Regular Separators for VASS Coverability Languages

Chris Köcher
Max Planck Institute for Software Systems, Kaiserslautern

Georg Zetzsche
Max Planck Institute for Software Systems, Kaiserslautern

Abstract

We study regular separators of vector addition systems (VASS, for short) with coverability semantics. A regular language R is a *regular separator* of languages K and L if $K \subseteq R$ and $L \cap R = \emptyset$. It was shown by Czerwiński, Lasota, Meyer, Muskalla, Kumar, and Saivasan (CONCUR 2018) that it is decidable whether, for two given VASS, there exists a regular separator. In fact, they show that a regular separator exists if and only if the two VASS languages are disjoint. However, they provide a triply exponential upper bound and a doubly exponential lower bound for the size of such separators and leave open which bound is tight.

We show that if two VASS have disjoint languages, then there exists a regular separator with at most doubly exponential size. Moreover, we provide tight size bounds for separators in the case of fixed dimensions and unary/binary encodings of updates and NFA/DFA separators. In particular, we settle the aforementioned question.

The key ingredient in the upper bound is a structural analysis of separating automata based on the concept of *basic separators*, which was recently introduced by Czerwiński and the second author. This allows us to determinize (and thus complement) without the powerset construction and avoid one exponential blowup.

2012 ACM Subject Classification Theory of computation → Models of computation; Theory of computation → Regular languages

Keywords and phrases Vector Addition System, Separability, Regular Language

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2023.15

Funding Funded by the European Union (ERC, FINABIS, 101077902). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2023).
Editors: Patricia Bouyer and Srikanth Srinivasan; Article No. 15; pp. 15:1–15:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Safety verification of concurrent systems typically consists of deciding whether two languages $K, L \subseteq \Sigma^*$ are disjoint: If each of the languages describes the set of event sequences that (i) are consistent with the behavior of a some system component and (ii) reach an undesirable state, then their intersection is exactly the set of event sequences that are consistent with both components and reach the undesirable state.

If we wish to not only decide, but *certify* disjointness of languages $K, L \subseteq \Sigma^*$, then a natural kind of certificate is a *regular separator*: a regular language $R \subseteq \Sigma^*$ such that $K \subseteq R$ and $L \cap R = \emptyset$. Regular separators can indeed act as disjointness certificates: Deciding whether a given language intersects (resp. is included in) a regular language is usually simple.

The *regular separability* problem asks whether for two given languages there exists a regular separator. This decision problem has recently attracted a significant amount of interest. After the problem was shown to be undecidable for context-free languages in the 1970s [18, 33], recent work has a strong focus on *vector addition systems* (VASS), which are automata with counters that can be incremented, decremented, but not tested for zero. Typically, VASS are considered with two possible semantics: With the *reachability semantics*, where a target configuration has to be reached exactly, and the *coverability semantics*, where the target only has to be covered. Decidability of regular separability remains an open problem for reachability semantics. However, decidability has been established for coverability languages of VASS [10] and several other subclasses, such as one-dimensional VASS [9], integer VASS [6] (where counters can become negative), and commutative VASS languages [7]. Moreover, for each of these subclasses, decidability is retained if one of the input languages is an arbitrary VASS reachability language [13].

The decidability result about VASS coverability languages is a consequence of a remarkable and surprising result by Czerwiński, Lasota, Meyer, Muskalla, Kumar, and Saivasan [10]: Two languages of finitely-branching well-structured transition systems (WSTS) are separable by a regular language if and only if they are disjoint. (In fact, very recently, Keskin and Meyer [20] have even shown that the finite branching assumption is not required.) Moreover, VASS (with coverability semantics) are a standard example of (finitely branching) WSTS.

Despite this range of work on decidability, very little is known about a fundamental aspect of the separators: *What is the size of the separator, if they exist?* Here, by size, we mean the number of states in an NFA or DFA. In fact, the only result we are aware of is a partial answer for VASS coverability languages: In [10] a triply exponential upper bound and a doubly exponential lower bound is shown for NFA separating VASS coverability languages, leaving open whether there always exists a doubly-exponential separator.

Contribution. We study the size of regular separators in VASS coverability languages. Our first main result is that if two VASS coverability languages are disjoint, then there exists a doubly exponential-sized separating NFA. We then provide a comprehensive account of separator sizes for VASS languages: We study separator sizes in (i) fixed/arbitrary dimension, (ii) with unary/binary counter updates and (iii) deterministic/non-deterministic separators. In each case, we provide a tight polynomial or singly, doubly, or triply exponential bound.

Related work. There also exists some work on separability of relations by recognizable relations [1, 5] (which, in some precise sense, is also an instance of regular separability).

The equivalence between regular separability and disjointness for WSTS [10, 20] and the fact that decidability of the two problems usually coincide, raise the question of whether they are inter-reducible in general. However, there are language classes where disjointness is decidable and regular separability is undecidable [21, 34] and vice-versa [34].

Decidability of separability by piecewise testable languages is quite well understood. There is a language theoretic characterization [12] (which also holds for more general separator classes [35]) and a more abstract characterization (that also applies to trees) [15] of when separability is decidable.

There is long line of work on separability of regular languages of finite words by languages from smaller subclasses [11, 23–31]. Beyond finite words, separability has been studied for languages of infinite words (for regular languages [17] and Büchi VASS [2]) and for regular languages of finite trees [3] and infinite trees [8].

2 Preliminaries

Let $d \in \mathbb{N}^+$ be a positive number. A *vector* $\vec{v}$ over $\mathbb{Z}$ is an element $\vec{v} \in \mathbb{Z}^d$. For a vector $\vec{v} = (v_1, \dots, v_d) \in \mathbb{Z}^d$ and a number $1 \le i \le d$ we write $\vec{v}[i]$ for the i-th component v_i of $\vec{v}$. By $\vec{0} \in \mathbb{Z}^d$ we denote the *zero vector* satisfying $\vec{0}[i] = 0$ for each $1 \le i \le d$. For two vectors $\vec{u}, \vec{v} \in \mathbb{Z}^d$ we write $\vec{u} + \vec{v}$ for the vector $\vec{w} \in \mathbb{Z}^d$ with $\vec{w}[i] = \vec{u}[i] + \vec{v}[i]$ for each $1 \le i \le d$, i.e., $+$ is the component-wise addition. We write $\vec{u} \le \vec{v}$ if, and only if, we have $\vec{u}[i] \le \vec{v}[i]$ (for the natural ordering in $\mathbb{Z}$) for each $1 \le i \le d$. Note that $\le$ is a partial ordering on $\mathbb{Z}^d$, but in the case of $d > 1$ no linear ordering.

Now, let $c, d \in \mathbb{N}^+$, $\vec{u} \in \mathbb{Z}^c$, and $\vec{v} \in \mathbb{Z}^d$. By $(\vec{u}, \vec{v})$ we denote the vector $\vec{w} \in \mathbb{Z}^{c+d}$ with $\vec{w}[i] = \vec{u}[i]$ for each $1 \le i \le c$ and $\vec{w}[i + c] = \vec{v}[i]$ for each $1 \le i \le d$, i.e., $(\vec{u}, \vec{v})$ is the concatenation of $\vec{u}$ and $\vec{v}$.

Vector Addition Systems. Let $d \in \mathbb{N}^+$. A *(d-dimensional) vector addition system with states* or *(d-)VASS* is a tuple $\mathfrak{V} = (Q, \Sigma, \Delta, s, t)$ where Q is a finite set of *states*, Σ is an alphabet, $\Delta \subseteq Q \times \Sigma_\varepsilon \times \mathbb{Z}^d \times Q$ is a finite set of transitions, and $s, t \in Q$ are its *source* resp. *target states*. Here, Σ_ε denotes the set $\Sigma \cup \{\varepsilon\}$. The vector $\vec{x} \in \mathbb{Z}^d$ of a transition $(p, a, \vec{x}, q) \in \Delta$ is called the *counter update* of this transition.

A *pseudo-configuration* is a tuple from $Q \times \mathbb{Z}^d$; it is called a *configuration* if this tuple is even contained in $Q \times \mathbb{N}^d$. A *pseudo-run* is a sequence $(q_i, \vec{v_i})_{0 \le i \le \ell}$ of pseudo-configurations such that for each $1 \le i \le \ell$ there is a transition $(q_{i-1}, a_i, \vec{x_i}, q_i) \in \Delta$ with $\vec{v_i} = \vec{v_{i-1}} + \vec{x_i}$. The *label* of such pseudo-run is $a_1 a_2 \dots a_\ell \in \Sigma^*$; its *length* is ℓ (note that due to ε-labeled transitions we have $\ell \ge |a_1 a_2 \dots a_\ell|$). A pseudo-run is called a *run* if we have $\vec{v_i} \in \mathbb{N}^d$ for each $0 \le i \le n$, i.e., if each intermediate pseudo-configuration is actually a configuration. For two configurations $(p, \vec{u}), (q, \vec{v}) \in Q \times \mathbb{N}^d$ and $w \in \Sigma^*$ we write $(p, \vec{u}) \xrightarrow{w}_{\mathfrak{V}} (q, \vec{v})$ if there is a run $(q_i, \vec{v_i})_{0 \le i \le \ell}$ with label w, $(p, \vec{u}) = (q_0, \vec{v_0})$, and $(q, \vec{v}) = (q_\ell, \vec{v_\ell})$. For a natural number $\ell \in \mathbb{N}$ we write $(p, \vec{u}) \to^{\ell}_{\mathfrak{V}} (q, \vec{v})$ if there is a run from $(p, \vec{u})$ to $(q, \vec{v})$ of length ℓ. We write $(p, \vec{u}) \to_{\mathfrak{V}} (q, \vec{v})$ if there exists such an ℓ.

The *(coverability) language* of $\mathfrak{V}$ is $\mathrm{L}(\mathfrak{V}) = \{w \in \Sigma^* \mid \exists \vec{v} \in \mathbb{N}^d \colon (s, \vec{0}) \xrightarrow{w}_{\mathfrak{V}} (t, \vec{v})\}$ (note that $\vec{v} \ge \vec{0}$ holds for any $\vec{v} \in \mathbb{N}^d$; we say that $(t, \vec{v})$ *covers* the so-called target configuration $(t, \vec{0})$). We say $L \subseteq \Sigma^*$ is a *(coverability) d-VASS-language* if there is a d-VASS $\mathfrak{V}$ with $L = \mathrm{L}(\mathfrak{V})$.

Now, let $\mathfrak{V}_i = (Q_i, \Sigma, \Delta_i, s_i, t_i)$ be two d-VASS ($i = 1, 2$). We want to construct the product VASS $\mathfrak{V}_1 \times \mathfrak{V}_2$ which simulates $\mathfrak{V}_1$ and $\mathfrak{V}_2$ in parallel. To this end, set the $2d$-VASS $\mathfrak{V}_1 \times \mathfrak{V}_2 := (Q_1 \times Q_2, \Sigma, \Delta, (s_1, s_2), (t_1, t_2))$ with the following transitions in Δ:

- $((p_1, p_2), a, (\vec{v_1}, \vec{v_2}), (q_1, q_2)) \in \Delta$ if $(p_1, a, \vec{v_1}, q_1) \in \Delta_1$ and $(p_2, a, \vec{v_2}, q_2) \in \Delta_2$,
- $((p_1, p_2), \varepsilon, (\vec{v_1}, \vec{0}), (q_1, p_2)) \in \Delta$ if $(p_1, \varepsilon, \vec{v_1}, q_1) \in \Delta_1$, and

- $((p_1, p_2), \varepsilon, (\vec{0}, \vec{v_2}), (p_1, q_2)) \in \Delta$ if $(p_2, \varepsilon, \vec{v_2}, q_2) \in \Delta_2$.

Then the following statement is easy to see:

▶ **Lemma 2.1.** *Let $\mathfrak{V}_1$ and $\mathfrak{V}_2$ be two d-VASS. Then $\mathrm{L}(\mathfrak{V}_1 \times \mathfrak{V}_2) = \mathrm{L}(\mathfrak{V}_1) \cap \mathrm{L}(\mathfrak{V}_2)$, i.e., the intersection of two d-VASS-languages is a $2d$-VASS-language.* ◀

For a vector $\vec{v} \in \mathbb{Z}^d$ let $\|\vec{v}\| = \max\{|\vec{v}[i]| : 1 \leq i \leq d\}$ be the *norm* of $\vec{v}$ (where $|x|$ is the absolute value of $x \in \mathbb{Z}$). We also define the norm of the transition relation Δ as follows: $\|\Delta\| := \max\{\|\vec{v}\| : (p, a, \vec{v}, q) \in \Delta\}$. Then the *size* $|\mathfrak{V}|$ of the d-VASS $\mathfrak{V}$ is $|Q| + d \cdot |\Delta| \cdot \|\Delta\|$.

