter in a VASS, with carefully chosen transition effects and reachability condition, can be used to implicitly perform a limited number of zero-tests [16].

Recently, some work has been dedicated to the coverability problem for low-dimensional VASS [3, 42]. Furthermore, reachability in low-dimensional VASS has been given plenty of attention, in particular for 1-VASS [48, 26] and for 2-VASS [28, 6]. In the restricted class of flat VASS, other fixed dimensions have also been studied [15, 17].

Another studied variant, *bidirected* VASS, has the property that for every transition  $(p, \mathbf{x}, q)$ , the reverse transition  $(q, -\mathbf{x}, p)$  is also present. The reachability problem in bidirected VASS is equivalent to the uniform word problem in commutative semigroups, both of which are EXPSPACE-complete [40]; not to be confused with the reversible reachability problem in general VASS which is also EXPSPACE-complete [33]. In 1982, Meyer and Mayr listed an open problem that stated, in terms of commutative semigroups, the best known upper

bound for coverability in general VASS [45], the best known lower bound for coverability in bidirected VASS [37], and asked for improvements to these bounds [40, Section 8, Problem 3]. Subsequently, Rosier and Yen refined the upper bound for coverability in general VASS to  $2^{\mathcal{O}(d \log d)} \cdot \log(n)$ -space [46]. Finally, Koppenhagen and Mayr showed that the coverability problem in bidirected VASS can be decided in  $2^{\mathcal{O}(n)}$ -space [30], matching the lower bound.

## 2 Preliminaries

We use bold font for vectors. We index the  $i$ -th component of a vector  $\mathbf{v}$  by writing  $\mathbf{v}[i]$ . Given two vectors  $\mathbf{u}, \mathbf{v} \in \mathbb{Z}^d$  we write  $\mathbf{u} \leq \mathbf{v}$  if  $\mathbf{u}[i] \leq \mathbf{v}[i]$  for each  $1 \leq i \leq d$ . For every  $1 \leq i \leq d$ , we write  $\mathbf{e}_i \in \mathbb{Z}^d$  to represent the  $i$ -th standard basis vector that has  $\mathbf{e}_i[i] = 1$  and  $\mathbf{e}_i[j] = 0$  for all  $j \neq i$ . Given a vector  $\mathbf{v} \in \mathbb{Z}^d$  we define  $\|\mathbf{v}\| = \max\{1, |\mathbf{v}[1]|, \dots, |\mathbf{v}[d]|\}$ . Throughout, we assume that  $\log$  has base 2. We use  $\text{poly}(n)$  to denote  $n^{\mathcal{O}(1)}$ .

A  $d$ -dimensional Vector Addition System with States ( $d$ -VASS)  $\mathcal{V} = (Q, T)$  consists of a non-empty finite set of states  $Q$  and a non-empty set of transitions  $T \subseteq Q \times \mathbb{Z}^d \times Q$ . A *configuration* of a  $d$ -VASS is a pair  $(q, \mathbf{v}) \in Q \times \mathbb{N}^d$  consisting of the current state  $q$  and current counter values  $\mathbf{v}$ , denoted  $q(\mathbf{v})$ . Given two configurations  $p(\mathbf{u}), q(\mathbf{v})$ , we write  $p(\mathbf{u}) \rightarrow q(\mathbf{v})$  if there exists  $t = (p, \mathbf{x}, q) \in T$  where  $\mathbf{x} = \mathbf{v} - \mathbf{u}$ . We may refer to  $\mathbf{x}$  as the *update* of a transition and may also write  $p(\mathbf{v}) \xrightarrow{t} q(\mathbf{w})$  to emphasise the transition  $t$  taken.

A *path* in a VASS is a (possibly empty) sequence of transitions  $((p_1, \mathbf{x}_1, q_1), \dots, (p_\ell, \mathbf{x}_\ell, q_\ell))$ , where  $(p_i, \mathbf{x}_i, q_i) \in T$  for all  $1 \leq i \leq \ell$  and such that the start and end states of consecutive transitions match  $q_i = p_{i+1}$  for all  $1 \leq i \leq \ell - 1$ . A *run*  $\pi$  in a VASS is a sequence of configurations  $\pi = (q_0(\mathbf{v}_0), \dots, q_\ell(\mathbf{v}_\ell))$  such that  $q_i(\mathbf{v}_i) \rightarrow q_{i+1}(\mathbf{v}_{i+1})$  for all  $1 \leq i \leq \ell - 1$ . We denote the length of the run by  $\text{len}(\pi) = \ell + 1$ . If there is such a run  $\pi$ , we can write  $q_0(\mathbf{v}_0) \xrightarrow{\pi} q_\ell(\mathbf{v}_\ell)$ . We may also write  $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$  if there exists a run from  $p(\mathbf{u})$  to  $q(\mathbf{v})$ . The *underlying path* of a run  $\pi$  is sequence of transitions  $(t_1, \dots, t_\ell)$  taken between each of the configurations in  $\pi$ , so  $q_i(\mathbf{v}_i) \xrightarrow{t_{i+1}} q_{i+1}(\mathbf{v}_{i+1})$  for all  $0 \leq i \leq \ell - 1$ .

A  $B$ -bounded  $d$ -VASS, in short  $(B, d)$ -VASS, is given as an integer upper bound on the counter values  $B \in \mathbb{N}$  and  $d$ -VASS  $\mathcal{V}$ . A configuration in a  $(B, d)$ -VASS is a pair  $q(\mathbf{v}) \in Q \times \{0, \dots, B\}^d$ . The notions of paths and runs in bounded VASS remain the same as for VASS, but are accordingly adapted for the appropriate bounded configurations. We note that one should think that  $B$  forms part of the problem statement, not the input, as it will be given implicitly by a function depending on the size of the VASS. For example, we later consider *linearly-bounded*  $d$ -VASS, that represent occasions where  $B = \mathcal{O}(\|\mathcal{V}\|)$ .

We do allow for zero-dimensional VASS, that is VASS with no counters, which can be seen as just directed graphs. A *hypergraph* is a generalisation of the graph. Formally, a hypergraph is a tuple  $H = (V, E)$  where  $V$  is a set of vertices and  $E$  is a collection of non-empty subsets of  $V$  called *hyperedges*. For an integer  $\mu$ , a hypergraph is  $\mu$ -uniform if each hyperedge has cardinality  $\mu$ . Note that a 2-uniform hypergraph is a standard graph.

We study the complexity of the *coverability problem*. An instance  $(\mathcal{V}, p(\mathbf{u}), q(\mathbf{v}))$  of coverability asks whether there is a run in the given VASS  $\mathcal{V}$  from the given initial configuration  $p(\mathbf{u})$  to a configuration  $q(\mathbf{v}')$  with at least the counter values  $\mathbf{v}' \geq \mathbf{v}$  of the given target configuration  $q(\mathbf{v})$ . At times, we also consider the *reachability problem* that additionally requires  $\mathbf{v}' = \mathbf{v}$  so that the target configuration is reached exactly.

To measure the complexity of these problems we need to discuss the encoding used. In *unary encoding*, a  $d$ -VASS  $\mathcal{V} = (Q, T)$  has *size*  $\|\mathcal{V}\| = |Q| + \sum_{(p, \mathbf{x}, q) \in T} \|\mathbf{x}\|$ . We define a *unary*  $d$ -VASS  $\mathcal{U} = (Q', T')$  to have restricted transitions  $T' \subseteq Q' \times \{-1, 0, 1\}^d \times Q'$ , the size is therefore  $\|\mathcal{U}\| = |Q'| + |T'|$ . For any unary encoded  $d$ -VASS  $\mathcal{V}$  there exists an equivalent

unary  $d$ -VASS  $\mathcal{U}$  such that  $\|\mathcal{U}\| = \|\mathcal{V}\|$ . An instance  $(\mathcal{V}, p(\mathbf{u}), q(\mathbf{v}))$  of coverability has size  $n = \|\mathcal{V}\| + \|\mathbf{u}\| + \|\mathbf{v}\|$ . An equal in size, equivalent instance  $(\mathcal{V}', p'(\mathbf{0}), q'(\mathbf{0}))$  of coverability exists; consider adding an initial transition  $(p', \mathbf{s}, p)$  and a final transition  $(q, -\mathbf{t}, q')$ .

