# Universality Problem for Unambiguous VASS

Wojciech Czerwiński, Diego Figueira, Piotr Hofman

# Universality Problem for Unambiguous VASS

**Wojciech Czerwiński**
University of Warsaw
wczerwin@mimuw.edu.pl

**Diego Figueira**
Univ. Bordeaux, CNRS, LaBRI
diego.figueira@labri.fr

**Piotr Hofman**
University of Warsaw
piotrek.hofman@gmail.com

## Abstract

We study languages of unambiguous VASS, that is, Vector Addition Systems with States, whose transitions read letters from a finite alphabet, and whose acceptance condition is defined by a set of final states (*i.e.*, the coverability language). We show that the problem of universality for unambiguous VASS is ExpSpace-complete, in sheer contrast to Ackermann-completeness for arbitrary VASS, even in dimension 1. When the dimension $d \in \mathbb{N}$ is fixed, the universality problem is PSpace-complete if $d \geq 2$, and coNP-hard for 1-dimensional VASSes (also known as *One Counter Nets*).

**2012 ACM Subject Classification** Theory of computation → Formal languages and automata theory

**Keywords and phrases** unambiguity, vector addition systems, universality problems

**Funding** *Diego Figueira*: ANR BraVAS (ANR-17-CE40-0028); ANR DéLTA (ANR-16-CE40-0007).

**Acknowledgements** We thank Lorenzo Clemente for leading us to the $NC^2$ membership for UFA universality problem.

# 1 Introduction

Determinism is a central notion of computational models, it ensures that there is one way to proceed for every input. It often enables constructions which would not be possible without it and allows for efficient algorithms. While the relation between deterministic vs non-deterministic models is extensively studied, there exists also a less understood middle ground of *unambiguous* systems. In the case of models accepting word languages, a model is said to be *unambiguous* if for every word in its language, there is exactly one accepting run, which is a much weaker restriction than determinism. Unambiguity, although featuring non-determinism, often causes some problems to be computationally easier. As a prominent example, the universality problem for finite automata (*i.e.*, whether all words over the alphabet are accepted by the automaton), which is PSpace-complete in general, is known to be in PTime in the unambiguous case [12] and even in $NC^2$ [13]. While the study of unambiguous models of computation has lately attracted some attention, in some settings it remains, by and large, an unexplored area.

In particular, there has been considerable volume of research on unambiguous *finite automata* (see [1] for a nice overview). One way to design a polynomial time algorithm for the universality problem on finite automata is to show that the shortest word which is not in the language, if any, is of at most linear length. Then, by counting the number of linear length runs one may answer the problem. The existence of a linear counterexamples for universality and its PTime algorithm, led to the conjecture, formulated by Colcombet [1], that for every unambiguous finite automaton (UFA) there exists another UFA of polynomial size accepting the complement of its language. This conjecture was later shown false by Raskin [10]. As it turns out, there is a family of UFA such that for accepting the complement of UFA with $n$ states even nondeterministic finite automaton (NFA) needs a super-polynomial number of states —at least $\Theta(n^{\log\log\log n})$. The universality problem for UFA is actually known to be not only in PTime, but even in $NC^2$ [13], the class of problems solvable by uniform families of circuits width $\mathcal{O}(\log^2 n)$ depth and binary branching. The work [13] in fact solves the more general problem of *path equivalence* for two NFA: is the number of accepting runs on $w$ the same for both automata, for every word $w$? However, to the best of our knowledge the best known lower bound for the problem is NL-hardness, so the exact complexity of universality problem for UFA is still open even in the simplest possible setting of finite automata.

There was also research about the universality problem and related ones for unambiguous register automata. In [7] authors have shown that the containment problem for unambiguous register automata is in 2ExpSpace and even in ExpSpace if the number of register is fixed, which implies similar upper bounds for the universality problem. Without the unambiguity assumption, even the universality problem (and even with just one register) can be easily shown undecidable [8].

It is not by accident that existing research focuses on universality, equivalence and containment of languages of unambiguous systems, and that there are efficient algorithms for these problems under the assumption of unambiguity. Unambiguity speaks about the language of a system, so it is natural to hope that problems related to the language of the systems may become more tractable. But for the most natural problem concerning the language, i.e., for the emptiness problem one cannot hope for an improvement. This is because for most of the systems one can relabel transitions giving each one a unique label. Then the system becomes deterministic and in consequence unambiguous. The language changes, but it is empty iff the original language was empty, which intuitively explains why the emptiness problem shouldn't be any easier for unambiguous systems compared

to general non-deterministic ones. One the other hand, it is more reasonable to expect that the universality problem might be easier, since both the universality problem and the unambiguity property are universal properties of the form "*For all words, [...]*".

### Our contribution

The foremost goal of this paper is to push the understanding of unambiguity further. We focus on the universality problem, which is arguably the most natural first step, that may open the way for further studies on the equivalence, co-finiteness, containment and other problems for languages. The universality problem was studied for finite automata and register automata under the unambiguity assumption. In our opinion, the most interesting yet unsolved cases in which one can expect a progress assuming unambiguity are One Counter Nets (called also 1-dimensional VASS here) and its generalization Vector Addition Systems with States (VASS).

The universality checking for VASS with state acceptance is known to be decidable by the use of well quasi-order techniques [4] (the paper shows decidability of trace universality, but language universality can be reduced to that problem). However the problem is also known to be Ackermann-complete even for 1-dimensional VASS [3], so hardly tractable. For deterministic VASS it is quite easy to show that universality problem can be decided in PTime. Therefore, it is natural to hope for an improvement under the unambiguity restriction.

Our main contribution is ExpSpace membership of the universality problem for unambiguous VASS. The believe that it is the most interesting result and it was as well the most challenging problem and technically involved solution. We actually have shown that this problem is ExpSpace-complete. For the completeness of the picture we have also analyzed the complexity of the problem for $d$-dimensional VASS for fixed $d \in \mathbb{N}$. We have shown that the problem is PSpace-complete for every $d \geq 2$. For $d = 1$ we have shown coNP-hardness, although we do not have the matching upper bound, we conjecture that it is coNP-complete. We additionally consider the variant of the problem in which the numbers in the input are encoded in unary. Finally, we study also the problem of unambiguity checking (*i.e.*, given a VASS, is it unambiguous?). All our results are listed in Section 3.

## 2 Preliminaries

We use the letter $\Sigma$ to denote a finite alphabet, $\mathbb{Z}$ to denote the set of all integers, and $\mathbb{N}$ the set of non-negative integers. We use $\varepsilon$ to denote the empty string, and $\Sigma_\varepsilon$ to denote $\Sigma \cup \{\varepsilon\}$. We use $A \subseteq_{fin} B$ to denote that $A$ is a finite subset of $B$, and $\wp_{fin}(A)$ to denote the set of all finite subsets of $A$. We use $\bar{u}, \bar{v}, \bar{w}, \ldots$ to denote vectors of numbers, and we use $\bar{0}$ to denote the all-0 vector and $\bar{1}$ to denote the all-1 vector. We use $[i, j]$ for $i, j \in \mathbb{N}$, $i \leq j$ to denote the set $\{i, i+1, \ldots, j-1, j\}$. For a vector $\bar{u} \in \mathbb{Z}^d$ and $i \in [1, d]$ we denote by $\bar{u}[i]$ the $i$-th coordinate of $\bar{u}$. For a word $w \in \Sigma^*$ and $i \in \mathbb{N}$ we denote by $w[i]$ the $i$-th letter of $w$. For $\bar{u}, \bar{v} \in \mathbb{Z}^d$ we write $\bar{u} \preceq \bar{v}$ if for all $i \in [1, d]$ we have $\bar{u}[i] \leq \bar{v}[i]$. We define minimum of $\bar{u}$ and $\bar{v}$ as $\min(\bar{u}, \bar{v})[i] = \min(\bar{u}[i], \bar{v}[i])$ for any $i \in [1, d]$.

