

An Approach to Regular Separability in Vector Addition Systems

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

Georg Zetsche
Max Planck Institute for Software Systems (MPI-SWS)
georg@mpi-sws.org

Abstract

We study the problem of regular separability of languages of vector addition systems with states (VASS). It asks whether for two given VASS languages K and L , there exists a regular language R that includes K and is disjoint from L . While decidability of the problem in full generality remains an open question, there are several subclasses for which decidability has been shown: It is decidable for (i) one-dimensional VASS, (ii) VASS coverability languages, (iii) languages of integer VASS, and (iv) commutative VASS languages.

We propose a general approach to deciding regular separability. We use it to decide regular separability of an arbitrary VASS language from any language in the classes (i), (ii), and (iii). This generalizes all previous results, including (iv).

CCS Concepts: • Theory of computation → Models of computation; Formal languages and automata theory.

Keywords: separability problem, vector addition systems, decidability

ACM Reference Format:

Wojciech Czerwiński and Georg Zetsche. 2020. An Approach to Regular Separability in Vector Addition Systems. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '20)*, July 8–11, 2020, Saarbrücken, Germany. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3373718.3394776>

1 Introduction

Vector addition systems with states. Vector addition systems with states (VASS) [17] are one of the most intensively studied model for concurrent systems. They can be seen as automata with finitely many counters, which can be increased or decreased whenever its values is non-negative, but not tested for zero. Despite their fundamental nature

and the extensive interest, core aspects remain obscure. A prominent example is the reachability problem, which was shown decidable in the early 1980s [30]. However, its complexity remains unsettled. The best known upper bounds are non-primitive-recursive [29], whereas the best known lower bound is tower hardness [8], and reachability seems far from being understood.

There is also a number of other natural problems concerning VASS where the complexity or even decidability remains unresolved. An example is the structural liveness problem, which asks whether there exists a configuration such that for every configuration c reachable from it and every transition t one can reach from c some configuration in which t is enabled. Its decidability status was settled only recently [21], but the complexity is still unknown. For closely related extensions of VASS, namely branching VASS and pushdown VASS even decidability status is unknown with the best lower bound being tower-hardness [25, 26]. This all suggests that there is still a lot to understand about VASS.

Separability problem. One way to gain a fresh perspective and deeper understanding of the matter is to study decision problems that generalize reachability. It seems to us that here, a natural choice is the problem of *regular separability*. It asks whether for two given languages K and L there exists a *regular separator*, i.e. a regular language R such that $K \subseteq R$ and $R \cap L = \emptyset$. Decidability of this problem for general VASS languages appears to be difficult. It has been shown decidable for several subclasses, namely for (a) *commutative VASS languages* [5] (equivalently, separability of sections of reachability sets by recognizable sets), for (b) *one-counter nets* [7] i.e. VASS with one counter, (c) integer VASS [4], i.e. VASS where we allow counters to become negative, and finally for (d) *coverability languages*, which follows from the general decidability for well-structured transition systems [9]. However, in full generality, decidability remains a challenging open question. It should be mentioned that this line of research has already led to unforeseen insights: The closely related problem of separability by bounded regular languages prompted methods that turned out to yield decidability results that were deeply unexpected [6].

Contribution. We present a general approach to deciding separability by regular languages and prove three new results, which generalize all four regular separability results

*Supported by the European Research Council under the European Union's Horizon 2020 research and innovation programme (ERC Consolidator Grant LIPA, agreement no. 683080).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

LICS '20, July 8–11, 2020, Saarbrücken, Germany

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-7104-9/20/07.

<https://doi.org/10.1145/3373718.3394776>

shown until now. Specifically, we show decidability of regular separability of (i) VASS languages from languages of one-counter nets, (ii) VASS languages from coverability VASS languages, and (iii) VASS languages from integer VASS languages. This clearly generalizes results (b), (c), and (d) above, and we will see that this also strengthens (a).

Main ingredients. The starting point of our approach is the observation that for many language classes \mathcal{K} , deciding regular separability of a language L from a given language K in \mathcal{K} can be reduced to deciding regular separability of L from some fixed language G in \mathcal{K} . In all three cases (i)–(iii), this allows us to interpret the words in L as walks in the grid \mathbb{Z}^n . For (i), we then have to decide separability from those walks in $\mathbb{Z} = \mathbb{Z}^1$ that remain in \mathbb{N} and arrive at zero. For (ii), we decide separability from the set of walks that remain in \mathbb{N}^n and arrive somewhere in \mathbb{N}^n . For (iii), we want to separate from all walks in \mathbb{Z}^n that end at the origin. The corresponding fixed languages are denoted D_1 (for (i)), C_n (for (ii)), and Z_n (for (iii)), respectively.

In order to decide separability from a fixed language G (i.e. D_1 , C_n , or Z_n), we first classify those regular languages that are disjoint from G . Second, the classifications are used to decide whether a given VASS language L is included in such a regular language. These decision procedures employ either the previous result (a) above or reduce to the simultaneous unboundedness problem (which is known to be decidable for VASS languages [6, 16]).

VASS vs. integer VASS The result (iii) is significantly more involved than (i) and (ii). First, the classification of regular languages disjoint from Z_n leads to a geometric characterization of regular separability. This is then applied in a decision procedure that employs the KLMST decomposition from the algorithms by Sacerdote and Tenney [37], Mayr [30], Kosaraju [22], and Lambert [23] (and recast by Leroux and Schmitz [29]) for reachability in VASS. Previous algorithms for VASS languages that use this decomposition (by Habermehl, Meyer, and Wimmel [16] and by Czerwiński, Hofman, and Zetsche [6]) perform the decomposition once, which yields regular overapproximations that contain all information needed for their purposes. Our procedure requires an additional refinement: Depending on a property of each overapproximation, we can either reduce separability to the commutative case and apply (a) or we can reduce the dimension of the input language (i.e. transform it into a set of walks in \mathbb{Z}^m for $m < n$) and invoke our algorithm recursively.

Connection to VASS reachability. We hope that this approach can be used to decide regular separability for VASS in full generality in the future. This would amount to deciding regular separability of a given VASS language from the set of all walks in \mathbb{Z}^n that remain in \mathbb{N}^n and arrive in the origin. The corresponding language is denoted D_n . We emphasize that an algorithm along these lines might directly yield new insights concerning reachability: Classifying those regular

languages that are disjoint from D_n would yield an algorithm for reachability because the latter reduces to intersection of a given regular language with D_n . Such an algorithm would look for a certificate for *non-reachability* (like Leroux’s algorithm [27]) instead of a run.

Related work. Aside from *regular* separability, separability problems in a more general sense have also attracted significant attention in recent years. Here, the class of sought separators can differ from the regular languages. A series of recent works has concentrated on separability of regular languages by separators from subclasses [31–36], and work in this direction has been started for trees as well [2, 13].

In the case of non-regular languages as input languages, it was shown early that regular separability is undecidable for context-free languages [18, 39]. Moreover, aside from the above mentioned results on regular separability, infinite-state systems have also been studied with respect to separability by bounded regular languages [6] and piecewise testable languages [10] and generalizations thereof [41].

2 Preliminaries

By $\mathbb{Q}(\mathbb{Q}_+)$, we denote the set of (non-negative) rational numbers. Let Σ be an alphabet and let ε denote the empty word. If $\Sigma = \{x_1, \dots, x_n\}$, then the *Parikh image* of a word $w \in \Sigma^*$ is defined as $\Psi(w) = (|w|_{x_1}, \dots, |w|_{x_n})$, where $|w|_x$ denotes the number of occurrences of x in w . The *commutative closure* of a language $L \subseteq \Sigma^*$ is $\Pi(L) = \{u \in \Sigma^* \mid \exists v \in L: \Psi(v) = \Psi(u)\}$.

A (*n-dimensional*) *vector addition system with states* (VASS) is a tuple $V = (Q, T, s, t)$, where Q is a finite set of *states*, $T \subseteq Q \times \Sigma_\varepsilon \times \mathbb{Z}^n \times Q$ is a finite set of *transitions*, $s \in Q$ is its *source state*, $t \in Q$ is its *target state*. Here, Σ_ε denotes $\Sigma \cup \{\varepsilon\}$. A *configuration* of V is a pair $(q, \mathbf{u}) \in Q \times \mathbb{N}^n$. For each transition $t = (q, x, \mathbf{v}, q') \in T$ and configurations (q, \mathbf{u}) , (q', \mathbf{u}') with $\mathbf{u}' = \mathbf{u} + \mathbf{v}$, we write $(q, \mathbf{u}) \xrightarrow{x} (q', \mathbf{u}')$. For a word $w \in \Sigma^*$, we write $(q, \mathbf{u}) \xrightarrow{w} (q', \mathbf{u}')$ if there are $x_1, \dots, x_n \in \Sigma_\varepsilon$ and configurations (q_i, \mathbf{v}_i) for $i \in [0, n]$ with $(q_{i-1}, \mathbf{v}_{i-1}) \xrightarrow{x_i} (q_i, \mathbf{v}_i)$ for $i \in [1, n]$, $(q_0, \mathbf{v}_0) = (q, \mathbf{u})$, and $(q_n, \mathbf{v}_n) = (q', \mathbf{u}')$. The language of V is then $L(V) = \{w \in \Sigma^* \mid (s, \mathbf{0}) \xrightarrow{w} (t, \mathbf{0})\}$. An (*n-dimensional*) *integer vector addition system with states* (\mathbb{Z} -VASS) [15] is syntactically a VASS, but for \mathbb{Z} -VASS, the configurations are pairs in $Q \times \mathbb{Z}^n$. This difference aside, the language is defined verbatim. Likewise, an *n-dimensional coverability vector addition system with states* (coverability VASS) is syntactically a VASS. However, if we regard a VASS V as a coverability VASS, we define its language as $L(V) = \{w \in \Sigma^* \mid (s, \mathbf{0}) \xrightarrow{w} (t, \mathbf{u}) \text{ for some } \mathbf{u} \in \mathbb{N}^n\}$. Let \mathcal{V}_n (\mathcal{Z}_n , \mathcal{C}_n) denote the class of languages of n -dim. VASS (\mathbb{Z} -VASS, coverability VASS).

Let $\Sigma_n = \{a_i, \bar{a}_i \mid i \in [1, n]\}$ and define the homomorphism $\varphi_n: \Sigma_n^* \rightarrow \mathbb{Z}^n$ by $\varphi_n(a_i) = \mathbf{e}_i$ and $\varphi_n(\bar{a}_i) = -\mathbf{e}_i$. Here, $\mathbf{e}_i \in \mathbb{Z}^n$ is the vector with 1 in coordinate i and 0 everywhere

else. By way of φ_n , we can regard words from Σ_n^* as walks in the grid \mathbb{Z}^n that start in the origin. Later, we will only write φ when the n is clear from the context. With this, let $Z_n = \{w \in \Sigma_n^* \mid \varphi(w) = \mathbf{0}\}$. Hence, Z_n is the set of walks that start and end in the origin.

For $w \in \Sigma_1^*$, let $\text{drop}(w) = \min\{\varphi(v) \mid v \text{ is a prefix of } w\}$. Thus, if w is interpreted as walking along \mathbb{Z} , then $\text{drop}(w)$ is the lowest value attained on the way. Note that $\text{drop}(w) \in [-|w|, 0]$ for every $w \in \Sigma_1^*$. We define $C_1 = \{w \in \Sigma_1^* \mid \text{drop}(w) = 0\}$. For each $i \in [1, n]$, let $\lambda_i: \Sigma_n^* \rightarrow \Sigma_1^*$ be the homomorphism with $\lambda_i(a_i) = a_1$, $\lambda_i(a_j) = \varepsilon$ for $j \neq i$, and $\lambda_i(\bar{a}_j) = \bar{\lambda}_i(a_j)$ for every $j \in [1, n]$. Then we define $C_n = \bigcap_{i=1}^n \lambda_i^{-1}(C_1)$. Thus, C_n is the set of walks in \mathbb{Z}^n that start in the origin and remain in the positive orthant \mathbb{N}^n . Finally, let $D_n = C_n \cap Z_n$. Hence, D_n collects those walks that start in the origin, always remain in \mathbb{N}^n and arrive in the origin. For $w \in \Sigma_n^*$, $w = w_1 \cdots w_m$, $w_1, \dots, w_m \in \Sigma_n$, let $\bar{w} = \bar{w}_1 \cdots \bar{w}_m$ and $w^{\text{rev}} = w_m \cdots w_1$. Here, we set $\bar{a}_i = a_i$ for $a_i \in \Sigma_n$. For $L \subseteq \Sigma^*$ we define $\bar{L} = \{\bar{w} \mid w \in L\}$.

For alphabets Σ, Γ , a subset $T \subseteq \Sigma^* \times \Gamma^*$ is a *rational transduction* if it is a homomorphic image of a regular language, i.e. if there is an alphabet Δ , a regular $K \subseteq \Delta^*$, and a morphism $h: \Delta^* \rightarrow \Sigma^* \times \Gamma^*$ such that $T = h(K)$. Typical examples of rational transductions are the relation $\{(w, g(w)) \mid w \in \Sigma^*\}$ for some morphism $g: \Sigma^* \rightarrow \Gamma^*$ or $\{(w, w) \mid w \in R\}$ for some regular language $R \subseteq \Sigma^*$ [1].

It is well-known that if $S \subseteq \Sigma^* \times \Gamma^*$ and $T \subseteq \Delta^* \times \Sigma^*$ are rational transductions, then the relation $S \circ T$, which is defined $\{(u, v) \in \Delta^* \times \Gamma^* \mid \exists w \in \Sigma^*: (u, w) \in T, (w, v) \in S\}$ and also $T^{-1} = \{(v, u) \in \Sigma^* \times \Delta^* \mid (u, v) \in T\}$ are rational transductions as well [1].