We can define the *Rackoff-number* $\mathrm{Rack}(\mathfrak{V})$ of $\mathfrak{V}$: $\mathrm{Rack}(\mathfrak{V}) := (|Q| \cdot \|\Delta\| + 2)^{(3d)!+1}$. Then we can show that for each run from a configuration $c \in Q \times \mathbb{N}^d$ covering the target configuration $(t, \vec{0})$ there is also such run of length bounded by the Rackoff-number. This is the following central statement:

▶ **Theorem 2.2** ([4, 32]). *Let $\mathfrak{V} = (Q, \Sigma, \Delta, s, t)$ be a d-VASS and $c \in Q \times \mathbb{N}^d$ be a configuration such that there is a vector $\vec{v} \in \mathbb{N}^d$ with $c \to_{\mathfrak{V}} (t, \vec{v})$. Then there are $\ell \in \mathbb{N}$ and $\vec{w} \in \mathbb{N}^d$ with $0 \leq \ell \leq \mathrm{Rack}(\mathfrak{V})$ and $c \to^{\ell}_{\mathfrak{V}} (t, \vec{w})$.* ◀

The bound above is due to Bozelli and Ganty [4], which is slightly tighter than Rackoff's original bound of $2^{2^{\mathcal{O}(\|\Delta\| \log \|\Delta\|)}}$ [32]. It should be noted that very recently, a significantly better upper bound has been obtained [22]).

Regular languages. A *non-deterministic finite automaton* or *NFA* is a tuple $\mathfrak{A} = (Q, \Sigma, \delta, I, F)$ where Q is a finite set of *states*, Σ is an alphabet, $\delta \subseteq Q \times \Sigma \times Q$ is a set of *transitions*, and $I, F \subseteq Q$ are the sets of *initial* resp. *accepting* states. It is called *deterministic* or *DFA* if $|I| = 1$ and for each $p \in Q$ and $a \in \Sigma$ there is exactly one $q \in Q$ with $(p, a, q) \in \delta$. The size $|\mathfrak{A}|$ of $\mathfrak{A}$ is $|Q|$. For $p, q \in Q$ and $w \in \Sigma^*$ we write $p \xrightarrow{w}_{\mathfrak{A}} q$ if there are $a_1, \ldots, a_\ell \in \Sigma$ and $q_0, q_1, \ldots, q_\ell \in Q$ with $w = a_1 a_2 \ldots a_\ell$, $p = q_0$, $q = q_\ell$, and $(q_{i-1}, a_i, q_i) \in \delta$ for each $1 \leq i \leq \ell$. The *accepted language* of $\mathfrak{A}$ is $\mathrm{L}(\mathfrak{A}) = \{w \in \Sigma^* \mid \exists \iota \in I, f \in F : \iota \xrightarrow{w}_{\mathfrak{A}} f\}$. A language $L \subseteq \Sigma^*$ is called *regular* if there is an NFA $\mathfrak{A}$ with $L = \mathrm{L}(\mathfrak{A})$.

Regular Separability. Let Σ be an alphabet. Two languages $K, L \subseteq \Sigma^*$ are called *regular separable* (denoted $K \mid L$) if there is a regular language $R \subseteq \Sigma^*$ with $K \subseteq R$ and $L \cap R = \emptyset$. In this case R is called a *regular separator* of K and L. We say that any NFA accepting R *separates* K and L. Since the class of regular languages is closed under complement, we learn that if $K \mid L$ holds, then also $L \mid K$ (via the complementary separator).

The following equivalence is known about the languages of coverability VASS. Note that actually Czerwiński et al. [10] have shown this result for the languages of a more general notion—so-called *well structured transition systems* (or *WSTS* for short, cf. e.g. [14]).

▶ **Theorem 2.3** ([10]). *Let $\mathfrak{V}$ and $\mathfrak{W}$ be two VASS. Then we have $\mathrm{L}(\mathfrak{V}) \mid \mathrm{L}(\mathfrak{W})$ if, and only if, $\mathrm{L}(\mathfrak{V}) \cap \mathrm{L}(\mathfrak{W}) = \emptyset$.* ◀

3 Main Results

In this section, we present the main results of this work. An overview can be found in Table 1. Here, by i-exp, we mean that there is an i-fold exponential upper bound. More precisely, there exists a separator with at most $\exp_i(\mathrm{poly}(n))$ states for input VASS of size n. Here $\exp_0(n) = n$ and $\exp_{i+1}(n) = 2^{\exp_i(n)}$ for $i \geq 0$. All our bounds are tight in the sense that for each i-fold exponential upper bound with $i \geq 1$, we present a sequence of VASS pairs of size polynomial in n such that the smallest separator requires $\exp_i(n)$ states. Proofs can be found in Sections 5 and 6 (upper bounds and lower bounds, resp.).

		NFAs		DFAs	
		unary	binary	unary	binary
d as input		2-exp.	2-exp.	3-exp.	3-exp.
d fixed	$d \geq 2$	poly.	exp.	exp.	2-exp.
	$d = 1$	poly.	exp.	exp.	exp.

Table 1 An overview of the (matching) upper and lower bounds for finite automata separating two disjoint d-VASS. We distinguish between (i) whether the dimension $d \in \mathbb{N}^+$ is part of the input, (ii) whether the separating automaton should be an NFA or a DFA, and (iii) whether counter updates are encoded in unary or binary. The colors denote the employed lower bound technique.

First upper bound. Our first upper bound result is the following.

▶ **Theorem 3.1.** *Let $\mathfrak{V}_1$ and $\mathfrak{V}_2$ be d-VASS with at most $n \geq 1$ states and updates of norm at most $m \geq 1$. If $\mathrm{L}(\mathfrak{V}_1) \cap \mathrm{L}(\mathfrak{V}_2) = \emptyset$, then $\mathrm{L}(\mathfrak{V}_1)$ and $\mathrm{L}(\mathfrak{V}_2)$ are separated by an NFA with at most $(n+m)^{2^{\mathrm{poly}(d)}}$ states.*

This provides almost all upper bounds in Table 1. In particular, it closes the gap left by [10] by providing a doubly exponential upper bound for NFA separators in the general case.

Let us explain how we avoid one exponential blow-up compared to [10]. In [10], the authors first construct VASS $\mathfrak{V}'_1$ and $\mathfrak{V}'_2$ such that (i) $\mathfrak{V}'_2$ is deterministic, (ii) $\mathrm{L}(\mathfrak{V}'_1) \cap \mathrm{L}(\mathfrak{V}'_2) = \emptyset$ and (iii) any separator for $\mathrm{L}(\mathfrak{V}'_1)$ and $\mathrm{L}(\mathfrak{V}'_2)$ can be transformed into a separator for $\mathrm{L}(\mathfrak{V}_1)$ and $\mathrm{L}(\mathfrak{V}_2)$. Then, relying on Rackoff-style bounds for covering runs in VASS, they construct a doubly exponential NFA separator for $\mathrm{L}(\mathfrak{V}'_1)$ and $\mathrm{L}(\mathfrak{V}'_2)$. The latter step yields an inherently non-deterministic separator. However, the transformation mentioned in (iii) requires a complementation, which results in a triply exponential bound overall.

Instead, roughly speaking, we first apply an observation from [13] to reduce to an even more specific case: We construct $\mathfrak{V}$ such that for the language C_d of all counter instruction sequences that keep the d counters above zero, we have (a) $\mathrm{L}(\mathfrak{V}) \cap C_d = \emptyset$ and (b) any separator of $\mathrm{L}(\mathfrak{V})$ and C_d can be transformed into a separator for $\mathrm{L}(\mathfrak{V}_1)$ and $\mathrm{L}(\mathfrak{V}_2)$. Then, we rely on the fact that a particular family $(B_k)_{k \in \mathbb{N}}$ of regular languages is a family of *basic separators* (a concept introduced by Czerwiński and the second author in [13]): Every language regularly separable from C_d is included in a finite union of sets B_k. Here, B_k contains all sequences of counter instructions such that at least one counter at some point falls below zero, but before that, it never exceeds the value k. We prove a version of this with complexity bounds: We show that $\mathrm{L}(\mathfrak{V}) \cap C_d = \emptyset$ implies that $\mathrm{L}(\mathfrak{V})$ is included in B_k for some doubly exponential bound k. Here, the key advantage is that we understand the structure of the B_k so well that we can just observe that the separator B_k is already deterministic. Thus, the complementation step will not result in another exponential blow-up.

Second upper bound. Theorem 3.1 provides all upper bounds for NFA separators in Table 1. It also provides all upper bounds for DFAs where the DFA bound is exponential in the corresponding NFA bound (via the powerset construction). The only exception to this is the dark gray entry: Here, the tight DFA bound is actually the same as for NFA.

▶ **Theorem 3.2.** *Let $\mathfrak{V}_1$ and $\mathfrak{V}_2$ be 1-VASS with binary updates. If $\mathrm{L}(\mathfrak{V}_1) \cap \mathrm{L}(\mathfrak{V}_2) = \emptyset$, then there exists a separating DFA with at most exponentially many states.*

For this, we observe that the states of NFA resulting from Theorem 3.1 for $d = 1$ can be equipped with a partial ordering $\leq$ such that (i) if $p \leq q$, then all words accepted from p are also accepted from q and (ii) every anti-chain in this ordering has at most polynomial size. This permits determinization without a blow-up.

Lower bounds. The lower bounds for the first row of Table 1 are known from [10]. For the others, we use two types of pairs. The first is similar to the language pairs in [10]:

$$\begin{aligned} K_{f,n} &= \{w \in \{a,b\} \mid \text{the } f(n)\text{-th last letter of } w \text{ is an } a \text{ and } |w| \geq f(n)\} \\ L_{f,n} &= \{w \in \{a,b\} \mid \text{the } f(n)\text{-th last letter of } w \text{ is a } b \text{ or } |w| < f(n)\} \end{aligned} \tag{1}$$

where $f\colon \mathbb{N} \to \mathbb{N}$ is one of the functions $n \mapsto n$ (a separating DFA needs 2^n states; the blue entries) or $n \mapsto 2^n$ (a separating DFA needs 2^{2^n} states, the yellow entry). In [10], these are used for $n \mapsto 2^{2^n}$. The second language pair consists of $L_n = \{a^m \mid m \geq 2^n\}$, and $K_n = \{a^m \mid m < 2^n\}$ (an NFA needs 2^n states, the light and dark gray entries).

4 Basic Separators

As already mentioned in the previous section we want to apply the approach from [13] to show our main theorem. To this end, we first have to introduce languages following the courses of the counters of our VASS. We introduce two (basic) actions a_i and $\overline{a_i}$ for each $1 \leq i \leq d$ to indicate that counter i gets increased resp. decreased by one. By $\Gamma_d := \{a_i, \overline{a_i} \mid 1 \leq i \leq d\}$ we denote the alphabet of basic actions. Then a word $w \in \Gamma_d^*$ encodes the course of updates of the d counters on some pseudo-run of a d-VASS. For $1 \leq i \leq d$ we introduce a homomorphism $\phi_i\colon \Gamma_d^* \to \mathbb{Z}$ induced by the equations $\phi_i(a_i) = 1$, $\phi_i(\overline{a_i}) = -1$, and $\phi_i(b) = 0$ for $b \in \Gamma_d \setminus \{a_i, \overline{a_i}\}$. In other words, $\phi_i(w)$ is the value of counter i after application of the actions specified in w.

