The complexity of separability for semilinear sets and Parikh automata

Elias Rojas Collins[a], Chris Köcher[b], Georg Zetzsche[b]

[a] *Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, 02139, MA, USA*
[b] *Max Planck Institute for Software Systems, Paul-Ehrlich-Str. 26, Kaiserslautern, 67663, RP, Germany*

Abstract

In a *separability problem*, we are given two sets K and L from a class $\mathcal{C}$, and we want to decide whether there exists a set S from a class $\mathcal{S}$ such that $K \subseteq S$ and $S \cap L = \emptyset$. In this case, we speak of *separability of sets in* $\mathcal{C}$ *by sets in* $\mathcal{S}$.

We study two types of separability problems. First, we consider separability of semilinear sets (i.e. subsets of $\mathbb{N}^d$ for some d) by sets definable by quantifier-free monadic Presburger formulas (or equivalently, the recognizable subsets of $\mathbb{N}^d$). Here, a formula is monadic if each atom uses at most one variable. Second, we consider separability of languages of Parikh automata by regular languages. A Parikh automaton is a machine with access to counters that can only be incremented, and have to meet a semilinear constraint at the end of the run. Both of these separability problems are known to be decidable with elementary complexity.

Our main results are that both problems are coNP-complete. In the case of semilinear sets, coNP-completeness holds regardless of whether the input sets are specified by existential Presburger formulas, quantifier-free formulas, or semilinear representations. Our results imply that recognizable separability of rational subsets of $\Sigma^* \times \mathbb{N}^d$ (shown decidable by Choffrut and Grigorieff) is coNP-complete as well. Another application is that regularity of deterministic Parikh automata (where the target set is specified using a quantifier-free Presburger formula) is coNP-complete as well.

Keywords: Vector Addition System, Separability, Regular Language

1. Introduction

Separability In a *separability problem*, we are given two sets K and L from a class $\mathcal{C}$, and we want to decide whether there exists a set S from a class $\mathcal{S}$ such that $K \subseteq S$ and $S \cap L = \emptyset$. Here, the sets in $\mathcal{S}$ are the admissible separators, and S is said to *separate* the sets K and L. In the case where $\mathcal{C}$ is a class of non-regular languages and $\mathcal{S}$ is the class of regular languages, then the problem is called *regular separability (problem) for* $\mathcal{C}$. While the problem turned out to be undecidable for context-free languages in the 1970s [1, 2], the last decade saw a significant amount of attention on regular separability for subclasses (or variants) of *vector addition systems with states (VASS)*. Regular separability was studied for coverability languages of VASS (and, more generally, well-structured transition systems) [3–5], one-counter automata and one-dimensional VASS [6], Parikh automata [7], commutative VASS languages [8], concerning its relationship with the intersection problem [9], Büchi VASS [10, 11], and also for settings where one input language is an arbitrary VASS and the other is from some subclass [12]. Recently, this line of work culminated in the breakthrough result that regular separability for general VASS languages is decidable and Ackermann-complete [13]. However, for subclasses of VASS languages, the complexity landscape is far from understood.

Separating Parikh automata An important example of such a subclass is the class of languages accepted by *Parikh automata*, which are non-deterministic automata equipped with counters that can only be incremented. Here, a run is accepting if the final counter values belong to a particular semilinear set. Languages of Parikh automata have received significant attention over many decades [14–25], including a lot of work in recent years [26–31]. This is because they are expressive enough to model non-trivial counting behavior, but still enjoy low complexity for many algorithmic tasks (e.g. the emptiness problem is coNP-complete). Example applications are monadic second-order logic with cardinalities [32] (this paper introduced the specific model of Parikh automata), solving subword constraints [33], and model-checking FIFO channel systems [34]. Moreover, these languages have other equivalent characterizations, such as reversal-bounded counter automata—a classic (and intensely studied) type of infinite-state systems with nice decidability properties [14, 22]—and automata with $\mathbb{Z}$-counters, also called $\mathbb{Z}$-VASS [15, 35][1].

Decidability of regular separability was shown by Clemente, Czerwiński, Lasota, and Paperman [7] in 2017 as one of the first decidability results for regular separability. Moreover, this result was a key ingredient in Keskin and Meyer's algorithm to decide regular separability for general VASS [13]. However, despite the strong interest in Parikh automata and in regular separability, the complexity of this problem remained unknown. In [7, Section 7], the authors provide an elementary complexity upper bound.

Separating semilinear sets: Monadic interpolants One of the steps in the algorithm from [7] is to decide separability of sets defined in Presburger arithmetic, the first-order theory of $(\mathbb{N}; +, \leq, 0, 1)$. Separators of logically defined sets can also be viewed as *interpolants*. If $\varphi(\boldsymbol{x}, \boldsymbol{y})$ and $\psi(\boldsymbol{y}, \boldsymbol{z})$ are (first-order or propositional) formulas such that $\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{z}\, (\varphi(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \psi(\boldsymbol{y}, \boldsymbol{z}))$ holds, then a formula $\chi(\boldsymbol{y})$ is a *Craig interpolant* if $\forall \boldsymbol{x} \forall \boldsymbol{y}\, (\varphi(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \chi(\boldsymbol{y}))$ and

[1]See [16] for efficient translation among Parikh automata, reversal-bounded counter automata, and $\mathbb{Z}$-VASS.

$\forall \boldsymbol{y} \forall \boldsymbol{z}\, (\chi(\boldsymbol{y}) \rightarrow \psi(\boldsymbol{y}, \boldsymbol{z}))$ both hold. Here, $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ are each a vector of variables, meaning χ only mentions variables that occur both in φ and ψ. Equivalently, the set defined by χ is a separator of the sets defined by the existential formulas $\exists \boldsymbol{x} \colon \varphi(\boldsymbol{x}, \boldsymbol{y})$ and $\exists \boldsymbol{z} \colon \neg\psi(\boldsymbol{y}, \boldsymbol{z})$. In Interpolation-Based Model Checking (ITP) [36, 37], Craig interpolants are used to safely overapproximate sets of states: If φ describes reachable states and ψ describes the set of safe states, then χ overapproximates φ without adding unsafe states. Note that in Presburger logic there are implications that do not have a Craig interpolant (this is in contrast to propositional logic). So, before constructing an interpolant, a first step of ITP is to decide whether there even exists such an interpolant.

In the case of Presburger arithmetic, the definable sets are the *semilinear sets*. For many infinite-state systems, the step relation (or even the reachability relation) is semilinear, and thus, separators can play the role of Craig interpolants in infinite-state model checking. For the separators, a natural choice is the class of *recognizable sets*, which are those defined by *monadic* Presburger formulas, meaning each atom refers to at most one variable. Monadic formulas have recently received attention [38–41] because of their applications in query optimization in constraint databases [42, 43] and symbolic automata [38]. Thus, deciding recognizable separability of semilinear sets can be viewed as synthesizing monadic Craig interpolants.

Recognizable separability was shown decidable by Choffrut and Grigorieff [44] (see [8] for an extension beyond semilinear sets). This was a key ingredient for separability of Parikh automata in [7]. Choffrut and Grigorieff's algorithm has elementary complexity [7, Section 7], but the exact complexity of recognizable separability of semilinear sets remained unknown.

Contribution Our *first main result* is that for given existential Presburger formulas, recognizable separability (i.e. monadic separability) is coNP-complete. In particular, recognizable separability is coNP-complete for given semilinear representations. Moreover, our result implies that recognizable separability is coNP-complete for rational subsets of monoids $\Sigma^* \times \mathbb{N}^d$ as considered by Choffrut and Grigorieff [44]. Building on the methods of the first result, our *second main result* is that regular separability for Parikh automata is coNP-complete.