For a language $L \subseteq \Sigma^*$ and a subset $T \subseteq \Sigma^* \times \Gamma^*$, we define $TL = \{v \in \Gamma^* \mid \exists u \in L: (u, v) \in T\}$. A language class \mathcal{K} is called *full trio* if for every $L \subseteq \Sigma^*$ from \mathcal{K} , and every rational transduction $T \subseteq \Sigma^* \times \Gamma^*$, we also have TL in \mathcal{K} . The full trio *generated by* L , denoted by $\mathcal{M}(L)$, is the class of all languages TL , where $T \subseteq \Sigma^* \times \Gamma^*$ is a rational transduction for some Γ . It is well-known that \mathcal{V}_n , C_n , and Z_n are (effectively) the full trios generated by D_n , C_n , and Z_n , respectively [14, 19].

State of the art. We now give a brief overview of previous results on regular separability for subclasses of VASS languages. Two languages $K, L \subseteq \Sigma^*$ are called *regularly separable* if there exists a regular language $S \subseteq \Sigma^*$ with $K \subseteq S$ and $L \cap S = \emptyset$. In that case, we write $K \mid L$. The *regular separability problem* asks, given languages K and L , whether $K \mid L$. The first studied subclass of VASS was that of *commutative VASS languages*, i.e. those of the form $\Pi(L)$ for a VASS language L .

Theorem 1 ([5]). *Given VASS languages $K, L \subseteq \Sigma^*$, it is decidable whether $\Pi(K) \mid \Pi(L)$.*

As observed in [6], Theorem 1 also implies the following.

1-VASS	\geq -VASS	\mathbb{Z} -VASS	c-VASS	VASS
D [7]	D	D	D	D
	D [9]	D	D	D
		D [4]	D	D
			D [5]	?
				?
				VASS

Table 1. Overview of the decidability of regular separability for VASS subclasses. Here, \geq -VASS and c-VASS are short for coverability VASS languages, and commutative VASS languages, respectively. The entry in the column for class \mathcal{K}_0 and the row for class \mathcal{K}_1 denotes decidability of regular separability of languages of \mathcal{K}_0 from languages of \mathcal{K}_1 . The entries in bold are new consequences of results in this paper.

Corollary 2 ([5, 6]). *Given VASS languages K, L and words $w_1, \dots, w_m \in \Sigma^*$ such that $K, L \subseteq w_1^* \cdots w_m^*$, it is decidable whether $K \mid L$.*

After Theorem 1, the investigation went on to 1-dim. VASS [7]:

Theorem 3 ([7]). *Given 1-VASS languages K and L , it is decidable whether $K \mid L$.*

Moreover, the next theorem has been established in [4].

Theorem 4 ([4]). *Given \mathbb{Z} -VASS languages $K, L \subseteq \Sigma^*$, it is decidable whether $K \mid L$.*

It should be noted that the authors of [4] speak of Parikh automata, but these are equivalent to \mathbb{Z} -VASS: Parikh automata are equivalent to reversal-bounded counter machines [3, Prop. 3.13] and the latter are equivalent to blind counter machines [14, Theorem 2], which are the same as \mathbb{Z} -VASS.

Finally, a recent general result shows that any two coverability languages of well-structured transition systems [11, 12] fulfilling some mild conditions are regular separable if and only if they are disjoint [9]. In particular it applies to the situation when the systems are upward-compatible and one of them is finitely branching, which is the case for coverability VASS languages:

Theorem 5 ([9]). *Given coverability VASS languages $K, L \subseteq \Sigma^*$, it is decidable whether $K \mid L$.*

3 Main Results

In this section, we record the main results of this work. See Table 1 for an overview. Our first main result is that regular separability is decidable if one input language is a VASS language and the other is the language of a 1-VASS.

Theorem 6. *Given a VASS V_0 and a 1-dim. VASS V_1 , it is decidable whether $L(V_0) \mid L(V_1)$.*

This generalizes Theorem 3, because here, one of the input languages can be an arbitrary VASS language. Our second

main result generalizes Theorem 5 in the same way as Theorem 6 extends Theorem 3:

Theorem 7. *Given a VASS V_0 and a coverability VASS V_1 , it is decidable whether $L(V_0) \mid L(V_1)$.*

Our third main result is decidability of regular separability of a given VASS language from a given \mathbb{Z} -VASS language.

Theorem 8. *Given a VASS V_0 and a \mathbb{Z} -VASS V_1 , it is decidable whether $L(V_0) \mid L(V_1)$.*

As before, this significantly generalizes Theorem 4. Our proof of Theorem 8 relies on Theorem 1. At first glance, it might seem that Theorem 8 is unrelated to regular separability of commutative VASS languages. However, a simple observation shows that Theorem 8 also strengthens Theorem 1. This is because for deciding regular separability of commutative VASS languages, one may assume that one of the input languages is Z_n :

Proposition 9. *Let $\Gamma_n = \{a_1, \dots, a_n\} \subseteq \Sigma_n$. For any $K, L \subseteq \Gamma_n^*$, we have $\Pi(K) \mid \Pi(L)$ if and only if $\Pi(K\bar{L}) \mid Z_n$.*

Since $\Pi(K\bar{L})$ is a VASS language and Z_n is a \mathbb{Z} -VASS language, this means Theorem 8 indeed strengthens Theorem 1.

The rest of this section proves Proposition 9, which follows from two simple observations. The first concerns separability of subsets of monoids. If M is a monoid and $K, L \subseteq M$ are subsets, then K and L are called *separable* if there is a morphism $\varphi: M \rightarrow F$ into a finite monoid F such that $\varphi(K) \cap \varphi(L) = \emptyset$. Clearly, if $K, L \subseteq \Sigma^*$, then K and L are separable if and only if $K \mid L$. Therefore, it creates no inconsistencies to write $K \mid L$ whenever $K, L \subseteq M$ are separable. Let $\Delta = \{(m, m) \mid m \in M\} \subseteq M \times M$.

Lemma 10. *Let $K, L \subseteq M$. Then $K \mid L$ if and only if $K \times L \mid \Delta$.*

Proof. If $K \mid L$ with $\varphi: M \rightarrow F$, define $\varphi': M \times M \rightarrow F \times F$ by $\varphi'(u, v) = (\varphi(u), \varphi(v))$. Then clearly $\varphi'(K \times L) \cap \varphi'(\Delta) = \emptyset$. Conversely, if $K \times L \mid \Delta$ with a morphism $\varphi: M \times M \rightarrow F$, let $\varphi': M \rightarrow F$ be the morphism with $\varphi'(u) = \varphi(u, 1)$ for $u \in M$. Then we have $\varphi'(K) \cap \varphi'(L) = \emptyset$, because if there were $u \in K$, $v \in L$ with $\varphi'(u) = \varphi'(v)$, then

$$\begin{aligned} \varphi(K \times L) \ni \varphi(u, v) &= \varphi(u, 1)\varphi(1, v) \\ &= \varphi(v, 1)\varphi(1, v) = \varphi(v, v) \in \varphi(\Delta), \end{aligned}$$

and thus $\varphi(K \times L) \cap \varphi(\Delta) \neq \emptyset$, which is impossible. \square

For subsets $S, T \subseteq \mathbb{N}^\Sigma$, separability is equivalent to unary separability as studied by [5]. We now have:

$$\begin{aligned} \Pi(K) \mid \Pi(L) &\Leftrightarrow \Psi(K) \mid \Psi(L) \Leftrightarrow \Psi(K) \times \Psi(L) \mid \Delta \\ &\Leftrightarrow \Psi(K) \times \Psi(\bar{L}) \mid \Delta' \Leftrightarrow \Pi(K\bar{L}) \mid Z_n, \end{aligned}$$

where $\Delta' = \{\mathbf{u} \in \mathbb{N}^{\Sigma \cup \bar{\Sigma}} \mid \mathbf{u}(a_i) = \mathbf{u}(\bar{a}_i) \text{ for } i \in [1, n]\}$. Here, the third equivalence is just renaming components. The second equivalence is Lemma 10; the first and last equivalence are due to an observation from [5]: In [5, Lemma 11], it is

shown that for languages $K, L \subseteq \Sigma^*$, we have $\Pi(K) \mid \Pi(L)$ if and only if $\Psi(K) \mid \Psi(L)$. This completes Proposition 9.

4 VASS vs. 1-VASS

In this section, we introduce our approach to regular separability together with the first application: Regular separability of VASS languages and 1-dim. VASS languages.

Our approach is inspired by the decision procedure for regular separability for one dimensional VASS [7]. There, given languages K and L , the idea is to construct *approximants* K_k and L_k for $k \in \mathbb{N}$. Here, K_k and L_k are regular languages with $K \subseteq K_k$ and $L \subseteq L_k$ for which one can show that $K \mid L$ if and only if there is a $k \in \mathbb{N}$ with $K_k \cap L_k = \emptyset$. The latter condition is then checked algorithmically.

We simplify this idea in two ways. First, we show that for many language classes, one may assume that one of the two input languages is fixed (or fixed up to a parameter). Roughly speaking, if a language class \mathcal{K} is defined by machines involving a finite-state control, then \mathcal{K} is typically a full trio since a transduction can be applied using a product construction in the finite-state control. Moreover, there is often a simple set \mathcal{G} of languages so that \mathcal{K} is the full trio generated by \mathcal{G} . For example, as mentioned above, \mathcal{V}_n is generated by D_n for each $n \geq 1$. This makes the following simple lemma very useful.

Lemma 11. *Let T be a rational transduction. Then $L \mid TK$ if and only if $T^{-1}L \mid K$.*

Proof. Suppose $L \subseteq R$ and $R \cap TK = \emptyset$ for some regular R . Then clearly $T^{-1}L \subseteq T^{-1}R$ and $T^{-1}R \cap K = \emptyset$. Therefore, the regular set $T^{-1}R$ witnesses $T^{-1}L \mid K$. Conversely, if $T^{-1}L \mid K$, then $K \mid T^{-1}L$ and hence, by the first direction, $(T^{-1})^{-1}K \mid L$. Since $(T^{-1})^{-1} = T$, this reads $TK \mid L$ and thus $L \mid TK$. \square

Suppose we have full trios \mathcal{K}_0 and \mathcal{K}_1 generated by languages G_0 and G_1 , respectively. Then, to decide if $T_0G_0 \mid T_1G_1$, we can check whether $T_1^{-1}T_0G_0 \mid G_1$. Since $T_1^{-1}T_0$ is also a rational transduction and hence $T_1^{-1}T_0G_0$ belongs to \mathcal{K}_0 , this means we may assume that one of the input languages is G_1 . This effectively turns separability into a decision problem with one input language L where we ask whether $L \mid G_1$.

Going further in this direction, instead of considering approximants of two languages, we just consider regular overapproximations of G_1 and decide whether L intersects all of them. However, we find it more convenient to switch to the complement and think in terms of “basic separators of G_1 ” instead of overapproximations of G_1 . Informally, we call a family of regular languages *basic separators of G_1* if (i) each of them is disjoint from G_1 and (ii) every regular language R disjoint from G_1 is included in a finite union of basic separators. This implies that $L \mid G_1$ if and only if there exists a finitely many basic separators S_1, \dots, S_k such that L is contained in the union $\bigcup_{i \in [1, k]} S_i$. Note that for each language G_1 there trivially exists a family of basic separators;

just take the family of all regular languages disjoint from G_1 . Our approach is to identify a family of basic separators for which it is decidable whether a language from \mathcal{K}_0 is included in a finite union of them.

Basic separators for one-dimensional VASS. Let us see this approach in an example and prove Theorem 6. Since \mathcal{V}_1 is generated as a full trio by D_1 , Lemma 11 tells us that it suffices to decide whether a given VASS language L fulfills $L \mid D_1$. Now the first step is to develop a notion of basic separators for D_1 .

Since $D_1 \subseteq \Sigma_1^*$, we assume now that $n = 1$, meaning $\varphi: \Sigma_1^* \rightarrow \mathbb{Z}$. One way a finite automaton can guarantee non-membership in D_1 is by modulo counting. For $k \in \mathbb{N}$, let

$$M_k = \{w \in \Sigma_1^* \mid \varphi(w) \not\equiv 0 \pmod{k}\},$$

which is regular. Another option for an automaton to make sure an input word w avoids D_1 is to guarantee (i) for prefixes v of w , that $\varphi(v)$ does not exceed some k if $\text{drop}(v) = 0$ and (ii) $\varphi(w) \neq 0$. For $w \in \Sigma_1^*$, let $\mu(w) = \max\{\varphi(v) \mid v \text{ is a prefix of } w \text{ and } \text{drop}(v) = 0\}$ and

$$B_k = \{w \in \Sigma_1^* \mid w \notin D_1 \text{ and } \mu(w) \leq k\}$$

Here, the B stands for “bounded counter value”. It is obvious that the languages B_k are disjoint from D_1 . We observe that they are regular: A word w with $\mu(w) \leq k$ avoids D_1 if and only if (i) φ drops below zero after a prefix where φ is confined to $[0, k]$ or (ii) φ stays above zero and thus assumes values in $[0, k]$ throughout. The third type of separator is a symmetric right-to-left version of B_k , namely

$$\begin{aligned} \bar{B}_k^{\text{rev}} &= \{\bar{w}^{\text{rev}} \mid w \in B_k\} \\ &= \{w \in \Sigma_1^* \mid w \notin D_1 \text{ and } \mu(\bar{w}^{\text{rev}}) \leq k\} \end{aligned}$$

Then we have indeed a family of basic separators for D_1 :

Lemma 12. *Let $R \subseteq \Sigma_1^*$ be a regular language. Then $R \cap D_1 = \emptyset$ if and only if R is included in $M_k \cup B_\ell \cup \bar{B}_m^{\text{rev}}$ for some $k, \ell, m \in \mathbb{N}$.*

The proof of Lemma 12 is very similar to the proof of Lemma 8 in [7], but phrased in a slightly different setting. It is a relatively simple pumping argument: If R evades M_k , B_ℓ , and \bar{B}_m^{rev} for every $k, \ell, m \in \mathbb{N}$, then it has words w with $\varphi(w) \equiv 0 \pmod{k}$, $\mu(w) \geq \ell$, $\mu(\bar{w}^{\text{rev}}) \geq m$ for arbitrarily high $k, \ell, m \in \mathbb{N}$. Then w has a prefix u with $\text{drop}(u) = 0$ and $\varphi(u) \geq \ell$ and a suffix v with $\text{drop}(\bar{v}^{\text{rev}}) = 0$ and $\varphi(\bar{v}^{\text{rev}}) \geq m$. One can then pump an infix in the prefix and an infix in the suffix so as to (i) drive up the φ -values in the middle of the word and (ii) compensate the non-zero value of $\varphi(w)$. If k is a multiple of $n!$ and $\ell, m > n$, where n is the number of states of an automaton for R , then this yields a word in D_1 .