For $w \in \Gamma_d^*$ define $\mathrm{drop}_i(w) := \min\{\phi_i(v) \mid v \text{ is a prefix of } w\} \in [-|w|, 0]$, i.e., $\mathrm{drop}_i(w)$ is the lowest value the counter i had while applying the actions in w. In a run counter i starts with value 0 and stays non-negative. Therefore, any run $w \in \Gamma_d^*$ of a d-VASS satisfies $\mathrm{drop}_i(w) = 0$. By $G_i := \{w \in \Gamma_d^* \mid \mathrm{drop}_i(w) = 0\}$ we define the language of all action sequences where counter i never falls below zero. Then the language of all runs of a d-VASS is $C_d := \bigcap_{i=1}^d G_i$. Next, we want to describe the courses $w \in \Gamma_d^*$ of pseudo-runs of a given VASS $\mathfrak{V}$. To this end, we first have to recall the notion of rational transductions:

Rational Transductions. Let Σ and Γ be two alphabets. A *transducer* is a tuple $\mathfrak{T} = (Q, \delta, I, F)$ where Q is a finite set of *states*, $\delta \subseteq Q \times \Gamma^* \times \Sigma^* \times Q$ is a finite set of *transitions*, and $I, F \subseteq Q$ are the *initial* resp. *accepting* states. A pair $(v, w) \in \Gamma^* \times \Sigma^*$ is *accepted* by $\mathfrak{T}$ if there are $q_0, q_1, \ldots, q_n \in Q$, $v_1, \ldots, v_n \in \Sigma^*$, and $w_1, \ldots, w_n \in \Gamma^*$ with $v = v_1 \ldots v_n$, $w = w_1 \ldots w_n$, $q_0 \in I$, $q_n \in F$, and $(q_{i-1}, v_i, w_i, q_i) \in \delta$ for each $1 \leq i \leq n$. The *accepted relation* of $\mathfrak{T}$ is $\mathrm{R}(\mathfrak{T}) = \{(v, w) \in \Gamma^* \times \Sigma^* \mid (v, w) \text{ is accepted by } \mathfrak{T}\}$. A relation $T \subseteq \Gamma^* \times \Sigma^*$ is called a *rational transduction* if there is a transducer $\mathfrak{T}$ with $\mathrm{R}(\mathfrak{T}) = T$. For a relation $T \subseteq \Gamma^* \times \Sigma^*$ and a language $L \in \Gamma^*$ we write $T(L)$ for the language $\{w \in \Sigma^* \mid \exists v \in L\colon (v, w) \in T\}$. Additionally, we write T^{-1} for the relation $\{(w, v) \in \Sigma^* \times \Gamma^* \mid (v, w) \in T\}$. The following connection between d-VASS and transducers is well-known:

▶ **Lemma 4.1** (cf. [16,19]). *A language $L \subseteq \Sigma^*$ is a coverability d-VASS-language if, and only if, there is a rational transduction $T \subseteq \Gamma_d^* \times \Sigma^*$ with $L = T(C_d)$.*

Proof idea. We only show the implication "⇒" (for the converse implication cf. e.g. [13]). So, let $\mathfrak{V} = (Q, \Sigma, \Delta, s, t)$ be a d-VASS with $\mathrm{L}(\mathfrak{V}) = L$. We construct the following transducer $\mathfrak{T}_{\mathfrak{V}} = (Q, \delta, \{s\}, \{t\})$: set $\delta = \{(p, \mathrm{code}(\vec{v}), a, q) \mid (p, a, \vec{v}, q) \in \Delta\}$, where $\mathrm{code}(\vec{v}) = a_1^{\vec{v}[1]} a_2^{\vec{v}[2]} \ldots a_d^{\vec{v}[d]}$ and $a_i^n := \overline{a_i}^{|n|}$ holds for $n < 0$. Then we can see $(\mathrm{R}(\mathfrak{T}_{\mathfrak{V}}))(C_d) = \mathrm{L}(\mathfrak{V})$. ◀

Regular separability is, in some sense, compatible with rational transductions:

▶ **Lemma 4.2** ([13]). *Let $K \subseteq \Sigma^*$, $L \subseteq \Gamma^*$ be two languages and $T \subseteq \Gamma^* \times \Sigma^*$ be a rational transduction. Then $K \mid T(L)$ if, and only if, $T^{-1}(K) \mid L$.*

We include the (very simple) proof of this lemma, as it hints at our proof of Theorem 5.1:

Proof. For the "only if", suppose $K \mid T(L)$ with a regular separator $R \subseteq \Sigma^*$. It is easy to check that then, $T^{-1}(R) \subseteq \Gamma^*$ is a regular separator of $T^{-1}(K)$ and L. Thus $T^{-1}(K) \mid L$ holds. Conversely, assume $T^{-1}(K) \mid L$ via the regular separator $R \subseteq \Gamma^*$. Then we also know $L \mid T^{-1}(K)$ via $\Gamma^* \setminus R$. The proof of the "only if" direction yields $T(L) \mid K$ via $T(\Gamma^* \setminus R) \subseteq \Sigma^*$. Finally, we obtain $K \mid T(L)$ via $\Sigma^* \setminus T(\Gamma^* \setminus R)$. ◀

Basic separators. From Theorem 2.3 and Lemma 4.2 we learn that two d-VASS-languages $L, K \subseteq \Sigma^*$ are regular separable if, and only if, $T^{-1}(K) \mid C_d$ holds, where T is a rational transduction with $L = T(C_d)$. Czerwiński and the second author of this work have introduced in [13] the notion of *basic separators* of any language from the language C_d. These are families of regular languages disjoint from C_d such that each regular language, which is disjoint from C_d, is included in a finite union of basic separators. For coverability d-VASS suitable basic separators are the languages $B_k \subseteq \Gamma_d^*$ which contain all action sequences having one counter $1 \le i \le d$ falling below zero, but before that, counter i never exceeds the value of k. To this end, we first define the value $\mu_i(w) := \max\{\phi_i(v) \mid v \text{ is a prefix of } w \text{ with } \mathrm{drop}_i(v) = 0\}$ of a word $w \in \Gamma_d^*$. This is the greatest value of counter i before it falls below zero for the first time (or it is the maximal value of counter i if it always stays non-negative). Then B_k (for $k \in \mathbb{N}$) is defined as follows:

$$B_k := \{w \in \Gamma_d^* \mid \exists 1 \le i \le d \colon w \notin G_i \text{ and } \mu_i(w) \le k\}\,.$$

As shown in [13], the following equivalence holds for coverability d-VASS:

▶ **Corollary 4.3.** *Let $\mathfrak{V}$ and $\mathfrak{W}$ be two d-VASS and $T \subseteq \Gamma_d^* \times \Sigma^*$ be a rational transduction with $\mathrm{L}(\mathfrak{W}) = T(C_d)$. Then the following properties are equivalent:*

1. $\mathrm{L}(\mathfrak{V}) \cap \mathrm{L}(\mathfrak{W}) = \emptyset$
2. $\mathrm{L}(\mathfrak{V}) \mid \mathrm{L}(\mathfrak{W})$
3. $T^{-1}(\mathrm{L}(\mathfrak{V})) \mid C_d$
4. *there is $k \in \mathbb{N}$ such that B_k is a regular separator of $T^{-1}(\mathrm{L}(\mathfrak{V}))$ and C_d.* ◀

In the proof of our main result Theorem 5.1, we will show that there is a "small" $k \in \mathbb{N}$ such that B_k separates $T^{-1}(\mathrm{L}(\mathfrak{V}))$ and C_d.

5 Upper Bounds

We now prove Theorems 3.1 and 3.2. For Theorem 3.1, we prove a more concrete bound:

▶ **Theorem 5.1.** *Let $\mathfrak{V}_i = (Q_i, \Sigma, \Delta_i, s_i, t_i)$ (for $i = 1, 2$) be two d-VASS with $\mathrm{L}(\mathfrak{V}_1) \cap \mathrm{L}(\mathfrak{V}_2) = \emptyset$. Then there is an NFA of size at most $\mathcal{O}(|Q_1| \cdot \|\Delta_1\|^d \cdot \mathrm{Rack}(\mathfrak{V}_1 \times \mathfrak{V}_2)^d)$ separating $\mathrm{L}(\mathfrak{V}_1)$ and $\mathrm{L}(\mathfrak{V}_2)$.*

This clearly implies Theorem 3.1. To show this, let $\mathfrak{V}_i = (Q_i, \Sigma, \Delta_i, s_i, t_i)$ (for $i = 1, 2$) be two disjoint d-VASS and let $\mathfrak{V}_1 \times \mathfrak{V}_2 = (Q_1 \times Q_2, \Sigma, \Delta, (s_1, s_2), (t_1, t_2))$ be the product VASS as constructed in Lemma 2.1. Note that $\mathrm{L}(\mathfrak{V}_1 \times \mathfrak{V}_2) = \emptyset$ holds due to the disjointness of $\mathfrak{V}_1$ and $\mathfrak{V}_2$. Additionally, let $\mathfrak{T}_{\mathfrak{V}_1}$ be the transducer constructed in the proof of Lemma 4.1

satisfying the property $(\mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1}))(C_d) = \mathrm{L}(\mathfrak{V}_1)$. According to Corollary 4.3 the assumption $\mathrm{L}(\mathfrak{V}_1) \cap \mathrm{L}(\mathfrak{V}_2) = \emptyset$ implies $(\mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1}))^{-1}(\mathrm{L}(\mathfrak{V}_2)) \mid C_d$. Our aim is to find a "small" number $\hat{k} \in \mathbb{N}$ such that $B_{\hat{k}}$ is a regular separator of $K(\mathfrak{V}_1, \mathfrak{V}_2) := (\mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1}))^{-1}(\mathrm{L}(\mathfrak{V}_2))$ and C_d.

Let $w \in K(\mathfrak{V}_1, \mathfrak{V}_2) \subseteq \Gamma_d^*$ be some word. Then there is another word $w' \in \Sigma^*$ with $(w, w') \in R(\mathfrak{T}_{\mathfrak{V}_1})$ and $w' \in L(\mathfrak{V}_2)$. From this fact we obtain a w'-labeled pseudo-run of $\mathfrak{V}_1$ from s_1 to t_1 such that w encodes the counter updates of this pseudo-run. Additionally, we obtain a w'-labeled run of $\mathfrak{V}_2$ from s_2 to t_2. We can compose these two pseudo-runs to one w'-labeled pseudo-run of $\mathfrak{V}_1 \times \mathfrak{V}_2$ from (s_1, s_2) to (t_1, t_2). So, there is a sequence $\pi_w := ((p_i, q_i), (\vec{x_i}, \vec{y_i}))_{0 \le i \le n}$ of pseudo-configurations and transitions $((p_{i-1}, q_{i-1}), b_i, (\vec{u_i}, \vec{v_i}), (p_i, q_i)) \in \Delta$ (for $1 \le i \le n$) with $(p_0, q_0) = (s_1, s_2)$, $(p_n, q_n) = (t_1, t_2)$, $(\vec{x_0}, \vec{y_0}) = (\vec{0}, \vec{0})$, $(\vec{x_i}, \vec{y_i}) = (\vec{x_{i-1}} + \vec{u_i}, \vec{y_{i-1}} + \vec{v_i})$ for each $1 \le i \le n$, $w = \mathrm{code}(\vec{u_1}) \ldots \mathrm{code}(\vec{u_n})$ (note that this equation holds by the choice of our transducer $\mathfrak{T}_{\mathfrak{V}_1}$ from Lemma 4.1), and $w' = b_1 b_2 \ldots b_n$. Since we have $\mathrm{L}(\mathfrak{V}_1) \cap \mathrm{L}(\mathfrak{V}_2) = \emptyset$ by assumption, π_w is a pseudo-run of $\mathfrak{V}_1 \times \mathfrak{V}_2$, but actually not a run, i.e., at least one counter of $\mathfrak{V}_1 \times \mathfrak{V}_2$ falls below zero at some time. As stated above, w' labels a run of $\mathfrak{V}_2$, i.e., no counter of $\mathfrak{V}_2$ ever falls below zero. This implies the existence of $0 \le i \le n$ with $\vec{x_i} \in \mathbb{Z}^d \setminus \mathbb{N}^d$.