Application I: Monadic decomposability Our first main result strengthens a recent result on monadic decomposability. A formula in Presburger arithmetic is *monadically decomposable* if it has a monadic equivalent. It was shown recently that (i) deciding whether a given quantifier-free formula is *monadically decomposable* (i.e. whether it has a monadic equivalent) is coNP-complete [40, Theorem 1] (see [39, Corollary 8.1] for an alternative proof; and see [45, Proposition 3] for improved bounds for the approach in [40]), whereas (ii) for existential formulas, the problem is coNEXP-complete [41, Corollary 3.6]. Our first main result strengthens (i): If $\varphi(\boldsymbol{x})$ is a quantifier-free formula, then the sets defined by $\varphi(\boldsymbol{x})$ and $\neg\varphi(\boldsymbol{x})$ are separable by a monadic formula if and only if $\varphi(\boldsymbol{x})$ is monadically decomposable. Perhaps surprisingly, our coNP upper bound still holds for existential Presburger formulas, for which monadic decomposability is known to be coNEXP-complete[2].

Application II: Regularity of Parikh automata Another consequence of our results is that regularity of deterministic Parikh automata, i.e. deciding whether a given deterministic Parikh automaton accepts a regular language, is coNP-complete: Given a deterministic Parikh automaton for a language $L \subseteq \Sigma^*$, one can construct in polynomial time a Parikh automaton for $K = \Sigma^* \setminus L$. Then, L is regular if and only if L and K are regularly separable. Here, we assume that the semilinear target set is given as a quantifier-free Presburger formula. Decidability of this problem has been shown by Cadilhac, Finkel, and McKenzie [20, Theorem 25] (even in the more general case of unambiguous constrained automata).

Key ingredients The existing elementary-complexity algorithm for recognizable separability of semilinear sets works with semilinear representations and distinguishes two cases: If in one component j, one of the input sets $S_1, S_2 \subseteq \mathbb{N}^d$ is bounded by some $b \geq 0$, then it considers each $x \in [0, b]$ and recursively decides separability of $S_1[j \mapsto x]$ and $S_2[j \mapsto x]$, where $S_i[j \mapsto x]$ is just S_i restricted to having x in this bounded component. If, however, all components in both sets are unbounded, then it checks feasibility of a system of linear Diophantine equations. This approach leads to repeated intersection of semilinear sets, and thus exponential time. We provide a characterization (Proposition 4.5) that describes inseparability directly as the non-empty intersection of two semilinear sets $\hat{S}_1, \hat{S}_2 \subseteq \mathbb{N}^d$ associated with S_1, S_2. This easily yields an NP procedure for inseparability, even if the input sets are given as existential Presburger formulas.

This characterization is then the first key ingredient for deciding regular separability of Parikh automata in coNP. This is because in [7], it is shown that, after some preprocessing, the languages of Parikh automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are separable if and only if two semilinear sets $C_1, C_2 \subseteq \mathbb{N}^d$ associated with $\mathcal{A}_1$ and $\mathcal{A}_2$ are separable by a recognizable set. These semilinear sets consist of vectors, each of which counts for some run of $\mathcal{A}_i$, how many times each simple cycles occurs in this run. Thus, our first result tells us that it suffices to decide whether $\hat{C}_1$ and $\hat{C}_2$ are disjoint. Unfortunately, the vectors of C_1, C_2 have exponential dimension d, since there are exponentially many simple cycles in each $\mathcal{A}_i$. Thus, applying our first result directly using existential Presburger arithmetic would only yield a coNEXP upper bound.

To avoid this blowup, the second key idea is to *encode the vectors in $\hat{C}_1$ and $\hat{C}_2$ as words*, where the cycle occurrences appear as a concatenation in some order. By constructing $\mathbb{Z}$-VASS $\mathcal{W}_1, \mathcal{W}_2$ for the encodings of the vectors in $\hat{C}_1, \hat{C}_2$, we reduce separability to intersection emptiness of $\mathcal{W}_1$ and $\mathcal{W}_2$. The

[2] This is not a contradiction to the above reduction from monadic decomposability to recognizable separation, since this reduction would require complementing an existential formula.

latter, in turn, easily reduces to non-reachability in a product $\mathbb{Z}$-VASS, which is in coNP.

2. Preliminaries

By $\mathbb{N} = \{0, 1, 2, \ldots\}$ we denote the set of all non-negative integers. Let $d \in \mathbb{N}$ be a number and $I \subseteq [1, d]$ be a set of indices. By $\pi_I : \mathbb{N}^d \to \mathbb{N}^I$ we denote the *projection* of vectors in $\mathbb{N}^d$ to vectors in $\mathbb{N}^I$, i.e., $\pi_I(\boldsymbol{v})[i] = \boldsymbol{v}[i]$ for each $\boldsymbol{v} \in \mathbb{N}^d$ and $i \in I$. The *support* of a vector $\boldsymbol{v} \in \mathbb{N}^d$ is the set of all coordinates in $\boldsymbol{v}$ with non-zero value, i.e. $\mathrm{supp}(\boldsymbol{v}) = \{i \in [1, d] \mid \boldsymbol{v}[i] \neq 0\}$.

Semilinear sets A set $S \subseteq \mathbb{N}^d$ is *linear* if there is a vector $\boldsymbol{u} \in \mathbb{N}^d$ and a finite set $P \subseteq \mathbb{N}^d$ of so-called *periods* such that $S = \boldsymbol{u} + P^*$ holds. Here, for $P = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$, the set P^* is defined as $P^* = \{\lambda_1\boldsymbol{u}_1 + \cdots + \lambda_n\boldsymbol{u}_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$. A subset $S \subseteq \mathbb{N}^d$ is called *semilinear* if it is a finite union of linear sets. In case we specify S by way of a finite union of linear sets, then we call this description a *semilinear representation.* The set $S \subseteq \mathbb{N}^d$ is called *hyperlinear* if there are finite sets $B, P \subseteq \mathbb{N}^d$ such that $S = B + P^*$ holds. It is well known that the semilinear sets are precisely those definable in *Presburger arithmetic* [46], the first-order theory of the structure $(\mathbb{N}; +, \leq, 0, 1, (\equiv_m)_{m \in \mathbb{N} \setminus \{0\}})$. Here $\equiv_m$ is the predicate where $x \equiv_m y$ if and only if $x - y$ is divisible by m. By quantifier elimination, every formula in Presburger arithmetic has a quantifier-free equivalent.

Parikh automata Intuitively, a Parikh automaton has finitely many control states and access to $d \geq 0$ counters. Upon reading a letter (or the empty word), it can add a vector $\boldsymbol{u} \in \mathbb{N}^d$ to its counters. Moreover, for each state $q \in Q$, it specifies a target set $C_q \subseteq \mathbb{N}^d$. An input word is accepted if at the end of the run, the accumulated counter values belong to C_q, where q is the state at the end of the run. Formally, a *Parikh automaton* is a tuple $\mathcal{A} = (Q, \Sigma, T, q_0, (C_q)_{q \in Q})$, where Q is a finite set of states, $T \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \mathbb{N}^d \times Q$ is its finite set of *transitions*, $q_0 \in Q$ is the *initial* state, and $C_q \subseteq \mathbb{N}^d$ is the *target set* in state q, for each $q \in Q$. For an input word $w \in \Sigma^*$, a *run on* w is a sequence $(q_0, w_1, \boldsymbol{u}_1, q_1) \cdots (q_{n-1}, w_n, \boldsymbol{u}_n, q_n)$ of transitions in T with $w = w_1 \cdots w_n$. The run is *accepting* if $\boldsymbol{u}_1 + \cdots + \boldsymbol{u}_n \in C_{q_n}$. The *language* of $\mathcal{A}$ is then the set of all words $w \in \Sigma^*$ such that $\mathcal{A}$ has an accepting run on w.