It is well known that for  $d$ -VASS, the coverability problem can be reduced to the reachability problem. Indeed, for an instance  $(\mathcal{V}, p(\mathbf{u}), q(\mathbf{v}))$  of coverability, define  $\mathcal{V}' = (Q, T')$  that has additional decremental transitions at the target states  $T' = T \cup \{(q, \mathbf{e}_i, q) : 1 \leq i \leq d\}$ . It is clear that  $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v}')$ , for some  $\mathbf{v}' \geq \mathbf{v}$ , in  $\mathcal{V}$  if and only if  $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$  in  $\mathcal{V}'$ .

► **Lemma 2.1** (folklore). *Let  $(\mathcal{V}, p(\mathbf{u}), q(\mathbf{v}))$  be an instance of coverability. It can be reduced to an instance of reachability  $(\mathcal{V}', p(\mathbf{u}), q(\mathbf{v}))$  such that  $\|\mathcal{V}'\| = \mathcal{O}(\|\mathcal{V}\|)$ .*

A  $d$ -dimensional Vector Addition System ( $d$ -VAS)  $\mathcal{V}$  is a system without states, consisting only of a non-empty collection of transitions  $\mathcal{V} \subseteq \mathbb{Z}^d$ . All definitions, notations, and problems carry over for VAS except that, for simplicity, we drop the states across the board. For example, a configuration in a VAS is just a vector  $\mathbf{v} \in \mathbb{N}^d$ . Another well-known result from the seventies by Hopcroft and Pansiot, one can simulate the states of a VASS at the cost of three extra dimensions in a VAS [28]. For clarity, the VAS obtained has an equivalent reachability relation between configurations; a configuration  $q(\mathbf{x})$  in the original VASS corresponds with a configuration  $(\mathbf{x}, a, b, c)$  in the VAS, where  $a, b$ , and  $c$  represent the state  $q$ .

► **Lemma 2.2** ([28, Lemma 2.1]). *A  $d$ -VASS  $\mathcal{V}$  can be simulated by  $(d+3)$ -VAS  $\mathcal{V}'$  such that  $\|\mathcal{V}'\| = \text{poly}(\|\mathcal{V}\|)$ .*

### 3 Improved Bounds on the Maximum Counter Value

This section is devoted to our improvement of the seminal result of Rackoff. Throughout, we fix our attention to the arbitrary instance  $(\mathcal{V}, p(\mathbf{s}), q(\mathbf{t}))$  of the coverability problem in a  $d$ -VASS  $\mathcal{V} = (Q, T)$  from the initial configuration  $p(\mathbf{s})$  to a configuration  $q(\mathbf{t}')$  with at least the counter values of the target configuration  $q(\mathbf{t})$ . We denote  $n = \|\mathcal{V}\| + \|\mathbf{s}\| + \|\mathbf{t}\|$ . Informally,  $n$  may as well be the number of states plus the absolute value of the greatest update on any transition, for these differences can be subsumed by the second exponent in our following upper bounds. The following two theorems follow from Rackoff's technique and subsequent work by Rosier and Yen, in particular see [45, Lemma 3.4 and Theorem 3.5] and [46, Theorem 2.1 and Lemma 2.2].

► **Theorem 3.1** (Corollary of [45, Lemma 3.4] and [46, Theorem 2.1]). *Suppose  $p(\mathbf{s}) \xrightarrow{*} q(\mathbf{t}')$  for some  $\mathbf{t}' \geq \mathbf{t}$ . Then there exists a run  $\pi$  such that  $p(\mathbf{s}) \xrightarrow{\pi} q(\mathbf{t}'')$  for some  $\mathbf{t}'' \geq \mathbf{t}$  and  $\text{len}(\pi) \leq n^{2^{\mathcal{O}(d \log d)}}$ .*

► **Theorem 3.2** (cf. [45, Theorem 3.5]). *For a given  $d$ -VASS  $\mathcal{V}$ , integer  $\ell$ , and two configurations  $p(\mathbf{s})$  and  $q(\mathbf{t})$ , there is an algorithm that determines the existence of a run  $\pi$  of length  $\text{len}(\pi) \leq \ell$  that witnesses coverability, so  $p(\mathbf{s}) \xrightarrow{\pi} q(\mathbf{t}')$  for some  $\mathbf{t}' \geq \mathbf{t}$ . The algorithm can be implemented to run in non-deterministic  $\mathcal{O}(d \log(n \cdot \ell))$ -space or deterministic  $2^{\mathcal{O}(d \log(n \cdot \ell))}$ -time.*

Note that Theorem 3.1 combined with Theorem 3.2, that is proved in the full version [31], yield non-deterministic  $2^{\mathcal{O}(d \log d)}$ -space and deterministic  $n^{2^{\mathcal{O}(d \log(d))}}$ -time algorithms for coverability. Our result improves this by a  $\mathcal{O}(\log(d))$  factor in the second exponent.

► **Theorem 3.3.** *Suppose  $p(\mathbf{s}) \xrightarrow{*} q(\mathbf{t}')$  for some  $\mathbf{t}' \geq \mathbf{t}$ . Then there exists a run  $\pi$  such that  $p(\mathbf{s}) \xrightarrow{\pi} q(\mathbf{t}'')$  for some  $\mathbf{t}'' \geq \mathbf{t}$  and  $\text{len}(\pi) \leq n^{2^{\mathcal{O}(d)}}$ .*

This combined with Theorem 3.2 yields the following corollary.

► **Corollary 3.4.** *Coverability in  $d$ -VASS can be decided by both a non-deterministic  $2^{\mathcal{O}(d)} \cdot \log(n)$ -space algorithm and a deterministic  $n^{2^{\mathcal{O}(d)}}$ -time algorithm.*

Note that by Lemma 2.2, we may handle VAS instead of VASS. Recall that, as there are no states, a  $d$ -VAS consists only of a set of vectors in  $\mathbb{Z}^d$  that we still refer to as transitions. A configuration is just a vector in  $\mathbb{N}^d$ . Accordingly, we may fix our attention on the instance  $(\mathcal{V}, \mathbf{s}, \mathbf{t})$  of the coverability problem in a  $d$ -VAS  $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_m\}$  from the initial configuration  $\mathbf{s}$  to a configuration  $\mathbf{t}'$  that is at least as great as the target configuration  $\mathbf{t}$ . The rest of this section is dedicated to the proof of Theorem 3.3. Imitating Rackoff's proof, we proceed by induction on the dimension  $d$ . Formally, we prove a stronger statement; Theorem 3.3 is a direct corollary of the following lemma.

► **Lemma 3.5.** *Define  $L_i := n^{4^i}$ , and let  $\mathbf{t} \in \mathbb{N}^d$  such that  $\|\mathbf{t}\| \leq n$ . For any  $\mathbf{s} \in \mathbb{N}^d$ , if  $\mathbf{s} \xrightarrow{*} \mathbf{t}'$  for some  $\mathbf{t}' \geq \mathbf{t}$  then there exists a run  $\pi$  such that  $\mathbf{s} \xrightarrow{\pi} \mathbf{t}''$  for some  $\mathbf{t}'' \geq \mathbf{t}$  and  $\text{len}(\pi) \leq L_d$ .*

The base case is  $d = 0$ . In a 0-dimensional VAS, the only possible configuration is the empty vector  $\varepsilon$  and therefore there is only the trivial run  $\varepsilon \xrightarrow{*} \varepsilon$ . This trivially satisfies the lemma.

For the inductive step, when  $d \geq 1$ , we assume that Lemma 3.5 holds for all lower dimensions  $0, \dots, d-1$ . Let  $\pi = (\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_\ell)$  be a run with minimal length such that  $\mathbf{s} \xrightarrow{\pi} \mathbf{t}'$  for some  $\mathbf{t}' \geq \mathbf{t}$ , so in particular,  $\mathbf{c}_0 = \mathbf{s}$  and  $\mathbf{c}_\ell = \mathbf{t}'$ . Our objective is to prove that  $\text{len}(\pi) = \ell + 1 \leq L_d$ . Observe that configurations  $\mathbf{c}_i$  need to be distinct, else  $\pi$  could be shortened trivially. We introduce the notion of a *thin configuration*.