Set $\hat{k} := \|\Delta_1\| \cdot \mathrm{Rack}(\mathfrak{V}_1 \times \mathfrak{V}_2)$. By definition we know that $B_{\hat{k}} \cap C_d = \emptyset$ holds. So, we only have to prove $K(\mathfrak{V}_1, \mathfrak{V}_2) \subseteq B_{\hat{k}}$, i.e., we show that for each $w \in K(\mathfrak{V}_1, \mathfrak{V}_2)$ there is a counter $1 \le i \le d$ having a value at most $\hat{k}$ before falling below zero. We show this result by contradiction: assume there is $w \in K(\mathfrak{V}_1, \mathfrak{V}_2) \setminus B_{\hat{k}}$. This means, in w for each counter $1 \le i \le d$ we have two possibilities: (i) the counter i stays non-negative (i.e., $w \in G_i$) or (ii) the counter i falls below zero and before this happens for the first time it exceeds the value $\hat{k}$. We construct then another word $v \in K(\mathfrak{V}_1, \mathfrak{V}_2) \setminus B_{\hat{k}}$ having more counters $1 \le i \le d$ satisfying $v \in G_i$ than w. By induction we obtain a word $v \in K(\mathfrak{V}_1, \mathfrak{V}_2) \setminus B_{\hat{k}}$ with $v \in G_i$ for each $1 \le i \le d$. This implies $v \in C_d$ and therefore the existence of another word $v' \in \Sigma^*$ with $(v, v') \in R(\mathfrak{T}_{\mathfrak{V}_1})$ and $v' \in L(\mathfrak{V}_2)$. Then v' is the label of runs in $\mathfrak{V}_1$ (since $v \in C_d$) and $\mathfrak{V}_2$ (since $v' \in L(\mathfrak{V}_2)$). Hence, we obtain $v' \in \mathrm{L}(\mathfrak{V}_1) \cap \mathrm{L}(\mathfrak{V}_2)$, which is a contradiction to our assumption that $\mathfrak{V}_1$ and $\mathfrak{V}_2$ accept disjoint languages.

▶ **Lemma 5.2.** *We have $K(\mathfrak{V}_1, \mathfrak{V}_2) \subseteq B_{\hat{k}}$.*

To show this lemma we first have to introduce another notion: let $I \subseteq \{1, \ldots, d\}$. For a vector $\vec{v} \in \mathbb{Z}^d$ we define the *projection* $\vec{v}^I$ to the components specified in I as follows: $\vec{v}^I[j] = \vec{v}[j]$ if $j \in I$ and $\vec{v}^I[j] = 0$ if $j \notin I$. Now, let $\mathfrak{V} = (Q, \Sigma, \Delta, s, t)$ be a d-VASS. For a pseudo-configuration $c = (q, \vec{v}) \in Q \times \mathbb{Z}^d$ we define $c^I := (q, \vec{v}^I)$. The *projection* $\mathfrak{V}^I$ of $\mathfrak{V}$ to I is the d-VASS $\mathfrak{V}^I = (Q, \Sigma, \Delta^I, s, t)$ with $\Delta^I := \{(p, a, \vec{v}^I, q) \mid (p, a, \vec{v}, q) \in \Delta\}$.

Proof. Let $w \in K(\mathfrak{V}_1, \mathfrak{V}_2)$. Towards a contradiction we suppose that $w \notin B_{\hat{k}}$ holds. Then for each $1 \le i \le d$ we have either $w \in G_i$ (i.e., the i-th counter never falls below 0), or $w \notin G_i$ and $\mu_i(w) > \hat{k}$ (i.e., the i-th counter reaches a value $> \hat{k}$ before falling below 0 for its first time). Let $I_w \subseteq \{1, \ldots, d\}$ be the set of indices $1 \le i \le d$ with $w \in G_i$. Assuming $|I_w| < d$ we want to construct from w another word $v \in K(\mathfrak{V}_1, \mathfrak{V}_2) \setminus B_{\hat{k}}$ with $|I_v| > |I_w|$.

So, let $i \in \{1, \ldots, d\} \setminus I_w$ be the index of the last counter exceeding the upper bound $\hat{k}$ before it falls below zero for its first time, i.e., i is the number of the counter having the longest prefix w_1 of w with $\mu_i(w_1) \le \hat{k}$. Additionally, let $0 \le j < n$ be the first computational step in which counter i exceeds $\hat{k}$, i.e., $\vec{x_j}[i] > \hat{k}$ and $\vec{x_h}[i] \le \hat{k}$ for each $0 \le h < j$. Now, restrict the pseudo-run π_w to the counters in I_w and all of $\mathfrak{V}_2$'s counters. Since none of these counters falls below zero, the pseudo-run π_w is actually a run in $\mathfrak{V}_1^{I_w} \times \mathfrak{V}_2$. This especially holds for π_w's sub-run from $((p_j, q_j), (\vec{x_j}^{I_w}, \vec{y_j}))$ to $((p_n, q_n), (\vec{x_n}^{I_w}, \vec{y_n}))$ in $\mathfrak{V}_1^{I_w} \times \mathfrak{V}_2$. Since $(p_n, q_n) = (t_1, t_2)$ holds, there is—according to Theorem 2.2—also a run from $((p_j, q_j), (\vec{x_j}^{I_w}, \vec{y_j}))$ to some

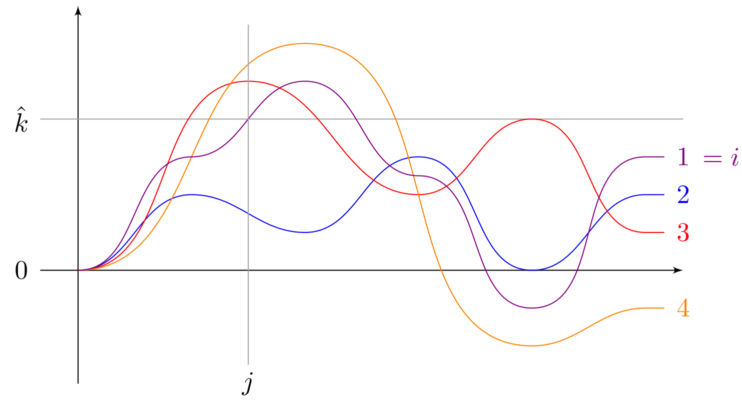

Figure 1 The values of the counters in the course of a run encoded by some word $w \in K(\mathfrak{V}_1, \mathfrak{V}_2) \setminus B_{\hat{k}}$. The values of the counters 2 and 3 never fall below zero implying $w \in G_2 \cap G_3$. The counters 1 and 4 exceed the value $\hat{k}$ before they fall below zero the first time (i.e., $w \notin G_1 \cup G_4$ and $\mu_1(w), \mu_4(w) > \hat{k}$). Since counter 1 exceeds $\hat{k}$ after counter 4 does, we choose $i = 1$ in our proof. The first intersection of counter 1's curve and $\hat{k}$ marks the step j.

configuration $((p_n, q_n), (\vec{x_m'}^{I_w}, \vec{y_m'}))$ of length at most $\operatorname{Rack}(\mathfrak{V}_1^{I_w} \times \mathfrak{V}_2) \leq \operatorname{Rack}(\mathfrak{V}_1 \times \mathfrak{V}_2)$. Since $\mathfrak{V}_1^{I_w}$ is a projection of $\mathfrak{V}_1$ to the counters in I_w, we can also extend this run to a pseudo-run with all counters. Let $((p_h', q_h'), (\vec{x_h'}, \vec{y_h'}))_{j \leq h \leq m}$ be this pseudo-run extended to all $2d$ counters satisfying $((p_j, q_j), (\vec{x_j}, \vec{y_j})) = ((p_j', q_j'), (\vec{x_j'}, \vec{y_j'}))$, $p_m' = p_n = t_1$, and $q_m' = q_n = t_2$. Let $((p_{h-1}', q_{h-1}'), b_h', (\vec{u_h'}, \vec{v_h'}), (p_h', q_h')) \in \Delta$ be the corresponding transitions (for $j < h \leq m$). Set $v := \operatorname{code}(\vec{u_1}) \dots \operatorname{code}(\vec{u_j}) \operatorname{code}(\vec{u_{j+1}'}) \dots \operatorname{code}(\vec{u_m'})$. Our next aim is to prove that $|I_v| > |I_w|$ and $v \in K(\mathfrak{V}_1, \mathfrak{V}_2) \setminus B_{\hat{k}}$ holds.

▷ Claim 5.3. $|I_v| > |I_w|$

Proof. We show $I_w \uplus \{i\} \subseteq I_v$. By the choice of our pseudo-run we have $I_w \subseteq I_v$ (recall that all counters from I_w always stay ≥ 0). So, we only have to show $i \in I_v$. We have $\vec{x_j}[i] = \vec{x_j'}[i] > \hat{k} = \|\Delta_1\| \cdot \operatorname{Rack}(\mathfrak{V}_1 \times \mathfrak{V}_2)$, $m - j \leq \operatorname{Rack}(\mathfrak{V}_1 \times \mathfrak{V}_2)$, and $\vec{u_h'}[i] \leq \|\Delta_1\|$ for each $j < h \leq m$. Hence, we obtain $\vec{x_h'}[i] \geq 0$ for each $j \leq h \leq m$, i.e., on our new run the counter i never falls below zero. We infer $v \in G_i$ and, therefore, $i \in I_v$. ◁

▷ Claim 5.4. $v \in K(\mathfrak{V}_1, \mathfrak{V}_2) \setminus B_{\hat{k}}$

Proof. First, we show $v \in K(\mathfrak{V}_1, \mathfrak{V}_2)$. The sequence

$$((p_{h-1}, q_{h-1}), b_h, (\vec{u_h}, \vec{v_h}), (p_h, q_h))_{1 \leq h \leq j}, ((p_{h-1}', q_{h-1}'), b_h', (\vec{u_h'}, \vec{v_h'}), (p_h', q_h'))_{j < h \leq m}$$

of transitions in $\mathfrak{V}_1 \times \mathfrak{V}_2$ induces some accepting run $(q_0, \vec{0}) \xrightarrow{b_1 \dots b_j b_{j+1}' \dots b_m'}_{\mathfrak{V}_2} (q_m', \vec{v_m'})$ in $\mathfrak{V}_2$, i.e., we have $b_1 \dots b_j b_{j+1}' \dots b_m' \in \mathrm{L}(\mathfrak{V}_2)$. Additionally, the word v encodes the counters' course of updates in the transition sequence $(p_{h-1}, b_h, \vec{u_h}, p_h)_{1 \leq h \leq j}, (p_{h-1}', b_h', \vec{u_h'}, p_h')_{j < h \leq m}$ in $\mathfrak{V}_1$. According to $p_0 = s_1$, $p_j = p_j'$, and $p_m' = t_1$ this transition sequence is a pseudo-run of $\mathfrak{V}_1$ labeled by $b_1 \dots b_j b_{j+1}' \dots b_m'$. By the choice of our transducer $\mathfrak{T}_{\mathfrak{V}_1}$ (which is the one from the proof of Lemma 4.1), we learn $(v, b_1 \dots b_j b_{j+1}' \dots b_m') \in \mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1})$ implying $v \in (\mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1}))^{-1}(b_1 \dots b_j b_{j+1}' \dots b_m')$. We finally obtain $v \in K(\mathfrak{V}_1, \mathfrak{V}_2)$.

Now, we prove $v \notin B_{\hat{k}}$. If we have $I_v = \{1, \dots, d\}$ then we learn $v \in \bigcap_{h=1}^{d} G_h = C_d$. Since $C_d \cap B_{\hat{k}} = \emptyset$ holds, we have $v \notin B_{\hat{k}}$ in this case. Now, assume $I_v \neq \{1, \dots, d\}$. Let

$i' \in \{1, \ldots, d\} \setminus I_v$ be arbitrary. From $I_w \cup \{i\} \subseteq I_v$ we learn $i' \notin I_w$ and $i' \neq i$. Hence, we have $v, w \notin G_{i'}$. Since $w \notin B_{\hat{k}}$ holds (by the assumption at the outset of this claim's proof), we infer $\mu_{i'}(w) > \hat{k}$, i.e., the counter i' exceeds $\hat{k}$ in w before it falls below zero for the first time. Additionally, in v the counter i' falls below zero sometime. We have to show that it exceeds the value $\hat{k}$ before it first drops below zero.