Deciding separability. The next step in our approach is to decide whether a given VASS language L is contained in $M_k \cup B_\ell \cup \bar{B}_m^{\text{rev}}$ for some $k, \ell, m \in \mathbb{N}$. Of course this is the case if and only if $L \subseteq M_k \cup B_k \cup \bar{B}_k^{\text{rev}}$ for some $k \in \mathbb{N}$. Thus,

Lemma 12 essentially tells us that whether $L \mid D_1$ holds only depends on three numbers associated to each word from L . Consider the function $\sigma: \Sigma_1^* \rightarrow \mathbb{N}^3$ with $\sigma(w) = (\mu(w), \varphi(w), \mu(\bar{w}^{\text{rev}}))$. We call a subset $S \subseteq \mathbb{N}^3$ *separable* if there is a $k \in \mathbb{N}$ so that for every $(x_1, x_2, x_3) \in S$, we have $x_1 \leq k$ or $x_3 \leq k$ or $x_2 \not\equiv 0 \pmod{k}$. Then, Lemma 12 can be formulated as:

Lemma 13. *Let $L \subseteq \Sigma_1^*$. If $L \cap D_1 = \emptyset$, then $L \mid D_1$ if and only if $\sigma(L)$ is separable.*

This enables us to transform L into a bounded language \hat{L} that behaves the same in terms of separability from D_1 . Let

$$\begin{aligned} \hat{L} &= \{a_1^m \bar{a}_1^{m+1} a_1^r \bar{a}_1^s a_1^{n+1} \bar{a}_1^n \mid \exists w \in L: \\ &\quad m \leq \mu(w), n \leq \mu(\bar{w}^{\text{rev}}), r - s = \varphi(w)\}. \end{aligned}$$

Note that if $v = a_1^m \bar{a}_1^{m+1} a_1^r \bar{a}_1^s a_1^{n+1} \bar{a}_1^n$, then we have $\mu(v) = m$ and $\mu(\bar{v}^{\text{rev}}) = n$ and $\varphi(v) = r - s$. Therefore, the set $\sigma(\hat{L})$ is separable if and only if $\sigma(L)$ is separable. Hence, we have:

Lemma 14. *For every $L \subseteq \Sigma_1^*$ with $L \cap D_1 = \emptyset$, we have $L \mid D_1$ if and only if $\hat{L} \mid D_1$.*

Using standard VASS constructions, we can turn L into \hat{L} .

Lemma 15. *Given a VASS language $L \subseteq \Sigma_1^*$, one can construct a VASS for \hat{L} .*

This leaves us with the task of deciding whether $\hat{L} \mid D_1$. Since $\hat{L} \subseteq B$ with $B = a_1^* \bar{a}_1^* a_1^* \bar{a}_1^* a_1^* \bar{a}_1^*$, we have $\hat{L} \mid D_1$ if and only if $\hat{L} \mid (D_1 \cap B)$. As subsets of B , both \hat{L} and $D_1 \cap B$ are bounded languages and we can decide whether $\hat{L} \mid (D_1 \cap B)$ using Corollary 2.

5 VASS vs. Coverability VASS

Let us now show Theorem 7. In [9] it was shown that two coverability VASS languages K and L are regularly separable if and only if $K \cap L = \emptyset$ (the result in [9] applies to all languages of well-structured transition systems). However, when deciding $K \mid L$ for a VASS language K and a coverability VASS language L , a simple disjointness check is not enough: If $K = \{a^n b^m \mid n < m\}$ and $L = \{a^n b^m \mid n \geq m\}$, then K is in \mathcal{V} and L is in \mathcal{C} and we have $K \cap L = \emptyset$, but not $K \mid L$.

Instead, we use our approach to reduce separability to the simultaneous unboundedness problem [10, 40]. A language $L \subseteq a_1^* \cdots a_n^*$ is *simultaneously unbounded* if for every $k \in \mathbb{N}$, there is a word $a_1^{x_1} \cdots a_n^{x_n} \in L$ with $x_1, \dots, x_n \geq k$. The *simultaneous unboundedness problem (SUP)* asks, given a language $L \subseteq a_1^* \cdots a_n^*$, whether L is simultaneously unbounded. For VASS languages, this problem is decidable. This follows from computability of downward closures [16] or from general results on unboundedness problems for VASS [6].

Basic separators for coverability VASS. As before, we develop a notion of basic separators. We start with a version of the sets B_k , where we use C_1 instead of D_1 . We set

$$B'_k = \{w \in \Sigma_1^* \mid w \notin C_1 \text{ and } \mu(w) \leq k\}.$$

Just like for B_k , the set B'_k is clearly disjoint from C_1 . Moreover, we need variants of these in higher dimension: For $i \in [1, n]$ and $k \in \mathbb{N}$, let $B_{i,k} = \lambda_i^{-1}(B'_k)$. This means, $B_{i,k}$ is the set of walks through \mathbb{Z}^n which have a prefix p such that in coordinate i , the walk p goes below zero, but also never exceeds k before it does so. The sets $B_{i,k}$ are clearly regular, because an automaton can maintain the φ -value in coordinate i of the read prefix in its state as long as it stays positive: During that time, the value belongs to $[0, k]$.

Lemma 16. *Let $R \subseteq \Sigma_n^*$ be a regular language. Then $R \cap C_n = \emptyset$ if and only if R is included in a finite union of sets of the form $B_{i,k}$ for $i \in [1, n]$ and $k \in \mathbb{N}$.*

Roughly speaking, we argue as follows. If R evades $B_{i,k}$ for each $k \in \mathbb{N}$ and each $i \in [1, n]$, then this yields an infinite sequence of walks in R that attain unbounded positive values in each coordinate. Using arguments from the theory of well-quasi orderings, we can transform this into a sequence of walks in R that in some coordinate $i \in [1, n]$ stay non-negative and are still unbounded in the other coordinates. Repeating this process results in a walk that stays non-negative in every coordinate and thus belongs to C_n .

Lemma 16 tells us that to decide whether $L \mid C_n$ for a given language L , we have to check whether L is included in $B_{1,k} \cup \dots \cup B_{n,k}$ for some $k \in \mathbb{N}$. Like in Theorem 6, we turn L into a different language. Using standard methods, we can show the following:

Lemma 17. *Given a VASS language $L \subseteq \Sigma_n^*$, one can construct a VASS for the language*

$$\hat{L} = \{a_1^{x_1} \dots a_n^{x_n} \mid \exists w \in L: \mu(\lambda_i(w)) \geq x_i \text{ for } i \in [1, n]\}.$$

Note that $w \in B_{i,k}$ if and only if $\lambda_i(w) \notin C_1$ and $\mu(\lambda_i(w)) \leq k$. Therefore, Lemma 16 implies that $L \mid C_n$ if and only if $L \cap C_n = \emptyset$ and \hat{L} is not simultaneously unbounded.

Remark 18. *In our decidability proof, we use a polynomial-time Turing reduction from regular separability of a VASS language and a coverability VASS language to the SUP for VASS languages. There is also such a reduction in the converse direction. This is because given a VASS language $L \subseteq a_1^* \dots a_n^*$, it is easy to construct in polynomial time a VASS for*

$$\tilde{L} = \{a_1^{x_1} \bar{a}_1^{x_1+1} \dots a_n^{x_n} \bar{a}_n^{x_n+1} \mid a_1^{x_1} \dots a_n^{x_n} \in L\}.$$

By Lemma 16, L is simultaneously unbounded if and only if $\tilde{L} \mid C_n$. Thus, the problems (i) regular separability of VASS languages and coverability VASS languages and (ii) the SUP for VASS languages are polynomially inter-reducible.

6 VASS vs. Integer VASS

In this section, we apply our approach to solving regular separability between a VASS language and a \mathbb{Z} -VASS language. Here, the collection of basic separators serves as a geometric characterization of separability. Proving that it is a set of basic separators is more involved than in Sections 4 and 5.

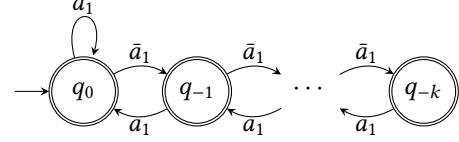


Figure 1. An automaton for $I_k \subseteq \Sigma_1^*$

6.1 A geometric characterization

Lemma 11 tells us that regular separability between a VASS language and a \mathbb{Z} -VASS language amounts to checking whether a given VASS language $L \subseteq \Sigma_n^*$ is included in some regular language $R \subseteq \Sigma_n^*$ with $R \cap Z_n = \emptyset$. Therefore, in this section, we classify the regular languages $R \subseteq \Sigma_n^*$ with $R \cap Z_n = \emptyset$.

A very simple type of such languages is given by modulo counting. For $\mathbf{u}, \mathbf{v} \in \mathbb{Z}^n$, we write $\mathbf{u} \equiv \mathbf{v} \pmod k$ if \mathbf{u} and \mathbf{v} are component-wise congruent modulo k . The language

$$M_k = \{w \in \Sigma_n^* \mid \varphi(w) \not\equiv \mathbf{0} \pmod k\}$$

is clearly regular and disjoint from Z_n .

Since Z_n is commutative (i.e. $\Pi(Z_n) = Z_n$), one might expect that it suffices to consider commutative separators. This is not the case: The language $L = (a_1 \bar{a}_1)^* a_1^+$ is regularly separable from Z_1 , but every commutative regular language including L intersects Z_1 . Therefore, our second type of regular languages disjoint from Z_n is non-commutative and we start to describe it in the case $n = 1$. Consider the language

$$D_{1,k} = \{w \in \Sigma_1^* \mid \varphi(w) \neq 0 \text{ and}$$

$$\text{for every infix } v \text{ of } w: \varphi(v) \geq -k\}.$$

The set $D_{1,k}$ is clearly disjoint from Z_1 . To see that $D_{1,k}$ is regular, let us first observe that the language $I_k = \{w \in \Sigma_1^* \mid \text{for every infix } v \text{ of } w: \varphi(v) \geq -k\}$ is regular, because the automaton in Fig. 1 accepts I_k : After reading a word w , the automaton's state reflects the difference $M - \varphi(w)$, where M is the maximal value $\varphi(v)$ for prefixes v of w . Second, the automaton \mathcal{A}_k in Fig. 2 satisfies $L(\mathcal{A}_k) \cap I_k = D_{1,k}$: As long as the seen prefix w satisfies $\varphi(w) \in [-k, k]$, the state of \mathcal{A}_k reflects $\varphi(w)$ exactly. However, as soon as \mathcal{A}_k encounters a prefix w with $\varphi(w) > k$, it enters q_∞ . From there, it accepts every suffix, because an input from I_k can never reach 0 under φ with such a prefix w . Thus, $D_{1,k}$ is regular.

The language $D_{1,k}$ has analogs in higher dimension. Instead of making sure the value of φ never drops more than k along one particular axis, one can impose this condition in an arbitrary direction $\mathbf{u} \in \mathbb{Z}^n$. For $\mathbf{u}, \mathbf{v} \in \mathbb{Q}^n$, $\mathbf{u} = (u_1, \dots, u_n)$, $\mathbf{v} = (v_1, \dots, v_n)$, we define $\langle \mathbf{u}, \mathbf{v} \rangle = u_1 v_1 + \dots + u_n v_n$. For every vector $\mathbf{u} \in \mathbb{Z}^n$ and $k \in \mathbb{N} \setminus \{0\}$, let

$$D_{\mathbf{u},k} = \{w \in \Sigma_n^* \mid \langle \varphi(w), \mathbf{u} \rangle \neq 0 \text{ and}$$

$$\text{for every infix } v \text{ of } w: \langle \varphi(v), \mathbf{u} \rangle \geq -k\}.$$

We think of the walks in $D_{\mathbf{u},k}$ as “drifting in direction \mathbf{u} ”, hence the name. To see that $D_{\mathbf{u},k}$ is regular, consider the

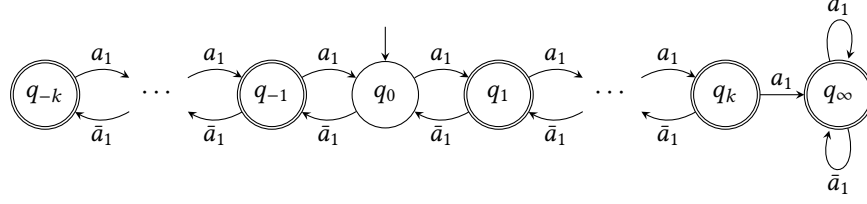


Figure 2. Automaton \mathcal{A}_k with $L(\mathcal{A}_k) \cap I_k = D_{1,k}$.

morphism $h_u: \Sigma_n^* \rightarrow \{a_1, \bar{a}_1\}^*$ with $x \mapsto a_1^{\langle \varphi(x), u \rangle}$ for $x \in \Sigma_n^*$. Here, we mean $a_1^\ell = \bar{a}_1^{|\ell|}$ and $\bar{a}_1^\ell = a_1^{|\ell|}$ in case $\ell \in \mathbb{Z}$, $\ell < 0$. Then we have $\langle \varphi(w), u \rangle = \varphi(h_u(w))$ for any $w \in \Sigma_n^*$ and hence $D_{u,k} = h_u^{-1}(D_{1,k})$. Therefore, $D_{u,k}$ inherits regularity from $D_{1,k}$.

The main result of this section is that the sets M_k and $D_{u,k}$ suffice to explain disjointness of regular languages from Z_n in the following sense.