► **Definition 3.6 (Thin Configuration).** *In a  $d$ -VAS, we say that a configuration  $\mathbf{c} \in \mathbb{N}^d$  is thin if there exists a permutation  $\sigma$  of  $\{1, \dots, d\}$  such that  $\mathbf{c}[\sigma(i)] < M_i$  for every  $i \in \{1, \dots, d\}$ , where  $M_0 := n$  and for  $i \geq 1$ ,  $M_i := L_{i-1} \cdot n$ .*

Recall, from above, the run  $\pi = (\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_\ell)$ . Let  $t \in \{0, \dots, \ell\}$  be the first index where  $\mathbf{c}_t$  is not thin, otherwise let  $t = \ell + 1$  if every configuration in  $\pi$  is thin. We decompose the run about the  $t$ -th configuration  $\pi = \pi_{\text{thin}} \cdot \pi_{\text{tail}}$ , where  $\pi_{\text{thin}} := (\mathbf{c}_0, \dots, \mathbf{c}_{t-1})$  and  $\pi_{\text{tail}} := (\mathbf{c}_t, \dots, \mathbf{c}_\ell)$ . Note that  $\pi_{\text{thin}}$  or  $\pi_{\text{tail}}$  can be empty. Subsequently, we individually analyse the lengths of  $\pi_{\text{thin}}$  and  $\pi_{\text{tail}}$  (see Figure 1). We will also denote  $\mathbf{m} = \mathbf{c}_t$  to be the first configuration that is not thin.

▷ **Claim 3.7.**  $\text{len}(\pi_{\text{thin}}) \leq d! \cdot n^d \cdot L_{d-1} \cdot \dots \cdot L_0$ .

Proof. By definition, every configuration in  $\pi_{\text{thin}}$  is thin. Moreover, since  $\pi$  has a minimal length, no configurations in  $\pi$  repeat, let alone in  $\pi_{\text{thin}}$ . We now count the number of possible thin configurations. There are  $d!$  many permutations of  $\{1, \dots, d\}$ . For a given permutation  $\sigma$  and an index  $i \in \{1, \dots, d\}$ , we know that for a thin configuration  $\mathbf{c}$ ,  $0 \leq \mathbf{c}[\sigma(i)] < M_i$ , so there are at most  $M_i = L_{i-1} \cdot n$  many possible values on the  $\sigma(i)$ -th counter. Hence the total number of thin configurations is at most  $d! \cdot \prod_{i=1}^d (L_{i-1} \cdot n) = d! \cdot n^d \cdot L_{d-1} \cdot \dots \cdot L_0$ . ◁

▷ **Claim 3.8.**  $\text{len}(\pi_{\text{tail}}) \leq L_{d-1}$ .

Proof. Consider  $\mathbf{m} \in \mathbb{N}^d$ , the first configuration of  $\pi_{\text{tail}}$ . Let  $\sigma$  be a permutation such that  $\mathbf{m}[\sigma(1)] \leq \mathbf{m}[\sigma(2)] \leq \dots \leq \mathbf{m}[\sigma(d)]$ . Given that  $\mathbf{m}$  is not thin, for every permutation  $\sigma'$  there exists an  $i \in \{1, \dots, d\}$  such that  $\mathbf{m}[\sigma'(i)] \geq M_i$ ; in particular, this holds for  $\sigma$ . Note that this also implies  $M_i \leq \mathbf{m}[\sigma(i+1)] \leq \dots \leq \mathbf{m}[\sigma(d)]$ .



Let  $j$  be any of the remaining components. Recall that by the choice of  $\mathbf{m}$ ,  $\mathbf{m}[j] \geq M_i = n \cdot L_{i-1}$ . Since  $n > \|\mathcal{V}\| \geq \|\mathbf{v}_j\|$  for every  $1 \leq j \leq \text{len}(\rho')$ , this means that in a single step, the value of a counter can change by at most  $n$ . Given that  $\text{len}(\rho) = \text{len}(\rho') \leq L_{i-1}$ , the value on each of the remaining components must be at least  $n$  for every configuration in  $\rho$ . In particular, observing that  $\|\mathbf{t}\| \leq n$ , the final configuration of  $\rho$  satisfies

$$\mathbf{m} + \sum_{j=1}^{\text{len}(\rho')} \mathbf{v}_j \geq \mathbf{t}.$$

Finally, observe that  $\text{len}(\rho) = \text{len}(\rho') \leq L_{i-1} \leq L_{d-1}$ . ◁

To conclude this section, we show that Lemma 3.5 follows from Claim 3.7 and Claim 3.8.

**Proof of Lemma 3.5.** From Claim 3.7 and Claim 3.8,

$$\begin{aligned} \text{len}(\pi) &\leq \text{len}(\pi_{\text{thin}}) + \text{len}(\pi_{\text{tail}}) \leq d! \cdot n^d \cdot L_{d-1} \cdot \dots \cdot L_0 + L_{d-1} \\ &\leq 2 \cdot d! \cdot n^d \cdot L_{d-1} \cdot \dots \cdot L_0. \end{aligned}$$

Recall that  $n \geq 2$  and observe that  $2 \cdot d! \cdot n^d \leq n^{2^d}$ . Hence,

$$\text{len}(\pi) \leq n^{2^d} \cdot L_{d-1} \cdot \dots \cdot L_0.$$

Next, we use the definition of  $L_i := n^{4^i}$  to show

$$\text{len}(\pi) \leq n^{2^d} \cdot \prod_{i=0}^{d-1} n^{4^i} \leq n^{(2^d + \sum_{i=0}^{d-1} 4^i)}.$$

Finally, when  $d \geq 1$ ,  $2^d + \sum_{i=0}^{d-1} 4^i \leq 4^d$  holds, therefore

$$\text{len}(\pi) \leq n^{4^d} = L_d. \quad \blacktriangleleft$$

## 4 Conditional Time Lower Bound for Coverability

In this section, we present a conditional lower bound based on the *Exponential Time Hypothesis* (ETH) [29]. Roughly speaking, ETH is a conjecture that an  $n$ -variable instance of 3-SAT cannot be solved by a deterministic  $2^{o(n)}$ -time algorithm (for a modern survey, see [38]). In our reductions, it will be convenient for us to work with the  $k$ -clique problem instead. In the  $k$ -clique problem we are given a graph  $G = (V, E)$  as an input and the task is to decide whether there is a set of  $k$  pairwise adjacent vertices in  $V$ . The naive algorithm for  $k$ -clique runs in  $\mathcal{O}(n^k)$  time. Even though the exact constant in the dependence on  $k$  can be improved [44], ETH implies that the exponent must have a linear dependence on  $k$ .

► **Theorem 4.1** ([11, Theorem 4.2], [12, Theorem 4.5], and [14, Theorem 14.21]). *Assuming the Exponential Time Hypothesis, there is no algorithm running in  $f(k) \cdot n^{o(k)}$ -time for the  $k$ -clique problem for any computable function  $f$ . Moreover one can assume that  $G$  is  $k$ -partite, i.e.  $G = (V_1 \cup \dots \cup V_k, E)$  and edges belong to  $V_i \times V_j$  for  $i \neq j \in \{1, \dots, k\}$ .*

We will use Theorem 4.1 to show the following conditional lower bound for coverability in unary  $d$ -VASS, which is proved at the end of this section.

► **Theorem 4.2.** *Assuming the Exponential Time Hypothesis, there does not exist an  $n^{2^{o(d)}}$ -time algorithm deciding coverability in a unary  $d$ -VASS with  $n$  states.*

We first reduce the  $k$ -clique problem to coverability in bounded 2-VASS with the ability to perform a fixed number of zero-tests. We will then leverage a result by Rosier and Yen to construct an equivalent, with respect to coverability,  $(\mathcal{O}(\log k))$ -VASS without zero-tests.

## 131:10 Coverability in VASS Revisited

► **Lemma 4.3.** *Given a  $k$ -partite graph  $G = (V_1 \cup \dots \cup V_k, E)$  with  $n$  vertices, there exists a unary  $(\mathcal{O}(n^{2k}), 2)$ -VASS  $\mathcal{T}$  such that there is a  $k$ -clique in  $G$  if and only if there exists a run from  $q_I(\mathbf{0})$  to  $q_F(\mathbf{v})$  in  $\mathcal{T}$ , for some  $\mathbf{v} \geq \mathbf{0}$ . Moreover,  $\|\mathcal{T}\| \leq \text{poly}(n+k)$  and  $\mathcal{T}$  can be constructed in  $\text{poly}(n+k)$ -time.*

**Proof.** Without loss of generality, we may assume that each of the  $k$  vertex subsets in the graph has the same size  $|V_1| = \dots = |V_k| = \ell$ . Thus  $n = k \cdot \ell$ . For convenience, we denote  $V = \{1, \dots, k\} \times \{1, \dots, \ell\}$ .