We consider a Vector Addition Systems with States (VASS) of dimension $d \in \mathbb{N}$ as a tuple $\mathcal{A} = (\Sigma, d, Q, q_0, \delta, F)$ where $\Sigma$ is a finite alphabet, $Q$ is a finite state space, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and $\delta \subseteq_{fin} Q \times \Sigma_\varepsilon \times \mathbb{Z}^d \times Q$ is the set of transitions. We often write transition $(p, a, v, q)$ as $p \xrightarrow{a;v} q$. We will henceforth write $d$-**VASS** to denote a VASS of fixed dimension $d$. A **configuration** of $\mathcal{A}$ is a pair of a state $q \in Q$ and a vector $\bar{u} \in \mathbb{N}^d$, that we usually note $q(\bar{u})$. If $c$ is a configuration,

write $c[i]$ to denote the $i$-th coordinate of the vector it contains. A **run** of $\mathcal{A}$ from a configuration $q(\bar{u})$ to a configuration $q'(\bar{v})$ reading the word $w \in \Sigma^*$ is a sequence of transitions $(r_1, \alpha_1, \bar{v}_n, r'_1) \cdots (r_n, \alpha_n, \bar{v}_n, r'_n) \in \delta^*$ such that: (i) $r_1 = q$ and $r'_n = q'$, (ii) $r'_i = r_{i+1}$ for every $1 \leq i < n$; (iii) $w = \alpha_1 \cdots \alpha_n$; (iv) $\bar{u} + \sum_{i \leq j} \bar{v}_i \in \mathbb{N}^k$ for every $1 \leq j \leq n$; and (v) $\bar{v} = \bar{u} + \sum_{i \leq n} \bar{v}_i$. If we further have $q' \in F$, we say that such run is **accepting**. We henceforth say that a configuration $c$ is **reachable** from a configuration $c'$ if there is a run from $c'$ to $c$. The **effect** of a transition $(r, \alpha, \bar{v}, r')$ is the vector $\bar{v} \in \mathbb{Z}^d$, the **effect of a run** is the sum of effects of the transitions therein. The **norm** of a VASS $\mathcal{A}$ is the maximal absolute value of a number occurring in its transition, and we denote it by $|\mathcal{A}|$. The **language** of a configuration $c$ in $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A}, c)$, is the set of all $w \in \Sigma^*$ with an accepting run from $c$. We call $q_0(\bar{0})$ the **initial configuration** where $q_0$ is the initial state. If $c$ is the initial configuration then we say just language of $\mathcal{A}$ and write $\mathcal{L}(\mathcal{A})$ instead of $\mathcal{L}(\mathcal{A}, c)$. A VASS $\mathcal{A}$ is **unambiguous** if for every $w \in \Sigma^*$ there is no more than one accepting run starting from the initial configuration and reading $w$. The **unambiguity checking problem** for VASS is the problem of, given a VASS $\mathcal{A}$, decide whether it is unambiguous. An automaton over $\Sigma$ (finite automaton or VASS) is **universal** if it accepts the language $\Sigma^*$. The **universality problem** for VASS is the problem of, given a VASS $\mathcal{A}$, decide whether it is universal. We will henceforth assume that the numbers contained in the transitions of VASSes are always encoded in binary if not explicitly indicated otherwise.

Let us recall now the main result of the Rackoff construction [9]. Let us denote $A_{M,d,n} = (2n^2(M+1)^2)^{(4d)^{d-1}}$. We present here an adaptation of the Rackoff argument with an explicit bound on the length of an accepting run, its proof can be found in the Appendix.

▶ **Proposition 1** (Adaptation of the Rackoff construction)**.** *If a language of a $d$-VASS with norm $M$ and $n$ states is nonempty then there exists an accepting run of length at most $A_{M,d,n}$.*

The language emptiness problem for VASS (*i.e.*, given a VASS, does it accept at least one word?) is, basically, equivalent to the coverability problem, which is known to be ExpSpace-complete as shown by the lower bound of Lipton [6] and the upper-bound of Rackoff [9]. The coverability problem is the problem of, given a VASS $\mathcal{A}$ and two configurations $c_1, c_2$, whether there is a run from $c_1$ to some some configuration $c'_2$ such that $c'_2 \succeq c_2$. In our setting, this result can be restated as the language emptiness problem for VASS being ExpSpace-complete, even when all transitions are $\varepsilon$-transitions, and hence the language is either $\{\}$ or $\{\varepsilon\}$. What is more, the construction of Lipton is unambiguous: if there is an accepting run, there is exactly one. This is formalized in the next lemma. Let us denote by $\varepsilon$-VASS, a VASS whose every transition reads $\varepsilon$ (and thus the alphabet is not important here).

▶ **Lemma 2** (consequence of [6, 9])**.** *The problem of whether an unambiguous $\varepsilon$-VASS has an empty language is* ExpSpace*-complete.*

## 3 Results

We summarize all our results in the next two theorems. Detailed proofs will come in the sections that follow.

▶ **Theorem 3.** *The universality problem for*

*(i) VASS is* ExpSpace*-complete, both with binary and unary encodings;*
*(ii) $d$-VASS with unary encoding is in* $\mathrm{NC}^2$ *and* NL*-hard, for every $d \geq 1$;*

*(iii) d-VASS with binary encoding is* PSPACE*-complete, for every* $d \geq 2$*;*
*(iv)* 1*-VASS (One Counter Net) with binary encoding is* coNP*-hard.*

▶ **Theorem 4.** *The unambiguity checking problem for*

*(i) VASS is* EXPSPACE*-complete, both with binary and unary encodings;*
*(ii) d-VASS with unary encoding is* NL*-complete, for every* $d \geq 1$*;*
*(iii) d-VASS with binary encoding is* PSPACE*-complete, for every* $d \geq 2$*;*
*(iv)* 1*-VASS with binary encoding is* coNP*-hard.*

The main technical contribution lies in the EXPSPACE bounds on the universality problem in Theorem 3(i). The upper bound will need some insights on the structure of accepting runs in unambiguous VASS which happen to have a universal language. The remaining upper bounds will follow easily from this one. The EXPSPACE, PSPACE, and coNP lower bounds of items (i), (iii), and (iv) are also of interest, as they reveal different ways in which unambiguity can encode non-trivial properties. All the remaining results of Theorems 3 and 4 are either easy, or follow from simple adaptations of the four results just mentioned.

It is interesting to observe that complexity results on universality seem to coincide with the complexity of emptiness for the non-deterministic version of the considered classes. Notice also that closing the 'gap' between $\mathrm{NC}^2$ and NL in Theorem 3(ii) would imply in particular solving the corresponding problem for UFA, which is an open question.

### Organization

We will prove Theorem 3 in Section 4 and Theorem 4 in Section 5. Each of these sections is divided into an "upper bounds" and "lower bounds" subsections. For reference, the upper and lower bounds of item (i) of Theorem 3 are shown in Propositions 5 and 17 respectively; item (ii) in Propositions 16 and 21; item (iii) in Propositions 15 and 19; and item (iv) in Proposition 20. The upper and lower bounds of item (i) of Theorem 4 are shown in Propositions 22 and 23 respectively; item (ii) in Propositions 22 and 24; item (iii) in Propositions 22 and 25; and item (iv) in Proposition 26.

## 4 Testing for Universality

In this section we will prove Theorem 3. Most of the section will be dedicated to proving the EXPSPACE upper bound of item (i).

### 4.1 Upper bounds

▶ **Proposition 5** (Theorem 3(i) upper bound)**.** *The universality problem for unambiguous VASSes is in* EXPSPACE*.*

The proof strategy is as follows. First, we define an abstraction of a configuration, called an $N$-profile, for $N \in \mathbb{N}$, which is the result of replacing every number bigger or equal to $N$ with $N$ in a configuration. The intuition is that any number bigger or equal $N$ is so big that we can disregard its exact value. We next show that in certain circumstances, for any unambiguous $d$-VASS $V$ with $n$ states two configurations having equal $f(|V|, d, n)$-profile have also the same language, where $f$ is some fixed doubly-exponential function. This fact allows us to construct an unambiguous finite automaton $\mathcal{A}$ of doubly-exponential size, whose every state corresponds to one $f(|V|, d, n)$-profile, and such that $\mathcal{A}$ is universal if, and only

if, $V$ is universal. As universality of UFAs is in PolyLogSpace, this gives us an ExpSpace algorithm for checking universality.