Theorem 19. *Let $R \subseteq \Sigma_n^*$ be a regular language. Then $R \cap Z_n = \emptyset$ if and only if R is included in a finite union of languages of the form M_k and $D_{u,k}$ for $k \in \mathbb{N}$ and $u \in \mathbb{Z}^n$.*

We therefore say that $L \subseteq \Sigma_n^*$ is *geometrically separable* if L is contained in a finite union of languages of the form M_k and $D_{u,k}$. Then, we can formulate Theorem 19 as a geometric characterization of separability from Z_n .

Corollary 20. *For $L \subseteq \Sigma_n^*$, we have $L \mid Z_n$ if and only if L is geometrically separable.*

The rest of Section 6.1 is devoted to proving Theorem 19.

Overview of the proof. The “if” direction of Theorem 19 is clear. We show the “only if” direction for $L \subseteq \Sigma_n^*$ in several steps. We first associate to each finite automaton over Σ_n a (rational) cone in \mathbb{Q}^n , which must either encompass all of \mathbb{Q}^n or be included in a halfspace (Lemma 21).

The next step is to decompose the automaton for L into automata whose strongly connected components form a path. We then prove Theorem 19 in the case that such an automaton has the cone \mathbb{Q}^n (Lemma 22). In the case that the cone of an automaton \mathcal{A} is included in some halfspace, we show that $L(\mathcal{A})$ further decomposes into a part inside some $D_{u,k}$ and a part that stays close to some strict linear subspace $U \subseteq \mathbb{Q}^n$ (Lemma 23).

It thus remains to treat regular languages $L \subseteq \Sigma_n^*$ whose walks remain close to U . To this end, we transform $L(\mathcal{A})$ into a language in Σ_m^* where $m = \dim U < n$. The transformation does not affect disjointness from Z_n (resp. Z_m), regularity, or geometric separability (Lemmas 24 to 27 and Proposition 28). Since $m < n$, this allows us to apply induction.

Cones of automata. For a set $S \subseteq \mathbb{Q}^n$, the *cone generated* by S consists of all vectors $x_1 u_1 + \dots + x_\ell u_\ell$ where $x_1, \dots, x_\ell \in \mathbb{Q}_+$ and $u_1, \dots, u_\ell \in S$. To each automaton \mathcal{A} over Σ_n , we associate a cone as follows. If $w \in \Sigma_n^*$ labels a

path in \mathcal{A} , then $\varphi(w)$ is the *effect* of that path. Let $\text{cone}(\mathcal{A})$ be the cone generated by the effects of cycles of \mathcal{A} . Since every cycle effect is the sum of effects of simple cycles, we know that $\text{cone}(\mathcal{A})$ is generated by the effects of simple cycles. In particular, $\text{cone}(\mathcal{A})$ is finitely generated and the set of simple cycle effects can serve as a representation of $\text{cone}(\mathcal{A})$. A key ingredient in our proof is a dichotomy of cones (Lemma 21), which is a consequence of the well-known Farkas’ lemma [38, Corollary 7.1d]. A *half-space* is a subset of \mathbb{Q}^n of the form $\{x \in \mathbb{Q}^n \mid \langle x, u \rangle \geq 0\}$ for some $u \in \mathbb{Q}^n$, $u \neq 0$.

Lemma 21. *For every \mathcal{A} , either $\text{cone}(\mathcal{A}) = \mathbb{Q}^n$ or $\text{cone}(\mathcal{A})$ is included in some half-space.*

Farkas’ lemma states that if a system $Ax = b$ of linear inequalities has no solution over \mathbb{Q}_+ , then this is certified by a half-space that contains Ax for every x over \mathbb{Q}_+ , but does not contain b . To show Lemma 21, we express the condition $b \notin \text{cone}(\mathcal{A}) = \mathbb{Q}^n$ as non-solvability of such a system. Then, the half-space provided by Farkas’ lemma includes $\text{cone}(\mathcal{A})$.

Linear automata. For an automaton \mathcal{A} , consider the directed acyclic graph (dag) consisting of strongly connected components of \mathcal{A} . If this dag is a path, then \mathcal{A} is called *linear*. Given an automaton \mathcal{A} , we can construct linear automata $\mathcal{A}_1, \dots, \mathcal{A}_\ell$ with $L(\mathcal{A}) = L(\mathcal{A}_1) \cup \dots \cup L(\mathcal{A}_\ell)$.

Lemma 22. *Let \mathcal{A} be a linear automaton with $\text{cone}(\mathcal{A}) = \mathbb{Q}^n$. If $L(\mathcal{A}) \cap Z_n = \emptyset$, then $L(\mathcal{A}) \subseteq M_k$ for some k .*

Proof. Since $\text{cone}(\mathcal{A}) = \mathbb{Q}^n$, we know that in particular the vectors $e_1, -e_1, \dots, e_n, -e_n$ belong to $\text{cone}(\mathcal{A})$. This means there are cycles labeled w_1, \dots, w_p such that both e_i and $-e_i$ are linear combinations of $\varphi(w_1), \dots, \varphi(w_p)$ with coefficients in \mathbb{Q}_+ , for every $i \in [1, n]$. Therefore, there is a $k \in \mathbb{N}$ such that $k \cdot e_i$ and $-k \cdot e_i$ are linear combinations of $\varphi(w_1), \dots, \varphi(w_p)$ with coefficients in \mathbb{N} , for every $i \in [1, n]$. We claim that $L(\mathcal{A}) \subseteq M_k$. Towards a contradiction, suppose $w \in L(\mathcal{A})$ with $\varphi(w) \equiv 0 \pmod k$. Since \mathcal{A} is linear, we can take the run for w and insert cycles so that the resulting run visits every state in \mathcal{A} . Instead of inserting every cycle once, we insert it k times, so that the resulting run (i) visits every state in \mathcal{A} and (ii) reads a word $w' \in \Sigma_n^*$ with $\varphi(w') \equiv \varphi(w) \pmod k$. Now since $\varphi(w') \equiv \varphi(w) \equiv 0 \pmod k$, we can write $-\varphi(w') = x_1 \varphi(w_1) + \dots + x_p \varphi(w_p)$ with coefficients $x_1, \dots, x_p \in \mathbb{N}$. Since in the run for w' , every state of \mathcal{A} is visited, we can insert cycles corresponding

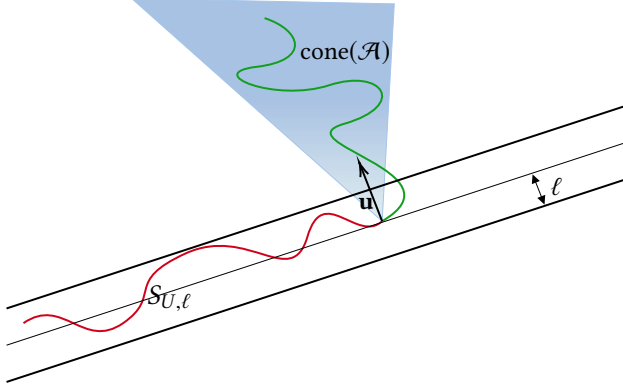


Figure 3. Two runs (red and green) inside $D_{u,k} \cup S_{U,\ell}$.

to the w_1, \dots, w_p : For each $i \in [1, p]$, insert the cycle for w_i exactly x_i times. Let w'' be the word read by the resulting run and note that $w'' \in L(\mathcal{A})$. Then we have $\varphi(w'') = \varphi(w') + x_1\varphi(w_1) + \dots + x_p\varphi(w_p) = \mathbf{0}$ and thus $w'' \in Z_n$, contradicting $L(\mathcal{A}) \cap Z_n = \emptyset$. \square

Walks that stay close to a subspace. Suppose we are given a vector space $U \subseteq \mathbb{Q}^n$ (represented by a basis) with $m = \dim U < n$ and a bound $\ell \geq 0$. Let $\|\mathbf{u}\| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$. For $U \subseteq \mathbb{Q}^n$ and $\mathbf{v} \in \mathbb{Q}^n$, we set $d(\mathbf{v}, U) = \inf \{\|\mathbf{v} - \mathbf{x}\| \mid \mathbf{x} \in U\}$. Then we define the set

$$S_{U,\ell} = \{w \in \Sigma_n^* \mid \text{for every prefix } v \text{ of } w: d(\varphi(v), U) \leq \ell\}.$$

Hence, $S_{U,\ell}$ collects those walks whose prefixes stay close to the subspace U .

Lemma 23. *Let \mathcal{A} be an automaton such that $\text{cone}(\mathcal{A})$ is contained in some half-space. One can compute $k, \ell \in \mathbb{N}$, $\mathbf{u} \in \mathbb{Z}^n$, and a strict subspace $U \subseteq \mathbb{Q}^n$ with $L(\mathcal{A}) \subseteq D_{\mathbf{u},k} \cup S_{U,\ell}$.*

Proof. Suppose $\text{cone}(\mathcal{A}) \subseteq H$, where $H = \{\mathbf{v} \in \mathbb{Q}^n \mid \langle \mathbf{v}, \mathbf{u} \rangle \geq 0\}$ for some vector $\mathbf{u} \in \mathbb{Q}^n \setminus \{\mathbf{0}\}$. Without loss of generality, we may assume $\mathbf{u} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$. Let $U = \{\mathbf{v} \in \mathbb{Q}^n \mid \langle \mathbf{v}, \mathbf{u} \rangle = 0\}$. Then clearly $\dim U = n-1$. Observe that since $\text{cone}(\mathcal{A}) \subseteq H$, we have $\langle \mathbf{v}, \mathbf{u} \rangle \geq 0$ for every cycle effect $\mathbf{v} \in \mathbb{Z}^n$ of \mathcal{A} . Let k be the number of states in \mathcal{A} . Now, whenever $w \in L(\mathcal{A})$ and v is an infix of w , then $\langle \varphi(v), \mathbf{u} \rangle \geq -k$: If $\langle \varphi(v), \mathbf{u} \rangle < -k$, then the path reading v must contain a cycle reading $v' \in \Sigma_n^*$ with $\langle \varphi(v'), \mathbf{u} \rangle < 0$, which contradicts $\text{cone}(\mathcal{A}) \subseteq H$.

We claim that $L(\mathcal{A}) \subseteq D_{\mathbf{u},k} \cup S_{U,k}$. Let $w \in L(\mathcal{A})$. We distinguish two cases. *Case 1:* Suppose w has a prefix v with $\langle \varphi(v), \mathbf{u} \rangle > k$. Write $w = vv'$. As argued above, we have $\langle \varphi(v'), \mathbf{u} \rangle \geq -k$. Hence, $\langle \varphi(w), \mathbf{u} \rangle = \langle \varphi(v), \mathbf{u} \rangle + \langle \varphi(v'), \mathbf{u} \rangle > 0$. Thus, we have $w \in D_{\mathbf{u},k}$. *Case 2:* Suppose for every prefix v of w , we have $\langle \varphi(v), \mathbf{u} \rangle \leq k$. Then, for every prefix v of w , we have $-k \leq \langle \varphi(v), \mathbf{u} \rangle \leq k$ and thus $d(\varphi(v), U) = |\langle \varphi(v), \mathbf{u} \rangle| / \|\mathbf{u}\| \leq k$. Thus, $w \in S_{U,k}$. \square

Mapping to lower dimension. Lemma 23 tells us that if $\text{cone}(\mathcal{A})$ is included in some halfspace, then $L(\mathcal{A})$ can be split into (i) a part $L \cap D_{\mathbf{u},k}$ that is already geometrically separable and (ii) a part $L \cap S_{U,\ell}$ that stays close to a strict subspace $U \subseteq \mathbb{Q}^n$. Therefore, to complete the proof that regular languages disjoint from Z_n are geometrically separable, it remains to treat subsets of $S_{U,\ell}$. We will now show that they can be transformed into a language in Σ_m^* , where $m = \dim U < n$. This transformation will not affect regularity, geometric separability or disjointness from Z_n (resp. Z_m) and thus allow us to apply induction.

This transformation will be performed by a transducer. The transducer will consist of three steps, *coordinate transformation* (f), *intersection* ($R_{V,p}$), and *projection* (π_m). In the coordinate transformation, we translate L from walks that stay close to U into walks that stay close to $V = \{(v_1, \dots, v_n) \in \mathbb{Q}^n \mid v_{m+1} = \dots = v_n = 0\}$. The intersection will then select only those walks that not only stay close to V , but even arrive in V . Finally, we project away the coordinates $m+1, \dots, n$ and thus have walks in \mathbb{Z}^m .

We now describe each of the three steps. For the coordinate transformation, we apply a linear map A to the walk in L that maps U to V . Let us define this map as a matrix $A \in \mathbb{Z}^{n \times n}$. We choose an orthogonal basis $\mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{Z}^n$ of \mathbb{Q}^n such that $\mathbf{b}_1, \dots, \mathbf{b}_m$ is a basis for U . This can be done, e.g. using Gram-Schmidt orthogonalisation [24]. If $B \in \mathbb{Z}^{n \times n}$ is the matrix whose columns are $\mathbf{b}_1, \dots, \mathbf{b}_n$, then B is invertible and maps V to U . Thus the inverse $B^{-1} \in \mathbb{Q}^n$ maps U to V . We can clearly choose an $\alpha \in \mathbb{Z}$ such that $\alpha B^{-1} \in \mathbb{Z}^{n \times n}$ and we set $A = \alpha B^{-1}$. For each $i \in [1, n]$, choose a word $w_i \in \Sigma_n^*$ with $\varphi(w_i) = A\varphi(a_i)$ and let $f: \Sigma_n^* \rightarrow \Sigma_n^*$ be the morphism with $f(a_i) = w_i$ and $f(\bar{a}_i) = \bar{w}_i$. Now f indeed transforms walks close to U into walks close to V :

Lemma 24. *We can compute $p \in \mathbb{N}$ with $f(S_{U,\ell}) \subseteq S_{V,p}$.*

Moreover, applying f does not introduce geometric separability and preserves disjointness with Z_n .

Lemma 25. *If $f(L)$ is geometrically separable for $L \subseteq \Sigma_n^*$, then so is L . We have $L \cap Z_n = \emptyset$ if and only if $f(L) \cap Z_n = \emptyset$.*

The first statement holds because $f^{-1}(M_k) \subseteq M_k$ and $f^{-1}(D_{\mathbf{u},k}) \subseteq D_{A^T \mathbf{u},k}$, where A^T is the transpose of A . The second is due to A being injective and linear.