Remark 2.1. For our results on general Parikh automata, we assume that the target sets are specified using existential Presburger formulas. However, this is not an important aspect: Given a Parikh automaton, one can in polynomial time modify the automaton (and the target set) so that the target set is given, e.g. by a semilinear representation, or a quantifier-free Presburger formula. This is a simple consequence of the fact that one can translate Parikh automata into integer VASS in logarithmic space [16, Corollary 1]. However, this conversion does not preserve determinism, and for deterministic Parikh automata, it can be important how target sets are given (see Corollary 3.7 and the discussion after it). Therefore, for deterministic Parikh automata, we always specify how the targets sets are given.

Separability A subset $L \subseteq M$ of a monoid M is *recognizable* if there is a morphism $\varphi\colon M \to F$ into some finite monoid F such that $\varphi^{-1}(\varphi(L)) = L$. The recognizable subsets of M form a Boolean algebra [47, Chapter III, Prop. 1.1]. We say that sets $K, L \subseteq M$ are *(recognizably) separable*, denoted $K \mid L$, if there is a morphism $\varphi\colon M \to F$ into some finite monoid F such that $\varphi(K) \cap \varphi(L) = \emptyset$. Equivalently, we have $K \mid L$ if and only if there is a recognizable $S \subseteq M$ with $K \subseteq S$ and $S \cap L = \emptyset$. Here, S is called a *separator* of K and L. Clearly, we have $K \mid L$ if and only if $L \mid K$: if S is a separator of K and L then $M \setminus S$ separates L and K.

In the case $M = \Sigma^*$ for some alphabet Σ, the recognizable sets in Σ^* are exactly the regular languages (cf. [48, Theorem II.2.1]), and thus we speak of *regular separability*. In the case $M = \mathbb{N}^d$ for some $d \geq 0$, then the recognizable subsets of $\mathbb{N}^d$ are precisely the finite unions of cartesian products $U_1 \times \cdots \times U_d$, where each $U_i \subseteq \mathbb{N}$ is ultimately periodic [47, Theorem 5.1]. Here, a set $U \subseteq \mathbb{N}$ is *ultimately periodic* if there are $n_0, p \in \mathbb{N} \setminus \{0\}$ such that for all $n \geq n_0$, we have $n \in U$ if and only if $n + p \in U$. This implies that the recognizable subsets of $\mathbb{N}^d$ are precisely those definable by a *monadic Presburger formula*, i.e. one where every atom only refers to one variable [38]. For these reasons, in the case of $M = \mathbb{N}^d$, we also sometimes speak of *monadic separability*.

In a *recognizable separability problem*, we are given two subsets K and L from a monoid M as input, and we want to decide whether K and L are recognizably separable. Again, in the case of $M = \Sigma^*$, we also call this the *regular separability problem*.

3. Main results

Recognizable separability of semilinear sets Our first main result is the following.

Theorem 3.1. *Given two semilinear sets defined by existential Presburger formulas, recognizable separability is* coNP*-complete.*

The lower bound follows with a simple reduction from the emptiness problem for sets defined by existential Presburger formulas: If φ defines a subset $K \subseteq \mathbb{N}^d$, then $K \mid \mathbb{N}^d$ if and only if K is empty. We prove the coNP upper bound in Section 5. By the same argument, recognizable separability is coNP-hard for input sets given by quantifier-free formulas. Thus:

Corollary 3.2. *Given two semilinear sets defined by quantifier-free Presburger formulas, recognizable separability is* coNP*-complete.*

In particular, this re-proves the coNP upper bound for monadic decomposability of quantifier-free formulas, as originally shown by Hague, Lin, Rümmer, and Wu [40, Theorem 1].

Remark 3.3. Our result also implies that for existential Presburger formulas over $(\mathbb{Z};+,\leq,0,1,(\equiv_m)_{m\in\mathbb{N}\setminus\{0\}})$ defining $K,L\subseteq\mathbb{Z}^d$, it is coNP-complete to decide whether they are separable by a monadically defined subset of $\mathbb{Z}^d$. Indeed, consider the injective map $\nu\colon\mathbb{Z}^d\to\mathbb{N}^{2d}$, where $\nu(x_1,\ldots,x_d)=(\sigma(x_1),|x_1|,\ldots,\sigma(x_d),|x_d|)$ with $\sigma(x)=0$ for $x\geq 0$ and $\sigma(x)=1$ for $x<0$. Then $S\subseteq\mathbb{Z}^d$ is monadically definable if and only if $\nu(S)$ is monadically definable[3]. Thus, $K,L\subseteq\mathbb{Z}^d$ are monadically separable if and only if $\nu(K),\nu(L)\subseteq\mathbb{N}^{2d}$ are monadically separable. Finally, one easily constructs existential formulas for $\nu(K),\nu(L)$.

Since for a given semilinear representation of a set $S\subseteq\mathbb{N}^d$, it is easy to construct an existential Presburger formula defining S, Theorem 3.1 also implies the following.

Corollary 3.4. *Given two semilinear representations, recognizable separability is* coNP*-complete.*

In this case, the coNP lower bound comes from the NP-hard membership problem for semilinear sets (even if all numbers are written in unary) [49, Lemma 10]: For a semilinear subset $S\subseteq\mathbb{N}^d$ and a vector $\boldsymbol{u}\in\mathbb{N}^d$, we have $\boldsymbol{u}\notin S$ if and only if $S\mid\{\boldsymbol{u}\}$. Finally, Theorem 3.1 allows us to settle the complexity of recognizable separability of rational subsets of $\Sigma^*\times\mathbb{N}^d$.

Corollary 3.5. *Given $d\in\mathbb{N}$ and two rational subsets of $\Sigma^*\times\mathbb{N}^d$, deciding recognizable separability is* coNP*-complete.*

Decidability was first shown by Choffrut and Grigorieff [44, Theorem 1]. The coNP upper bound follows because Choffrut and Grigorieff [44, Theorem 10] reduce recognizable separability of subsets of $\Sigma^*\times\mathbb{N}^d$ to recognizable separability of rational subsets of $\mathbb{N}^{2d}$ (and their reduction is clearly in polynomial time). Moreover, for a given rational subset of $\mathbb{N}^{2d}$, one can construct in polynomial time an equivalent existential Presburger formula [50, Theorem 1]. Thus, the upper bound follows from Theorem 3.1. Since semilinear sets in $\mathbb{N}^d$ (given by a semilinear representation) can be viewed as rational subsets of $\mathbb{N}^d$ (and hence of $\Sigma^*\times\mathbb{N}^d$), the coNP lower bound is inherited from Corollary 3.4.

Regular separability of Parikh automata Our second main result is the following:

Theorem 3.6. *Regular separability for Parikh automata is* coNP*-complete.*

The coNP lower bound comes via the coNP-complete emptiness problem: For a given Parikh automaton accepting a language $K\subseteq\Sigma^*$, we have $K\mid\Sigma^*$

[3]This is easily shown by translating each atomic formula (over a single variable) into a monadic formula in each direction. However, note that within $\mathbb{Z}^d$, monadic definability is not the same as recognizability. For example, the sets $\{0\}$ and $\mathbb{Z}\setminus\{0\}$ are monadically separable, but not separable by a recognizable subset of $\mathbb{Z}$, since every non-empty recognizable subset of $\mathbb{Z}$ is infinite [47, Chapter III, Example 1.4].

if and only if $K = \emptyset$. Thus, the interesting part is the upper bound, which we prove in Section 6. This is a significant improvement to the previously known elementary (or finitely iterated exponential time) complexity upper bound by Clemente, Czerwiński, Lasota, and Paperman [7].

Theorem 3.6 can also be applied to deciding regularity of deterministic Parikh automata.