For any number $N \in \mathbb{N}$, the **$N$-profile** of a configuration $(q, \bar{v}) \in Q \times \mathbb{N}^d$ is the pair $(q, \min(\bar{v}, N \cdot \bar{1}))$. Let $B_{M,d,n} = M \cdot A_{M,2d,2n^2}$, and let $C_{M,d,n} = M \cdot (B_{M,d,n} + 1)^d$.

We start with a useful lemma which bounds the length of runs witnessing ambiguity.

▶ **Lemma 6.** *Let $V$ be a $d$-VASS with norm $M$ and $n$ states. If $V$ is ambiguous then there exists two different runs accepting the same word of length at most $A_{M,2d,2n^2}$ each.*

**Proof.** Consider the following $2d$-VASS $V'$, which accepts exactly these words, which have at least two different accepting runs from initial configuration of $V$. The VASS $V'$ guesses two different runs of $V$ and simulates them, it is quite similar to a synchronized product of $V$ with itself. In its $2d$ counters $V'$ keeps counter valuations of two configurations of $V$ of the simulated runs. State of $V'$ is a pair of states of $V$ together with one bit of information indicating whether the two simulated runs have already differed or they are the same till that moment. VASS $V'$ accepts if states of both simulated runs are accepting and the bit indicates that they have differed (even if now they are in the same state). It is easy to see that $V'$ indeed accepts words, which have two different accepting runs in $V$. Therefore if $V$ is ambiguous then $\mathcal{L}(V')$ is nonempty. Notice that norm of $V'$ is bounded by $M$, as norm of $V$ is. Therefore by Proposition 1 if $\mathcal{L}(V')$ is nonempty then there is an accepting run of $V'$ of length at most $A_{M,2d,2n^2}$. Notice that existence of such a run implies existence of two different runs of $V$ over the same word, which additionally also have length bounded by $A_{M,2d,2n^2}$. This finishes the proof. ◀

We state two basic properties of VASS which will be useful throughout.

▷ Claim 7. For any two configurations $c$ and $c'$ of a VASS $V$ with equal $(|V| \cdot N)$-profile, if $\rho$ is an accepting run from $c$ of length at most $N$ then $\rho$ is also accepting from $c'$.

▷ Claim 8 (language monotonicity). If $q(\bar{u})$ and $q(\bar{v})$ are two configurations of a VASS $V$ with $\bar{u} \preceq \bar{v}$ then $\mathcal{L}(V, q(\bar{u})) \subseteq \mathcal{L}(V, q(\bar{v}))$.

The following is the key lemma which will enable the improved complexity for the universality problem.

▶ **Lemma 9.** *Let $V$ be a universal, unambiguous $d$-VASS with $n$ states. Then, any two configurations with equal $B_{|V|,d,n}$-profile reachable from the initial configuration have the same set of accepting runs (in particular, they have the same language).*

**Proof.** By means of contradiction, let $c_1, c_2 \in Q \times \mathbb{N}^d$ be two configurations reachable from the initial configuration $c_{\text{init}}$ with the same $B_{|V|,d,n}$-profile, but different sets of accepting runs. Let $\rho$ be an accepting run from $c_1$ but not from $c_2$, reading the word $w$.

Let $c_{\text{init}} \xrightarrow{u} c_2$. The word $uw$ is accepted by $V$ since it is universal, so there must be a configuration $c_2'$ such that $c_{\text{init}} \xrightarrow{u} c_2'$ and $w \in \mathcal{L}(V, c_2')$. Therefore $w$ is accepted both from configuration $c_1$ with the run $\rho$ and from configuration $c_2'$ with some accepting run $\hat{\rho}$. There are two cases to consider: either (i) $c_2 \neq c_2'$, or (ii) $c_2 = c_2'$ and $\hat{\rho} \neq \rho$.

For (i), let us first consider an (ambiguous) VASS $\tilde{V}$, being the result of adding $\varepsilon$-labelled self-loops with effect $\bar{0}$ in every state to $V$. Clearly, for every configuration $c$ we have $\mathcal{L}(V, c) = \mathcal{L}(\tilde{V}, c)$. Let us confider a $2d$-VASS $V'$, which is a synchronized product of $\tilde{V}$ with itself: transitions, initial and accepting states are defined in a natural way. Product is synchronized, so for any $a \in \Sigma_\varepsilon$ there is an $a$-labelled transition in the product $V'$ iff there exist $a$-labelled transitions in the components, both identical with $\tilde{V}$. For two configurations

$c = q(\bar{u})$ and $c' = q'(\bar{u}')$ of $V$ we denote by $\mathcal{L}(V', c, c')$ the language $\mathcal{L}(V', (q, q')(\bar{u}, \bar{u}'))$. Notice that, by construction, $\mathcal{L}(V', c, c')$ is the intersection of $\mathcal{L}(V, c)$ and $\mathcal{L}(V, c')$. Therefore the word $w$ belongs to $\mathcal{L}(V', c_1, c_2')$. By Proposition 1 there exists an accepting run $\rho'$ of $V'$ of length at most $A_{|V|,2d,n^2}$ reading a word $w'$ from $\mathcal{L}(V', c_1, c_2') = \mathcal{L}(V, c_1) \cap \mathcal{L}(V, c_2)$. Consider the projection $\rho_1$ of $\rho'$ onto the first copy of $\tilde{V}$. We know thus that $\rho_1$ is accepting from $c_1$. Further, the absolute value of the effect of $\rho_1$ on every coordinate is at most $|V| \cdot A_{|V|,2d,n^2} \leq |V| \cdot A_{|V|,2d,2n^2} = B_{|V|,d,n}$. Recall that $c_1$ and $c_2$ have the same $B_{|V|,d,n}$-profile, so if $\rho_1$ is accepting from $c_1$ then it is also accepting from $c_2$. Therefore $w' \in \mathcal{L}(V, c_2)$ and $w' \in \mathcal{L}(V, c_2')$, which means that there are two distinct accepting runs over $uw'$ in $V$, contradicting the fact that it is unambiguous.

For (ii), we have that there are two distinct accepting runs for $w$ from $\max(c_1, c_2)$, namely $\rho$ and $\hat{\rho}$. Then, by Lemma 6, there exist two different runs $\rho_1$ and $\rho_2$ from $\max(c_1, c_2)$ of length at most $A_{|V|,2d,2n^2}$ accepting the same word $w'$. Since $c_1$ and $c_2$ have the same $B_{|V|,d,n}$-profile, where $B_{|V|,d,n} = |V| \cdot A_{|V|,2d,2n^2}$, by Claim 7 both $\rho_1$ and $\rho_2$ are accepting from configuration $c_2$, and thus there are two distinct accepting runs over $uw'$ in $V$, contradicting the fact that it is ambiguous. ◀

▶ **Corollary 10.** *If a universal, unambiguous $d$-VASS $V$ with $n$ states contains an accepting run with two configurations $c_1$ and $c_2$ such that $c_1$ occurs before $c_2$, then*

*(i) if $c_1$ and $c_2$ have equal $B_{|V|,d,n}$-profile, then $c_1 \preceq c_2$;*
*(ii) for every $i \in [1, d]$, $c_1[i] - c_2[i] < C_{|V|,d,n}$.*

**Proof.** (i) By means of contradiction, let $c_1$ and $c_2$ be configurations with the same profile such that $c_1 \not\preceq c_2$, meaning that $c_1[i] > c_2[i]$ for some $i$. Let $\rho_1\rho_2\rho_3$ be an accepting run of $V$, such that $\rho_1$ reaches the configuration $c_1$ from the initial configuration, and $\rho_2$ reaches the configuration $c_2$ from configuration $c_1$. Since the effect of $\rho_2$ decrements component $i$, it is easy to see that there is some $k \in \mathbb{N}$ such that $(\rho_2)^k\rho_3$ is an accepting run from $c_1$ but not from $c_2$, contradicting Lemma 9 above.