Recall that i was the counter with the longest prefix w_1 of w with $\mu_i(w_1) \leq \hat{k}$. This implies $\mu_{i'}(w_1) > \hat{k}$. Note that w_1 is a prefix of $\mathrm{code}(\vec{u_1}) \ldots \mathrm{code}(\vec{u_j})$ and therefore also of v. Hence, we have $\mu_{i'}(v) > \hat{k}$. Since i' was arbitrary, this holds for all counters in $\{1, \ldots, d\} \setminus I_v$. In other words, for each $h \in \{1, \ldots, d\}$ we have either $v \in G_h$ or $\mu_h(v) > \hat{k}$. Hence, $v \notin B_{\hat{k}}$ holds in this case. ◁

So, we have learned that there is another word $v \in K(\mathfrak{V}_1, \mathfrak{V}_2) \setminus B_{\hat{k}}$ having more non-negative counters I_v than w. Finally, induction yields a word $\hat{v} \in K(\mathfrak{V}_1, \mathfrak{V}_2) \setminus B_{\hat{k}}$ with $I_{\hat{v}} = \{1, \ldots, d\}$, i.e., $\hat{v} \in \bigcap_{h=1}^{d} G_h = C_d$. This implies $\hat{v} \in C_d \cap K(\mathfrak{V}_1, \mathfrak{V}_2)$ - a contradiction to $K(\mathfrak{V}_1, \mathfrak{V}_2) \mid C_d$. ◀

With the help of Lemma 5.2 we are able to finally prove our main result Theorem 5.1.

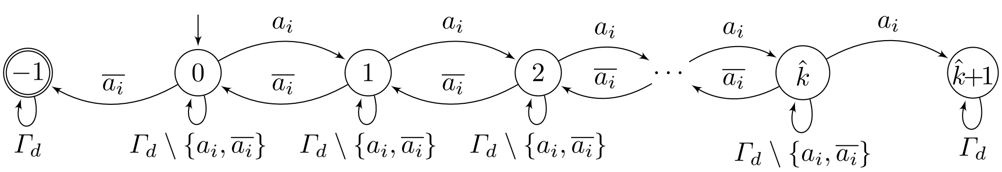

Figure 2 A DFA $\mathfrak{A}_i$ accepting the language $B_{\hat{k},i}$. It simulates the counter i bounded by $\hat{k}$.

Proof of Theorem 5.1. Since we have $K(\mathfrak{V}_1, \mathfrak{V}_2) \subseteq B_{\hat{k}}$ and $B_{\hat{k}} \cap C_d = \emptyset$, the set $B_{\hat{k}}$ is a separator of $K(\mathfrak{V}_1, \mathfrak{V}_2)$ and C_d. This language is also regular: in Figure 2 we depict a DFA $\mathfrak{A}_i$ accepting the language $B_{\hat{k},i} := \{w \in \Gamma_d^* \mid w \notin G_i \text{ and } \mu_i(w) \leq \hat{k}\}$ for $1 \leq i \leq d$. Since $B_{\hat{k}} = \bigcup_{i=1}^{d} B_{\hat{k},i}$ holds, we obtain a DFA accepting $B_{\hat{k}}$ using the classical product construction. The resulting DFA has the size $\prod_{i=1}^{d} |\mathfrak{A}_i| = \prod_{i=1}^{d} (\hat{k} + 3) \in \mathcal{O}(\hat{k}^d)$.

We have seen that $B_{\hat{k}}$ separates $K(\mathfrak{V}_1, \mathfrak{V}_2)$ and C_d. With the same arguments as in Lemma 4.2, one shows that if R witnesses $T^{-1}(K) \mid L$, then $T(\Sigma^* \setminus R)$ witnesses $T(L) \mid K$. Since $K(\mathfrak{V}_1, \mathfrak{V}_2) = \mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1})^{-1}(\mathrm{L}(\mathfrak{V}_2))$ and $\mathrm{L}(\mathfrak{V}_1) = \mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1})(C_d)$, we conclude that

$$\mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1})(\Gamma_d^* \setminus B_{\hat{k}}) \tag{2}$$

witnesses $\mathrm{L}(\mathfrak{V}_1) \mid \mathrm{L}(\mathfrak{V}_2)$. Since we have a DFA of size $\mathcal{O}(\hat{k}^d)$ for $B_{\hat{k}}$ and thus such a DFA for $\Gamma_d^* \setminus B_{\hat{k}}$, we obtain an NFA for (2) of size $\mathcal{O}(|Q_1| \cdot \hat{k}^d) = \mathcal{O}(|Q_1| \cdot \|\Delta_1\|^d \cdot \mathrm{Rack}(\mathfrak{V}_1 \times \mathfrak{V}_2)^d)$. ◀

The term $\mathcal{O}(|Q_1| \cdot \|\Delta_1\|^d \cdot \mathrm{Rack}(\mathfrak{V}_1 \times \mathfrak{V}_2)^d)$ from Theorem 5.1 is doubly exponential in d (and polynomial in the remaining numbers). In other words, for two given disjoint d-VASS $\mathfrak{V}_1$ and $\mathfrak{V}_2$ there is a doubly exponential sized NFA separating their languages $\mathrm{L}(\mathfrak{V}_1)$ and $\mathrm{L}(\mathfrak{V}_2)$. If we are looking for a deterministic automaton separating these languages, we can use the power set construction to obtain a DFA of triply exponential size. The lower bounds by Czerwiński et. al. [10] show that these upper bounds are tight.

▶ **Corollary 5.5.** *From a given number $d \in \mathbb{N}^+$ and two disjoint d-VASS $\mathfrak{V}_1$ and $\mathfrak{V}_2$ we can compute*

(1) *an NFA separating* $\mathrm{L}(\mathfrak{V}_1)$ *and* $\mathrm{L}(\mathfrak{V}_2)$ *of size doubly exponential in* d*,* $|\mathfrak{V}_1|$*, and* $|\mathfrak{V}_2|$*.*
(2) *a DFA separating* $\mathrm{L}(\mathfrak{V}_1)$ *and* $\mathrm{L}(\mathfrak{V}_2)$ *of size triply exponential in* d*,* $|\mathfrak{V}_1|$*, and* $|\mathfrak{V}_2|$*.*

Proof. (1) By Theorem 5.1 we can compute an NFA separating $\mathrm{L}(\mathfrak{V}_1)$ and $\mathrm{L}(\mathfrak{V}_2)$ with the following number of states:

$$\begin{aligned} &\mathcal{O}(|Q_1| \cdot \|\Delta_1\|^d \cdot \mathrm{Rack}(\mathfrak{V}_1 \times \mathfrak{V}_2)^d) \\ &= \mathcal{O}(|Q_1| \cdot \|\Delta_1\|^d \cdot (|Q_1| \cdot |Q_2| \cdot \max\{\|\Delta_1\|, \|\Delta_2\|\} + 2)^{((6d)!+1)\cdot d}) \,, \end{aligned}$$

which is doubly exponential in d, $|\mathfrak{V}_1|$, and $|\mathfrak{V}_2|$.

(2) We can determinize the NFA from (1) using the classical power set construction. This results in an equivalent DFA of size exponential in the size of the NFA. ◀

Since the exponent of the term $\mathcal{O}(|Q_1| \cdot \|\Delta_1\|^d \cdot \mathrm{Rack}(\mathfrak{V}_1 \times \mathfrak{V}_2)^d)$ only depends on the dimension d, we could also ask for an upper bound of an NFA or DFA separating the languages of two VASS of fixed dimension. In this scenario we have to distinguish two cases: the numbers in our VASS are encoded in unary or binary. First, we consider the unary case. Here, we can construct a separating NFA of polynomial size and a DFA of exponential size.

▶ **Corollary 5.6.** *Fix a number* $d \in \mathbb{N}^+$*. From two disjoint* d*-VASS* $\mathfrak{V}_1$ *and* $\mathfrak{V}_2$ *in which the numbers are encoded in unary, we can compute*
(1) *an NFA separating* $\mathrm{L}(\mathfrak{V}_1)$ *and* $\mathrm{L}(\mathfrak{V}_2)$ *of size polynomial in* $|\mathfrak{V}_1|$ *and* $|\mathfrak{V}_2|$*.*
(2) *a DFA separating* $\mathrm{L}(\mathfrak{V}_1)$ *and* $\mathrm{L}(\mathfrak{V}_2)$ *of size exponential in* $|\mathfrak{V}_1|$ *and* $|\mathfrak{V}_2|$*.*

Proof. (1) Since d is assumed to be fixed, the size of the regular separator of $\mathrm{L}(\mathfrak{V}_1)$ and $\mathrm{L}(\mathfrak{V}_2)$ from Theorem 5.1 is a polynomial in the sizes of $\mathfrak{V}_1$ and $\mathfrak{V}_2$.
(2) To achieve this result, we only have to determinize the NFA from statement (1). ◀

Now, we have to consider VASS of fixed dimension d with binary encoded numbers. To this end, we first have to introduce a binary norm: for a vector $\vec{v} \in \mathbb{Z}^d$ set $\|\vec{v}\|_2 := \log \|\vec{v}\|$. Based on this, we define the binary norm $\|\Delta\|_2$ of a set of transitions Δ. Slightly abusing terminology, when we speak of *VASS with binary encoding* (or *with binary encoded numbers*), then this only means we measure its size with $\|\cdot\|_2$ in place of $\|\cdot\|$. In this case, for two given VASS we find a separating NFA of exponential size and a separating DFA of doubly exponential size.

▶ **Corollary 5.7.** *Fix a number* $d \in \mathbb{N}^+$*. From two disjoint* d*-VASS* $\mathfrak{V}_1$ *and* $\mathfrak{V}_2$ *with binary encoding, we can compute*
(1) *an NFA separating* $\mathrm{L}(\mathfrak{V}_1)$ *and* $\mathrm{L}(\mathfrak{V}_2)$ *of size exponential in* $|\mathfrak{V}_1|$ *and* $|\mathfrak{V}_2|$*.*
(2) *a DFA separating* $\mathrm{L}(\mathfrak{V}_1)$ *and* $\mathrm{L}(\mathfrak{V}_2)$ *of size doubly exponential in* $|\mathfrak{V}_1|$ *and* $|\mathfrak{V}_2|$*.*

Proof. (1) Since we encode numbers in binary the values $\|\Delta_1\|$ and $\|\Delta_2\|$ are exponential in the description size of $\mathfrak{V}_1$ resp. $\mathfrak{V}_2$. Hence, the NFA constructed in Theorem 5.1 has size exponential in the sizes of $\mathfrak{V}_1$ and $\mathfrak{V}_2$.
(2) Again, this is a direct consequence of the first statement using the classical power set construction to determinize the constructed NFA. ◀

5.1 Upper Bound for Binary Encoded 1-VASS

Interestingly, the given upper bound for a DFA separating the languages of two given binary encoded VASS of dimension 1 is not tight, yet. We can use a better construction than the classical power set construction to determinize our constructed separating NFA. In this case, we obtain a DFA which also has exponential size (in comparison to doubly exponential size with the power set construction).

▶ **Theorem 5.8.** *Given disjoint* 1-*VASS* $\mathfrak{V}_1$ *and* $\mathfrak{V}_2$ *with binary-encoded numbers, we can compute a DFA separating* $\mathrm{L}(\mathfrak{V}_1)$ *and* $\mathrm{L}(\mathfrak{V}_2)$ *of size exponential in* $|\mathfrak{V}_1|$ *and* $|\mathfrak{V}_2|$.

We take a closer look at the resulting NFA constructed in the last step of the proof of Theorem 5.1 (resp. in Corollary 5.7(1)). With the knowledge about this special NFA, we will apply an improved power set construction resulting in a DFA separating $\mathrm{L}(\mathfrak{V}_1)$ and $\mathrm{L}(\mathfrak{V}_2)$ without the exponential blowup.