For the second step of our transformation (intersection), we observe the following. Since the walks in $S_{V,p}$ stay close to V , there is a finite set $F \subseteq \mathbb{Z}^{n-m}$ of possible difference vectors between a point $\varphi(v)$ reached by a prefix v of a word in $S_{V,p}$ and the point closest to $\varphi(v)$ in V . Therefore, the set $R_{V,p}$ of walks in $S_{V,p}$ that also arrive in V is regular: One can maintain the current distance vector in the state. To make this formal, let $\pi_j: \mathbb{Q}^n \rightarrow \mathbb{Q}^j$ denote the projection on the last j coordinates, $\pi_j(v_1, \dots, v_n) = (v_{n-j+1}, \dots, v_n)$. Then we have $d(\mathbf{v}, V) = \|\pi_{n-m}(\mathbf{v})\|$ for every $\mathbf{v} \in \mathbb{Q}^n$. If v is a prefix of $w \in S_{V,p}$, then $d(\varphi(v), V) \leq p$ implies $\|\pi_{n-m}(\varphi(v))\| \leq p$ and

hence there is a finite set $F \subseteq \mathbb{Z}^{n-m}$ such that $\bar{\pi}_{n-m}(\varphi(v)) \in F$ for every prefix v of some $w \in S_{V,p}$. Thus, the set $R_{V,p} = \{w \in S_{V,p} \mid \varphi(w) \in V\}$ is regular. The second step of our transformation is to intersect with $R_{V,p}$.

Lemma 26. *Let $L \subseteq S_{V,p}$. If $L \cap R_{V,p}$ is geometrically separable, then so is L . We have $L \cap Z_n = \emptyset$ if and only if $(L \cap R_{V,p}) \cap Z_n = \emptyset$.*

The first statement is due to $L \setminus R_{V,p}$ consisting of words w where for every prefix v , we have $\|\bar{\pi}_{n-m}(\varphi(w))\| \leq p$ and also $\bar{\pi}_{n-m}(\varphi(w)) \neq \mathbf{0}$. Thus, $L \setminus R_{V,p}$ is included in M_k for some $k \in \mathbb{N}$. In other words, by intersecting with $R_{V,p}$ we cut out words that are separable from Z_n via M_k and thus we do not affect geometric separability. The same reasoning gives the second statement.

In our third step, we project onto the first m coordinates: We define $\pi_m: \Sigma_n^* \rightarrow \Sigma_m^*$ as the morphism with $\pi_m(a_i) = a_i$, $\pi_m(\bar{a}_i) = \bar{a}_i$ for $i \in [1, m]$, and $\pi_m(a_i) = \pi_m(\bar{a}_i) = \varepsilon$ for $i \in [m+1, n]$. In other words, π_m deletes the letters a_i and \bar{a}_i for $i \in [m+1, n]$.

Lemma 27. *Let $L \subseteq R_{V,p}$. If $\pi_m(L)$ is geometrically separable, then so is L . Moreover, $L \cap Z_n = \emptyset$ if and only if $\pi_m(L) \cap Z_n = \emptyset$.*

The coordinates $m+1, \dots, n$ are always zero at the end of walks in $R_{V,p}$. This permits a straightforward translation of basic separators for $\pi_m(L)$ to basic separators for L in the proof of Lemma 27.

We are now prepared to define our transformation: Let $T_{U,\ell} \subseteq \Sigma_n^* \times \Sigma_m^*$ be the transduction with $T_{U,\ell}L = \pi_m(f(L) \cap R_{V,p})$. Then Lemmas 25 to 27 clearly imply:

Proposition 28. *Let $L \subseteq S_{U,\ell}$. If $T_{U,\ell}L \subseteq \Sigma_m^*$ is geometrically separable, then so is L . Also, $L \cap Z_n = \emptyset$ if and only if $(T_{U,\ell}L) \cap Z_m = \emptyset$. Thus, $L \mid Z_n$ if and only if $(T_{U,\ell}L) \mid Z_m$.*

Let us now prove Theorem 19. Suppose $R \subseteq \Sigma_n^*$ and $R \cap Z_n = \emptyset$. We show by induction on the dimension n that then, R is included in a finite union of sets of the form M_k and $D_{u,k}$. Let $R = L(\mathcal{A})$ for an automaton \mathcal{A} . Since \mathcal{A} can be decomposed into a finite union of linear automata, it suffices to prove the claim in the case that \mathcal{A} is linear. If $\text{cone}(\mathcal{A}) = \mathbb{Q}^n$, then Lemma 22 tells us that $L(\mathcal{A}) \subseteq M_k$ for some $k \in \mathbb{N}$. If $\text{cone}(\mathcal{A})$ is contained in some half-space, then according to Lemma 23, we have $R \subseteq D_{u,k} \cup S_{U,\ell}$ for some $u \in \mathbb{Q}^n \setminus \{\mathbf{0}\}$, $k, \ell \in \mathbb{N}$, and strict subspace $U \subseteq \mathbb{Q}^n$. This implies that the regular language $R \setminus D_{u,k}$ is included in $S_{U,\ell}$. We may therefore apply Proposition 28, which yields $T_{U,\ell}(R \setminus D_{u,k}) \cap Z_m = \emptyset$. Since $T_{U,\ell}(R \cap S_{U,\ell}) \subseteq \Sigma_m^*$ with $m = \dim U < n$, induction tells us that $T_{U,\ell}(R \setminus D_{u,k})$ is geometrically separable and hence, by Proposition 28, $R \setminus D_{u,k}$ is geometrically separable. Since $R \subseteq D_{u,k} \cup (R \setminus D_{u,k})$, R is geometrically separable.

6.2 The decision procedure

In this section, we apply Corollary 20 to prove Theorem 8.

Before we prove Theorem 8 let us explain why a particular straightforward approach does not work. Corollary 20 tells us that in order to decide whether $L \mid Z_n$ for $L \subseteq \Sigma_n^*$, it suffices to check whether there are $k \in \mathbb{N}$, $\ell_1, \dots, \ell_m \in \mathbb{N}$, and vectors $u_1, \dots, u_m \in \mathbb{Z}^n$ such that $L \subseteq M_k \cup D_{u_1, \ell_1} \cup \dots \cup D_{u_m, \ell_m}$. It is tempting to conjecture that there is a finite collection of direction vectors $F \subseteq \mathbb{Z}^n$ (such as a basis together with negations) so that for a given language L , such an inclusion holds only if it holds with some $u_1, \dots, u_m \in F$. In that case we would only need to consider scalar products of words in L with vectors in F and thus reformulate the problem over sections of reachability sets of VASS. However, this is not the case. For $u, v \in \mathbb{Q}^n$, we write $u \sim v$ if $\mathbb{Q}_+u = \mathbb{Q}_+v$. Since \sim has infinitely many equivalence classes and every class intersects \mathbb{Z}^n , the following shows that there is no fixed set of directions.

Proposition 29. *For each $u \in \mathbb{Z}^n$, there is a $k_0 \in \mathbb{N}$ such that for $k \geq k_0$, the following holds. For every $\ell, \ell_1, \dots, \ell_m \geq 1$, $u_1, \dots, u_m \in \mathbb{Z}^n$ with $u_i \sim u$ for $i \in [1, m]$, we have $D_{u,k} \not\subseteq M_\ell \cup D_{u_1, \ell_1} \cup \dots \cup D_{u_m, \ell_m}$.*

Outline of the algorithm. We now turn to the proof of Theorem 8. According to Corollary 20, we have to decide whether a given VASS language $L \subseteq \Sigma_n^*$ satisfies $L \subseteq M_k \cup D_{u_1, \ell_1} \cup \dots \cup D_{u_m, \ell_m}$ for some $k \in \mathbb{N}$, $u_1, \dots, u_m \in \mathbb{Z}^n$ and $\ell_1, \dots, \ell_m \in \mathbb{N}$. Our algorithm employs the KLMST decomposition (so named by Leroux and Schmitz [29] after its inventors) used by Sacerdote and Tenney [37], Mayr [30], Kosaraju [22], and Lambert [23] and recently cast in terms of ideal decompositions by Leroux and Schmitz [29]. The decomposition yields VASS languages L_1, \dots, L_p with $L = L_1 \cup \dots \cup L_p$, together with finite automata $\mathcal{A}_1, \dots, \mathcal{A}_p$ whose languages overapproximate L_1, \dots, L_p , respectively. We show that the \mathcal{A}_i are not only overapproximations, but are what we call “modular envelopes” (Theorem 30). This allows us to proceed similarly to the proof of Theorem 19. It suffices to check regular separability for each L_i . If $\text{cone}(\mathcal{A}_i) = \mathbb{Q}^n$, then it suffices to check whether $\Pi(L_i) \mid Z_n$ using Theorem 1 (see Lemma 31). If $\text{cone}(\mathcal{A}_i)$ is contained in some halfspace, then we transform L_i into a VASS language $\hat{L}_i \subseteq \Sigma_m^*$. Here, \hat{L}_i essentially captures the walks of L_i that stay close to a strict linear subspace $U \subseteq \mathbb{Q}^n$ with $m = \dim U < n$. Since $m < n$, we can then apply our algorithm recursively to \hat{L}_i .

We first explain the concept of modular envelopes. We then describe the algorithm for regular separability and finally, we show how to construct modular envelopes.

Modular envelopes. For a finite automaton \mathcal{A} with input alphabet Σ , let $\text{Loop}(\mathcal{A}) \subseteq \Sigma^*$ be the set of words that can be read on a cycle in \mathcal{A} . Recall that $\Psi(w)$ denotes the Parikh image of w . We say that an automaton \mathcal{A} is a *modular envelope* for a language $L \subseteq \Sigma^*$ if (i) $L \subseteq L(\mathcal{A})$ and (ii) for every selection $u_1, \dots, u_m \in \Sigma^*$ of words from $\text{Loop}(\mathcal{A})$ and every $w \in L$ and every $k \in \mathbb{N}$, there is a word $w' \in L$ so that

each u_j is an infix of w' and $\Psi(w') \equiv \Psi(w) \bmod k$, where the congruence is defined component-wise.

In other words, \mathcal{A} describes a regular overapproximation that is small enough that we can find every selection of \mathcal{A} 's loops as infixes in a word from L whose Parikh image is congruent modulo k to a given word from L . Using the KLMST decomposition, we prove:

Theorem 30. *Given a VASS language L , one can construct VASS languages L_1, \dots, L_p , together with a modular envelope \mathcal{A}_i for each L_i such that $L = L_1 \cup \dots \cup L_p$.*

We postpone the proof of Theorem 30 until Section 6.3 and first show how it is used to decide geometric separability.

Modular envelopes with cone \mathbb{Q}^n . By Lemma 21 we know that every cone either equals \mathbb{Q}^n or is included in some half-space. The following lemma will be useful in the first case.

Lemma 31. *Let $L \subseteq \Sigma_n^*$ be a language with a modular envelope \mathcal{A} . If $\text{cone}(\mathcal{A}) = \mathbb{Q}^n$ then the following are equivalent: (i) $L \mid Z_n$, (ii) $L \subseteq M_k$ for some $k \in \mathbb{N}$, (iii) $\Pi(L) \mid Z_n$.*

Proof. Note that (ii) implies (iii) immediately and that (iii) implies (i) because $L \subseteq \Pi(L)$. Thus, we only need to show that (i) implies (ii). By Corollary 20 if $L \mid Z_n$, then L is included in some $M_k \cup D_{u_1,k} \cup \dots \cup D_{u_m,k}$. We show that in our case we even have $L \subseteq M_k$.

Take any $w \in L$. We aim at constructing $w' \in L$ such that $w' \notin D_{u_i,k}$ for every $i \in [1, m]$ and additionally $\varphi(w') \equiv \varphi(w) \bmod k$. Since $\text{cone}(\mathcal{A}) = \mathbb{Q}^n$, for every u_j , there exist a loop v_j in \mathcal{A} such that $\langle \varphi(v_j), u_j \rangle < 0$. Since $\varphi(v_j), u_j \in \mathbb{Z}^n$, we even have $\langle \varphi(v_j), u_j \rangle \leq -1$. Since each v_j belongs to $\text{Loop}(\mathcal{A})$, the words v_j^{k+1} also belong to $\text{Loop}(\mathcal{A})$.

Since \mathcal{A} is a modular envelope, there exists a word $w' \in L$ that has all the words $v_1^{k+1}, \dots, v_m^{k+1}$ as infixes and where $\Psi(w) \equiv \Psi(w') \bmod k$. Recall that every infix u of every word in $D_{u_i,k}$ has $\langle \varphi(u), u_i \rangle \geq -k$. However, we have the inequality $\langle \varphi(v_j^{k+1}), u_j \rangle \leq -(k+1)$. Thus, w' cannot belong to $D_{u_i,k}$ for $i \in [1, m]$. Since $L \subseteq M_k \cup D_{u_1,k} \cup \dots \cup D_{u_m,k}$, this only leaves $w' \in M_k$. Since $\Psi(w') \equiv \Psi(w) \bmod k$, we also have $\varphi(w') \equiv \varphi(w) \bmod k$ and thus $w \in M_k$. \square

We are now prepared to explain the decision procedure for Theorem 8. The algorithm is illustrated in Algorithm 1. If $n = 0$, then $\Sigma_n = \emptyset$ and thus either $L = \emptyset$ or $L = \{\varepsilon\}$, meaning $L \mid Z_0$ if and only if $L \neq \emptyset$. If $n \geq 1$, we perform the KLMST decomposition, which, as explained in Section 6.3, yields languages $L_1 \cup \dots \cup L_p$ and modular envelopes $\mathcal{A}_1, \dots, \mathcal{A}_p$ such that $L = L_1 \cup \dots \cup L_p$. Since then $L \mid Z_n$ if and only if $L_i \mid Z_n$ for each $i \in [1, p]$, we check for the latter. For each $i \in [1, p]$, the dichotomy of Lemma 21 guides a case distinction: If $\text{cone}(\mathcal{A}_i) = \mathbb{Q}^n$, then by Lemma 31, $L_i \mid Z_n$ if and only if $\Pi(L_i) \mid Z_n$, which can be checked via Theorem 1.

