THEORIE-TAG 2023

A. Lin, G. Zetzsche (Eds.): Theorietag 2023, Kaiserslautern, 4. – 6.10.2023
RPTU & MPI-SWS Kaiserslautern, Technical Report, pp. 33–36.

Regular Separators for VASS Coverability Languages

Chris Köcher$^{(A)}$ Georg Zetzsche$^{(A)}$

$^{(A)}$Max Planck Insitute for Software Systems, Kaiserslautern
{ckoecher,georg}@mpi-sws.org

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 [8, 6], recent work had 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 [4] and several other subclasses, such as one-dimensional VASS [3], integer VASS [1] (where counters can become negative), and commutative VASS languages [2]. Moreover, for each of these subclasses, decidability is retained if one of the input languages is an arbitrary VASS reachability language [5].

The decidability result about VASS coverability languages is a consequence of a remarkable and surprising result by Czerwiński, Lasota, Meyer, Muskalla, Kumar, and Saivasan [4]: 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 [7] have even shown that the finite branching assumption can be lifted.) Moreover, VASS (with coverability semantics) are a standard example of (finitely branching) WSTS.

Funded by the European Union (ERC, FINABIS, 101077902).

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 [4] 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.

2. 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 $\Sigma \cup \{\varepsilon\}$.

A *configuration* is a tuple from $Q \times \mathbb{N}^d$. 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 $\ell \in \mathbb{N}$, configurations $(q_i, \vec{v_i}) \in Q \times \mathbb{N}^d$ for each $0 \leq i \leq \ell$ and transitions $(q_{i-1}, a_i, \vec{x_i}, q_i) \in \Delta$ with $\vec{v_i} = \vec{v_{i-1}} + \vec{x_i}$ for each $1 \leq i \leq \ell$ such that $w = a_1 a_2 \ldots a_\ell$ holds. Here, $+$ is the component-wise addition of integers in d-dimensional vectors.

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 $\vec{v} \geq \vec{0}$ holds for any $\vec{v} \in \mathbb{N}^d$; we say that $(t, \vec{v})$ *covers* the target configuration $(t, \vec{0})$. We call $L \subseteq \Sigma^*$ a *(coverability) d-VASS-language* if there is a d-VASS $\mathfrak{V}$ with $L = \mathrm{L}(\mathfrak{V})$.

The following equivalence is known about regular separability of coverability VASS-languages:

Theorem 2.1 ([4]) *Let $\mathfrak{V}$ and $\mathfrak{W}$ be two VASS. The languages $\mathrm{L}(\mathfrak{V})$ and $\mathrm{L}(\mathfrak{W})$ are regular separable if, and only if, $\mathrm{L}(\mathfrak{V}) \cap \mathrm{L}(\mathfrak{W}) = \emptyset$ holds.*

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. All our bounds are tight in the sense that for each i-exp upper bound, there is also an i-fold exponential lower bound.

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

		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 over the 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.

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 [4] 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 [4]. In [4], 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 [5] to reduce to an even more specific case: Namely, we construct $\mathfrak{V}$ such that for the language C_d of all counter instruction sequences that keep d counter 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 [5]): 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 antichain in this ordering has at most polynomial size. This permits determinization without a blow-up.

Lower bounds. The lower bounds for the first row in our table have already been shown in [4]. For the others, we use two types of pairs. The first is similar to the language pairs in [4]:

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

where $f \colon \mathbb{N} \to \mathbb{N}$ is one of the functions $n \mapsto n$ (a separating DFA needs 2^n states; the purple entries) or $n \mapsto 2^n$ (a separating DFA needs 2^{2^n} states, the yellow entry). In [4], 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).

References

[1] L. CLEMENTE, W. CZERWINSKI, S. LASOTA, C. PAPERMAN, Regular Separability of Parikh Automata. In: I. CHATZIGIANNAKIS, P. INDYK, F. KUHN, A. MUSCHOLL (eds.), *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*. LIPIcs 80, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 117:1–117:13.

[2] L. CLEMENTE, W. CZERWINSKI, S. LASOTA, C. PAPERMAN, Separability of Reachability Sets of Vector Addition Systems. In: H. VOLLMER, B. VALLÉE (eds.), *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*. LIPIcs 66, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 24:1–24:14.

[3] W. CZERWINSKI, S. 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*. IEEE Computer Society, 2017, 1–12.

[4] W. CZERWINSKI, S. LASOTA, R. MEYER, S. MUSKALLA, K. N. KUMAR, P. SAIVASAN, Regular Separability of Well-Structured Transition Systems. In: S. SCHEWE, L. ZHANG (eds.), *29th International Conference on Concurrency Theory (CONCUR 2018)*. Leibniz International Proceedings in Informatics (LIPIcs) 118, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2018, 35:1–35:18.

[5] W. CZERWIŃSKI, G. ZETZSCHE, An Approach to Regular Separability in Vector Addition Systems. In: *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. Association for Computing Machinery, New York, NY, USA, 2020, 341–354.

[6] H. B. HUNT III, On the Decidability of Grammar Problems. *Journal of the ACM* **29** (1982) 2, 429–447.

[7] E. KESKIN, R. MEYER, Separability and Non-Determinizability of WSTS. *CoRR* **abs/2305.02736** (2023).

[8] T. G. SZYMANSKI, J. H. WILLIAMS, Noncanonical Extensions of Bottom-up Parsing Techniques. *SIAM Journal on Computing* **5** (1976) 2.