We begin by sketching the main ideas behind the reduction before they are implemented. We start by finding the first  $n = k \cdot \ell$  primes and associating a distinct prime  $p_{i,j}$  to each vertex  $(i, j) \in V$ . Note that a product of  $k$  different primes uniquely corresponds to selecting  $k$  vertices. Thus the idea is to guess such a product, and test whether the corresponding vertices form a  $k$ -clique. To simplify the presentation we present VASS also as counter programs, inspired by Esparza's presentation of Lipton's lower bound [20, Section 7].

We present an overview of our construction in Algorithm 1. Note that the counter  $y$  is used only by subprocedures. Initially both counter values are 0, as in the initial configuration of the coverability instance. The program is non-deterministic and we are interested in the existence of a certain run. One should think that coverability holds if and only if there is a run through the code without getting stuck so to say. In this example a run can be stuck only in the **Edge** $[e]$  subprocedure, that will be explained later. The precise final counter values are not important, as we are simply aiming to cover the target counter values  $\mathbf{0}$ . The variable  $i$  (in the first loop) and variables  $i$  and  $j$  (in the second loop) are just syntactic sugar for copying similar code multiple times. The variables  $j$  (in the first loop) and  $e$  (in the second loop) allow us to neatly represent non-determinism in a VASS.

■ **Algorithm 1** A counter program for a VASS with zero tests with two counters  $x$  and  $y$ .

---

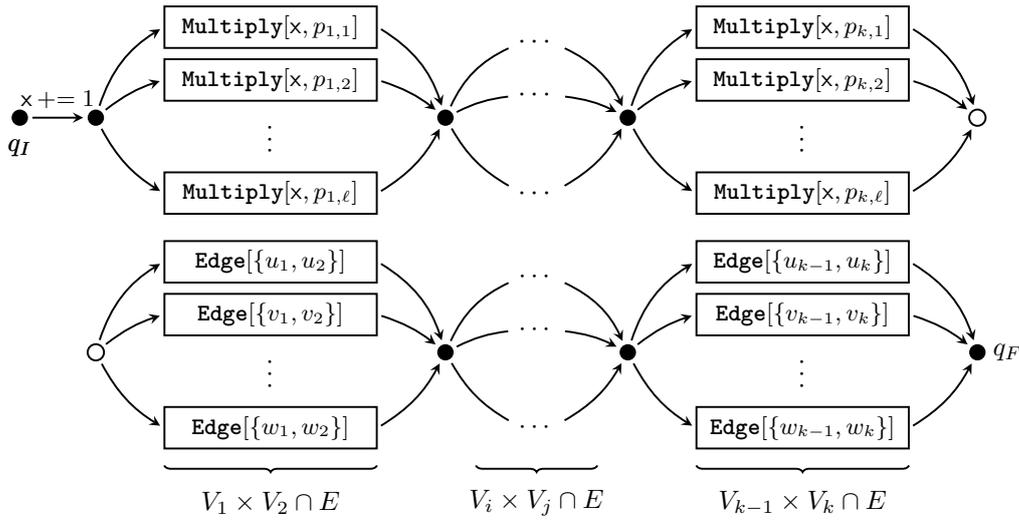
```

input :  $x = 0, y = 0$ 
 $x += 1$ 
for  $i \leftarrow 1$  to  $k$  do
  | guess  $j \in \{1, \dots, \ell\}$ 
  | Multiply $[x, p_{i,j}]$ 
end
for  $(i, j) \in \{1, \dots, k\}^2, i \neq j$  do
  | guess  $e \in E \cap (V_i \times V_j)$ 
  | Edge $[e]$ 
end

```

---

Algorithm 1 uses the **Multiply** $[x, p]$  and **Edge** $[e]$  subprocedures. These two subprocedures will be implemented later. Note that **Multiply** $[x, p]$  takes a counter  $x$  as input as we later reuse this subprocedure when there is more than one counter subject to multiplication. The intended behaviour of **Multiply** $[x, p]$  is that it can be performed if and only if as a result we get  $x = x \cdot p$ , despite the fact that VASS can only additively increase and decrease counters. The subprocedure **Edge** $[e]$  can be performed if and only if both vertices of the edge  $e$  are encoded in the value of the counter  $x$ . Overall, Algorithm 1 is designed so that in the first part the variable  $x$  is multiplied by  $p_{i,j}$ , where for every  $i$  one  $j$  is guessed. This equates to selecting one vertex from each  $V_i$ . Then the second part the algorithm checks whether between every pair of selected vertices from  $V_i$  and  $V_j$  there is an edge. Clearly there is a run through the program that does not get stuck if and only if there is  $k$ -clique in  $G$ .



■ **Figure 2** The top part of the VASS implements the first line and the first loop in Algorithm 1. The variable  $x$  is multiplied by  $k$  non-deterministically chosen primes  $p_{i,j}$ , each corresponding to a vertex in  $V_i$ . The bottom part of the VASS implements the second loop in Algorithm 1. For every pair  $i \neq j$  the VASS non-deterministically chooses  $e \in V_i \cap V_j$  and invokes the subprocedure **Edge**[ $e$ ].

In Figure 2 we present a VASS with zero-tests implementing Algorithm 1. The construction will guarantee that  $q_F(\mathbf{0})$  can be covered from  $q_I(\mathbf{0})$  if and only if there is a  $k$ -clique in  $G$ .

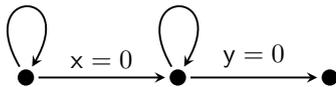
It remains to define the subprocedures. One should think that every call of a subprocedure corresponds to a unique part of the VASS, like a gadget of sorts. To enter and leave the subprocedure one needs to add trivial transitions that do not change the counter values. All subprocedures rely on the invariant  $y = 0$  at the beginning and admit the invariant at the end.

■ **Algorithm 2** The counter program of **Multiply**[ $x, p$ ] above its VASS implementation (left) and the counter program of **Divide**[ $x, p$ ] above its VASS implementation (right).

**input** :  $x = v, y = 0$   
**output** :  $x = v \cdot p, y = 0$

```
repeat
  | x -= 1; y += 1
until x = 0
repeat
  | x += p; y -= 1
until y = 0
```

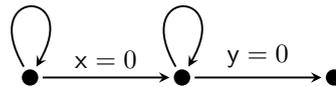
$x -= 1$        $x += p$   
 $y += 1$        $y -= 1$



**input** :  $x = v \cdot p, y = 0$   
**output** :  $x = v, y = 0$

```
repeat
  | x -= p; y += 1
until x = 0
repeat
  | x += 1; y -= 1
until y = 0
```

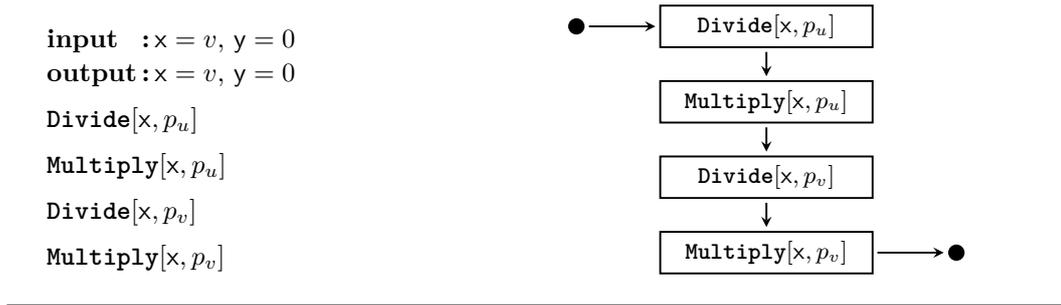
$x -= p$        $x += 1$   
 $y += 1$        $y -= 1$



## 131:12 Coverability in VASS Revisited

We start with  $\text{Multiply}[x, p]$  and  $\text{Divide}[x, p]$  that indeed multiply and divide  $x$  by  $p$ , respectively. See Algorithm 2 for the counter program and VASS implementations. Notice that the repeat loops correspond to the self-loops in the VASS. In the  $\text{Multiply}[x, p]$  gadget, it is easy to see that a run passes through the procedure if and only if the counter  $x$  is multiplied by  $p$ . Similarly, in the  $\text{Divide}[x, p]$  gadget, it is easy to see that a run pass through the procedure if and only if the counter  $x$  is divided by  $p$  wholly. Indeed, the division procedure would get stuck if  $p \nmid x$  because it will be impossible to exit the first loop.