Corollary 3.7. *For deterministic Parikh automata with target sets given as quantifier-free Presburger formulas, deciding regularity is* coNP*-complete.*

Decidability of regularity was shown by Cadilhac, Finkel, and McKenzie [20, Theorem 25] (in the slightly more general setting of unambiguous constrained automata). For the coNP upper bound, note that for a language $L \subseteq \Sigma^*$ given by a deterministic Parikh automaton (with quantifier-free formulas for the target sets), one can in polynomial time construct the same type of automaton for the complement $\Sigma^* \setminus L$. Since L is regular if and only if L and $\Sigma^* \setminus L$ are separable by a regular language, we can invoke Theorem 3.6. The coNP lower bound is inherited from monadic decomposability of quantifier-free formulas. Indeed, given a quantifier-free Presburger formula $\varphi(x_1, \ldots, x_n)$ with free variables $(x_1, \ldots, x_n)$, one easily constructs a deterministic Parikh automaton (with quantifier-free target sets) for the language $L_\varphi = \{a_1^{x_1} \cdots a_n^{x_n} \mid \varphi(x_1, \ldots, x_n)\}$. As shown by Ginsburg and Spanier [51, Theorem 1.2], L_φ is regular if and only if φ is monadically decomposable. However, monadic decomposability for quantifier-free formulas is coNP-complete [40, Theorem 1].

For the coNP upper bound in Corollary 3.7, we cannot drop the assumption that the formula be quantifier-free. This is because if the target sets can be existential Presburger formulas, then the regularity problem is coNEXP-hard. This follows by the same reduction from monadic decomposability: If we construct L_φ as above using an existential formula φ, then again, L_φ is regular if and only if φ is monadically decomposable. Moreover, monadic decomposability for existential formulas is coNEXP-complete [41, Corollary 3.6].

4. A characterization of separability in hyperlinear sets

Before we prove our two main results, Theorems 3.1 and 3.6, we should recall the ideas of the existing algorithms [8, 44] for recognizable separability of linear sets. We will use these ideas to obtain a new characterization of separability in hyperlinear sets.

Let $L_1, L_2 \subseteq \mathbb{N}^d$ be two linear sets. The algorithms [8, 44] rely on a procedure that successively eliminates "bounded components": If, say, L_1 is bounded in component j by some $b \in \mathbb{N}$, then one can observe that $L_1 \mid L_2$ if, and only if, $L_1[j \mapsto x] | L_2[j \mapsto x]$ for every $x \in [0, b]$. Here, $L_i[j \mapsto x]$ is L_i restricted to those vectors that have x in the j-th component, and then projected to all components $\neq j$. Therefore, the algorithms of [8, 44] recursively check separability of $L_1[j \mapsto x]$ and $L_2[j \mapsto x]$ for each $x \in [0, b]$. This process invokes several expensive intersection operations on semilinear sets and thus has high complexity. Instead, our approach immediately guesses and verifies the set of components that remain after the elimination process. The corresponding checks involve the notion of twin-unboundedness.

Twin-unbounded components Our notion applies, slightly more generally, to hyperlinear sets. Hence, let $R = A + U^* \subseteq \mathbb{N}^d$ and $S = B + V^* \subseteq \mathbb{N}^d$ be two hyperlinear sets where $A, B, U, V \subseteq \mathbb{N}^d$ are finite sets.

Definition 4.1. A coordinate $j \in [1, d]$ is *twin-unbounded for R and S* if there exist $\boldsymbol{p} \in U^*$ and $\boldsymbol{q} \in V^*$ such that $j \in \text{supp}(\boldsymbol{p}) = \text{supp}(\boldsymbol{q})$.

Hence, intuitively, twin-unbounded coordinates are those that can be made large/driven up in R in the same way as in S. We will present yet another characterization of twin-unbounded coordinates. Let $j \in [1, d]$. We say the j-th coordinate of the hyperlinear set $S = B + V^*$ is *bounded* if there is no period vector in V with support on j, i.e., $j \notin \text{supp}(\boldsymbol{p})$ for all $\boldsymbol{p} \in V$. We say that a subset $J \subseteq [1, d]$ of coordinates is bounded in S if each $j \in J$ is bounded in S.

Consider the following process: Given two hyperlinear sets R and S. We modify R and S by performing each of the following three steps for each coordinate $j \in [1, d]$ until the sets of remaining period vectors in R and S stabilize:

- If neither R nor S is bounded at j, we leave S and R untouched.
- If only R is bounded at j, we remove all period vectors from S which have support on j.
- If only S is bounded at j, we remove all period vectors from R which have support on j.

Then, the coordinates that remain unbounded are precisely the twin-unbounded ones.

Example 4.2. Consider $R = \{(1, 0, 1)\}^*$ and $S = \{(1, 1, 0), (0, 0, 1)\}^*$. Then R is bounded by the value 0 at coordinate 2. So R and S are separable if and only if R and S restricted to the vectors having the value 0 in the second coordinate. So, we only consider this restriction of S—in our algorithm this is reflected by the deletion of the period vector $(1, 1, 0)$ of S. After deletion of the period vector $(1, 1, 0)$, S is bounded at coordinate 1 by the value 0. So, we remove the period vector $(1, 0, 1)$ from R. Finally, the period vector $(0, 0, 1)$ of S gets removed since R is now bounded at coordinate 3. Hence, our algorithm terminates in this case with no twin-unbounded coordinates. This example shows that even if R and S both are unbounded in coordinates 1 and 3, none of these coordinates is twin-unbounded.

If $R = \{(1, 0, 1), (0, 1, 0)\}^*$ and $S = \{(1, 1, 0), (0, 0, 1)\}^*$, then no coordinate is bounded in R and S. Hence, all coordinates are twin-unbounded and no period vector gets removed.

For $J \subseteq [1, d]$, we write $U_J = \{\boldsymbol{p} \in U \mid \text{supp}(\boldsymbol{p}) \subseteq J\}$ and $V_J = \{\boldsymbol{q} \in V \mid \text{supp}(\boldsymbol{q}) \subseteq J\}$.

Separating by modular constraints As observed in [8, 44], if all coordinates of two linear sets L_1, L_2 are unbounded, then separability holds if and only if the two sets can be separated by modulo constraints. This relies on the well known fact that finitely generated abelian groups are *subgroup separable*, i.e. that for every element $\boldsymbol{u} \in \mathbb{Z}^d$ that does not belong to a subgroup $A \subseteq \mathbb{Z}^d$, there exists a homomorphism $\varphi\colon \mathbb{Z}^d \to \mathbb{F}$ into a finite group $\mathbb{F}$ such that (i) A is included in the kernel of φ and (ii) $\varphi(\boldsymbol{u}) \neq 0$. In our characterization (Proposition 4.5) we will use similar arguments and therefore we will recall subgroup separability here.

Lemma 4.3 (Subgroup separability)**.** *If $A \subseteq \mathbb{Z}^d$ is a subgroup and $\boldsymbol{u} \in \mathbb{Z}^d \setminus A$, then there is an $s \in \mathbb{N}$, $s > 0$, and a morphism $\varphi\colon \mathbb{Z}^d \to \mathbb{Z}/s\mathbb{Z}$ with (i) $\varphi(A) = 0$ and (ii) $\varphi(\boldsymbol{u}) \neq 0$.*

Proof. Consider the quotient group $\mathbb{Z}^d/A$. It is finitely generated and abelian and thus isomorphic to a group $\bigoplus_{j=1}^{n} \mathbb{Z}/r_j\mathbb{Z}$ for some numbers $r_1, \ldots, r_n \in \mathbb{N}$. The projection map $\pi\colon \mathbb{Z}^d \to \mathbb{Z}^d/A$ can thus be composed with the isomorphism above to yield a morphism $\psi\colon \mathbb{Z}^d \to \bigoplus_{j=1}^{n} \mathbb{Z}/r_j\mathbb{Z}$ with $\ker\psi = A$. Since $\boldsymbol{u} \notin A$ and thus $\psi(\boldsymbol{u}) \neq 0$, say the j-th component of $\psi(\boldsymbol{u})$ is not zero. We distinguish two cases:

(1) If $r_j > 0$, then we can choose $\varphi\colon \mathbb{Z}^d \to \mathbb{Z}/r_j\mathbb{Z}$ to be ψ followed by the projection to the j-th component.
(2) If $r_j = 0$, then $\mathbb{Z}/r_j\mathbb{Z} = \mathbb{Z}$ and thus the j-th component of $\psi(\boldsymbol{u})$ is an integer $k \in \mathbb{Z}$. We pick some $s > |k|$ and let $\varphi\colon \mathbb{Z}^d \to \mathbb{Z}/s\mathbb{Z}$ yield the j-th component of ψ, modulo s.