If $\text{cone}(\mathcal{A}_i)$ is contained in some half-space $H = \{x \in \mathbb{Q}^n \mid \langle x, u \rangle \geq 0\}$ with $u \in \mathbb{Z}^n \setminus \{0\}$, then Lemma 23 tells

Algorithm 1: Deciding separability of a VASS language L from Z_n

Input: $n \in \mathbb{N}$ and VASS language $L = L(V) \subseteq \Sigma_n^*$
if $n = 0$ **and** $L = \emptyset$ **then return** “yes”
if $n = 0$ **and** $L \neq \emptyset$ **then return** “no”
 Use KLMST decomposition to compute VASS languages L_1, \dots, L_p , together with modular envelopes $\mathcal{A}_1, \dots, \mathcal{A}_p$.
for $i \in [1, p]$ **do**
 if $\text{cone}(\mathcal{A}_i) = \mathbb{Q}^n$ **then**
 Check whether $\Pi(L_i) \mid Z_n$
 if not $\Pi(L_i) \mid Z_n$ **then return** “no”
 end
 if $\text{cone}(\mathcal{A}_i) \subseteq H = \{x \in \mathbb{Q}^n \mid \langle x, u \rangle \geq 0\}$ **for some** $u \in \mathbb{Z}^n \setminus \{0\}$ **then**
 Let $U = \{x \in \mathbb{Q}^n \mid \langle x, u \rangle = 0\}$.
 /* $\dim U = n - 1$ */
 Compute k, ℓ with $L_i \subseteq L(\mathcal{A}_i) \subseteq D_{u,k} \cup S_{U,\ell}$
 /* Now we have $L_i \setminus D_{u,k} \subseteq S_{U,\ell}$ */
 Compute transduction $T_{U,\ell} \subseteq \Sigma_n^* \times \Sigma_{n-1}^*$
 Compute VASS for $\hat{L}_i = T_{U,\ell}(L_i \setminus D_{u,k})$
 /* \hat{L}_i has dimension $n - 1$: $\hat{L}_i \subseteq \Sigma_{n-1}^*$ */
 Check recursively whether $\hat{L}_i \mid Z_{n-1}$
 if not $\hat{L}_i \mid Z_{n-1}$ **then return** “no”
 end
end
return “yes”

us that $L_i \subseteq L(\mathcal{A}_i) \subseteq D_{u,k} \cup S_{U,\ell}$ for some computable $k, \ell \in \mathbb{N}$ and $U = \{x \in \mathbb{Q}^n \mid \langle x, u \rangle = 0\}$. In particular, we have $L_i \mid Z_n$ if and only if $L_i \setminus D_{u,k} \mid Z_n$. Note that $L_i \setminus D_{u,k} = L_i \cap (\Sigma_n^* \setminus D_{u,k})$ is a VASS language and is included in $S_{U,\ell}$. Thus, the walks in L_i always stay close to the hyperplane U , which has dimension $n - 1$. We can therefore use the transduction $T_{U,\ell}$ to transform L_i into a set \hat{L}_i of walks in $(n - 1)$ -dimensional space and decide separability recursively for the result: We have $\hat{L}_i = T_{U,\ell}(L_i \setminus D_{u,k}) \subseteq \Sigma_{n-1}^*$ and Proposition 28 tells us that $L_i \mid Z_n$ if and only if $\hat{L}_i \mid Z_{n-1}$.

6.3 Constructing modular envelopes

Petri nets. We now prove Theorem 30. Since our proof crucially relies on Lambert’s iteration lemma (Lemma 35), we adopt in this section the notation of Lambert and phrase our proof in terms of Petri nets. A Petri net $N = (P, T, \text{PRE}, \text{POST})$ consists of a finite set P of places, a finite set T of transitions and two mappings $\text{PRE}, \text{POST}: T \rightarrow \mathbb{N}^P$. Configurations of Petri net are elements of \mathbb{N}^P , called markings. The effect of a transition $t \in T$ is $\text{POST}(t) - \text{PRE}(t) \in \mathbb{Z}^P$, denoted $\text{eff}(t)$. If for every place $p \in P$ we have $\text{PRE}(t)[p] \leq \mathbf{M}[p]$ for a transition $t \in T$ then t is fireable in \mathbf{M} and the result of firing t in marking \mathbf{M} is $\mathbf{M}' = \mathbf{M} + \text{eff}(t)$, we write $\mathbf{M} \xrightarrow{t} \mathbf{M}'$. We extend notions of fireability and firing naturally to sequences

of transitions, we also write $\mathbf{M} \xrightarrow{w} \mathbf{M}'$ for $w \in T^*$. The effect of $w \in T^*$, $w = t_1 \cdots t_m$, $t_1, \dots, t_m \in T$ is $\text{eff}(w) = \text{eff}(t_1) + \cdots + \text{eff}(t_m)$.

For a Petri net $N = (P, T, \text{PRE}, \text{POST})$ and markings $\mathbf{M}_0, \mathbf{M}_1$, we define the language $L(N, \mathbf{M}_0, \mathbf{M}_1) = \{w \in T^* \mid \mathbf{M}_0 \xrightarrow{w} \mathbf{M}_1\}$. Hence, $L(N, \mathbf{M}_0, \mathbf{M}_1)$ is the set of transition sequences leading from \mathbf{M}_0 to \mathbf{M}_1 . A *labeled Petri net* is a Petri net $N = (P, T, \text{PRE}, \text{POST})$ together with an *initial marking* \mathbf{M}_I , a *final marking* \mathbf{M}_F , and a *labeling*, i.e. a homomorphism $h : T^* \rightarrow \Sigma^*$. The language *recognized* by the labeled Petri net is then defined as $L_h(N, \mathbf{M}_I, \mathbf{M}_F) = h(L(N, \mathbf{M}_I, \mathbf{M}_F))$.

It is folklore (and easy to see) that a language is a VASS language if and only if it is recognized by a labeled Petri net (and the translation is effective). Thus, it suffices to show Theorem 30 for languages of the form $L = h(L(N, \mathbf{M}_I, \mathbf{M}_F))$. Moreover, it is already enough to prove Theorem 30 for languages of the form $L(N, \mathbf{M}_I, \mathbf{M}_F)$: If \mathcal{A} is a modular envelope for L , then applying h to the edges of \mathcal{A} yields a modular envelope for $h(L)$. Thus from now on, we assume $L = L(N, \mathbf{M}_I, \mathbf{M}_F)$ for a fixed Petri net $N = (P, T, \text{PRE}, \text{POST})$.

Basic notions. Let us introduce some notions used in Lambert's proof. We extend the set of configurations \mathbb{N}^d into \mathbb{N}_ω^d , where $\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\}$. We extend the notion of transition firing into \mathbb{N}_ω^d , by defining $\omega - k = \omega = \omega + k$ for every $k \in \mathbb{N}$. For $\mathbf{u}, \mathbf{v} \in \mathbb{N}_\omega^d$ we write $\mathbf{u} \leq_\omega \mathbf{v}$ if $\mathbf{u}[i] = \mathbf{v}[i]$ or $\mathbf{v}[i] = \omega$. Intuitively reaching a configuration with ω at some places means that it is possible to reach configurations with values ω substituted by arbitrarily high values.

A key notion in [23] is that of MGTS, which formulate restrictions on paths in Petri nets. A *marked graph-transition sequence (MGTS)* for our Petri net $N = (P, T, \text{PRE}, \text{POST})$ is a finite sequence $C_0, t_1, C_1 \dots C_{n-1}, t_n, C_n$, where t_i are transitions from T and C_i are precovering graphs, which are defined next. A *precovering graph* is a quadruple $C = (G, \mathbf{m}, \mathbf{m}^{\text{init}}, \mathbf{m}^{\text{fin}})$, where $G = (V, E, h)$ is a finite, strongly connected, directed graph with $V \subseteq \mathbb{N}_\omega^P$ and labeling $h : E \rightarrow T$, and three vectors: a *distinguished* vector $\mathbf{m} \in V$, an *initial* vector $\mathbf{m}^{\text{init}} \in \mathbb{N}_\omega^P$, and a *final* vector $\mathbf{m}^{\text{fin}} \in \mathbb{N}_\omega^P$. A precovering graph has to meet two conditions: First, for every edge $e = (\mathbf{m}_1, \mathbf{m}_2) \in E$, there is an $\mathbf{m}_3 \in \mathbb{N}_\omega^P$ with $\mathbf{m}_1 \xrightarrow{h(e)} \mathbf{m}_3 \leq_\omega \mathbf{m}_2$. Second, we have $\mathbf{m}^{\text{init}}, \mathbf{m}^{\text{fin}} \leq_\omega \mathbf{m}$. Additionally we impose the restriction on MGTS that the initial vector of C_0 equals \mathbf{M}_I and the final vector of C_n equals \mathbf{M}_F .

Languages of MGTS. Each precovering graph can be treated as a finite automaton. For $\mathbf{m}_1, \mathbf{m}_2 \in V$, $L(C, \mathbf{m}_1, \mathbf{m}_2)$ denotes the set of all $w \in T^*$ read on a path from \mathbf{m}_1 to \mathbf{m}_2 . Moreover, let $L(C) = L(C, \mathbf{m}, \mathbf{m})$. MGTS have associated languages as well. Let $\mathcal{N} = C_0, t_1, C_1 \dots C_{n-1}, t_n, C_n$ be an MGTS of a Petri net N , where $C_i = (G_i, \mathbf{m}_i, \mathbf{m}_i^{\text{init}}, \mathbf{m}_i^{\text{fin}})$. Its language $L(\mathcal{N})$ is the set of all words of the form $w = w_0 t_1 w_1 \cdots w_{n-1} t_n w_n \in T^*$ where: $w_i \in L(C_i)$ for each $i \in [0, n]$ and (ii) there exist markings $\mathbf{u}_0, \mathbf{u}'_0, \mathbf{u}_1, \mathbf{u}'_1, \dots, \mathbf{u}_n, \mathbf{u}'_n \in$

\mathbb{N}^P such that $\mathbf{u}_i \leq_\omega \mathbf{m}_i^{\text{init}}$ and $\mathbf{u}'_i \leq_\omega \mathbf{m}_i^{\text{fin}}$ and

$$\mathbf{u}_0 \xrightarrow{w_0} \mathbf{u}'_0 \xrightarrow{t_1} \mathbf{u}_1 \xrightarrow{w_1} \dots \xrightarrow{w_{n-1}} \mathbf{u}'_{n-1} \xrightarrow{t_n} \mathbf{u}_n \xrightarrow{w_n} \mathbf{u}'_n. \quad (1)$$

In this situation, the occurrences of t_1, \dots, t_n shown in Eq. (1) are called the *bridges* and w_0, \dots, w_n are called the *graph parts*. Notice that by (ii) and the restriction that $\mathbf{m}_0^{\text{init}} = \mathbf{M}_I$ and $\mathbf{m}_n^{\text{fin}} = \mathbf{M}_F$, we have $L(\mathcal{N}) \subseteq L(N, \mathbf{M}_I, \mathbf{M}_F)$ for any MGTS \mathcal{N} . Hence roughly speaking, $L(\mathcal{N})$ is the set of runs that contain the transitions t_1, \dots, t_n and additionally markings before and after firing these transitions are prescribed on some places: this is exactly what the restrictions $\mathbf{u}_i \leq_\omega \mathbf{m}_i^{\text{init}}, \mathbf{u}'_i \leq_\omega \mathbf{m}_i^{\text{fin}}$ impose. As an immediate consequence of the definition, we observe that for every MGTS $\mathcal{N} = C_0, t_1, C_1 \dots C_{n-1}, t_n, C_n$ we have

$$L(\mathcal{N}) \subseteq L(C_0) \cdot \{t_1\} \cdot L(C_1) \cdots L(C_{n-1}) \cdot \{t_n\} \cdot L(C_n). \quad (2)$$

Perfect MGTS. Lambert calls MGTS with a particular property *perfect* [23]. Since the precise definition is involved and we do not need all the details, it is enough for us to mention a selection of facts about perfect MGTS (Property 32 and Proposition 33). One of these is the intuitive property that in perfect MGTSes, the value ω on place p in \mathbf{m}_i means that inside of the precovering graph C_i , the token count in place p can be made arbitrarily high. The precise formulation involves the notion of covering sequences, which we define next. Let C be a precovering graph for a Petri net $N = (P, T, \text{PRE}, \text{POST})$ with a distinguished vector $\mathbf{m} \in \mathbb{N}_\omega^P$ and initial vector $\mathbf{m}^{\text{init}} \in \mathbb{N}_\omega^P$. For a marking \mathbf{M}_0 let $L(N, \mathbf{M}_0) = \bigcup_{\mathbf{M} \in \mathbb{N}^P} L(N, \mathbf{M}_0, \mathbf{M})$, i.e. the set of all the transition sequences fireable in \mathbf{M}_0 . A sequence $x \in L(C) \cap L(N, \mathbf{m}^{\text{init}})$ is called a *covering sequence* for C if x is enabled in \mathbf{m}^{init} and for every place $p \in P$ we have either (i) $\mathbf{m}^{\text{init}}[p] = \omega$, or (ii) $\mathbf{m}[p] = \mathbf{m}^{\text{init}}[p] \in \mathbb{N}$ and $\text{eff}(x)[p] = 0$, or (iii) $\mathbf{m}^{\text{init}}[p] < \mathbf{m}[p] = \omega$ and $\text{eff}(x)[p] > 0$. The property of perfect MGTS that we need is the following:

Property 32. *In a perfect MGTS \mathcal{N} , each precovering graph possesses a covering sequence.*

This is part of the definition of perfect MGTS, see [23, page 92]. The second fact about perfect MGTS that we will use is that one can decompose each Petri net into finitely many perfect MGTS. In [23] the following is shown (Theorem 4.2 (page 94) together with the preceding definition).