■ **Algorithm 3** The counter program for  $\text{Edge}[\{u, v\}]$  and its VASS implementation.



The procedure  $\text{Edge}[\{u, v\}]$  is very simple, it is a sequence of four subprocedures, see Algorithm 3. Indeed, to check if the vertices from edge  $e$  are encoded in  $x$  we simply check whether  $x$  is divisible by the corresponding primes. Afterwards we multiply  $x$  with the same primes so that the value does not change and it is ready for future edge checks.

It remains to analyse the size of the VASS and its construction time in this reduction time. In every run from  $q_I(\mathbf{0})$  to  $q_F(\mathbf{v})$ , for some  $\mathbf{v} \geq \mathbf{0}$ , the greatest counter value observable can be bounded above by  $p^k$  where  $p$  is the  $n$ -th prime. By the Prime Number Theorem (for example, see [51]), we know that  $p^k \leq \mathcal{O}((n \log(n))^k) \leq \mathcal{O}(n^{2k})$  is an upper bound on the counter values observed. Hence  $\mathcal{T}$  is an  $\mathcal{O}(n^{2k})$ -bounded unary 2-VASS.

Finally, the  $\text{Multiply}$  and  $\text{Divide}$  subprocedures contain three states and five transitions. Since the  $n$ -th prime is bounded above by  $\mathcal{O}(n \log(n))$ , we also get  $\|\mathcal{T}\| = \mathcal{O}(n \log(n))$ , hence our VASS can be represented using unary encoding. Analysing Algorithm 1, it is easy to see that overall the number of states is polynomial in  $n$ . Finally, the first  $n$  primes can be found in  $\mathcal{O}(n^{1+o(1)})$ -time [2]. Therefore, in total  $\mathcal{T}$  has size  $\|\mathcal{T}\| = \text{poly}(n + k)$  and can be constructed in  $\text{poly}(n + k)$ -time. ◀

To attain conditional lower bounds for coverability we must replace the zero-tests. We make use of a technique of Rosier and Yen [46] that relies on the construction of Lipton [37]. They show that a  $(2n)^{2k}$ -bounded counter machine with finite state control can be simulated by a unary  $(\mathcal{O}(k))$ -VASS with  $n$  states. As Rosier and Yen detail after their proof, it is possible to apply this technique to multiple counters with zero-tests at once [46]. This accordingly results in the number of VASS counters increasing, but we instantiate this with just two counters. We remark that the VASS constructed in Lemma 4.3 is structurally bounded, so for any initial configuration there is a limit on the largest observable counter, as is the case in the VASS Lipton constructed [37].

► **Lemma 4.4** (Corollary of [46, Lemma 4.3]). *Let  $\mathcal{T}$  be an  $n$ -state unary  $(n^{\mathcal{O}(k)}, 2)$ -VASS with zero-tests, for some parameter  $k$ . Then there exists an  $\mathcal{O}(n)$ -state  $(\mathcal{O}(\log k))$ -VASS  $\mathcal{V}$ , such that there is a run from  $q_I(\mathbf{0})$  to  $q_F(\mathbf{v})$ , for some  $\mathbf{v} \geq \mathbf{0}$ , in  $\mathcal{T}$  if and only if there is a run from  $q'_I(\mathbf{0})$  to  $q'_F(\mathbf{w})$ , for some  $\mathbf{w} \geq \mathbf{0}$ , in  $\mathcal{V}$ . Moreover,  $\mathcal{V}$  has size  $\mathcal{O}(|\mathcal{T}|)$  and can be constructed in the same time.*

**Proof of Theorem 4.2.** Let  $k = 2^d$ . We instantiate Lemma 4.3 on  $k$ -partite graphs  $G$  with  $n$  vertices. We therefore obtain a unary  $(n^{2^{\mathcal{O}(d)}}, 2)$ -VASS with zero tests  $\mathcal{T}$  such that  $G$  contains a  $k$ -clique if and only if there is a run from  $q_I(\mathbf{0})$  to  $q_F(\mathbf{v})$ , for some  $\mathbf{v} \geq \mathbf{0}$ , in  $\mathcal{T}$ .

Given the bound on the value of the counters, we can apply Lemma 4.4 to  $\mathcal{T}$ . This gives us an  $\mathcal{O}(n)$ -state  $(\mathcal{O}(d))$ -VASS  $\mathcal{V}$  such that  $G$  contains a  $k$ -clique if and only if there is a run from  $q'_I(\mathbf{0})$  to  $q'_F(\mathbf{w})$ , for some  $\mathbf{w} \geq \mathbf{0}$ , in  $\mathcal{V}$ .

By Theorem 4.1 we conclude that under the Exponential Time Hypothesis there does not exist an  $n^{2^{\mathcal{O}(d)}}$ -time algorithm deciding coverability in unary  $d$ -VASS. ◀

## 5 Coverability and Reachability in Bounded Unary VASS

In this section, we give even tighter bounds for coverability in *bounded* fixed dimension unary VASS. Specifically, for a time constructible function  $B(n)$ , the coverability problem in  $(B(n), d)$ -VASS asks, for a given  $(B(n), d)$ -VASS  $\mathcal{V} = (Q, T)$  of size  $n$  as well as configurations  $p(\mathbf{u})$ ,  $q(\mathbf{v})$ , whether there is a run in  $\mathcal{V}$  from  $p(\mathbf{u})$  to  $q(\mathbf{v}')$  for some  $\mathbf{v}' \geq \mathbf{v}$  such that each counter value remains in  $\{0, \dots, B(n)\}$  throughout. We would like to clarify the fact that the bound is not an input parameter. We focus on the natural setting of linearly-bounded fixed dimension VASS, that is  $(\mathcal{O}(n), d)$ -VASS. There is a simple algorithm, given in the proof of Observation 5.1 which can be found in the full version [31], that yields an immediate  $\mathcal{O}(n^{d+1})$  upper bound for the time needed to decide the coverability problem. We accompany this observation with closely matching lower bounds, see Table 1 for an overview.

■ **Table 1** Conditional lower bounds and upper bounds of the time complexity of coverability and reachability in unary  $(\mathcal{O}(n), d)$ -VASS. For clarity, we remark that Theorem 5.4 is subject to Hypothesis 5.2 and that Theorem 5.8 is subject to Hypothesis 5.7. Note that the lower bounds for dimensions  $d = 2$  and  $d = 3$  follow from Theorem 5.4 by just adding components consisting of only zeros. All upper bounds follow from Observation 5.1.

$d$	Lower Bound	Upper Bound
0	$\Omega(n)$ (trivial)	$\mathcal{O}(n)$
1	$n^{2-o(1)}$ (Theorem 5.4)	$\mathcal{O}(n^2)$
2	$n^{2-o(1)}$ (from above)	$\mathcal{O}(n^3)$
3	$n^{2-o(1)}$ (from above)	$\mathcal{O}(n^4)$
$d \geq 4$	$n^{d-2-o(1)}$ (Theorem 5.8)	$\mathcal{O}(n^{d+1})$

► **Observation 5.1.** *Coverability in an  $n$ -sized unary  $(B(n), d)$ -VASS can be solved in  $\mathcal{O}(n(B(n) + 1)^d)$ -time.*

### Lower Bounds for Coverability in Linearly-Bounded VASS

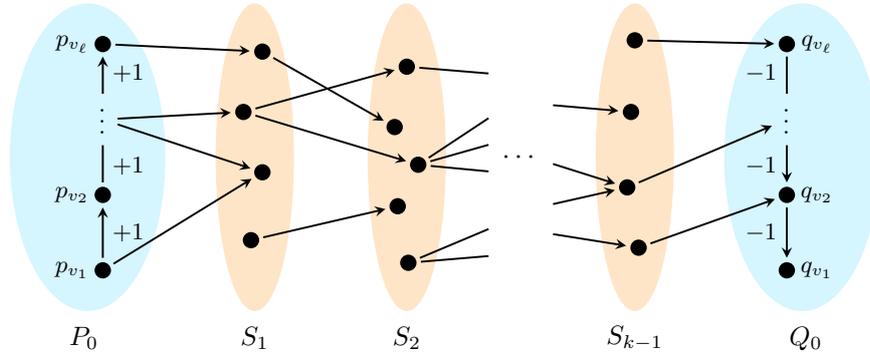
Now, we consider lower bounds for the coverability problem in linearly-bounded fixed dimension unary VASS. Firstly, in dimension one, we show that quadratic running time is conditionally optimal under the  $k$ -cycle hypothesis. Secondly, in dimensions four and higher, we require a running time at least  $n^{d-2-o(1)}$  under the 3-uniform hyperclique hypothesis. Together, this provides evidence that the simple  $\mathcal{O}(n^{d+1})$  algorithm for coverability in  $(\mathcal{O}(n), d)$ -VASS is close to optimal, as summarised in Table 1.