These choices clearly satisfy $\varphi(A) = 0$ and $\varphi(\boldsymbol{u}) \neq 0$. □

Separability vs. intersection emptiness We will now characterize inseparability of hyperlinear sets R, S via the intersection of two hyperlinear sets $\hat{R}$ and $\hat{S}$ associated with R, S. The proof will rely on an equivalence relation of vectors. For vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{N}^d$ and $k \in \mathbb{N} \setminus \{0\}$, we write $\boldsymbol{u} \sim_k \boldsymbol{v}$ if for every $i \in [1, d]$, we have

(1) $\boldsymbol{u}[i] = \boldsymbol{v}[i] \leq k$ or
(2) $\boldsymbol{u}[i], \boldsymbol{v}[i] > k$ and $\boldsymbol{u}[i] \equiv \boldsymbol{v}[i] \bmod k$.

The following was shown in [8, Prop. 18].

Lemma 4.4. *For any sets $X, Y \subseteq \mathbb{N}^d$, the following are equivalent:*

(1) *X and Y are not separable by a recognizable set.*
(2) *for each $k \in \mathbb{N} \setminus \{0\}$ there are $\boldsymbol{x}_k \in X$ and $\boldsymbol{y}_k \in Y$ with $\boldsymbol{x}_k \sim_k \boldsymbol{y}_k$.*

Let $k, \ell \in \mathbb{N} \setminus \{0\}$ be such that k divides ℓ. We can observe that $\boldsymbol{u} \sim_\ell \boldsymbol{v}$ implies $\boldsymbol{u} \sim_k \boldsymbol{v}$ in this case. Thus, to show recognizable inseparability of two sets $X, Y \subseteq \mathbb{N}^d$, it suffices to find $\boldsymbol{x}_k \in X$ and $\boldsymbol{y}_k \in Y$ for almost all numbers $k \in \mathbb{N} \setminus \{0\}$. We will use this fact in the proof of the following characterization of inseparability.

Proposition 4.5. *Let $R = A + U^* \subseteq \mathbb{N}^d$ and $S = B + V^* \subseteq \mathbb{N}^d$ be hyperlinear sets. Then R and S are* not *separable by a recognizable set if and only if the intersection*

$$(A + U^* - U_J^*) \cap (B + V^* - V_J^*) \tag{1}$$

is non-empty, where $J \subseteq [1, d]$ is the set of coordinates that are twin-unbounded for R, S.

Proof. Suppose there is a vector $\boldsymbol{x}$ in the intersection (1). Then we can write $\boldsymbol{x} = \boldsymbol{u} - \bar{\boldsymbol{u}}$ and $\boldsymbol{x} = \boldsymbol{v} - \bar{\boldsymbol{v}}$ with $\boldsymbol{u} \in A + U^*$, $\boldsymbol{v} \in B + V^*$, $\bar{\boldsymbol{u}} \in U_J^*$, and $\bar{\boldsymbol{v}} \in V_J^*$. Since J is twin-unbounded for R and S, there are—by definition—$\boldsymbol{p}_j \in U^*$ and $\boldsymbol{q}_j \in V^*$ with $j \in \operatorname{supp}(\boldsymbol{p}_j) = \operatorname{supp}(\boldsymbol{q}_j)$ for each $j \in J$. Then for $\boldsymbol{p} := \sum_{j \in J} \boldsymbol{p}_j$ and $\boldsymbol{q} := \sum_{j \in J} \boldsymbol{q}_j$ we infer $J \subseteq \operatorname{supp}(\boldsymbol{p}) = \operatorname{supp}(\boldsymbol{q})$. Now for each $k \in \mathbb{N} \setminus \{0\}$, consider the vectors

$$\boldsymbol{u}_k = \boldsymbol{u} - \bar{\boldsymbol{u}} + 2k \cdot \boldsymbol{p} + k \cdot \bar{\boldsymbol{u}} \quad \text{and} \quad \boldsymbol{v}_k = \boldsymbol{v} - \bar{\boldsymbol{v}} + 2k \cdot \boldsymbol{q} + k \cdot \bar{\boldsymbol{v}}\,.$$

Then we have $\boldsymbol{u}_k, \boldsymbol{v}_k \in \mathbb{N}^d$ for each $k \in \mathbb{N} \setminus \{0\}$. We claim that $\boldsymbol{u}_k \sim_k \boldsymbol{v}_k$ for all k. Indeed, on coordinates $j \in [1, d] \setminus \operatorname{supp}(\boldsymbol{p})$, the vectors $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ coincide with $\boldsymbol{x}$. Moreover, on coordinates $j \in \operatorname{supp}(\boldsymbol{p})$, both vectors $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ are larger than k and also congruent to $\boldsymbol{x}[j] \bmod k$. Hence, $\boldsymbol{u}_k \sim_k \boldsymbol{v}_k$. Since clearly $\boldsymbol{u}_k = \boldsymbol{u} + 2k \cdot \boldsymbol{p} + (k-1) \cdot \bar{\boldsymbol{u}} \in R$ and $\boldsymbol{v}_k = \boldsymbol{v} + 2k \cdot \boldsymbol{q} + (k-1) \cdot \bar{\boldsymbol{v}} \in S$, Lemma 4.4 implies that R and S are not separable.

Conversely, suppose that R and S are not separable. Then by Lemma 4.4 there are $\boldsymbol{u}_k \in R$ and $\boldsymbol{v}_k \in S$ with $\boldsymbol{u}_k \sim_k \boldsymbol{v}_k$ for every $k \in \mathbb{N} \setminus \{0\}$. We claim that the sequences $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ and $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots$ have subsequences $\boldsymbol{u}'_1, \boldsymbol{u}'_2, \ldots$ and $\boldsymbol{v}'_1, \boldsymbol{v}'_2, \ldots$ such that for every $k \geq 1$, we have (i) $\boldsymbol{u}'_{k+1} \in \boldsymbol{u}'_k + U_J^*$, (ii) $\boldsymbol{v}'_{k+1} \in \boldsymbol{v}'_k + V_J^*$ and (iii) $\boldsymbol{u}'_k \sim_k \boldsymbol{v}'_k$.

The claim is easy to observe: Note that by picking subsequences, we may assume that even $\boldsymbol{u}_k \sim_{k!} \boldsymbol{v}_k$ for every $k \geq 1$. Moreover, the latter property is preserved by taking subsequences. Thus, since A, B are finite, by picking subsequences again, we may assume that there are $\boldsymbol{r} \in A$ and $\boldsymbol{s} \in B$ such that $\boldsymbol{u}_k \in \boldsymbol{r} + U^*$ and $\boldsymbol{v}_k \in \boldsymbol{s} + V^*$ and $\boldsymbol{u}_k \sim_{k!} \boldsymbol{v}_k$ for $k \geq 1$. Then, by Dickson's lemma, we may assume that in addition $\boldsymbol{u}_{k+1} \in \boldsymbol{u}_k + U^*$ and $\boldsymbol{v}_{k+1} \in \boldsymbol{v}_k + V^*$ for every $k \geq 1$ (here, we apply Dickson's lemma to the $|U|$-dimensional vectors of coefficients at period vectors in U and similarly for V). Now since $\boldsymbol{u}_k \sim_{k!} \boldsymbol{v}_k$ for every k, it follows that the sequences $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ and $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots$ are unbounded on the same set $J \subseteq [1, d]$ of coordinates. Then clearly, J is twin-unbounded for R and S. This means, for all but finitely many k, we have $\boldsymbol{u}_{k+1} \in \boldsymbol{u}_k + U_J^*$ and $\boldsymbol{v}_{k+1} \in \boldsymbol{v}_k + V_J^*$. Hence, by picking another subsequence, we may assume that the latter holds for every $k \geq 1$. Then, $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ and $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots$ satisfy the properties (i–iii) above, establishing our claim.