So, let $\mathfrak{V}_i = (Q_i, \Delta_i, s_i, t_i)$ be two 1-VASS, $\mathfrak{T}_{\mathfrak{V}_1} = (Q_1, \delta_1, \{s_1\}, \{t_1\})$ be the rational transducer constructed from $\mathfrak{V}_1$ as described in the proof of Lemma 4.1 and let $\mathfrak{A} = (S, \Gamma_1, \delta_{\mathfrak{A}}, \{0\}, F_{\mathfrak{A}})$ be the DFA depicted in Figure 2 accepting $B_{\hat{k}}$ where $\hat{k} = \|\Delta_1\| \cdot \mathrm{Rack}(\mathfrak{V}_1 \times \mathfrak{V}_2)$. In other words, we have $S = \{-1, 0, 1, \dots, \hat{k}, \hat{k}+1\}$ and $F_{\mathfrak{A}} = \{-1\}$. The complement of $B_{\hat{k}}$ is accepted by the DFA $\overline{\mathfrak{A}} = (S, \Gamma_1, \delta_{\mathfrak{A}}, \{0\}, F_{\overline{\mathfrak{A}}})$ with $F_{\overline{\mathfrak{A}}} = \{0, 1, \dots, \hat{k}+1\}$. In the following let $\leq$ denote the natural ordering on $S \subseteq \mathbb{Z}$. Then we can observe that $\overline{\mathfrak{A}}$'s transition relation $\delta_{\mathfrak{A}}$ is compatible with the ordering $\leq$:

▶ **Observation 5.9.** *Let* $w \in \Gamma_1^*$ *be a word and* $m, m', n \in S$ *with* $m \xrightarrow{w}_{\overline{\mathfrak{A}}} n$ *and* $m' \geq m$. *Then there is* $n' \in S$ *with* $n' \geq n$ *and* $m' \xrightarrow{w}_{\overline{\mathfrak{A}}} n'$. ◀

In the next step we apply the rational transduction $\mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1})$ to $\mathrm{L}(\overline{\mathfrak{A}}) = \Gamma_1^* \setminus B_{\hat{k}}$. We do this with the help of the classical construction resulting in the following NFA $\mathfrak{B} = (Q_{\mathfrak{B}}, \Sigma, \delta_{\mathfrak{B}}, I_{\mathfrak{B}}, F_{\mathfrak{B}})$ accepting $(\mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1}))(\Gamma_1^* \setminus B_{\hat{k}})$:

- $Q_{\mathfrak{B}} = Q_1 \times S$ and $I_{\mathfrak{B}} = \{(s_1, 0)\}$,
- for all $(p,m), (q,n) \in Q_{\mathfrak{B}}$ and $a \in \Sigma$: $((p,m), a, (q,n)) \in \delta_{\mathfrak{B}}$ if, and only if, there is $w \in \Gamma_1^*$ with $p \xrightarrow{(w\,,\,a)}_{\mathfrak{T}_{\mathfrak{V}_1}} q$ and $m \xrightarrow{w}_{\overline{\mathfrak{A}}} n$, and
- for all $(p,m) \in Q_{\mathfrak{B}}$: $(p,m) \in F_{\mathfrak{B}}$ if, and only if, there are $w \in \Gamma_1^*$ and $n \in F_{\overline{\mathfrak{A}}}$ with $p \xrightarrow{(w\,,\,\varepsilon)}_{\mathfrak{T}_{\mathfrak{V}_1}} t_1$ and $m \xrightarrow{w}_{\overline{\mathfrak{A}}} n$.

Hence, $\mathfrak{B}$ is the separating NFA of exponential size from Corollary 5.7(1). We show next that the determinization of $\mathfrak{B}$ is possible without exponential blowup. To this end, we first need the following observation of $\mathfrak{B}$'s behavior namely that the compatibility of $\overline{\mathfrak{A}}$'s transition relation $\delta_{\mathfrak{A}}$ with $\leq$ is passed on to $\mathfrak{B}$'s transition relation $\delta_{\mathfrak{B}}$:

▶ **Lemma 5.10.** *Let* $w \in \Sigma^*$, $(p,m), (q,n) \in Q_{\mathfrak{B}}$, *and* $m' \in S$ *with* $(p,m) \xrightarrow{w}_{\mathfrak{B}} (q,n)$ *and* $m' \geq m$. *Then there is* $n' \in S$ *with* $n' \geq n$ *and* $(p,m') \xrightarrow{w}_{\mathfrak{B}} (q,n')$.

Proof. We prove this by induction on the length of w. If $w = \varepsilon$, then our statement is true: since $\mathfrak{B}$ has no ε-transitions, $(p,m) \xrightarrow{\varepsilon}_{\mathfrak{B}} (q,n)$ implies $p = q$ and $m = n$. Therefore, our statement holds for $n' = m'$.

Now, assume $w = w'a$ for some word $w' \in \Sigma^*$ and a letter $a \in \Sigma$. From $(p,m) \xrightarrow{w}_{\mathfrak{B}} (q,n)$ we learn that there is an intermediate state $(r,\ell) \in Q_{\mathfrak{B}}$ with $(p,m) \xrightarrow{w'}_{\mathfrak{B}} (r,\ell) \xrightarrow{a}_{\mathfrak{B}} (q,n)$. Since $|w'| < |w|$ holds, the induction hypothesis yields an $\ell' \in S$ with $\ell' \geq \ell$ and $(p,m') \xrightarrow{w'}_{\mathfrak{B}} (r,\ell')$. By the definition of the transition relation of $\mathfrak{B}$ we obtain from $(r,\ell) \xrightarrow{a}_{\mathfrak{B}} (q,n)$ a word $v \in \Gamma_1^*$ with $r \xrightarrow{(v,a)}_{\mathfrak{T}_{\mathfrak{V}_1}} q$ and $\ell \xrightarrow{v}_{\overline{\mathfrak{A}}} n$. According to Observation 5.9 there is $n' \in S$ with $n' \geq n$ and $\ell' \xrightarrow{v}_{\overline{\mathfrak{A}}} n'$. But this implies $(r,\ell') \xrightarrow{a}_{\mathfrak{B}} (q,n')$ and therefore $(p,m') \xrightarrow{w}_{\mathfrak{B}} (q,n')$. ◀

We can also show that the set of accepting states of $\mathfrak{B}$ is upwards closed wrt. the natural ordering of its set of states. This is the following lemma:

▶ **Lemma 5.11.** *Let* $(p,m) \in F_{\mathfrak{B}}$ *and* $m' \in S$ *with* $m' \geq m$. *Then we also have* $(p,m') \in F_{\mathfrak{B}}$.

Proof. By definition of $F_{\mathfrak{B}}$ there are $w \in \Gamma_1^*$ and $n \in F_{\overline{\mathfrak{A}}} = \{0, 1, \dots, \hat{k}+1\}$ with $p \xrightarrow{(w,\varepsilon)}_{\mathfrak{T}_{\mathfrak{V}_1}} t_1$ and $m \xrightarrow{w}_{\overline{\mathfrak{A}}} n$. Due to Observation 5.9 there is $n' \in S$ with $n' \geq n$ and $m' \xrightarrow{w}_{\overline{\mathfrak{A}}} n'$. Since $n' \geq n \geq 0$ holds, we also learn $n' \in F_{\overline{\mathfrak{A}}}$ implying $(p, m') \in F_{\mathfrak{B}}$. ◀

Finally, we have to determinize the NFA $\mathfrak{B}$. To this end, we recall the classical power set construction of $\mathfrak{B}$: the result of this construction is the DFA $\mathfrak{P} = (2^{Q_{\mathfrak{B}}}, \Sigma, \delta_{\mathfrak{P}}, \{\iota_{\mathfrak{P}}\}, F_{\mathfrak{P}})$ where

- $\iota_{\mathfrak{P}} = \{(s_1, 0)\}$,
- $(X, a, Y) \in \delta_{\mathfrak{P}}$ if, and only if, $Y = \{y \in Q_{\mathfrak{B}} \mid \exists x \in X \colon (x, a, y) \in \delta_{\mathfrak{B}}\}$, and
- $F_{\mathfrak{P}} = \{X \subseteq Q_{\mathfrak{B}} \mid X \cap F_{\mathfrak{B}} \neq \emptyset\}$.

By induction we learn that $X \xrightarrow{w}_{\mathfrak{P}} Y$ holds if, and only if, Y is exactly the set of states that are reachable from X via w, i.e., $y \in Y$ iff there is $x \in X$ with $x \xrightarrow{w}_{\mathfrak{B}} y$. In particular, if Y is accepting and we have $\iota_{\mathfrak{P}} \xrightarrow{w}_{\mathfrak{P}} Y$, then there is $y \in Y \cap F_{\mathfrak{B}} \neq \emptyset$ with $(s_1, 0) \xrightarrow{w}_{\mathfrak{B}} y$. This means, an accepting run in $\mathfrak{P}$ also witnesses an accepting run in $\mathfrak{B}$.

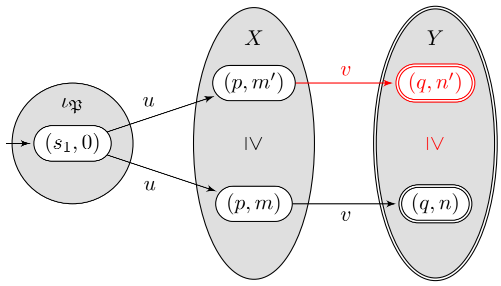

Figure 3 Visualization of the power set construction on the NFA $\mathfrak{B}$. The states in gray are states of the DFA $\mathfrak{P}$, the white ones are states of $\mathfrak{B}$. The reachability of and acceptance of (q, n') (red) is ensured by Lemmas 5.10 and 5.11.

Now, let $X \subseteq Q_{\mathfrak{B}}$ be some intermediate state of this w-labeled run from $\{(s_1, 0)\}$ to Y, i.e., we have $\iota_{\mathfrak{P}} \xrightarrow{u}_{\mathfrak{P}} X \xrightarrow{v}_{\mathfrak{P}} Y$ with $w = uv$. Let $(p, m) \in X$ and $(q, n) \in Y \cap F_{\mathfrak{B}}$ with $(s_1, 0) \xrightarrow{u}_{\mathfrak{B}} (p, m) \xrightarrow{v}_{\mathfrak{B}} (q, n)$. Assume that there is another state $(p, m') \in X$ with $m' \geq m$. Then Lemmas 5.10 and 5.11 state that there is also another state $(q, n') \in Y \cap F_{\mathfrak{B}}$ with $n' \geq n$ and $(s_1, 0) \xrightarrow{u}_{\mathfrak{B}} (p, m') \xrightarrow{v}_{\mathfrak{B}} (q, n')$, which also witnesses acceptance of w (cf. Figure 3, colored in red). This means, the set of w-labeled accepting runs of $\mathfrak{B}$ also is in some sense upwards closed. Therefore, it suffices to store only the greatest value $m \in S$ for each state $p \in Q_1$ such that $(p, m) \in X$ holds. This can be represented by a partial mapping from Q_1 into S. Here, we extend these partial mappings to maps with the help of a new symbol $\bot \notin S$, such that $f(q) = \bot$ means that f is undefined at q, that is, there is no such state (q, n). The result is a DFA having $(|S|+1)^{|Q_1|}$ many states, which is exponential in the sizes of $\mathfrak{V}_1$ and $\mathfrak{V}_2$. This is much smaller than the size of $\mathfrak{P}$: it contains $2^{|Q_1| \cdot |S|}$ many states which is doubly exponential in $|\mathfrak{V}_1|$ and $|\mathfrak{V}_2|$.

Concretely, our DFA $\mathfrak{C} = (Q_{\mathfrak{C}}, \Sigma, \delta_{\mathfrak{C}}, \{\iota_{\mathfrak{C}}\}, F_{\mathfrak{C}})$ accepting $\mathrm{L}(\mathfrak{B}) = (\mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1}))(\Gamma_1^* \setminus B_{\hat{k}})$ is defined as follows:

- $Q_{\mathfrak{C}} = (S \cup \{\bot\})^{Q_1}$, i.e., the set of all maps from Q_1 to $S \cup \{\bot\}$ with $\bot \notin S$; here, $f(q) = \bot$ means that no state $(q, n) \in Q_{\mathfrak{B}}$ is reachable,
- $\iota_{\mathfrak{C}} \colon Q_1 \to S \cup \{\bot\}$ with $\iota_{\mathfrak{C}}(s_1) = 0$ and $\iota_{\mathfrak{C}}(q) = \bot$ for each $q \in Q_1 \setminus \{s_1\}$,
- for all $f, g \in Q_{\mathfrak{C}}$ and $a \in \Sigma$: $(f, a, g) \in \delta_{\mathfrak{C}}$ if, and only if, for each $q \in Q_1$ we have $g(q) = \max\{n \in S \mid \exists p \in Q_1 \colon ((p, f(p)), a, (q, n)) \in \delta_{\mathfrak{B}}\}$ where $\max \emptyset := \bot$, and

- $F_{\mathfrak{C}} = \{f \in Q_{\mathfrak{C}} \mid \exists q \in Q_1 \colon (q, f(q)) \in F_{\mathfrak{B}}\}$.