► **Hypothesis 5.2** ( $k$ -Cycle Hypothesis). *For every  $\varepsilon > 0$ , there exists a  $k$  such that there does not exist a  $\mathcal{O}(m^{2-\varepsilon})$ -time algorithm for finding a  $k$ -cycle in directed graphs with  $m$  edges.*

The  $k$ -cycle hypothesis arises from the state-of-the-art  $\mathcal{O}(m^{2-\frac{c}{k}+o(1)})$ -time algorithms, where  $c$  is some constant [4, 50, 19]. It has been previously used as an assumption for hardness results, for example, see [36, 5, 18]. It is a standard observation, due to colour-coding arguments, that we may without loss of generality assume that the graph given is a  $k$ -circle-layered graph [36, Lemma 2.2]. Specifically, we can assume that the input graph  $G = (V, E)$  has vertex partition  $V = V_0 \cup \dots \cup V_{k-1}$  such that each edge  $\{u, v\} \in E$  is in  $V_i \times V_{i+1 \pmod k}$  for some  $0 \leq i < k$ . Furthermore, we may assume  $|V| \leq |E|$ .

The core of the upcoming lower bounds is captured in the following lemma; see Figure 3 for an overview and the full version [31] for the proof.

► **Lemma 5.3.** *Given a  $k$ -circle-layered graph  $G = (V_0 \cup \dots \cup V_{k-1}, E)$  with  $m$  edges, there exists a unary  $(\mathcal{O}(n), 1)$ -VASS  $\mathcal{V}$  such that there is a  $k$ -cycle in  $G$  if and only if there exists a run from  $p(0)$  to  $q(0)$  in  $\mathcal{V}$ . Moreover,  $\mathcal{V}$  has size  $n \leq \mathcal{O}(m)$  and can be constructed in  $\mathcal{O}(m)$ -time.*



■ **Figure 3** The  $(\mathcal{O}(n), 1)$ -VASS  $\mathcal{V}$  of size  $n \leq \mathcal{O}(m)$  for finding  $k$ -cycle in a  $k$ -circle-layered graphs with  $m$  edges. Note that unlabelled transitions have zero effect. Observe that the graph is mostly copied into the states and transitions of the linearly-bounded 1-VASS. Importantly, two copies of  $V_0$  are created. By starting at  $p_{v_1}(0)$  in the first copy, a vertex from  $V_0$  belonging to the  $k$ -cycle can be selected by loading the sole counter with a value corresponding to that vertex. Then, in the second copy,  $q_{v_1}(0)$  can only be reached if the state first arrived at corresponds to the vertex selected in the beginning. Accordingly, there is a run from  $p_{v_1}(0)$  to  $q_{v_1}(0)$  if and only if there exists a  $k$ -cycle, since the states visited in the underlying path of the run correspond to the vertices of the  $k$ -cycle.

► **Theorem 5.4.** *Assuming the  $k$ -cycle hypothesis, coverability and reachability in unary  $(\mathcal{O}(n), 1)$ -VASS of size  $n$  require  $n^{2-o(1)}$ -time.*

**Proof.** Assume for contradiction that reachability in a unary  $(\mathcal{O}(n), 1)$ -VASS of size  $n$  can be solved in  $\mathcal{O}(n^{2-\varepsilon})$ -time for some  $\varepsilon > 0$ . By the  $k$ -cycle hypothesis (Hypothesis 5.2), there exists a  $k$  such that the problem of finding a  $k$ -cycle in a  $k$ -circle layered graph with  $m$  vertices cannot be solved in  $\mathcal{O}(m^{2-\varepsilon})$ -time. Via the reduction presented above in Lemma 5.3, we create a  $(\mathcal{O}(n), 1)$ -VASS  $\mathcal{V}$  of size  $n \leq \mathcal{O}(m)$  together with an initial configuration  $p(0)$  and a target configuration  $q(0)$ , such that deciding reachability from  $p(0)$  to  $q(0)$  in  $\mathcal{V}$  determines the existence of a  $k$ -cycle in  $G$ . Thus the  $\mathcal{O}(n^{2-\varepsilon})$  algorithm for reachability would give a  $\mathcal{O}(m^{2-\varepsilon})$  algorithm for finding  $k$ -cycles, contradicting the  $k$ -cycle hypothesis.

By the equivalence of coverability and reachability in unary  $(\mathcal{O}(n), 1)$  VASS in Lemma 5.6, the same lower bound holds for coverability. ◀

► **Corollary 5.5.** *Assuming the  $k$ -cycle hypothesis, coverability in unary 2-VASS of size  $n$  requires  $n^{2-o(1)}$ -time.*

Reachability in  $(\mathcal{O}(n), d)$ -VASS can be decided in  $\mathcal{O}(n(B(n) + 1)^d)$ -time using the simple algorithm for Observation 5.1 with a trivially modified acceptance condition. It turns out that coverability and reachability are equivalent in unary  $(\mathcal{O}(n), d)$ -VASS. The following lemma is proved in the full version [31].

► **Lemma 5.6.** *For a  $(B(n), d)$ -VASS, let  $C^{B(n)}(n)$  and  $R^{B(n)}(n)$  denote the optimal running times for coverability and reachability, respectively. For any  $\gamma > 0$ , there exists some  $\delta > 0$  such that  $C^{\gamma n}(n) \leq \mathcal{O}(R^{\delta n}(n))$ . Conversely, for any  $\gamma > 0$ , there exists some  $\delta > 0$  such that  $R^{\gamma n}(n) \leq \mathcal{O}(C^{\delta n}(n))$ .*

## Lower Bounds for Reachability in Linearly-Bounded VASS

To obtain further lower bounds for the coverability problem in  $(\mathcal{O}(n), d)$ -VASS, by Lemma 5.6, we can equivalently find lower bounds for the reachability problem in  $(\mathcal{O}(n), d)$ -VASS. In Theorem 5.8, we will assume a well-established hypothesis concerning the time required to find hypercliques in 3-uniform hypergraphs. In fact, Lincoln, Vassilevska Williams, and Williams state and justify an even stronger hypothesis about  $\mu$ -uniform hypergraphs for every  $\mu \geq 3$  [36, Hypothesis 1.4]. We will use this computational complexity hypothesis to expose precise lower bounds on the time complexity of reachability in linearly-bounded fixed dimension unary VASS.

► **Hypothesis 5.7** (*k*-Hyperclique Hypothesis [36, Hypothesis 1.4]). *Let  $k \geq 3$  be an integer. On Word-RAM with  $\mathcal{O}(\log(n))$  bit words, finding an  $k$ -hyperclique in a 3-uniform hypergraph on  $n$  vertices requires  $n^{k-o(1)}$ -time.*

► **Theorem 5.8.** *Assuming Hypothesis 5.7, reachability in unary  $(\mathcal{O}(n), d + 2)$ -VASS of size  $n$  requires  $n^{d-o(1)}$ -time.*

For the remainder of this section, we focus on the proof of Theorem 5.8. The lower bound is obtained via reduction from finding hyperclique in 3-uniform hypergraphs, hence it is subject to the  $k$ -Hyperclique Hypothesis. We present our reduction in two steps. The first step is an intermediate step, in Lemma 5.9 we offer a reduction to an instance of reachability in unary VASS with a limited number of zero-tests (proved in the full version [31]). The second step extends the first, in Lemma 5.10 we modify the reduction by adding a counter so zero-tests are absented. This extension leverages the recently developed *controlling counter technique* of Czerwiński and Orlikowski [16]. This technique allows for implicit zero-tests to be performed in the presence of a dedicated counter whose transition effects and reachability condition ensure the implicit zero-tests were indeed performed correctly.