We now claim that $\boldsymbol{u}_1 - \boldsymbol{v}_1$ belongs to the group $\langle U_J \cup V_J \rangle$ generated by $U_J \cup V_J$. Towards a contradiction, suppose $\boldsymbol{u}_1 - \boldsymbol{v}_1$ does not belong to $\langle U_J \cup V_J \rangle$. By Lemma 4.3, there must be an $s \in \mathbb{N}$, $s > 0$, and a morphism $\varphi\colon \mathbb{Z}^d \to \mathbb{Z}/s\mathbb{Z}$

such that $\varphi(\langle U_J \cup V_J \rangle) = 0$ and $\varphi(\boldsymbol{u}_1 - \boldsymbol{v}_1) \neq 0$. However, the vector

$$(\boldsymbol{u}_s - \boldsymbol{v}_s) - (\boldsymbol{u}_1 - \boldsymbol{v}_1) = \underbrace{(\boldsymbol{u}_s - \boldsymbol{u}_1)}_{\in \langle U_J \rangle} - \underbrace{(\boldsymbol{v}_s - \boldsymbol{v}_1)}_{\in \langle V_J \rangle}$$

belongs to $\langle U_J \cup V_J \rangle$, but also agrees with $\boldsymbol{u}_1 - \boldsymbol{v}_1$ under φ (since all components of $\boldsymbol{u}_s - \boldsymbol{v}_s$ are divisible by s), contradicting Lemma 4.3. Hence $\boldsymbol{u}_1 - \boldsymbol{v}_1 \in \langle U_J \cup V_J \rangle$.

This means, we can write $\boldsymbol{u}_1 - \boldsymbol{v}_1 = \boldsymbol{v} - \bar{\boldsymbol{v}} - (\boldsymbol{u} - \bar{\boldsymbol{u}})$ with $\boldsymbol{u}, \bar{\boldsymbol{u}} \in U_J^*$ and $\boldsymbol{v}, \bar{\boldsymbol{v}} \in V_J^*$. But then the vector $\boldsymbol{u}_1 + \boldsymbol{u} - \bar{\boldsymbol{u}} = \boldsymbol{v}_1 + \boldsymbol{v} - \bar{\boldsymbol{v}}$ belongs to the intersection (1). ☐

With Proposition 4.5, we have now characterized inseparability of subsets of $\mathbb{N}^d$ via a particular intersection of two sets in $\mathbb{Z}^d$. It will later be more convenient to work with intersections of sets in $\mathbb{N}^d$, which motivates the following reformulation of Proposition 4.5.

Theorem 4.6. *Let $R = A + U^* \subseteq \mathbb{N}^d$ and $S = B + V^* \subseteq \mathbb{N}^d$ be hyperlinear sets. Then R and S are* not *separable by a recognizable set if and only if the intersection*

$$(A + U^* + V_J^*) \cap (B + V^* + U_J^*) \tag{2}$$

is non-empty, where $J \subseteq [1, d]$ is the set of coordinates that are twin-unbounded for R, S.

Proof. Direct consequence of Proposition 4.5, since clearly $A + U^* - U_J^*$ intersects $B + V^* - V_J^*$ if and only if $A + U^* + V_J^*$ intersects $B + V^* + U_J^*$. ☐

5. Separability of semilinear sets is in coNP

Using the characterization Theorem 4.6, we can now explain our algorithm for the coNP upper bound in Theorem 3.1. We describe an NP algorithm that establishes *inseparability.*

Algorithm Step I: Solution sets to linear Diophantine equations Let us first see that we can reduce the problem to the case where both input sets are given as projections of solution sets of linear Diophantine equations. We may assume that the input formulas are of the form $\exists \boldsymbol{x} \colon \kappa(\boldsymbol{x}, \boldsymbol{y})$, where κ is a formula consisting of conjunction and disjunction (i.e. no negation) of atoms of the form $t \geq a$, where t is a linear combination of variables $\boldsymbol{x} = (x_1, \ldots, x_n), \boldsymbol{y} = (y_1, \ldots, y_m)$ and integer coefficients, and a is a constant.

Let φ be a formula as described above. It is a well known fact that φ can be transformed into disjunctive normal form. This means, φ is equivalent to a formula $\varphi_1 \vee \cdots \vee \varphi_k$, where each φ_i (a so-called *clause*) has the form $\exists \boldsymbol{x} \colon \xi(\boldsymbol{x}, \boldsymbol{y})$ such that ξ is a conjunction of atoms appearing in φ. In general, the number of clauses of φ is exponential.

Now, let φ and ψ be the input formulas of the algorithm and let $\varphi_1 \vee \cdots \vee \varphi_k$ and $\psi_1 \vee \cdots \vee \psi_\ell$ be their equivalent formulas in disjunctive normal form. Since the number of clauses is exponential, we cannot compute all clauses for φ and ψ. However, the solution sets of φ and ψ are recognizably inseparable if, and only if, for some pair i, j, the solution sets of the formulas φ_i and ψ_j are recognizably inseparable. This is due to the following fact, which follows standard ideas.

Lemma 5.1. *Let $K, K_1, \ldots, K_n, L \subseteq M$ be sets from a monoid M such that $K = K_1 \cup \cdots \cup K_n$. Then $K \mid L$ if, and only if, $K_i \mid L$ for all $1 \leq i \leq n$.*

Proof. Assume $K \mid L$. Then there is a recognizable set $S \subseteq M$ separating K and L. Let $1 \leq i \leq n$ be arbitrary. Since $K_i \subseteq K$ holds, the set S is also a separator of K_i and L, i.e., $K_i \mid L$ for all $1 \leq i \leq n$.

Conversely, assume $K_i \mid L$ for all $1 \leq i \leq n$. Then there are recognizable sets $S_i \subseteq M$ separating K_i and L. Set $S := \bigcup_{1 \leq i \leq n} S_i$. Then S is recognizable (according to the closure properties of the class of recognizable sets). We also have

$$K = \bigcup_{1 \leq i \leq n} K_i \subseteq \bigcup_{1 \leq i \leq n} S_i = S$$

and

$$L \cap S = L \cap \left(\bigcup_{1 \leq i \leq n} S_i \right) = \bigcup_{1 \leq i \leq n} (L \cap S_i) = \bigcup_{1 \leq i \leq n} \emptyset = \emptyset \,.$$

In other words, S is a recognizable separator of K and L, i.e., $K \mid L$. ☐

Thus, for deciding the inseparability of the solution sets of φ and ψ in NP it is sufficient to guess (in polynomial time) clauses φ_i and ψ_j and show that inseparability of the solution sets of these two formulas is decidable in NP. Therefore, from now on we can assume that the input formulas are (existentially quantified) conjunctions of atoms of the form $t \geq a$.