(ii) Suppose there is a decrement of at least $C_{|V|,d,n}$ at some coordinate $i$. Since $C_{|V|,d,n} = |V| \cdot (B_{|V|,d,n} + 1)^d$ is at least the number of $B_{|V|,d,n}$-profiles times the biggest effect of a transition, this means that at least $k = B_{|V|,d,n}$ distinct configurations $c_1', \ldots, c_k'$ occur in the run between $c_1$ and $c_2$ such that $c_1[i] > c_1'[i] > c_2'[i] > \cdots > c_k'[i]$. Hence, among them $c_1, c_1', \ldots, c_k'$ there must be two equal $B_{|V|,d,n}$-profile configurations, contradicting the item (i) above. ◀

This last statement can be informally understood as follows: if $V$ is universal, then it is still universal if configurations are abstracted by its $C_{|V|,d,n}$-profiles. We now formalize what this means. Let us fix an unambiguous VASS $V$, and let us henceforth write $\omega$ as short for $C_{|V|,d,n}$. For any configuration $c$ let $\lfloor c \rfloor$ denote its $\omega$-profile, that is, $\lfloor q(\bar{u}) \rfloor = q(\min(\bar{u}, \omega \cdot \bar{1}))$. Let $V = (\Sigma, d, Q_V, q_V, \delta_V, F_V)$ be an unambiguous VASS. We construct a finite automaton $\mathcal{A}_V = (\Sigma, Q_\mathcal{A}, q_\mathcal{A}, \delta_A, F_\mathcal{A})$ in the following way:

- the set of states $Q_\mathcal{A}$ is the set of pairs $Q_V \times [0, \omega]^d$;
- the initial state $q_\mathcal{A}$ is $q_V(\bar{0})$;
- the set of final states $F_\mathcal{A}$ consists of all the pairs with first coordinate in $F_V$, namely $F_\mathcal{A} = F_V \times [0, \omega]^d$;
- $\delta_\mathcal{A}$ is the set of all transitions $p(\bar{u}) \xrightarrow{a} q(\lfloor \bar{u} + \bar{v} \rfloor)$ such that $(p, a, \bar{v}, q) \in \delta_V$ and $\bar{u} + \bar{v} \in \mathbb{N}^d$.

We now show that $\mathcal{A}_V$ is unambiguous, and that it is universal iff $V$ is universal.

▶ **Lemma 11.** *For every run $p_1(\bar{u}_1) \xrightarrow{a_1} p_2(\bar{u}_2) \xrightarrow{a_2} \cdots p_n(\bar{u}_n) \xrightarrow{a_n} p_{n+1}(\bar{u}_{n+1})$ of $\mathcal{A}_V$ there is a run $(p_1, a_1, \bar{v}_1, p_2) \cdots (p_n, a_n, \bar{v}_n, p_{n+1})$ of $V$ such that $\bar{v}_1 + \cdots + \bar{v}_i \geq \bar{u}_i$ for every $i \in [1, n]$.*

**Proof.** This can by shown by induction on $n$. It suffices to replace every transition $p_i(\bar{u}_i) \xrightarrow{a_i} p_{i+1}(\bar{u}_{i+1})$ of $\mathcal{A}_V$ by a transition $(p_i, a_i, \bar{v}, p_{i+1}) \in \delta_V$ such that $\bar{u}_{i+1} = \lfloor \bar{u}_i + \bar{v} \rfloor$, which exists by construction. ◀

As a consequence of the previous lemma, if there are two distinct accepting runs for a word $w$ in $\mathcal{A}_V$, then there are also two distinct accepting runs over $w$ in $V$. In other words:

▶ **Lemma 12.** *If $V$ is unambiguous then $\mathcal{A}_V$ is unambiguous.*

▶ **Lemma 13.** *$V$ is universal if, and only if, $\mathcal{A}_V$ is universal.*

**Proof.** First observe that $\mathcal{L}(\mathcal{A}_V) \subseteq \mathcal{L}(V)$ by Lemma 11. Hence, if $\mathcal{A}_V$ is universal, so is $V$. For the converse direction, suppose $V$ is universal, and let us show that $\mathcal{A}_V$ is universal as well. Let $\rho = (q_0, a_1, \bar{v}_1, q_1) \cdots (q_{n-1}, a_n, \bar{v}_n, q_n)$ be the accepting run of $w = a_1 \cdots a_n$ in $V$. Let us consider the run $\rho' = (q_0(\bar{x}_0), a_1, q_1(\bar{x}_1)) \cdots (q_{n-1}(\bar{x}_{n-1}), a_n, q_n(\bar{x}_n))$ of $\mathcal{A}_V$, where $x_0 = \bar{0}$ and for every $i > 0$, $\bar{x}_i = \lfloor \bar{x}_{i-1} + \bar{v}_i \rfloor$. We claim that $\rho'$ is an accepting run on $\mathcal{A}_V$. By means of contradiction, if $\rho'$ is not a run, there must be some $q_i(\bar{x}_i) \xrightarrow{a_{i+1}} q_{i+1}(\bar{x}_{i+1})$ which is not a transition of $\mathcal{A}_V$. This can only happen if in within configurations reachable through $\rho$ there are two configurations $c, c'$ such that $c$ appears before $c'$ and for some $j \in [1, k]$ we have $c[j] - c[j] > \omega$. But this would contradict Corollary 10-(ii). Hence, $\rho'$ is an accepting run and thus $\mathcal{A}_V$ is universal. ◀

Notice that the automaton $\mathcal{A}_V$ has a doubly-exponential number of states. As checking its universality is polynomial-time in its size [1], which is doubly exponential, the problem is in 2ExpTime. In order to design an ExpSpace algorithm we need a bit more work. The following lemma together with Lemma 13 finishes the proof of Proposition 5.

▶ **Lemma 14.** *Checking universality of $\mathcal{A}_V$ is in* ExpSpace.

**Proof.** First notice that the function $V \mapsto \mathcal{A}_V$ can be easily computed in ExpSpace. Indeed, a state of $\mathcal{A}_V$ is described by a pair consisting of a state from $Q_V$ and a vector $\bar{v} \in [0, \omega]^d$, where $\omega = C_{|V|,d,|Q_V|} = |V| \cdot (|V| \cdot (4|Q_V|^4(|V|+1)^2)^{(8d)^{2d-1}} + 1)^d$ is doubly exponential with respect to the description size of $V$, and therefore it can be kept in ExpSpace. It is then possible to iterate through all the possible pairs in $(Q_V, [0, \omega]^d)$ in ExpSpace and for every state output the transitions outgoing from of this state.

By [13] checking universality of UFA without cycles containing only $\varepsilon$-labelled transitions ($\varepsilon$-cycles) is in $\mathrm{NC}^2$, namely in the class of languages recognizable by uniform families of circuits of depth $\mathcal{O}(\log^2(n))$ and binary branching, where $n$ is the number of inputs. A simple procedure which eliminates all the $\varepsilon$-cycles (*i.e.*, all the transitions involved in $\varepsilon$-cycles) can be designed to be in NL. Observe that eliminating $\varepsilon$-cycles does not change the language of unambiguous automata, since no accepting run can contain a transition from an $\varepsilon$-cycle. Since $\mathrm{NL} \subseteq \mathrm{NC}^2$ and $\mathrm{NC}^2$ is closed under composition, we obtain that the universality problem for an arbitrary UFA (possibly with $\varepsilon$-transitions) is in $\mathrm{NC}^2$ as well. It is folklore that $\mathrm{NC}^2$ is included in poly-logarithmic space (actually in the deterministic space $\log^2 n$). Indeed, one can simply simulate a circuit of depth $D$ and binary branching in space $D$.