It has been shown that we may assume that the hypergraph is  $\ell$ -partite for the  $k$ -Hyperclique Hypothesis [36, Theorem 3.1]. Thus, we may assume that the vertices can be partitioned into  $\ell$  disjoint subsets  $V = V_1 \cup \dots \cup V_\ell$  and all hyperedges contain three vertices from distinct subsets  $\{u, v, w\} \in V_i \times V_j \times V_k$  for some  $1 \leq i < j < k \leq \ell$ .

► **Lemma 5.9.** *Let  $d \geq 1$  be a fixed integer. Given a  $4d$ -partite 3-uniform hypergraph  $H = (V_1 \cup \dots \cup V_{4d}, E)$  with  $n$  vertices, there exists a unary  $(\mathcal{O}(n^{4+o(1)}), d + 1)$ -VASS with  $\mathcal{O}(d^3)$  zero-tests  $\mathcal{T}$  such that there is a  $4d$ -hyperclique in  $H$  if and only if there is a run from  $q_I(\mathbf{0})$  to  $q_F(\mathbf{v})$ , for some  $\mathbf{v} \geq \mathbf{0}$ , in  $\mathcal{T}$ . Moreover,  $\mathcal{T}$  can be constructed in  $\text{poly}(d) \cdot n^{4+o(1)}$ -time.*

► **Lemma 5.10** ([16, Lemma 10]). *Let  $\rho$  be a run in a  $(d + 2)$ -VASS such that  $q_I(\mathbf{0}) \xrightarrow{\rho} q_F(\mathbf{0})$ . Further, let  $q_0(\mathbf{v}_0), q_1(\mathbf{v}_1), \dots, q_r(\mathbf{v}_r)$  be some distinguished configurations observed along the run  $\rho$  with  $q_0(\mathbf{v}_0) = q_I(\mathbf{0})$  and  $q_r(\mathbf{v}_r) = q_F(\mathbf{0})$  and let  $\rho_j$  be the segment of  $\rho$  that is between  $q_{j-1}(\mathbf{v}_{j-1})$  and  $q_j(\mathbf{v}_j)$ , so  $\rho$  can be described as*

$$q_I(\mathbf{0}) = q_0(\mathbf{v}_1) \xrightarrow{\rho_1} q_1(\mathbf{v}_1) \rightarrow \dots \rightarrow q_{r-1}(\mathbf{v}_{r-1}) \xrightarrow{\rho_r} q_r(\mathbf{v}_r) = q_F(\mathbf{0}).$$

## 131:16 Coverability in VASS Revisited

Let  $S_1, \dots, S_d, S_{d+1} \subseteq \{0, 1, \dots, r\}$  be the sets of indices of the distinguished configurations where zero-tests could be performed on counters  $x_1, \dots, x_d, x_{d+1}$ , respectively. Let  $t_{j,i} = |\{s \geq j : s \in S_i\}|$  be the number of zero-test for the counter  $x_i$  in the remainder of the run  $\rho_{j+1} \cdots \rho_r$ . Given that  $\mathbf{v}_0 = \mathbf{0}$  and  $\mathbf{v}_r = \mathbf{0}$ , if

$$\text{eff}(\rho_j)[d+2] = \sum_{i=1}^{d+1} t_{j,i} \cdot \text{eff}(\rho_j)[i], \quad (1)$$

then for every  $i \in \{1, \dots, d, d+1\}$  and  $j \in S_i$ , we know that  $\mathbf{v}_j[i] = 0$ .

With Lemma 5.10 in hand, we can ensure that the  $\mathcal{O}(d^3)$  zero-tests performed by  $\mathcal{T}$ , from Lemma 5.9, are executed correctly. We conclude this section with a proof of Theorem 5.8.

**Proof of Theorem 5.8.** Consider the reduction, presented in Lemma 5.9, from finding a  $4d$ -hyperclique in a  $4d$ -partite 3-uniform hypergraph  $H$  to reachability in  $(\mathcal{O}(n^{4+o(1)}), d+1)$ -VASS with  $\mathcal{O}(d^3)$  zero-tests. Now, given Lemma 5.10, we will add a controlling counter to  $\mathcal{T}$  so that the zero-tests on the  $d+1$  counters  $x_1, \dots, x_d, y$  are instead performed implicitly. So we introduce another counter  $z$  that receives updates on transitions, consistent with Equation 1, whenever any of the other counters are updated. Note that counters  $y$  and  $z$ , for the sake of a succinct and consistent description, are respectively referred to as counters  $x_{d+1}$  and  $x_{d+2}$  in the statement of Lemma 5.10. Moreover, notice that the maximum value of  $z$  is bounded by  $\text{poly}(d) \cdot \left(\sum_{i=1}^{d+1} x_i\right) \in \text{poly}(d) \cdot n^{4+o(1)}$ .

Therefore, we have constructed a unary  $(\text{poly}(d) \cdot n^{4+o(1)}, d+2)$ -VASS  $\mathcal{V}$  with the property that there  $H$  contains a  $4d$ -hyperclique if and only if there is a run from  $q'_I(\mathbf{0})$  to  $q'_F(\mathbf{0})$  in  $\mathcal{V}$ . Such a  $(\text{poly}(d) \cdot n^{4+o(1)}, d+2)$ -VASS  $\mathcal{V}$  has size  $\mathcal{O}(t \cdot |\mathcal{T}|)$  where  $t \in \text{poly}(d)$  is the number of zero-tests performed on the run from  $q_I(\mathbf{0})$  to  $q_F(\mathbf{0})$  in  $\mathcal{T}$ . Moreover,  $\mathcal{V}$  can be constructed in  $\text{poly}(d) \cdot n^{4+o(1)}$  time. Hence, if reachability in  $(\mathcal{O}(n), d+2)$ -VASS of size  $n$  can be solved faster than  $n^{d-o(1)}$ , then one can find a  $4d$ -hyperclique in a 3-uniform hypergraph faster than  $n^{4d-o(1)}$ , contradicting Hypothesis 5.7.  $\blacktriangleleft$

## 6 Conclusion

**Summary.** In this paper, we have revisited a classical problem of coverability in  $d$ -VASS. We have closed the gap left by Rosier and Yen [46] on the length of runs witnessing instances of coverability in  $d$ -VASS. We have lowered the upper bound of  $n^{2^{\mathcal{O}(d \log d)}}$ , from Rackoff's technique [45], to  $n^{2^{\mathcal{O}(d)}}$  (Theorem 3.3), matching the  $n^{2^{\Omega(d)}}$  lower bound from Lipton's construction [37]. This accordingly closes the gap on the exact space required for the coverability problem and yields a deterministic  $n^{2^{\mathcal{O}(d)}}$ -time algorithm for coverability in  $d$ -VASS (Corollary 3.4). We complement this with a matching lower bound conditional on ETH; there does not exist a deterministic  $n^{2^{\mathcal{O}(d)}}$ -time algorithm for coverability (Theorem 4.2). By and large, this settles the exact space and time complexity of coverability in VASS.

In addition, we study linearly-bounded unary  $d$ -VASS. Here, coverability and reachability are equivalent and the trivial exhaustive search  $\mathcal{O}(n^{d+1})$  algorithm is near-optimal. We prove that reachability in linearly-bounded 1-VASS requires  $n^{2-o(1)}$ -time under the  $k$ -cycle hypothesis (Theorem 5.4), matching the trivial upper bound. We further prove that reachability in linearly-bounded  $(d+2)$ -VASS requires  $n^{d-o(1)}$ -time under the 3-uniform hyperclique hypothesis (Theorem 5.8).

**Open Problems.** The *boundedness problem*, a problem closely related to coverability, asks whether, from a given initial configuration, the set of all reachable configurations is finite. This problem was also studied by Lipton then Rackoff and is EXPSPACE-complete [37, 45].

Boundedness was further analysed by Rosier and Yen [46, Theorem 2.1] and the same gap also exists for the exact space required. We leave the same improvement, to eliminate the same twice-exponentiated  $\log(d)$  factor, as an open problem.

Our lower bounds for the time complexity of coverability and reachability in linearly-bounded unary  $d$ -VASS, for  $d \geq 2$ , leave a gap of up to  $n^{3+o(1)}$ , see Table 1. We leave it as an open problem to either improve upon the upper bound  $\mathcal{O}(n^{d+1})$  given by the trivial algorithm, or to raise our conditional lower bounds.