Proposition 33 ([23]). *Given a Petri net N , one can compute finitely many perfect MGTS $\mathcal{N}_1, \dots, \mathcal{N}_p$ such that $L(N, \mathbf{M}_I, \mathbf{M}_F)$ equals $\bigcup_{i=1}^p L(\mathcal{N}_i)$.*

Building the automata. By Proposition 33, it suffices to construct a modular envelope for each $L(\mathcal{N}_i)$. Hence, we consider a single perfect MGTS $\mathcal{N} = C_0, t_1, C_1, \dots, t_n, C_n$ with distinguished vertices $\mathbf{m}_0, \dots, \mathbf{m}_n$ and construct a modular envelope \mathcal{A} for $L(\mathcal{N})$. We obtain \mathcal{A} by gluing together all the precovering graphs C_i along the transitions t_i . In other

words, \mathcal{A} is the disjoint union of all the graphs C_i and has an edge labeled t_i from \mathbf{m}_{i-1} to \mathbf{m}_i for each $i \in [1, n]$. The initial state of \mathcal{A} is \mathbf{m}_0 and its final state is \mathbf{m}_n .

Ingredient I: Run amalgamation. The first ingredient in for showing that \mathcal{A} is a modular envelope is a method for constructing runs in Petri nets: the amalgamation of runs as introduced by Leroux and Schmitz [29]. It is based on an embedding between Petri net runs introduced by Jančar [20] and Leroux [28]. A triple $(\mathbf{u}, t, \mathbf{v}) \in \mathbb{N}^P \times T \times \mathbb{N}^P$ is a *transition triple* if $\mathbf{v} = \mathbf{u} + \text{eff}(t)$. If there is no danger of confusion, we sometimes call $(\mathbf{u}, t, \mathbf{v})$ a transition. A triple $(\mathbf{u}, w, \mathbf{v})$ with $\mathbf{u}, \mathbf{v} \in \mathbb{N}^P$ and $w \in (\mathbb{N}^P \times T \times \mathbb{N}^P)^*$ is called a *prerun*. Let $\rho = (\mathbf{u}, w, \mathbf{v})$ and $\rho' = (\mathbf{u}', w', \mathbf{v}')$ be preruns with $w = (\mathbf{u}_0, t_1, \mathbf{v}_1)(\mathbf{u}_1, t_2, \mathbf{v}_2) \cdots (\mathbf{u}_{r-1}, t_r, \mathbf{v}_r)$ and $w' = (\mathbf{u}'_0, t'_1, \mathbf{v}'_1)(\mathbf{u}'_1, t'_2, \mathbf{v}'_2) \cdots (\mathbf{u}'_{s-1}, t'_s, \mathbf{v}'_s)$. An *embedding of ρ in ρ'* is a monotone map $\sigma: [1, r] \rightarrow [1, s]$ such that $t'_{\sigma(i)} = t_i$, $\mathbf{u}_i \leq \mathbf{u}'_{\sigma(i)}$ and $\mathbf{v}_i \leq \mathbf{v}'_{\sigma(i)}$ for $i \in [1, r]$, and $\mathbf{u} \leq \mathbf{u}'$ and $\mathbf{v} \leq \mathbf{v}'$. In this case, we call the words $t'_1 \cdots t'_{\sigma(1)-1}$, $t'_{\sigma(i)+1} \cdots t'_{\sigma(i+1)-1}$ for $i \in [1, r-1]$, and $t'_{\sigma(r)+1} \cdots t'_s$ the *inserted words of σ* . By $F(\sigma) \subseteq T^*$, we denote the set of all infixes of inserted words of σ . Furthermore, by $\Psi(\rho)$, we denote the Parikh image $\Psi(t_1 \cdots t_r) \in \mathbb{N}^T$.

Moreover, ρ is called a *run* if each $(\mathbf{u}_i, t_i, \mathbf{v}_i)$ is a transition and also $\mathbf{u} = \mathbf{u}_0$, $\mathbf{u}_i = \mathbf{v}_i$ for $i \in [1, r]$, and $\mathbf{v} = \mathbf{v}_r$. Note that this is equivalent to $\mathbf{u}_i = \mathbf{v}_i$ for $i \in [1, r]$ and $\mathbf{u} = \mathbf{u}_0 \xrightarrow{t_1} \mathbf{u}_1 \cdots \mathbf{u}_{r-1} \xrightarrow{t_r} \mathbf{u}_r$ and we sometimes use the latter notation to denote runs.

Suppose we have three runs ρ_0, ρ_1, ρ_2 and there are embeddings σ_1 of ρ_0 in ρ_1 and σ_2 of ρ_0 in ρ_2 . As observed in [29, Prop. 5.1] one can define a new run ρ_3 in which both ρ_1 and ρ_2 embed. Let ρ_0 be the run $\mathbf{u}_0 \xrightarrow{t_1} \mathbf{u}_1 \xrightarrow{t_2} \cdots \xrightarrow{t_r} \mathbf{u}_r$. Then ρ_1 and ρ_2 can be written as

$$\begin{aligned} \rho_1: \mathbf{u}_0 + \mathbf{v}_0 &\xrightarrow{w_0} \mathbf{u}_0 + \mathbf{v}_1 \xrightarrow{t_1} \mathbf{u}_1 + \mathbf{v}_1 \xrightarrow{w_1} \mathbf{u}_1 + \mathbf{v}_2 \cdots \\ &\cdots \xrightarrow{t_r} \mathbf{u}_r + \mathbf{v}_r \xrightarrow{w_r} \mathbf{u}_r + \mathbf{v}_{r+1} \\ \rho_2: \mathbf{u}_0 + \mathbf{v}'_0 &\xrightarrow{w'_0} \mathbf{u}_0 + \mathbf{v}'_1 \xrightarrow{t_1} \mathbf{u}_1 + \mathbf{v}'_1 \xrightarrow{w'_1} \mathbf{u}_1 + \mathbf{v}'_2 \cdots \\ &\cdots \xrightarrow{t_r} \mathbf{u}_r + \mathbf{v}'_r \xrightarrow{w'_r} \mathbf{u}_r + \mathbf{v}'_{r+1} \end{aligned} \quad (3)$$

for some $\mathbf{v}_i, \mathbf{v}'_i \in \mathbb{N}^P$, $i \in [0, r+1]$. Then the *amalgam* of ρ_1 and ρ_2 (along σ_1 and σ_2) is the run ρ_3 defined as

$$\begin{aligned} \mathbf{u}_0 + \mathbf{v}_0 + \mathbf{v}'_0 &\xrightarrow{w_0} \mathbf{u}_0 + \mathbf{v}_1 + \mathbf{v}'_0 \xrightarrow{w'_0} \mathbf{u}_0 + \mathbf{v}_1 + \mathbf{v}'_1 \xrightarrow{t_1} \\ \mathbf{u}_1 + \mathbf{v}_1 + \mathbf{v}'_1 &\xrightarrow{w_1} \mathbf{u}_1 + \mathbf{v}_2 + \mathbf{v}'_1 \xrightarrow{w'_1} \mathbf{u}_1 + \mathbf{v}_2 + \mathbf{v}'_2 \xrightarrow{t_2} \\ &\vdots \\ \mathbf{u}_{r-1} + \mathbf{v}_{r-1} + \mathbf{v}'_{r-1} &\xrightarrow{w_{r-1}} \mathbf{u}_{r-1} + \mathbf{v}_r + \mathbf{v}'_{r-1} \xrightarrow{w'_r} \mathbf{u}_{r-1} + \mathbf{v}_r + \mathbf{v}'_r \xrightarrow{t_r} \\ \mathbf{u}_r + \mathbf{v}_r + \mathbf{v}'_r &\xrightarrow{w_r} \mathbf{u}_r + \mathbf{v}_{r+1} + \mathbf{v}'_r \xrightarrow{w'_r} \mathbf{u}_r + \mathbf{v}_{r+1} + \mathbf{v}'_{r+1}. \end{aligned} \quad (4)$$

and the embedding τ of ρ_0 in ρ_3 is defined in the obvious way. Note that the run ρ_3 and the embedding τ satisfy

$$\begin{aligned} F(\sigma_1) \cup F(\sigma_2) &\subseteq F(\tau), \\ \Psi(\rho_3) - \Psi(\rho_0) &= (\Psi(\rho_1) - \Psi(\rho_0)) + (\Psi(\rho_2) - \Psi(\rho_0)). \end{aligned} \quad (5)$$

The following lemma is very much in the spirit of Leroux and Schmitz [29], which recasts the KLMST algorithm as the computation of an ideal decomposition. Specifically, their [29, Lemma VII.2] shows that the set of runs of \mathcal{N} is upward directed, meaning that for any two runs ρ_1 and ρ_2 , there exists ρ_3 in \mathcal{N} in which both ρ_1 and ρ_2 embed. We need precise control over the Parikh image of the runs we construct. Therefore, we introduce the notion of compatible embeddings, which guarantees that the amalgam of two runs from \mathcal{N} again belongs to \mathcal{N} .

If ρ and ρ' are runs in \mathcal{N} , then we can associate to each marking \mathbf{u}_i (\mathbf{u}'_i) in ρ (in ρ') a node $\tilde{\mathbf{v}}_i$ ($\tilde{\mathbf{v}}'_i$) in some C_j . We say that σ is *(N)-compatible* if (i) σ maps the k -th bridge transition in ρ to the k -th bridge transition in ρ' for each $k \in [1, n]$ and (ii) for $i \in [1, r]$, we have $\tilde{\mathbf{v}}'_{\sigma(i)} = \tilde{\mathbf{v}}_i$. In other words, σ does (i) preserve bridge transitions and (ii) map each marking in ρ to a marking in ρ' that visits the same node in \mathcal{N} .

Lemma 34. *Let ρ_0, ρ_1, ρ_2 be runs in \mathcal{N} where ρ_0 embeds compatibly in ρ_1 and ρ_2 . Then the amalgam ρ_3 of ρ_1 and ρ_2 is also a run in \mathcal{N} . Moreover, the induced embeddings of ρ_1, ρ_2 in ρ_3 are compatible.*

For Lemma 34, we roughly argue as follows. For runs as in Eq. (3), compatibility means that \mathbf{u}_i and $\mathbf{u}_i + \mathbf{v}_{i+1}$ and $\mathbf{u}_i + \mathbf{v}'_{i+1}$ correspond to the same node in a component of \mathcal{N} . Therefore, the difference vectors \mathbf{v}_{i+1} and \mathbf{v}'_{i+1} can be non-zero only in coordinates that have ω in these nodes. This means, the vector $\mathbf{u}_i + \mathbf{v}_{i+1} + \mathbf{v}'_{i+1}$ also differs from \mathbf{u}_i in only those coordinates and can again be associated to the same node to show that ρ_3 is a run in \mathcal{N} .

Proof. Let ρ_0 be the run $\mathbf{u}_0 \xrightarrow{t_1} \mathbf{u}_1 \xrightarrow{t_2} \cdots \xrightarrow{t_r} \mathbf{u}_r$ and let ρ_1 and ρ_2 be as in Eq. (3). Moreover, let τ be the resulting embedding of ρ_0 in ρ_3 and let σ'_j be the embedding of ρ_j in ρ_3 for $j \in \{1, 2\}$. Let us first argue that ρ_3 is a run in \mathcal{N} . To this end, we argue that the images of bridge transitions under τ satisfy condition (i) of a run. Suppose t_i in ρ_0 is a bridge transition. Then for some final marking \mathbf{m}^{fin} of some graph C_j and some initial marking \mathbf{m}^{init} of C_{j+1} , we have $\mathbf{u}_{i-1} \leq_{\omega} \mathbf{m}^{\text{fin}}$ and $\mathbf{u}_i \leq_{\omega} \mathbf{m}^{\text{init}}$. Note that \mathbf{m}^{fin} and \mathbf{m}^{init} have ω in the same set of places; we denote this set by $\Omega \subseteq P$. Since σ_1 is compatible and ρ_1 is a run, this also implies $\mathbf{u}_{i-1} + \mathbf{v}_i \leq_{\omega} \mathbf{m}^{\text{fin}}$ and $\mathbf{u}_i + \mathbf{v}_i \leq_{\omega} \mathbf{m}^{\text{init}}$. Note that this means $\mathbf{v}_i[p] = 0$ if $p \in P \setminus \Omega$, in other words $\mathbf{v}_i \in \mathbb{N}^{\Omega}$. By the same argument, we have $\mathbf{v}'_i \in \mathbb{N}^{\Omega}$. Therefore, we also have $\mathbf{u}_{i-1} + \mathbf{v}_i + \mathbf{v}'_i \leq \mathbf{m}^{\text{fin}}$ and $\mathbf{u}_i + \mathbf{v}_i + \mathbf{v}'_i \leq \mathbf{m}^{\text{init}}$, which proves condition (i).

We now continue with condition (ii). Since σ_1 and σ_2 are compatible, the bridge transitions of ρ_0 must be among the t_1, \dots, t_r . Therefore, each w_i is included in some graph part of ρ_1 ; and each w'_i is included in some graph part of ρ_2 . Let $\tilde{u}_i \in \mathbb{N}_\omega^P$ be the node in the graph C associated with u_i in ρ_0 . Since σ_1 is compatible, the \tilde{u}_i is also the node associated with $u_i + v_i$ and with $u_i + v_{i+1}$. In particular, we have $w_i \in L(C, \tilde{u}_i)$. By the same argument, we have $w'_i \in L(C, \tilde{u}_i)$ and therefore $w_i w'_i \in L(C, \tilde{u}_i)$. In other words, ρ_3 is obtained from ρ_2 by inserting loops w'_i in graphs right after loops at the same node \tilde{u}_i . Thus, ρ_3 is a run in \mathcal{N} .