It is now enough to argue that composition of ExpSpace and PolyLogSpace is included in ExpSpace. This result is also folklore, we sketch here a proof. Any algorithm in the composition of ExpSpace and PolyLogSpace can be seen as PolyLogSpace algorithm inputing the output of ExpSpace machine, potentially of a doubly exponential length. This

doubly exponential output cannot be kept by an ExpSpace algorithm, but one can simulate the composition by a PolyLogSpace algorithm asking ExpSpace oracles for particular letters of its input. Such an algorithm in turn can be simulated easily in ExpSpace. We keep three exponential size pieces of the information: (i) the space of the oracle, (ii) the index of the doubly exponential input being currently transferred to the oracle, and (iii) the space of the poly-logarithmic algorithm, which is poly-logarithmic with respect to the doubly exponential input, hence exponential. Therefore indeed ExpSpace $\circ$ PolyLogSpace $\subseteq$ ExpSpace, which finishes the proof. ◀

Let us now analyze the situation for a fixed dimension $d \in \mathbb{N}$. The number of states of $\mathcal{A}_V$ equals $|Q_V|$ times $|V| \cdot (|V| \cdot (4|Q_V|^4(|V|+1)^2)^{(8d)^{2d-1}} + 1)^d$, which for fixed $d$ is a polynomial depending on $|Q_V|$ and $|V|$. This immediately implies that for $V$ represented in unary the size of $\mathcal{A}_V$ is polynomial, while for $|V|$ represented in binary the size of $\mathcal{A}_V$ is exponential in the size of the input. A proof almost identical to that of Lemma 14, where we substitute ExpSpace with PSpace, yields the following result.

▶ **Proposition 15** (Theorem 3(iii) upper bound). *For every fixed $d \in \mathbb{N}$ the universality problem for binary represented, unambiguous $d$-VASS is in* PSpace.

In a similar way we solve the case of unary represented $d$-VASSes. In this case, we replace ExpSpace with the class of problems solvable in logarithmic space L. We also use the fact that L composed with $\mathrm{NC}^2$ is included in $\mathrm{NC}^2$, which is immediately implied by a trivial closure of $\mathrm{NC}^2$ by composition and inclusion $\mathrm{L} \subseteq \mathrm{NC}^2$. Then we get the following.

▶ **Proposition 16** (Theorem 3(ii) upper bound). *For every fixed $d \in \mathbb{N}$ the universality problem for unary represented, unambiguous $d$-VASS is in* $\mathrm{NC}^2$.

## 4.2 Lower bounds

▶ **Proposition 17** (Theorem 3(i) lower bound). *The universality problem for unambiguous VASS is* ExpSpace*-hard, even on a one letter alphabet.*

**Proof.** We reduce from the problem of whether an unambiguous $\varepsilon$-VASS has an empty language, which is ExpSpace-hard as observed in Lemma 2. Let $\mathcal{A} = (\{a\}, d, Q, q_0, \delta, F)$ be an unambiguous $\varepsilon$-VASS Observe that either there is exactly one accepting run in $\mathcal{A}$ (and hence its language is $\{\varepsilon\}$), or there are no accepting runs (and hence its language is $\emptyset$).

We build an unambiguous VASS $\mathcal{B}$ on a one letter alphabet $\{a\}$ such that $\mathcal{L}(\mathcal{B}) = a^*$ if $\mathcal{L}(\mathcal{A}) = \{\varepsilon\}$ and $\mathcal{L}(\mathcal{B}) = \emptyset$ otherwise. In fact $\mathcal{B}$ can be built as the union of two unambiguous, language-disjoint, VASSes $\mathcal{B}_1, \mathcal{B}_2$ such that

(i) $\mathcal{L}(\mathcal{B}_1) = \{a^\ell : \ell \leq N\}$ if $\mathcal{L}(\mathcal{A}) = \{\varepsilon\}$ or $\mathcal{L}(\mathcal{B}_1) = \emptyset$ otherwise; and
(ii) $\mathcal{L}(\mathcal{B}_2) = \{a^\ell : \ell > N\}$ if $\mathcal{L}(\mathcal{A}) = \{\varepsilon\}$ or $\mathcal{L}(\mathcal{B}_2) = \emptyset$ otherwise.

In fact, $\mathcal{B}_1$ and $\mathcal{B}_2$ can be easily constructed from $\mathcal{A}$.

Let $\mathcal{B}_1 = (\{a\}, d+1, Q \dot\cup \{q_f\}, q_0, \delta_1, \{q_f\})$ be an unambiguous VASS with the same state space and initial state as $\mathcal{A}$ but with of one more dimension. This transition increments along the $\varepsilon$-run, that is, for every transition $(q, \varepsilon, (u_1, \ldots, u_d), q') \in \delta$ we have a transition $(q, \varepsilon, (u_1, \ldots, u_d, 1), q')$ in $\delta_1$. Further, we have the transitions $(q, \varepsilon, \bar{0}, q_f)$ for every $q \in F$, which allows to go from a final state of $\mathcal{A}$ to the final state of $\mathcal{B}_1$ and $(q_f, a, (0, \ldots, 0, -1), q_f)$ which allows to read as many $a$'s as the length of the (sole) accepting run of $\mathcal{A}$.

Let $\mathcal{B}_2 = (\{a\}, d+1, Q \dot\cup \{q_f\}, q_0, \delta_2, \{q_f\})$ be an unambiguous VASS with the same state space and initial state. The set of transitions $\delta_2$ is defined as follows. First, we replace $\varepsilon$ with

$a$ in all transitions from $\delta$. That is, for every transition $(q, \varepsilon, (u_1, \ldots, u_d), q') \in \delta$, we have a transition $(q, a, (u_1, \ldots, u_d, 0), q')$ in $\delta_2$. Second, we add a transition to $q_f$, from which the language $a^*$ is accepted. That is, we add $(q, a, \bar{0}, q_f)$ for every $q \in F$, and $(q_f, a, \bar{0}, q_f)$ to $\delta_2$.

It is easy to see that $\mathcal{B}_1$ and $\mathcal{B}_2$ satisfy, respectively, the properties (i) and (ii), and that they are unambiguous assuming $\mathcal{A}$ is unambiguous. We build $\mathcal{B}$ as the union of $\mathcal{B}_1$ and $\mathcal{B}_2$. We can do this in the obvious way: taking disjoint sets of states for $\mathcal{B}_1$ and $\mathcal{B}_2$, taking the union of the final states sets, adding a new initial state $q_{init}$, and adding two transitions that allow to move from $q_{init}$ to either the initial state of $\mathcal{B}_1$ or the initial state of $\mathcal{B}_2$, by reading $\varepsilon$ and with $\bar{0}$-effect. Since the languages of $\mathcal{B}_1$ and $\mathcal{B}_2$ are a partition of $a^*$, we obtain that $\mathcal{B}$ is ambiguous, and $\mathcal{L}(\mathcal{B}) = a^*$ if, and only if, $\mathcal{L}(\mathcal{A})$ is non-empty. This concludes our proof. ◀

▶ **Corollary 18.** *The co-finiteness problem for unambiguous VASS, that is, whether the complement of its language is finite, is* ExpSpace*-hard.*

The following proposition proves the lower bound of Theorem 3(iii).

▶ **Proposition 19** (Theorem 3(iii) lower bound)**.** *The universality problem for unambiguous 2-VASS is* PSpace*-hard.*

**Proof.** We reduce from the bounded one-counter automata reachability problem, which is known to be PSpace-hard [2, Corollary 10]. This problem can be stated as follows: given a 1-VASS $\mathcal{A} = (\Sigma, 1, Q_\mathcal{A}, q, \delta_\mathcal{A}, F)$, a number $N \in \mathbb{N}$ encoded in binary, and a configuration $p(m)$, is there a run $(r_1, \alpha_1, u_1, r_1') \cdots (r_n, \alpha_n, u_n, r_n')$ from $q(0)$ to $p(m)$ such that $\sum_{i \leq j} u_i \leq N$ for every $1 \leq j \leq n$? The alphabet is not important for this problem, we can consider that every transition reads letter $a$.