In particular, each of the two input sets is a projection of the solution set of a system of linear Diophantine inequalities. By introducing slack variables (which will also be projected away), we can turn *inequalities* into *equations*. Thus, we have as input sets $K, L \subseteq \mathbb{N}^d$ with

$$K = \pi(\{\boldsymbol{x} \in \mathbb{N}^r \mid A\boldsymbol{x} = \boldsymbol{b}\}) \quad \text{and} \quad L = \pi(\{\boldsymbol{x} \in \mathbb{N}^r \mid C\boldsymbol{x} = \boldsymbol{d}\}) \,, \qquad (3)$$

where $\pi \colon \mathbb{Z}^r \to \mathbb{Z}^d$ is the projection to the first d components, and $A, C \in \mathbb{Z}^{s \times r}$ are integer matrices and $\boldsymbol{b}, \boldsymbol{d} \in \mathbb{Z}^s$ are integer vectors. Note that here, assuming that the number r of columns and the number s of rows are the same for K and L means no loss of generality.

Algorithm Step II: Recognizable inseparability as satisfiability In the second step, we will reduce recognizable inseparability of K and L to satisfiability of an existential Presburger formula. To this end, we use the fact that the solution sets to $A\boldsymbol{x} \geq \boldsymbol{b}$ (resp. $C\boldsymbol{x} \geq \boldsymbol{d}$) are hyperlinear sets, which allows us to apply Theorem 4.6.

Proposition 5.2. *K and L are recognizably inseparable if, and only if, there are vectors $\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{N}^r$ with*

(1) *$A\boldsymbol{p} = \boldsymbol{0}$, $C\boldsymbol{q} = \boldsymbol{0}$, and $\operatorname{supp}(\pi(\boldsymbol{p})) = \operatorname{supp}(\pi(\boldsymbol{q}))$,*
(2) *$\operatorname{supp}(\pi(\boldsymbol{u})), \operatorname{supp}(\pi(\boldsymbol{v})) \subseteq \operatorname{supp}(\pi(\boldsymbol{p}))$, $A\boldsymbol{u} = \boldsymbol{0}$, and $C\boldsymbol{v} = \boldsymbol{0}$,*
(3) *$A\boldsymbol{x} = \boldsymbol{b}$, $C\boldsymbol{y} = \boldsymbol{d}$, and $\pi(\boldsymbol{x} + \boldsymbol{v}) = \pi(\boldsymbol{y} + \boldsymbol{u})$.*

Proof. We apply Theorem 4.6. To this end, we use the standard hyperlinear representation for solution sets of systems of linear Diophantine equations. Let $A_0 \subseteq \mathbb{N}^r$ be the set of all (component-wise) minimal solutions to $A\boldsymbol{x} = \boldsymbol{b}$, and let $U \subseteq \mathbb{N}^r$ be the set of all minimal solutions to $A\boldsymbol{x} = \boldsymbol{0}$. Then it is well known that $K = \pi(A_0 + U^*) = \pi(A_0) + \pi(U)^*$. In the same way, we obtain a hyperlinear representation $L = \pi(B_0 + V^*) = \pi(B_0) + \pi(V)^*$. Then, the proposition follows from Theorem 4.6.

Indeed, observe that then $\pi(U)^*$ is exactly the set of $\pi(\boldsymbol{p}) \in \mathbb{N}^d$ with $A\boldsymbol{p} = \boldsymbol{0}$. Likewise, $\pi(V)^*$ is exactly the set of $\pi(\boldsymbol{q}) \in \mathbb{N}^d$ with $C\boldsymbol{q} = \boldsymbol{0}$. Therefore, if $J \subseteq [1, d]$ is the set of twin-unbounded components of K, L, and U_J, V_J are defined as in Theorem 4.6, then $\pi(U_J)^*$ consists of exactly those $\pi(\boldsymbol{u})$ for which (i) there are $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{N}^r$ with $A\boldsymbol{p} = \boldsymbol{0}$ and $C\boldsymbol{q} = \boldsymbol{0}$ with $\operatorname{supp}(\pi(\boldsymbol{u})) \subseteq \operatorname{supp}(\pi(\boldsymbol{p})) = \operatorname{supp}(\pi(\boldsymbol{q})) \subseteq J$, and (ii) $A\boldsymbol{u} = \boldsymbol{0}$. The set $\pi(V_J)^*$ has an analogous description.

Thus, if $\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{N}^r$ exist as in the proposition, then clearly $\pi(\boldsymbol{x} + \boldsymbol{v}) = \pi(\boldsymbol{y} + \boldsymbol{u})$ lies in the intersection $(\pi(A_0) + \pi(U)^* + \pi(V_J)^*) \cap (\pi(B_0) + \pi(V)^* + \pi(U_J)^*)$.

Conversely, an element in the intersection $(\pi(A_0)+\pi(U)^*+\pi(V_J)^*)\cap(\pi(B_0)+\pi(V)^* + \pi(U_J)^*)$ can be written as $\pi(\boldsymbol{x} + \boldsymbol{v}) = \pi(\boldsymbol{y} + \boldsymbol{u})$, such that $A\boldsymbol{x} = \boldsymbol{b}$, $C\boldsymbol{y} = \boldsymbol{d}$, and there are $\boldsymbol{p}_1, \boldsymbol{q}_1 \in \mathbb{N}^r$ witnessing $\boldsymbol{u} \in U_J^*$ and also $\boldsymbol{p}_2, \boldsymbol{q}_2 \in \mathbb{N}^r$ witnessing $\boldsymbol{v} \in V_J^*$. This means, $\operatorname{supp}(\pi(\boldsymbol{u})) \subseteq \operatorname{supp}(\pi(\boldsymbol{p}_1)) = \operatorname{supp}(\pi(\boldsymbol{q}_1))$, $A\boldsymbol{p}_1 = \boldsymbol{0}$, and $C\boldsymbol{q}_1 = \boldsymbol{0}$, but also $\operatorname{supp}(\boldsymbol{v}) \subseteq \operatorname{supp}(\pi(\boldsymbol{p}_2)) = \operatorname{supp}(\pi(\boldsymbol{q}_2))$, $A\boldsymbol{p}_2 = \boldsymbol{0}$, and $C\boldsymbol{q}_2 = \boldsymbol{0}$. But then we can use $\boldsymbol{p} := \boldsymbol{p}_1 + \boldsymbol{p}_2$ and $\boldsymbol{q} := \boldsymbol{q}_1 + \boldsymbol{q}_2$ to satisfy the requirements of the proposition. □

Finally, Proposition 5.2 can be used to complete the proof of our first main result:

Proof of Theorem 3.1. Let φ and ψ be two existential Presburger formulas without negation and using only atoms of the form $t \geq 0$, where t is a linear combination of variables and integer coefficients. We give an NP algorithm deciding inseparability by a recognizable set.

Since the solution sets of φ and ψ are inseparable if, and only if, their disjunctive normal forms have at least one pair of inseparable clauses, we guess such a pair of these clauses φ_i and ψ_j (cf. Lemma 5.1). We can transform φ_i and ψ_j into Diophantine equations $A\boldsymbol{x} = \boldsymbol{b}$ and $C\boldsymbol{x} = \boldsymbol{d}$. Using Proposition 5.2 we obtain in polynomial time an existential Presburger formula that is satisfiable if, and only if, the solution sets of $A\boldsymbol{x} = \boldsymbol{b}$ and $C\boldsymbol{x} = \boldsymbol{d}$ are inseparable if, and only if, φ_i and ψ_j are inseparable. Finally, the result follows from NP-completeness of the existential fragment of Presburger arithmetic. □

6. Regular separability of Parikh automata

We now prove our second main result: the coNP upper bound of regular separability of Parikh automata (Theorem 3.6). For this, it will be technically simpler to work with $\mathbb{Z}$-VASS, which are equivalent to Parikh automata. In [16, Corollary 1], it was shown that the two automata models can be converted (while preserving the language) into each other in logarithmic space. Therefore, showing the coNP upper bound for $\mathbb{Z}$-VASS implies it for Parikh automata.