We have to show now that our construction is correct, i.e., we show $\mathrm{L}(\mathfrak{C}) = \mathrm{L}(\mathfrak{B})$. We do this with the help of the following two propositions each proving one inclusion.

▶ **Proposition 5.12.** $\mathrm{L}(\mathfrak{C}) \subseteq \mathrm{L}(\mathfrak{B})$.

Proof. To prove this inclusion we first have to prove the following helping statement:

▷ Claim 5.13. Let $g \in Q_{\mathfrak{C}}$ and $w \in \Sigma^*$ with $\iota_{\mathfrak{C}} \xrightarrow{w}_{\mathfrak{C}} g$. Then we have $g(q) = \max\{n \in S \mid (s_1, 0) \xrightarrow{w}_{\mathfrak{B}} (q, n)\}$ for each $q \in Q_1$ with $g(q) \neq \bot$.

Proof. We first show $g(q) \geq \max\{n \in S \mid (s_1, 0) \xrightarrow{w}_{\mathfrak{B}} (q, n)\}$ for each $q \in Q_1$ with $g(q) \neq \bot$. We do this by induction on the length of w. So, if $w = \varepsilon$ the statement is obvious since $g = \iota_{\mathfrak{C}}$ holds in this case. Now, let $a \in \Sigma$ and $w' \in \Sigma^*$ with $w = w'a$. Then there is a state $f \in Q_{\mathfrak{C}}$ with $\iota_{\mathfrak{C}} \xrightarrow{w'}_{\mathfrak{C}} f \xrightarrow{a}_{\mathfrak{C}} g$.

Let $(p, m), (q, n) \in Q_{\mathfrak{B}}$ be arbitrary states with $(s_1, 0) \xrightarrow{w'}_{\mathfrak{B}} (p, m) \xrightarrow{a}_{\mathfrak{B}} (q, n)$. Since $|w'| < |w|$ holds, the induction hypothesis yields $f(p) \geq \max\{m' \in S \mid (s_1, 0) \xrightarrow{w'}_{\mathfrak{B}} (p, m')\}$ implying $f(p) \geq m$. From Lemma 5.10 we know that there is $n' \in S$ with $n' \geq n$ and $(p, f(p)) \xrightarrow{a}_{\mathfrak{B}} (q, n')$. Additionally, we know $(f, a, q) \in \delta_{\mathfrak{B}}$ implying $g(q) = \max\{n'' \in S \mid \exists p \in Q_1 \colon ((p, f(p)), a, (q, n'')) \in \delta_{\mathfrak{B}}\}$ and therefore $g(q) \geq n' \geq n$. Since $(q, n) \in Q_{\mathfrak{B}}$ was arbitrary, we infer $g(q) \geq \max\{n \in S \mid (s_1, 0) \xrightarrow{w}_{\mathfrak{B}} (q, n)\}$.

Now we show the inverse inequality. To this end, we show $(s_1, 0) \xrightarrow{w}_{\mathfrak{B}} (q, g(q))$ holds for all $q \in Q_1$ with $g(q) \neq \bot$. Again, we show this by induction on $|w|$. So, let $w = \varepsilon$. Then we have $g = \iota_{\mathfrak{C}}$ and $g(q) \neq \bot$ if, and only if, $q = s_1$. Obviously, we have $(s_1, 0) \xrightarrow{\varepsilon}_{\mathfrak{B}} (s_1, 0)$. Next, let $w = w'a$ for a letter $a \in \Sigma$ and a word $w' \in \Sigma^*$. There is $f \in Q_{\mathfrak{C}}$ with $\iota_{\mathfrak{C}} \xrightarrow{w'}_{\mathfrak{C}} f \xrightarrow{a}_{\mathfrak{C}} g$. The induction hypothesis yields $(s_1, 0) \xrightarrow{w'} (p, f(p))$ for each $p \in Q_1$ with $f(p) \neq \bot$. Let $q \in Q_1$ with $g(q) \neq \bot$. Due to $(f, a, g) \in \delta_{\mathfrak{C}}$ there is $p \in Q_1$ with $((p, f(p)), a, (q, g(q))) \in \delta_{\mathfrak{C}}$ (this also implies $f(p) \neq \bot$). Hence, we have $(s_1, 0) \xrightarrow{w'}_{\mathfrak{B}} (p, f(p)) \xrightarrow{a}_{\mathfrak{B}} (q, g(q))$. ◁

Let $w \in \mathrm{L}(\mathfrak{C})$. Then there is $g \in F_{\mathfrak{C}}$ with $\iota_{\mathfrak{C}} \xrightarrow{w}_{\mathfrak{C}} g$. Since g is accepting, there is a state $q \in Q_1$ with $(q, g(q)) \in F_{\mathfrak{B}}$. By definition of $\mathfrak{B}$ we know $g(q) \neq \bot$ in this case. By Claim 5.13 we know $g(q) = \max\{n \in S \mid (s_1, 0) \xrightarrow{w}_{\mathfrak{B}} (q, n)\}$. Hence, we learn $(s_1, 0) \xrightarrow{w}_{\mathfrak{B}} (q, g(q)) \in F_{\mathfrak{B}}$. This finally implies $w \in \mathrm{L}(\mathfrak{B})$. ◀

▶ **Proposition 5.14.** $\mathrm{L}(\mathfrak{B}) \subseteq \mathrm{L}(\mathfrak{C})$.

Proof. Again, we first need some helping statement:

▷ Claim 5.15. Let $w \in \Sigma^*$ and $(q, n) \in Q_{\mathfrak{B}}$ with $(s_1, 0) \xrightarrow{w}_{\mathfrak{B}} (q, n)$. Then there is $g \in Q_{\mathfrak{C}}$ with $\iota_{\mathfrak{C}} \xrightarrow{w}_{\mathfrak{C}} g$ and $g(q) \geq n$.

Proof. We show this statement by induction on the length $|w|$ of the word w. The case $w = \varepsilon$ is obvious since $g := \iota_{\mathfrak{C}}$ satisfies $\iota_{\mathfrak{C}} \xrightarrow{\varepsilon}_{\mathfrak{C}} \iota_{\mathfrak{C}}$ and $\iota_{\mathfrak{C}}(s_1) = 0 \geq 0$ (note that $(q, n) = (s_1, 0)$ holds). Now, let $a \in \Sigma$ and $w' \in \Sigma^*$ with $w = w'a$. Then there is a state $(p, m) \in Q_{\mathfrak{B}}$ with $(s_1, 0) \xrightarrow{w'}_{\mathfrak{B}} (p, m) \xrightarrow{a}_{\mathfrak{B}} (q, n)$. By induction hypothesis there is $f \in Q_{\mathfrak{C}}$ with $\iota_{\mathfrak{C}} \xrightarrow{w}_{\mathfrak{C}} f$ and $f(p) \geq m$. According to Lemma 5.10 we also know that there is $n' \in S$ with $n' \geq n$ and $(p, f(p)) \xrightarrow{a}_{\mathfrak{B}} (q, n')$. In this case, we have $((p, f(p)), a, (q, n')) \in \delta_{\mathfrak{B}}$.

Let $g \in Q_{\mathfrak{C}}$ be the uniquely defined state with $(f, a, g) \in \delta_{\mathfrak{C}}$. Then we have $g(q)$ is the maximal value $n \in S$ with $((r, f(r)), a, (q, n)) \in \delta_{\mathfrak{B}}$ for some $r \in Q_1$. In particular, we have $g(q) \geq n' \geq n$ and $\iota_{\mathfrak{C}} \xrightarrow{w'}_{\mathfrak{C}} f \xrightarrow{a}_{\mathfrak{C}} g$. ◁

Now, let $w \in \mathrm{L}(\mathfrak{B})$. Then there is a state $(q,n) \in F_{\mathfrak{B}}$ with $(s_1,0) \xrightarrow{w}_{\mathfrak{B}} (q,n)$. According to Claim 5.15 there is a state $g \in Q_{\mathfrak{C}}$ with $\iota_{\mathfrak{C}} \xrightarrow{w}_{\mathfrak{C}} g$ and $g(q) \geq n$. By Lemma 5.11 we also know $(q, g(q)) \in F_{\mathfrak{B}}$. But then we infer $g \in F_{\mathfrak{C}}$, i.e., $w \in \mathrm{L}(\mathfrak{C})$. ◀

Finally, we are able to prove the previously stated Theorem 5.8.

Proof of Theorem 5.8. The DFA $\mathfrak{C}$ as constructed above accepts the language $\mathrm{L}(\mathfrak{C}) = \mathrm{L}(\mathfrak{B}) = (\mathrm{R}(\mathfrak{T}_{\mathfrak{V}_1}))(\Gamma_1^* \setminus B_{\hat{k}})$ according to Propositions 5.12 and 5.14. This automaton has

$$|Q_{\mathfrak{C}}| = |(S \cup \{\bot\})^{Q_1}| = |S \cup \{\bot\}|^{|Q_1|} \in \mathcal{O}(\hat{k}^{|Q_1|}) = \mathcal{O}((\|\Delta_1\| \cdot \mathrm{Rack}(\mathfrak{V}_1 \times \mathfrak{V}_2))^{|Q_1|})$$
$$= \mathcal{O}((\|\Delta_1\| \cdot (|Q_1| \cdot |Q_2| \cdot \max\{\|\Delta_1\|, \|\Delta_2\|\} + 2)^{6!+1})^{|Q_1|})$$

many states. The last term is exponential in the size of $\mathfrak{V}_1$ and $\mathfrak{V}_2$. ◀

From the proof technique, we can extract a slightly more abstract statement that might be of independent interest. An *ordered NFA* is an NFA $\mathfrak{A}$ together with a quasi-ordering $(Q, \preccurlyeq)$ on its set of states Q such that if $p \preccurlyeq q$, then all words accepted from p are also accepted from q. An *anti-chain* of $\mathfrak{A}$ is an anti-chain in $(Q, \preccurlyeq)$.

▶ **Proposition 5.16.** *If $\mathfrak{A}$ is an ordered NFA whose anti-chains have at most ℓ states, then $\mathfrak{A}$ has an equivalent DFA with $|Q|^\ell$ states.*

Here, the states of the DFA are the anti-chains of $(Q, \preccurlyeq)$. This is useful whenever ℓ is small (e.g. logarithmic) in the size of $\mathfrak{A}$ (i.e., in $|Q|$). In our proof, for example, one can equip $Q_{\mathfrak{B}} = Q_1 \times [-1, \hat{k}+1]$ with the ordering $(p,m) \preccurlyeq (q,n)$ if and only if $p = q$ and $m \leq n$. Then clearly, an anti-chain in $(Q_{\mathfrak{B}}, \preccurlyeq)$ contains at most $|Q_1|$ states.

6 Lower Bounds

In this final section we want to show that all of the upper bounds shown in Section 5 are tight. This means, whenever Corollaries 5.5–5.7 and Theorem 5.8 gives an i-fold exponential upper bound for separators with $i \geq 1$, then we shall here provide a sequence $(\mathfrak{V}_1, \mathfrak{W}_1), (\mathfrak{V}_2, \mathfrak{W}_2), \ldots$ of VASS $\mathfrak{V}_n, \mathfrak{W}_n$ of size polynomial in n such that any separator of $L(\mathfrak{V}_n)$ and $L(\mathfrak{W}_n)$ requires at least $\exp_i(n)$ states. Recall that $\exp_0(n) = n$ and $\exp_{i+1}(n) = 2^{\exp_i(n)}$ for $i \geq 0$.