Let $\mathcal{A}$, $N$, $p(m)$ be the input of the aforementioned problem. We now construct, in polynomial time, an unambiguous 2-VASS $\mathcal{B} = (\Sigma, 2, Q, q_0, \delta, F)$, such that it is universal, if and only if, the answer to the input is negative —the statement then follows by closure under complement of PSpace. Concretely, the language of $\mathcal{B}$ is essentially the set of all sequences of transitions in $(\delta_\mathcal{A})^*$ which do not contain a run from $q(0)$ to $p(m)$ as prefix. Intuitively the construction of $\mathcal{B}$ from $\mathcal{A}$ can be divided in two steps. First we change $N$-bounded 1-VASS into 2-VASS by simulating configuration $q(i)$ by $q(i, N - i)$. However this 2-VASS might be far from being universal. Therefore we add to it a lot of transitions such that it is almost universal: the only way for a word to be not accepted is to reach a configuration corresponding to $p(m)$.

The construction of $\mathcal{B}$ is as follows. The alphabet $\Sigma$ is defined as $\delta_\mathcal{A} \dot{\cup} \{\star\}$; the state set $Q$ is defined as $Q_\mathcal{A} \dot{\cup} \{\bot, q_f, q_0\}$; and the set of final states $F = Q \setminus \{\bot\}$ being everything beside the *sink state* $\bot$. $\mathcal{B}$ will always keep the invariant that the sum of its two components is equal to $N$ on all configurations with state in $Q_\mathcal{A}$ reachable from the initial configuration $q_0(0, 0)$. Further, the transition graph is as in $\delta_\mathcal{A}$ but labels are used to enforce unambiguity. This is done by initializing the vector in $(0, N)$ as the first thing the automaton does (by adding a new initial state $q_0$ and transition $(q_0, t, (0, N), q)$ from it to the initial state of $\mathcal{A}$), and additionally translating every transition $t = (r, a, h, r') \in \delta_\mathcal{A}$ into $(r, t, (h, -h), r')$. Now we need to assure that the only way to be not accepted is to reach configuration $p(m, N - m)$. For that purpose we add a special transition reading $\star$ with effect $(-m, m - N)$ and going from $p$ to the sink state $\bot$. All the other sequences of transitions need to be made accepting. For that we add an extra accepting state $q_f$ and a lot of transitions leading to it. Concretely, $\mathcal{B}$ has these transitions:

(i) the initial transition $(q_0, t, (0, N), q)$ for every $t \in \Sigma \setminus \{\star\}$;
(ii) a 'simulating' transition $(r, t, (h, -h), r')$ for every $t = (r, a, h, r') \in \delta_\mathcal{A}$;

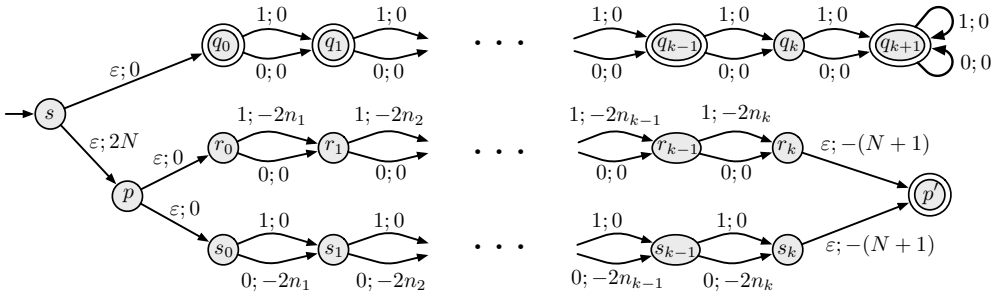


**Figure 1** Definition of $V_S$. An arrow labelled "$i;\ell$" denotes a transition reading $i$ with effect $\ell$. Double circled states are final.

(iii) a transition from $p$ to $\bot$ reading $\star$ with effect $(-m, m-N)$;
(iv) a transition from every $r \in Q_{\mathcal{A}} \setminus \{p\}$ to $q_f$ reading $\star$ with effect $(0,0)$;
(v) a transition from $p$ to $q_f$ reading $\star$ with effect $(-k, -(N-k))$ for every $k \neq m$;
(vi) a transition from every $r \in Q_{\mathcal{A}}$ to $q_f$ reading $t \in \Sigma \setminus \{\star\}$ if the $t$-labelled transition is not outgoing from $r$;
(vii) a transition from every $r \in Q_{\mathcal{A}}$ to $q_f$ reading $t \in \Sigma \setminus \{\star\}$ with effect $(0, -(N+\ell)-1)$ if the $t$-labelled transition is outgoing from $r$ and has effect $\ell < 0$;
(viii) $\bar{0}$-effect self-loops on $q_f$, with all possible letters of $\Sigma$.

The correctness of the construction is in the Appendix. ◀

Finally, we show coNP-hardness for universality of one counter nets.

▶ **Proposition 20** (Theorem 3(iv)). *The universality problem for unambiguous 1-VASS is* coNP*-hard.*

**Proof.** We equivalently will show that non-universality problem for unambiguous 1-VASSes is NP-hard. The reduction is from the Perfect Partition problem. In the Perfect Partition problem we are given a finite set of natural numbers $S = \{n_1, \ldots, n_k\} \subseteq_{\text{FIN}} \mathbb{N}$ and we are supposed to answer whether the set of indices $[1,k]$ can be partitioned into two subsets $I_1, I_2 \subseteq [1,k]$ such that $\sum_{i \in I_1} n_i = \sum_{i \in I_2} n_i$. Such a partition is called a *perfect partition*. All the numbers are binary represented. The Perfect Partition problem is known to be NP-hard [5].

For an instance of a Perfect Partition problem $S \subseteq_{\text{FIN}} \mathbb{N}$ we build an unambiguous 1-VASS $V_S$ such that perfect partition for $S$ exists if and only if the 1-VASS is not universal. Let $\sum_{i \in [1,k]} n_i = N$, note that the perfect partition exists iff there is a set of indices $I$ such that $\sum_{i \in I} n_i = N/2$. Every word of length $k$ encodes a subset of $S$ in the following way: for $w \in \{0,1\}^k$ we define $S_w = \{n_i \mid w[i] = 1\}$. We will design $V_S$ in such a way that $\mathcal{L}(V_S) \subseteq \{0,1\}^*$ will always contain all the words of length different than $k$. Among words of length $k$ language $\mathcal{L}(V_S)$ will contain exactly these for which $\sum_{n \in S_w} n \neq N/2$. Then indeed $\mathcal{L}(V_S)$ would be not universal iff set $S$ has a perfect partition.

The 1-VASS $V_S$ is defined in Figure 1. It consists of two parts: the top part, with states $q_0$ to $q_{k+1}$ accepts all the words of length different than $|S| = k$, while the bottom part accepts some words of length $k$.

It is immediate to see that the top part accepts all words of length different to $k$, and all of them by exactly one run.