---

## References

- 1 Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Inf. Comput.*, 160(1-2):109–127, 2000. doi:10.1006/inco.1999.2843.
- 2 Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in P. *Annals of Mathematics*, pages 781–793, 2004.
- 3 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.
- 4 Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997. doi:10.1007/BF02523189.
- 5 Bertie Ancona, Monika Henzinger, Liam Roditty, Virginia Vassilevska Williams, and Nicole Wein. Algorithms and hardness for diameter in dynamic graphs. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 13:1–13:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.ICALP.2019.13.
- 6 Michael Blondin, Matthias Englert, Alain Finkel, Stefan Göller, Christoph Haase, Ranko Lazić, 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.
- 7 Michael Blondin, Alain Finkel, Christoph Haase, and Serge Haddad. Approaching the Coverability Problem Continuously. In Marsha Chechik and Jean-François Raskin, editors, *Tools and Algorithms for the Construction and Analysis of Systems – 22nd International Conference, TACAS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9636 of *Lecture Notes in Computer Science*, pages 480–496. Springer, 2016. doi:10.1007/978-3-662-49674-9\_28.
- 8 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.
- 9 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.
- 10 Laura Bozzelli and Pierre Ganty. Complexity analysis of the backward coverability algorithm for VASS. In Giorgio Delzanno and Igor Potapov, editors, *Reachability Problems – 5th International Workshop, RP 2011, Genoa, Italy, September 28-30, 2011. Proceedings*, volume 6945 of *Lecture Notes in Computer Science*, pages 96–109. Springer, 2011. doi:10.1007/978-3-642-24288-5\_10.

- 11 Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.*, 201(2):216–231, 2005. doi:10.1016/j.ic.2005.05.001.
- 12 Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006. doi:10.1016/j.jcss.2006.04.007.
- 13 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.
- 14 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 15 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.
- 16 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.
- 17 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.
- 18 Mina Dalirrooyfard, Ce Jin, Virginia Vassilevska Williams, and Nicole Wein. Approximation Algorithms and Hardness for n-Pairs Shortest Paths and All-Nodes Shortest Cycles. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 – November 3, 2022*, pages 290–300. IEEE, 2022. doi:10.1109/FOCS54457.2022.00034.
- 19 Mina Dalirrooyfard, Thuy Duong Vuong, and Virginia Vassilevska Williams. Graph pattern detection: Hardness for all induced patterns and faster noninduced cycles. *SIAM J. Comput.*, 50(5):1627–1662, 2021. doi:10.1137/20M1335054.
- 20 Javier Esparza. Decidability and Complexity of Petri Net Problems – An Introduction. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996*, volume 1491 of *Lecture Notes in Computer Science*, pages 374–428. Springer, 1996. doi:10.1007/3-540-65306-6\_20.
- 21 Javier Esparza, Ruslán Ledesma-Garza, Rupak Majumdar, Philipp J. Meyer, and Filip Nikić. 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.
- 22 John Fearnley and Marcin Jurdziński. 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.

- 23 Diego Figueira, Santiago Figueira, Sylvain Schmitz, and Philippe Schnoebelen. Ackermannian and Primitive-Recursive Bounds with Dickson’s Lemma. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 269–278. IEEE Computer Society, 2011. doi:10.1109/LICS.2011.39.
- 24 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.
- 25 Steven M. German and A. Prasad Sistla. Reasoning about systems with many processes. *J. ACM*, 39(3):675–735, 1992. doi:10.1145/146637.146681.
- 26 Christoph Haase, Stephan Kreutzer, Joël Ouaknine, and James Worrell. Reachability in Succinct and Parametric One-Counter Automata. In Mario Bravetti and Gianluigi Zavattaro, editors, *CONCUR 2009 – Concurrency Theory, 20th International Conference, CONCUR 2009, Bologna, Italy, September 1-4, 2009. Proceedings*, volume 5710 of *Lecture Notes in Computer Science*, pages 369–383. Springer, 2009. doi:10.1007/978-3-642-04081-8\_25.
- 27 Christoph Haase, Joël Ouaknine, and James Worrell. On the relationship between reachability problems in timed and counter automata. In Alain Finkel, Jérôme Leroux, and Igor Potapov, editors, *Reachability Problems – 6th International Workshop, RP 2012, Bordeaux, France, September 17-19, 2012. Proceedings*, volume 7550 of *Lecture Notes in Computer Science*, pages 54–65. Springer, 2012. doi:10.1007/978-3-642-33512-9\_6.
- 28 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.
- 29 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.
- 30 Ulla Koppenhagen and Ernst W. Mayr. Optimal algorithms for the coverability, the subword, the containment, and the equivalence problems for commutative semigroups. *Inf. Comput.*, 158(2):98–124, 2000. doi:10.1006/inco.1999.2812.
- 31 Marvin Künnemann, Filip Mazowiecki, Lia Schütze, Henry Sinclair-Banks, and Karol Węgrzycki. Coverability in VASS Revisited: Improving Rackoff’s Bound to Obtain Conditional Optimality, 2023. arXiv:2305.01581.
- 32 Ranko Lazic and Sylvain Schmitz. The ideal view on Rackoff’s coverability technique. *Inf. Comput.*, 277:104582, 2021. doi:10.1016/j.ic.2020.104582.
- 33 Jérôme Leroux. Vector addition system reversible reachability problem. *Log. Methods Comput. Sci.*, 9(1), 2013. doi:10.2168/LMCS-9(1:5)2013.
- 34 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.
- 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 Andrea Lincoln, Virginia Vassilevska Williams, and R. Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1236–1252. SIAM, 2018. doi:10.1137/1.9781611975031.80.
- 37 Richard Lipton. The Reachability Problem Requires Exponential Space. *Department of Computer Science. Yale University*, 62, 1976.
- 38 Daniel Lokshantov, Dániel Marx, and Saket Saurabh. Lower bounds based on the Exponential Time Hypothesis. *Bulletin of EATCS*, 3(105), 2013.
- 39 Ernst W. Mayr. An Algorithm for the General Petri Net Reachability Problem. *SIAM J. Comput.*, 13(3):441–460, 1984. doi:10.1137/0213029.

- 40 Ernst W. Mayr and Albert R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in Mathematics*, 46(3):305–329, 1982. doi:10.1016/0001-8708(82)90048-2.
- 41 Filip Mazowiecki and Michał Pilipczuk. Reachability for Bounded Branching VASS. In Wan J. Fokkink and Rob van Glabbeek, editors, *30th International Conference on Concurrency Theory, CONCUR 2019, August 27-30, 2019, Amsterdam, the Netherlands*, volume 140 of *LIPICs*, pages 28:1–28:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.CONCUR.2019.28.
- 42 Filip Mazowiecki, Henry Sinclair-Banks, and Karol Węgrzycki. Coverability in 2-VASS with One Unary Counter is in NP. In Orna Kupferman and Paweł Sobociński, editors, *Foundations of Software Science and Computation Structures, 2023*. doi:10.1007/978-3-031-30829-1\_10.
- 43 Marvin L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, Inc., 1967.
- 44 Jaroslav Nešetřil and Svatopluk Poljak. On the complexity of the subgraph problem. *Commentationes Mathematicae Universitatis Carolinae*, 26(2):415–419, 1985.
- 45 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.
- 46 Louis E. Rosier and Hsu-Chun Yen. A Multiparameter Analysis of the Boundedness Problem for Vector Addition Systems. *J. Comput. Syst. Sci.*, 32(1):105–135, 1986. doi:10.1016/0022-0000(86)90006-1.
- 47 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>, doi:10.1145/2893582.2893585.
- 48 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.
- 49 Wil M. P. van der Aalst. Verification of Workflow Nets. In Pierre Azéma and Gianfranco Balbo, editors, *Application and Theory of Petri Nets 1997, 18th International Conference, ICATPN '97, Toulouse, France, June 23-27, 1997, Proceedings*, volume 1248 of *Lecture Notes in Computer Science*, pages 407–426. Springer, 1997. doi:10.1007/3-540-63139-9\_48.
- 50 Raphael Yuster and Uri Zwick. Detecting short directed cycles using rectangular matrix multiplication and dynamic programming. In J. Ian Munro, editor, *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, pages 254–260. SIAM, 2004. URL: <http://dl.acm.org/citation.cfm?id=982792.982828>.
- 51 Don Zagier. Newman’s short proof of the prime number theorem. *The American mathematical monthly*, 104(8):705–708, 1997.