The case where d is part of the input was already considered by Czerwiński et al. in [10] (they use the languages in Eq. (1) with $f\colon n \mapsto 2^{2^n}$). We mention this without proof:

▶ **Proposition 6.1** (Czerwiński et al. [10]). *For any $n \in \mathbb{N}$ there are disjoint VASS $\mathfrak{V}_n$ and $\mathfrak{W}_n$ of size polynomial in n such that*
(1) *any NFA separating $\mathrm{L}(\mathfrak{V}_n)$ and $\mathrm{L}(\mathfrak{W}_n)$ has at least 2^{2^n} states.*
(2) *any DFA separating $\mathrm{L}(\mathfrak{V}_n)$ and $\mathrm{L}(\mathfrak{W}_n)$ has at least $2^{2^{2^n}}$ states.* ◀

This provides the lower bounds of the first row in Table 1. In the construction of Czerwiński et al., the dimension of $\mathfrak{V}_n$ and $\mathfrak{W}_n$ grows (polynomially) with n. This means, we need different constructions for fixed dimension. Moreover, in fixed dimension, we cannot translate between VASS with unary and binary encodings. This means, we have to distinguish between the two encodings. Let us begin with unary encodings (i.e. the blue entries in Table 1). In the case of NFA separators, our upper bounds are polynomial, so we need not prove any lower bounds. For DFA separators, the exponential lower bound is already achieved for VASS that have no counters:

Figure 4 1-VASS $\mathfrak{V}_n$ (left) and $\mathfrak{W}_n$ (right) in the proof of Proposition 6.3.

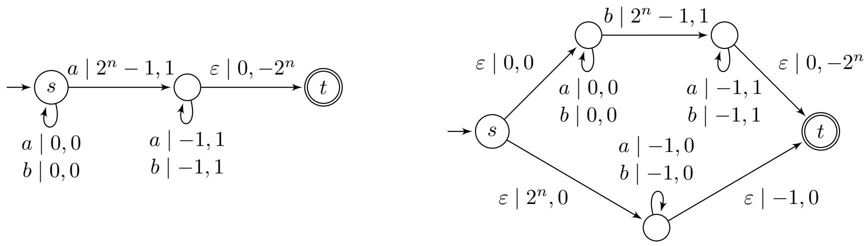

Figure 5 2-VASS $\mathfrak{V}_n$ (left) and $\mathfrak{W}_n$ (right) in the proof of Proposition 6.4.

▶ **Proposition 6.2.** *For any $n \in \mathbb{N}$ there are disjoint NFAs $\mathfrak{A}_n$ and $\mathfrak{B}_n$ of size polynomial in n such that any DFA separating $\mathrm{L}(\mathfrak{A}_n)$ and $\mathrm{L}(\mathfrak{B}_n)$ has at least 2^n states.*

Proof. For $n \in \mathbb{N}$ consider the languages $K_n = K_{f,n}$ and $L_n = L_{f,n}$ with $f : n \to n$, with $K_{f,n}, L_{f,n}$ as in Eq. (1). Both languages are regular and accepted by NFAs $\mathfrak{A}_n$ and $\mathfrak{B}_n$ with $\mathcal{O}(n)$ many states. Since we have $K_n = \{a,b\}^* \setminus L_n$, K_n is the only regular separator of K_n and L_n. But it is well-known that any DFA for K_n has at least 2^n states. ◀

This provides the lower bound for the two blue entries in Table 1. Let us now turn to binary encodings. For NFA separators, all lower bounds are achieved using 1-VASS: Our first proposition yields the lower bounds for the light and dark gray entries of Table 1.

▶ **Proposition 6.3.** *For any $n \in \mathbb{N}$, there are disjoint 1-VASS $\mathfrak{V}_n$ and $\mathfrak{W}_n$, with binary encoded numbers, of size polynomial in n such that any NFA (and thus any DFA) separating $\mathrm{L}(\mathfrak{V}_n)$ and $\mathrm{L}(\mathfrak{W}_n)$ has at least 2^n states.*

Proof. Consider the languages $K_n = \{a^m \mid m < 2^n\}$ and $L_n = \{a^m \mid m \geq 2^n\}$. These two languages are accepted by the 1-VASS $\mathfrak{V}_n$ and $\mathfrak{W}_n$ as depicted in Figure 4. The transitions increasing resp. decreasing the counter by 2^n can be encoded in binary using n bits, i.e., $\mathfrak{V}_n$ and $\mathfrak{W}_n$ have size $\mathcal{O}(n)$.

Since K_n and L_n are regular and $K_n = \{a\}^* \setminus L_n$ holds, the only regular separator of K_n and L_n is K_n itself. It is easy to see that any NFA (and thus any DFA) accepting K_n requires at least 2^n many states. ◀

It remains to show the lower bound for the yellow entry of Table 1:

▶ **Proposition 6.4.** *For any $n \in \mathbb{N}$, there are disjoint 2-VASS $\mathfrak{V}_n$ and $\mathfrak{W}_n$, with binary encoded numbers, of size polynomial in n such that any DFA separating $\mathrm{L}(\mathfrak{V}_n)$ and $\mathrm{L}(\mathfrak{W}_n)$ has at least 2^{2^n} states.*

Proof. Consider the languages $K_n = K_{f,n}$ and $L_n = L_{f,n}$ with $f : n \mapsto 2^n$ with $K_{f,n}, L_{f,n}$ as in Eq. (1). These two languages are accepted by the 2-VASS $\mathfrak{V}_n$ and $\mathfrak{W}_n$, resp., as depicted in Figure 5. It is clear that both VASS have size $\mathcal{O}(n)$. Similar to the proof of Proposition 6.2 we can see that any DFA accepting K_n has at least 2^{2^n} many states. ◀

References

1 Pablo Barceló, Diego Figueira, and Rémi Morvan. Separating automatic relations. *CoRR*, abs/2305.08727, 2023. arXiv:2305.08727, doi:10.48550/arXiv.2305.08727.

2 Pascal Baumann, Roland Meyer, and Georg Zetzsche. Regular separability in Büchi VASS. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 9:1–9:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.STACS.2023.9.

3 Mikołaj Bojańczyk. It is undecidable if two regular tree languages can be separated by a deterministic tree-walking automaton. *Fundam. Informaticae*, 154(1-4):37–46, 2017. doi:10.3233/FI-2017-1551.

4 Laura Bozzelli and Pierre Ganty. Complexity Analysis of the Backward Coverability Algorithm for VASS. In Giorgio Delzanno and Igor Potapov, editors, *Reachability Problems*, Lecture Notes in Computer Science, pages 96–109, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-24288-5_10.

5 Christian Choffrut and Serge Grigorieff. Separability of rational relations in $A^* \times \mathbb{N}^m$ by recognizable relations is decidable. *Inf. Process. Lett.*, 99(1):27–32, 2006. doi:10.1016/j.ipl.2005.09.018.

6 Lorenzo Clemente, Wojciech Czerwiński, Slawomir Lasota, and Charles Paperman. Regular Separability of Parikh Automata. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 117:1–117:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.117.

7 Lorenzo Clemente, Wojciech Czerwiński, Slawomir Lasota, and Charles Paperman. Separability of Reachability Sets of Vector Addition Systems. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPIcs*, pages 24:1–24:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.STACS.2017.24.

8 Lorenzo Clemente and Michał Skrzypczak. Deterministic and game separability for regular languages of infinite trees. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 126:1–126:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.126.

9 Wojciech Czerwiński and Slawomir Lasota. Regular separability of one counter automata. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005079.

10 Wojciech Czerwiński, Slawomir Lasota, Roland Meyer, Sebastian Muskalla, K. Narayan Kumar, and Prakash Saivasan. Regular Separability of Well-Structured Transition Systems. In Sven Schewe and Lijun Zhang, editors, *29th International Conference on Concurrency Theory (CONCUR 2018)*, volume 118 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 35:1–35:18, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CONCUR.2018.35.

11 Wojciech Czerwiński, Wim Martens, and Tomás Masopust. Efficient Separability of Regular Languages by Subsequences and Suffixes. 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 150–161. Springer, 2013. doi:10.1007/978-3-642-39212-2_16.

12 Wojciech Czerwiński, Wim Martens, Lorijn van Rooijen, Marc Zeitoun, and Georg Zetzsche. A Characterization for Decidable Separability by Piecewise Testable Languages. *Discrete Mathematics and Theoretical Computer Science*, 19(4), 2017. doi:10.23638/DMTCS-19-4-1.

13 Wojciech Czerwiński and Georg Zetzsche. An Approach to Regular Separability in Vector Addition Systems. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 341–354, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3373718.3394776.

14 Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1):63–92, 2001. doi:10.1016/S0304-3975(00)00102-X.

15 Jean Goubault-Larrecq and Sylvain Schmitz. Deciding Piecewise Testable Separability for Regular Tree Languages. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 97:1–97:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.97.

16 Sheila A. Greibach. Remarks on Blind and Partially Blind One-Way Multicounter Machines. *Theoretical Computer Science*, 7(3):311–324, 1978. doi:10.1016/0304-3975(78)90020-8.

17 Christopher Hugenroth. Separating Regular Languages over Infinite Words with Respect to the Wagner Hierarchy. In Mikołaj Bojańczyk and Chandra Chekuri, editors, *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021)*, volume 213 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 46:1–46:13, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2021.46.

18 Harry B. Hunt III. On the Decidability of Grammar Problems. *Journal of the ACM*, 29(2):429–447, 1982.

19 Matthias Jantzen. On the hierarchy of Petri net languages. *RAIRO - Theoretical Informatics and Applications*, 13(1):19–30, 1979. doi:10.1051/ita/1979130100191.

20 Eren Keskin and Roland Meyer. Separability and non-determinizability of WSTS. *CoRR*, abs/2305.02736, 2023. arXiv:2305.02736, doi:10.48550/arXiv.2305.02736.

21 Eryk Kopczyński. Invisible Pushdown Languages. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 867–872. ACM, 2016. doi:10.1145/2933575.2933579.

22 Marvin Künnemann, Filip Mazowiecki, Lia Schütze, Henry Sinclair-Banks, and Karol Węgrzycki. Coverability in VASS revisited: Improving rackoff's bound to obtain conditional optimality. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 131:1–131:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.131.

23 Tomás Masopust. Separability by piecewise testable languages is PTime-complete. *Theor. Comput. Sci.*, 711:109–114, 2018. doi:10.1016/j.tcs.2017.11.004.

24 Thomas Place, Lorijn van Rooijen, and Marc Zeitoun. Separating Regular Languages by Locally Testable and Locally Threshold Testable Languages. In Anil Seth and Nisheeth K. Vishnoi, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2013, December 12-14, 2013, Guwahati, India*, volume 24 of *LIPIcs*, pages 363–375. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPIcs.FSTTCS.2013.363.

25 Thomas Place and Marc Zeitoun. Separating regular languages with first-order logic. In Thomas A. Henzinger and Dale Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 75:1–75:10. ACM, 2014. doi:10.1145/2603088.2603098.

26 Thomas Place and Marc Zeitoun. Separation and the Successor Relation. In Ernst W. Mayr and Nicolas Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPIcs*, pages 662–675. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.STACS.2015.662.

27 Thomas Place and Marc Zeitoun. Separating Regular Languages with First-Order Logic. *Log. Methods Comput. Sci.*, 12(1), 2016. doi:10.2168/LMCS-12(1:5)2016.

28 Thomas Place and Marc Zeitoun. Separating Without Any Ambiguity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 137:1–137:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.137.

29 Thomas Place and Marc Zeitoun. Separation and covering for group based concatenation hierarchies. 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.2019.8785655.

30 Thomas Place and Marc Zeitoun. Separation for dot-depth two. *Log. Methods Comput. Sci.*, 17(3), 2021. doi:10.46298/lmcs-17(3:24)2021.

31 Thomas Place and Marc Zeitoun. A generic polynomial time approach to separation by first-order logic without quantifier alternation. In Anuj Dawar and Venkatesan Guruswami, editors, *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022, December 18-20, 2022, IIT Madras, Chennai, India*, volume 250 of *LIPIcs*, pages 43:1–43:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.FSTTCS.2022.43.

32 Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223–231, 1978. doi:10.1016/0304-3975(78)90036-1.

33 Thomas G. Szymanski and John H. Williams. Noncanonical extensions of bottom-up parsing techniques. *SIAM Journal on Computing*, 5(2), 1976.

34 Ramanathan S. Thinniyam and Georg Zetzsche. Regular separability and intersection emptiness are independent problems. In Arkadev Chattopadhyay and Paul Gastin, editors, *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019, December 11-13, 2019, Bombay, India*, volume 150 of *LIPIcs*, pages 51:1–51:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.FSTTCS.2019.51.

35 Georg Zetzsche. Separability by piecewise testable languages and downward closures beyond subwords. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 929–938. ACM, 2018. doi:10.1145/3209108.3209201.