The bottom part consists of states: $p$, $p'$, $r_0, r_1, \ldots, r_k$ and $s_0, s_1, \ldots, s_k$, where only the state $p'$ is accepting. Notice that transitions in states $r_i$ are mirrored with respect to

transitions in states $s_i$, namely effect of a transition over some letter from $r_i$ equals the effect of the transition over the other letter in $s_i$. Let us inspect now how an accepting run over $w \in \{0,1\}^k$ can look like. Every such run starts from $q_0(0)$ and then goes to $p(2N)$. Then it splits into two runs, to $r_0(2N)$ and $s_0(2N)$ and from this moment on there are two runs: one in some state $r_i$ and the other in the corresponding state $s_i$. Then after reading the whole $w$ the two runs are in configurations $r_k(2N - 2S_w)$ and $s_k(2N - 2(N - S_w)) = s_k(2S_w)$. Notice that $0 \leq S_w \leq N$, so both configurations are indeed always reachable. Now comes the last transition from either $r_k$ or $s_k$ to $p'$. Observe that if $S_w \neq N/2$ then exactly one of them can be fired. Indeed if $S_w \neq N/2$ so $2S_w \neq N$ then exactly one of the numbers $2S_w$ and $2N - 2S_w$ equals at least $N + 1$. Then from exactly one of the configurations $r_k(2N - 2S_w)$ and $s_k(2S_w)$ counter value $N + 1$ can be subtracted and the run over the word $w$ will reach an accepting configuration $p'(c)$ for some $c \geq 0$. Then we have $w \in \mathcal{L}(V_S)$ and exactly one accepting run over $w$. On the other hand assume now that $S_w = N/2$. Then the two reached configurations are $r_k(N)$ and $s_k(N)$. In none of them counter value $N + 1$ can be subtracted, which means that in that case no accepting run over $w$ exists and $w \notin \mathcal{L}(V_S)$. Therefore indeed $V_S$ is unambiguous and importantly $\mathcal{L}(V_S)$ is not universal iff there exists a perfect partition for $S$. This finishes the proof. ◀

▶ **Proposition 21** (Theorem 3(ii) lower bound). *The universality problem for d-VASS with unary encoding is* NL*-hard, for every $d \geq 1$.*

**Proof.** This already holds for UFA. ◀

## 5 Testing for Unambiguity

Here we will prove Theorem 4. As we will see, upper bounds follow from the emptiness problem and lower bounds from adaptations of the reductions from the previous section.

### 5.1 Upper bounds

We will next prove the upper bound of Theorem 4(i), (ii) and (iii) namely:

▶ **Proposition 22** (Theorem 4(i), (ii) and (iii) upper bound). *The unambiguity checking problem is:*

*(i) in* ExpSpace *for VASSes with binary encoding;*
*(ii) in* NL *for d-VASSes with unary encoding for any fixed $d \in \mathbb{N}$;*
*(iii) in* PSpace *for d-VASSes with binary encoding for any fixed $d \in \mathbb{N}$.*

**Proof.** By Lemma 6 if a $d$-VASS with norm $M$ and $n$ states is ambiguous then there exists two different runs of length at most $A_{M,2d,2n^2}$ accepting the same word. Number $A_{M,2d,2n^2} = (4n^4(M+1)^2)^{(8d)^{2d-1}}$ is doubly exponential wrt. the size of the VASS representation when $M$ is given in binary and $d$ not fixed. For fixed $d$ an $M$ given in binary $A_{M,2d,2n^2}$ is exponential wrt. the input and for fixed $d$ and $M$ given in unary it is polynomial wrt. the input. Therefore the algorithm, which enumerates all the pairs of different runs of length up to $A_{M,2d,2n^2}$ and checks whether some pair accepts the same word works in ExpSpace, NL and PSpace, respectively, which finishes the proof. ◀

## 5.2 Lower bounds

▶ **Proposition 23** (Theorem 4(i) lower bound). *The unambiguity checking problem for VASS with unary encoding is* ExpSpace*-hard.*

**Proof.** We reduce from the problem whether an unambiguous $\varepsilon$-VASS has an empty language, which is ExpSpace-complete as mentioned in Lemma 2 (it is a consequence of Lipton's construction [6]). To an unambiguous $\varepsilon$-VASS we add one state accepting the empty word $\varepsilon$. Then the constructed VASS is unambiguous iff the original one has empty language, which finishes the ExpSpace-hardness proof. ◀

▶ **Proposition 24** (Theorem 4(ii) lower bound). *The unambiguity checking problem for d-VASS with unary encoding is* NL*-hard.*

**Proof.** This is already true for finite automata. ◀

▶ **Proposition 25** (Theorem 4(iii) lower bound). *The unambiguity checking problem for 2-VASS is* PSpace*-hard.*

**Proof.** This is a corollary of the construction in the proof of Proposition 19. One can adapt the automaton by now having $\bot$ as sole accepting state, and all other states as non-accepting, and adding a transition $(\bot, \varepsilon, (0,0), \bot)$, in such a way that $\mathcal{B}$ is unambiguous if, and only if, there is no run that reaches $\bot$. ◀

▶ **Proposition 26** (Theorem 4(iv)). *The unambiguity checking problem for* 1*-VASS is* coNP*-hard.*

**Proof.** A construction very similar to the one used to show coNP-hardness of universality (Proposition 20) can be used to show that unambiguity checking for 1-VASS is coNP-hard. If instead of transitions $r_k \xrightarrow{\varepsilon;-(N+1)} p'$ and $s_k \xrightarrow{\varepsilon;-(N+1)} p'$ we have transitions $r_k \xrightarrow{\varepsilon;-N} p'$ and $s_k \xrightarrow{\varepsilon;-N} p'$, then $V_S$ is ambiguous if and only if there is a perfect partition for $S$. This shows that ambiguity checking is NP-hard and unambiguity checking is coNP-hard. ◀

# 6 Discussion

We leave open the question about the exact complexity of universality problem for unambiguous 1-VASS with transitions represented in binary, which we showed to be PSpace-easy and coNP-hard. We conjecture that it is coNP-complete. Another question that we leave open is the complexity of the universality problem for VASS without $\varepsilon$-transitions; our ExpSpace-hardness of Proposition 17 crucially uses $\varepsilon$-transitions, and it is not clear whether it can be adapted to avoid them. An open question related to the gap of Theorem 3(ii) is the one about the precise complexity of the universality problem for unambiguous finite automata, which is NL-hard and only known to be in $\mathrm{NC}^2$ [13].

While we have focused our study on the universality and unambiguity checking problems for unambiguous VASS, we point out that there are many intriguing unanswered problems on unambiguous systems. In particular, closely related to the universality problem are: co-finiteness, equivalence and inclusion problems. It is natural to ask about the decidability and complexity of the problem for most fundamental models of computation: finite automata, one counter nets, VASS or even pushdown automata (PDA) under the assumption of unambiguity. We give some examples. While equivalence of VASS languages is undecidable, is it be decidable for unambiguous VASS? Language equivalence is undecidable for PDA and decidable for deterministic PDA (by the celebrated result of Sénizergues [11]), but might it still be decidable for unambiguous PDA? And what about universality?

## References


1 Thomas Colcombet. Unambiguity in automata theory. In *Proceedings of DCFS 2015*, pages 3–18, 2015.

2 John Fearnley and Marcin Jurdziński. Reachability in two-clock timed automata is PSPACE-complete. *Inf. Comput.*, 243:26–36, 2015.

3 Piotr Hofman and Patrick Totzke. Trace inclusion for one-counter nets revisited. In Joël Ouaknine, Igor Potapov, and James Worrell, editors, *Proceedings of RP 2014*, volume 8762 of *Lecture Notes in Computer Science*, pages 151–162. Springer, 2014.

4 Petr Jancar, Javier Esparza, and Faron Moller. Petri nets and regular processes. *J. Comput. Syst. Sci.*, 59(3):476–503, 1999.

5 Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations 1972*, pages 85–103, 1972.

6 Richard Lipton. The reachability problem requires exponential space. *Department of Computer Science. Yale University*, 62, 1976.

7 Antoine Mottet and Karin Quaas. The containment problem for unambiguous register automata. In *Proceedings of STACS 2019*, pages 53:1–53:15, 2019.

8 Frank Neven, Thomas Schwentick, and Victor Vianu. Finite state machines for strings over infinite alphabets. *ACM Trans. Comput. Log.*, 5(3):403–435, 2004.

9 Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theor. Comput. Sci.*, 6:223–231, 1978.

10 Mikhail Raskin. A superpolynomial lower bound for the size of non-deterministic complement of an unambiguous automaton. In *Proceedings of ICALP 2018*, pages 138:1–138:11, 2018.

11 Géraud Sénizergues. L(A)=L(B)? decidability results from complete formal systems. *Theor. Comput. Sci.*, 251(1-2):1–166, 2001.