It remains to be shown that σ'_1, σ'_2 are compatible. By symmetry, it suffices to show this for σ'_1 . Consider the transitions in ρ_1 inside $u_i + v_i \xrightarrow{w_i} u_i + v_{i+1}$. Since the bridge transitions of ρ_1 must be among t_1, \dots, t_r , we know that w_i is included in some graph part of ρ_1 in some graph C . Observe that because of strong connectedness, all nodes in a graph of an MGTS have ω in exactly the same places. Let $\Omega \subseteq P$ be the set of those places. Let $\tilde{u}_i \in \mathbb{N}_\omega^P$ be the node associated to u_i in ρ_0 . Since σ_2 is compatible, the marking $u_i + v'_i$ is associated with the same node \tilde{u}_i . Therefore, we have $u_i \leq_\omega \tilde{u}_i$ and $u_i + v_i \leq_\omega \tilde{u}_i$. This means for every $p \in P$ with $v'_i[p] \neq 0$, we have $\tilde{u}_i[p] = \omega$. In other words, $v'_i \in \mathbb{N}^\Omega$. Note that every node associated to a transition in w_i belong to C and adding a vector from \mathbb{N}^Ω to a marking does not change its associated node. Since σ'_1 maps $u_i + v_i \xrightarrow{w_i} u_i + v_{i+1}$ to $u_i + v_i + v'_i \xrightarrow{w_i} u_i + v_{i+1} + v'_i$, σ'_1 has to preserve the nodes of the transitions in w_i . Similarly, one shows that σ'_1 preserves nodes of markings around non-bridge transitions among t_1, \dots, t_r . \square

Ingredient II: Lambert's iteration lemma. Our second ingredient for showing that \mathcal{A} is a modular envelope is Lambert's iteration lemma. It allows us to construct runs containing desired infixes, which can then be revised using amalgamation. Recall that we consider the marked graph-transition sequence $\mathcal{N} = C_0, t_1, C_1, \dots, C_{n-1}, t_n, C_n$. Let $C_i = (V_i, E_i, h_i)$ be a precovering graph, and let the distinguished vertex be m_i and initial vertex be m_i^{init} . The following is a simplified version of Lambert's iteration lemma (Lemma 4.1 in [23] (page 92)).

Lemma 35 (Lambert [23]). *Suppose $\mathcal{N} = C_0, t_1, C_1, \dots, t_n, C_n$ and let $x_i \in T^*$ be a covering sequence for C_i for $i \in [0, n]$. Then there exist $k_0 \in \mathbb{N}$ and sequences $\beta_i, y_i, z_i \in T^*$ for $i \in [0, n]$ such that for every $k \geq k_0$,*

$$x_0^k \beta_0 y_0^k z_0^k \cdot t_1 \cdot x_1^k \beta_1 y_1^k z_1^k \cdots t_n \cdot x_n^k \beta_n y_n^k z_n^k$$

is a run in \mathcal{N} , such that the shown occurrences of t_1, \dots, t_n are the bridges. Moreover, we have $\sum_{j=0}^i \text{eff}(x_i y_i z_i)[p] \geq 1$ for every $i \in [1, n]$ and $p \in P$ with $m_i^{\text{fin}}[p] = \omega$.

The inequalities $\sum_{j=0}^i \text{eff}(x_i y_i z_i)[p] \geq 1$ for each p with $m_i^{\text{fin}}[p] = \omega$ follow from item (i) in [23, Lemma 4.1]: In the

notation of [23], item (i) of Lemma 4.1 states that the effect of $x_0 y_0 z_0 \cdots x_j y_j z_j$ in p equals $\alpha(x_0(c'_j(p)) - x_0(c_0(p)))$. Since x_0 is a solution to the homogeneous characteristic equation (see page 91 for the definition), we have $x_0(c_0(p)) = 0$. Perfectness (see page 92 for the definition) and the fact that x_0 even has maximal support among all solutions imply that $x_0(c'_j(p)) \geq 1$ for such p . Finally, the proof of [23, Lemma 4.1] chooses $\alpha \in \mathbb{N}$ so as to be above certain thresholds. We may therefore assume that $\alpha \geq 1$.

We use Lemma 35 to construct a run in \mathcal{N} that contains the desired infixes and in which a given run embeds.

Lemma 36. *For every run ρ in \mathcal{N} and words $u_1, \dots, u_m \in \text{Loop}(\mathcal{A})$, there is a run ρ' in \mathcal{N} such that ρ embeds in ρ' via a compatible embedding σ with $u_1, \dots, u_m \in F(\sigma)$.*

Roughly speaking, the proof of Lemma 36 proceeds as follows. First, we take a covering sequence for every component and show that they can be prolonged so that each u_i appears as an infix of some covering sequence. Then, we prolongate the covering sequences further so that they contain the w_i if ρ is as in Eq. (1). Next, we iterate each covering sequence so that it creates enough tokens in places p with $m_i[p] = \omega$, $m_{i-1}^{\text{fin}}[p] \in \mathbb{N}$ so that the part w_i of ρ can embed. To make sure that there are enough tokens also in places p with $m_i^{\text{fin}}[p] = \omega$, we drive up k . Because of $\sum_{j=0}^i \text{eff}(x_i y_i z_i)[p] \geq 1$ for such places, this creates enough tokens to embed the run ρ .

Proof of the modular envelope property. We are finally ready to show that \mathcal{A} is a modular envelope for $L(\mathcal{N})$. Given a run ρ in \mathcal{N} for the transition sequence $w \in L(\mathcal{N})$, we first use Lemma 36 to obtain a run ρ_1 such that ρ_1 contains each u_i , $i \in [1, n]$, as an infix and ρ embeds via a compatible embedding σ_1 into ρ_1 .

For $j \geq 2$, let ρ_j be the run obtained by amalgamating ρ_{j-1} and ρ_1 along σ_{j-1} and σ_1 . Moreover, let σ_j be the resulting embedding. Then Lemma 34 tells us that ρ_j is a run in \mathcal{N} for every $j \geq 2$. Moreover, we have $u_i \in F(\sigma_{j-1}) \subseteq F(\sigma_j)$ for every $i \in [1, m]$ and also $\Psi(\rho_j) - \Psi(\rho) = \Psi(\rho_{j-1}) - \Psi(\rho) + \Psi(\rho_1) - \Psi(\rho)$ (Eq. (5)) and hence by induction $\Psi(\rho_j) = \Psi(\rho) + j \cdot (\Psi(\rho_1) - \Psi(\rho))$. In particular $\Psi(\rho_k) \equiv \Psi(\rho) \pmod k$. Therefore, the transition sequence of ρ_k is a word w' that has each u_i as an infix and satisfies $\Psi(w') \equiv \Psi(w) \pmod k$. This proves that \mathcal{A} is a modular envelope for $L(\mathcal{N})$.

References

- [1] Jean Berstel. 1979. *Transductions and context-free languages*. Teubner Studienbücher, Stuttgart.
- [2] Mikołaj Bojańczyk. 2017. It is Undecidable if Two Regular Tree Languages can be Separated by a Deterministic Tree-walking Automaton. *Fundam. Inform.* 154, 1-4 (2017), 37–46.
- [3] Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. 2012. Affine Parikh automata. *RAIRO - Theor. Inf. and Applic.* 46, 4 (2012), 511–545. <https://doi.org/10.1051/ita/2012013>

- [4] Lorenzo Clemente, Wojciech Czerwiński, Sławomir Lasota, and Charles Paperman. 2017. Regular Separability of Parikh Automata. In *Proceedings of ICALP '17*. 117:1–117:13.
- [5] Lorenzo Clemente, Wojciech Czerwiński, Sławomir Lasota, and Charles Paperman. 2017. Separability of Reachability Sets of Vector Addition Systems. In *Proceedings of STACS '17*. 24:1–24:14.
- [6] Wojciech Czerwiński, Piotr Hofman, and Georg Zetsche. 2018. Unboundedness Problems for Languages of Vector Addition Systems. In *Proceedings of ICALP '18*. 119:1–119:15. <https://doi.org/10.4230/LIPIcs.ICALP.2018.119>
- [7] Wojciech Czerwiński and Sławomir Lasota. 2017. Regular separability of one counter automata. In *Proceedings of LICS '17*. 1–12.
- [8] Wojciech Czerwiński, Sławomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. 2018. The Reachability Problem for Petri Nets is Not Elementary (Extended Abstract). *CoRR* abs/1809.07115 (2018).
- [9] Wojciech Czerwiński, Sławomir Lasota, Roland Meyer, Sebastian Muskalla, K. Narayan Kumar, and Prakash Saivasan. 2018. Regular Separability of Well-Structured Transition Systems. In *Proceedings of CONCUR '18*. 35:1–35:18. <https://doi.org/10.4230/LIPIcs.CONCUR.2018.35>
- [10] Wojciech Czerwiński, Wim Martens, Lrijn van Rooijen, Marc Zeitoun, and Georg Zetsche. 2017. A Characterization for Decidable Separability by Piecewise Testable Languages. *Discrete Mathematics & Theoretical Computer Science* 19, 4 (2017).
- [11] Alain Finkel and Philippe Schnoebelen. 2001. Well-structured transition systems everywhere! *Theoretical Computer Science* 256, 1-2 (2001), 63–92. [https://doi.org/10.1016/S0304-3975\(00\)00102-X](https://doi.org/10.1016/S0304-3975(00)00102-X)
- [12] Gilles Geeraerts, Jean-François Raskin, and Laurent Van Begin. 2007. Well-structured languages. *Acta Informatica* 44, 3-4 (2007), 249–288.
- [13] Jean Goubault-Larrecq and Sylvain Schmitz. 2016. Deciding Piecewise Testable Separability for Regular Tree Languages. In *Proceedings of ICALP '16*. 97:1–97:15.
- [14] Sheila A. Greibach. 1978. Remarks on blind and partially blind one-way multicounter machines. *Theoretical Computer Science* 7, 3 (1978), 311 – 324. [https://doi.org/10.1016/0304-3975\(78\)90020-8](https://doi.org/10.1016/0304-3975(78)90020-8)
- [15] Christoph Haase and Simon Halfon. 2014. Integer Vector Addition Systems with States. In *Proceedings of RP '14*. 112–124.
- [16] Peter Habermehl, Roland Meyer, and Harro Wimmel. 2010. The Downward-Closure of Petri Net Languages. In *Proceedings of ICALP '10*. 466–477.
- [17] John E. Hopcroft and Jean-Jacques Pansiot. 1979. On the Reachability Problem for 5-Dimensional Vector Addition Systems. *Theor. Comput. Sci.* 8 (1979), 135–159.
- [18] Harry B. Hunt III. 1982. On the Decidability of Grammar Problems. *J. ACM* 29, 2 (1982), 429–447.
- [19] Matthias Jantzen. 1979. On the hierarchy of Petri net languages. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications* 13, 1 (1979), 19–30. http://www.numdam.org/item?id=ITA_1979__13_1_19_0
- [20] Petr Jančar. 1990. Decidability of a Temporal Logic Problem for Petri Nets. *Theor. Comput. Sci.* 74, 1 (1990), 71–93.
- [21] Petr Jančar. 2017. Deciding Structural Liveness of Petri Nets. In *Proceedings of SOFSEM '17*. 91–102.
- [22] S. Rao Kosaraju. 1982. Decidability of Reachability in Vector Addition Systems (Preliminary Version). In *Proceedings of STOC '82*. 267–281.
- [23] Jean-Luc Lambert. 1992. A Structure to Decide Reachability in Petri Nets. *Theor. Comput. Sci.* 99, 1 (1992), 79–104.
- [24] Serge Lang. 1966. *Linear Algebra*. Addison-Wesley.
- [25] Ranko Lazic and Sylvain Schmitz. 2015. Nonelementary Complexities for Branching VASS, MELL, and Extensions. *ACM Trans. Comput. Log.* 16, 3 (2015), 20:1–20:30.
- [26] Ranko Lazic and Patrick Totzke. 2017. What Makes Petri Nets Harder to Verify: Stack or Data?. In *Concurrency, Security, and Puzzles — Essays Dedicated to Andrew William Roscoe on the Occasion of His 60th Birthday (LNCS)*, Vol. 10160. Springer, 144–161.
- [27] Jérôme Leroux. 2009. The General Vector Addition System Reachability Problem by Presburger Inductive Invariants. In *Proceedings of LICS '09*. 4–13.
- [28] Jérôme Leroux. 2011. Vector addition system reachability problem: a short self-contained proof. In *Proceedings of POPL '11*. 307–316.
- [29] Jérôme Leroux and Sylvain Schmitz. 2015. Demystifying Reachability in Vector Addition Systems. In *Proceedings of LICS'15*. 56–67.
- [30] Ernst W. Mayr. 1981. An Algorithm for the General Petri Net Reachability Problem. In *Proceedings of STOC'81*. 238–246.
- [31] Thomas Place. 2015. Separating Regular Languages with Two Quantifiers Alternations. In *Proceedings of LICS '15*. 202–213.
- [32] Thomas Place, Lrijn van Rooijen, and Marc Zeitoun. 2013. Separating Regular Languages by Piecewise Testable and Unambiguous Languages. In *Proceedings of MFCS '13*. 729–740.
- [33] Thomas Place and Marc Zeitoun. 2015. Separation and the Successor Relation. In *Proceedings of STACS '15*. 662–675.
- [34] Thomas Place and Marc Zeitoun. 2016. Separating Regular Languages with First-Order Logic. *Logical Methods in Computer Science* 12, 1 (2016).
- [35] Thomas Place and Marc Zeitoun. 2017. Concatenation Hierarchies: New Bottle, Old Wine. In *Proceedings of CSR '17*. 25–37.
- [36] Thomas Place and Marc Zeitoun. 2017. Separation for dot-depth two. In *Proceedings of LICS '17*. 1–12.
- [37] George S. Sacerdote and Richard L. Tenney. 1977. The decidability of the reachability problem for vector addition systems (preliminary version). In *Proceedings of STOC '77*. ACM, 61–76.
- [38] Alexander Schrijver. 1986. *Theory of linear and integer programming*. John Wiley & Sons.
- [39] Thomas G. Szymanski and John H. Williams. 1976. Noncanonical extensions of bottom-up parsing techniques. *SIAM J. Comput.* 5, 2 (1976).
- [40] Georg Zetsche. 2015. An Approach to Computing Downward Closures. In *Proceedings of ICALP '15*. 440–451.
- [41] Georg Zetsche. 2018. Separability by piecewise testable languages and downward closures beyond subwords. In *Proceedings of LICS '18*. 929–938.