12 Richard Edwin Stearns and Harry B. Hunt III. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM J. Comput.*, 14(3):598–611, 1985.

13 Wen-Guey Tzeng. On path equivalence of nondeterministic finite automata. *Inf. Process. Lett.*, 58(1):43–46, 1996.


## A Missing proofs

**Proof of Proposition 1.** Let $C = 2n^2(M+1)^2$. We proceed by induction on $d$. For $d = 1$ assume there is some accepting run with no configuration repeating. Then in its prefix of length $nM$ there is definitely first a configuration $q(x)$ and later a configuration $q(\bar{y})$ for some state $q$ and counter values $x < y$. Then we can change this accepting run into an accepting run of length at most $nM + (nM)^2 + n - 1$. We first pump the infix from $q(x)$ do $q(y)$ exactly $nM$ times obtaining then a configuration $q(z)$ with $z = y + nM(y - x) \geq nM$. As some accepting state is reachable from $q$ then it is also reachable by a run of length smaller than $n$. This run (and any of its prefixes) can, at worst, have a negative effect of value $(n-1)M$, and thus it can be triggered from $q(z)$, since $z \geq nM$. In this way, we get an accepting run of length at most $nM + (nM)^2 + n - 1 \leq C$, proving the base case.

For the inductive step, assume that there is an accepting run $s(\bar{0}) \xrightarrow{\rho} f(\bar{v})$ in a $(d+1)$-VASS with norm $M$ and $n$ states. Let $K_d = C^{(4d)^{d-1}}$. We distinguish two cases:

(i) the norm of every configuration on $\rho$ is bounded by $C \cdot K_d$;
(ii) the norm of some configuration on $\rho$ exceeds $C \cdot K_d$.

Without loss of generality we can assume that no configuration on $\rho$ appears more than once, otherwise we can "unpump" $\rho$ to obtain a shorter one. Observe that in the first case (i), the length of $\rho$ is bounded by $D = (C \cdot K_d)^{d+1}$ (we will bound $D$ later on).

In second case (ii), the run $\rho$ might be long, but we will show that there is another short accepting run $\rho'$. Let $p(\bar{u})$ be the first configuration on $\rho$ with norm exceeding $C \cdot K_d$. Let $s(\bar{0}) \xrightarrow{\rho_1} p(\bar{u}) \xrightarrow{\rho_2} f(\bar{v})$. Clearly, the length of $\rho_1$ is bounded by $D$ by a similar reasoning as in the case (i). We will replace $\rho_2$ with a "short" run $\pi$, so that $c \xrightarrow{\pi} f(\bar{v}')$. First note that some coordinate of $p(\bar{u})$ must have value greater or equal to $C \cdot K_d$; without loss of generality, assume it is the last one, that is, the $(d+1)$-st coordinate. Let us now ignore the last coordinate in the VASS. By inductive hypothesis, there is a sequence of transitions $\pi$ of length at most $K_d$ such that $p(\bar{u}_d) \xrightarrow{\pi_d} f(\bar{v}'_d)$, where $\pi_d$ is the result of ignoring the last coordinate of $\pi$, and $\bar{u}_d, \bar{v}_d \in \mathbb{N}^d$ are the results of ignoring the last coordinate of $\bar{u}, \bar{v}$. Consider now the sequence of transitions $\pi$ starting in $p(\bar{u})$. Its length is bounded by $K_d$, so its effect on the $(d+1)$-st coordinate is not smaller than $-M \cdot K_d$. Since $\bar{u}[d+1] \geq C \cdot K_d \geq M \cdot K_d$, then $\pi$ is indeed a valid run from $p(\bar{u})$ to $f(\bar{v}')$ for some $\bar{v}' \in \mathbb{N}^{d+1}$. Therefore, the run $\rho_1 \cdot \pi$ is accepting from $s(\bar{0})$ as $s(\bar{0}) \xrightarrow{\rho_1} p(\bar{u}) \xrightarrow{\pi} f(\bar{v}')$. Length of $\rho_1 \cdot \pi$ is at most $D + K_d$.

In order to finish the argument in case (ii) we need to show that $D + K_d \leq K_{d+1}$, through the following sequence of (very rough) estimations

$$\begin{aligned} D + K_d \leq 2D \leq C \cdot D = C \cdot (C \cdot K_d)^{d+1} = C \cdot \big((C \cdot C^{(4d)^{d-1}})^{d+1} \\ = C^{((4d)^{d-1}+1)(d+1)+1} \leq C^{4(4d)^{d-1} \cdot (d+1)} \leq C^{(4(d+1))^d} = K_{d+1}. \end{aligned}$$

Observe that in case (i), the bound $D \leq K_{d+1}$ is trivial. ◀

**Proof of Proposition 19 (correctness).** Observe first that, by construction, all configurations $c$ of $\mathcal{B}$ reachable from $q_0(0,0)$ are $N$-bounded. Further, if the state of $c$ is from $Q_{\mathcal{A}}$, then the sum of its components is equal to $N$.

We show that $\mathcal{B}$ is unambiguous. What is more, we will show that for every configuration $r(\bar{u}_0)$ reachable from $q_0(0,0)$ and for every letter $a \in \Sigma$ there is at most one outgoing transition from $r$ reading $a$ that can be applied to $r(u_0, u'_0)$. By means of contradiction, suppose that there are two distinct transitions $(r, a, (u_1, u'_1), r_1), (r, a, (u_2, u'_2), r_2) \in \delta$ such that $(u_0, u'_0) + (u_1, u'_1) \in \mathbb{N}^2$ and $(u_0, u'_0) + (u_2, u'_2) \in \mathbb{N}^2$. By construction, the only possibility is that one transition is a simulating transition as defined in (ii), and the other transition is as defined in (vii). In particular, $a$ must be a transition from $\delta_{\mathcal{A}}$, $r, r_1$ are states from $Q_{\mathcal{A}}$, and $r_2 = q_f$. By the above observation, $u_0 + u'_0 = N$, and by construction (item (vii)), $u_1 < 0$ and $u'_2 = -(N + u_1) - 1$. Since $u'_0 + u'_2 \geq 0$ by the hypothesis $(u_0, u'_0) + (u_2, u'_2) \in \mathbb{N}^2$, we can replace $u'_2$ with the equality $u'_2 = -(N + u_1) - 1$ just observed, and we obtain $u'_0 - (N + u_1) - 1 \geq 0$. Since we also know that $u_0 + u'_0 = N$ by the observation above, we can further replace $u'_0$ with $N - u_0$ in $u'_0 - (N + u_1) - 1 \geq 0$, and we obtain $u_0 + u_1 < 0$. Note that this contradicts the hypothesis $(u_0, u'_0) + (u_1, u'_1) \in \mathbb{N}^2$. The contradiction comes from assuming that both transitions were possible to trigger.

We finally show that $\mathcal{B}$ is universal if, and only if, there is no $N$-bounded run from $q(0)$ to $p(m)$ in $\mathcal{A}$. Observe first that for every word $w \in \Sigma^*$ there is exactly one run of $\mathcal{B}$ reading $w$. If there is an $N$-bounded run $\rho$ from $q(0)$ to $p(m)$ in $\mathcal{A}$, it follows that the run of $\mathcal{B}$ reading $\rho\star$ ends in the sink state $\bot$, and thus $\rho\star \notin \mathcal{L}(\mathcal{B})$, witnessing the fact that $\mathcal{B}$ is not universal. If, on the other hand, there is a run of $\mathcal{B}$ ending in state $\bot$, it must be reading a word of the form $\rho\star$ where $\rho$ is an $N$-bounded run from $q(0)$ to $p(m)$ in $\mathcal{A}$. Since $\bot$ is the sole state which is not accepting and since, as observed before, for all words there is a run, it follows that if $\mathcal{B}$ is universal, then there is no $N$-bounded run in from $q(0)$ to $p(m)$ in $\mathcal{A}$. ◀