Integer VASS A (d-dimensional) *integer vector addition system with states* (*$\mathbb{Z}$-VASS*, for short) is a quintuple $\mathcal{V} = (Q, \Sigma, T, \iota, f)$ where Q is a finite set of *states*, Σ is an *alphabet*, $T \subseteq Q \times \Sigma_\varepsilon \times \mathbb{Z}^d \times Q$ is a finite set of *transitions*, and $\iota, f \in Q$ are its *source* and *target state*, respectively. Here, $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$. A $\mathbb{Z}$-VASS $\mathcal{V} = (Q, \Sigma, T, \iota, f)$ is called *deterministic* if $\mathcal{V}$ has no ε-labeled transitions and for each $p \in Q$ and $a \in \Sigma$ there is at most one transition of the form $(p, a, \boldsymbol{v}, q) \in T$ (where $\boldsymbol{v} \in \mathbb{Z}^d$ and $q \in Q$).

A *configuration* of $\mathcal{V}$ is a tuple from $Q \times \mathbb{Z}^d$. For two configurations $(p, \boldsymbol{u}), (q, \boldsymbol{v})$ and a word $w \in \Sigma^*$ we write $(p, \boldsymbol{u}) \xrightarrow{w}_{\mathcal{V}} (q, \boldsymbol{v})$ if there are states $q_0, q_1, \ldots, q_\ell \in Q$, vectors $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_\ell \in \mathbb{Z}^d$, and letters $a_1, \ldots, a_\ell \in \Sigma_\varepsilon$ such that $w = a_1 a_2 \cdots a_\ell$, $(p, \boldsymbol{u}) = (q_0, \boldsymbol{v}_0)$, $(q, \boldsymbol{v}) = (q_\ell, \boldsymbol{v}_\ell)$, and for each $1 \le i \le \ell$ we have a transition $t_i = (q_{i-1}, a_i, \boldsymbol{x}_i, q_i) \in T$ with $\boldsymbol{v}_i = \boldsymbol{v}_{i-1} + \boldsymbol{x}_i$. In this case, the sequence $t_1 t_2 \cdots t_\ell$ is called a (w-labeled) run of $\mathcal{V}$. The *accepted language* of $\mathcal{V}$ is $L(\mathcal{V}) = \{w \in \Sigma^* \mid (\iota, \boldsymbol{0}) \xrightarrow{w}_{\mathcal{V}} (f, \boldsymbol{0})\}$.

Let $I \subseteq [1, d]$ be a set of indices. Then we can generalize the acceptance behavior of the $\mathbb{Z}$-VASS $\mathcal{V}$ as follows:

$$L(\mathcal{V}, I) = \left\{ w \in \Sigma^* \;\middle|\; \exists \boldsymbol{v} \in \mathbb{Z}^d \colon (\iota, \boldsymbol{0}) \xrightarrow{w}_{\mathcal{V}} (f, \boldsymbol{v}) \text{ and } \pi_I(\boldsymbol{v}) = \boldsymbol{0} \right\}.$$

Note that $L(\mathcal{V}, [1, d]) = L(\mathcal{V})$ holds.

An overview of the proof of Theorem 3.6 The remaining part of this section is dedicated to the proof of our second main result, Theorem 3.6. The first few steps (Lemmas 6.1, 6.3, 6.4 and 6.7) are essentially the same as in [7], for which we briefly give an overview: The authors reduce regular separability to recognizable separability of semilinear sets in $\mathbb{N}^d$ (for some dimension d). Concretely, instead of asking for the regular separability in two given $\mathbb{Z}$-VASS we separate quantities of cycles within runs of these $\mathbb{Z}$-VASS. Accordingly, the dimension corresponds to the number of (simple) cycles. Unfortunately, this number is exponential in the size of the input and therefore we cannot just use our first main result (Theorem 3.1) to prove the coNP upper complexity bound. Instead we will construct two $\mathbb{Z}$-VASS (of polynomial dimension) accepting sequences of cycles such that their language intersection corresponds to the intersection (2) from Theorem 4.6 (which is non-empty if, and only if, the $\mathbb{Z}$-VASS from the input are regularly inseparable). Intersection for $\mathbb{Z}$-VASS is known to be in NP implying also the NP upper complexity bound for the regular inseparability problem resp. the coNP upper bound for the separability problem of $\mathbb{Z}$-VASS.

6.1. Reduction to separability of semilinear sets

6.1.1. Determinizing the automata

As announced, we will first follow the reduction from [7]. In the first step, the regular separability problem of nondeterministic $\mathbb{Z}$-VASS can be reduced to the same problem in *deterministic* $\mathbb{Z}$-VASS. This reduction is possible in polynomial time which is a bit surprising at first glance since determinization typically requires at least an exponential blowup. However, in this reduction we determinize the $\mathbb{Z}$-VASS "up to some homomorphic preimage", i.e., from two given $\mathbb{Z}$-VASS $\mathcal{V}_1$ and $\mathcal{V}_2$ one constructs two deterministic $\mathbb{Z}$-VASS $\mathcal{W}_1$ and $\mathcal{W}_2$ with (i) $L(\mathcal{W}_i) = h^{-1}(L(\mathcal{V}_i))$ where $h\colon \Gamma^* \to \Sigma^*$ is a homomorphism and (ii) $L(\mathcal{V}_1) \,|\, L(\mathcal{V}_2)$ if, and only if, $L(\mathcal{W}_1) \,|\, L(\mathcal{W}_2)$ holds. Since our setting is technically slightly different from [7], we include a proof below.

Lemma 6.1 ([7, Lemma 7])**.** *Regular separability for $\mathbb{Z}$-VASS reduces in polynomial time to the regular separability problem for deterministic $\mathbb{Z}$-VASS.*

Before we can prove Lemma 6.1 we first need the following statement.

Claim 6.2. *Let $K, L \subseteq \Sigma^*$ be two languages and $h\colon \Gamma^* \to \Sigma^*$ be an alphabetic morphism[4]. If $K' \subseteq h^{-1}(K)$ with $h(K') = K$, then we have*

$$K \mid L \iff K' \mid h^{-1}(L)\,.$$

Proof. First, assume $K \mid L$. Then there is a regular separator $R \subseteq \Sigma^*$ of K and L, i.e., we have $K \subseteq R$ and $L \cap R = \emptyset$. Set $R' := h^{-1}(R) \subseteq \Gamma^*$. R' is regular since the class of regular languages is closed under inverse morphisms. We also have $K' \subseteq h^{-1}(K) \subseteq h^{-1}(R) = R'$. Additionally, we have $h^{-1}(L) \cap h^{-1}(R) = \emptyset$ since the existence of an element $w \in h^{-1}(L) \cap h^{-1}(R)$ would imply $h(w) \in L \cap R$. This means, R' is a regular separator of K' and $h^{-1}(L)$, i.e., $K' \mid h^{-1}(L)$.

Conversely, assume $K' \mid h^{-1}(L)$. Then there exists a regular separator $R' \subseteq \Gamma^*$ of K' and $h^{-1}(L)$, i.e., we have $K' \subseteq R'$ and $h^{-1}(L) \cap R' = \emptyset$. Set $R := h(R')$ which is a regular language since the class of regular languages is also closed under morphisms. Then we have $K = h(K') \subseteq h(R') = R$. Also $L \cap R = \emptyset$ holds: towards a contradiction suppose there is $w \in L \cap R$. From $w \in R = h(R')$ follows the existence of a word $w' \in R'$ with $h(w') = w$. We also infer $w' \in h^{-1}(L)$ from $w \in L$. Hence, we have $w' \in h^{-1}(L) \cap R' = \emptyset$—a contradiction. All in all, we proved that R is a regular separator of K and L, i.e., $K \mid L$. ☐

Proof of Lemma 6.1. The proof of this lemma is similar to [7, Lemma 7]: let $\mathcal{V}_i = (Q_i, \Sigma, T_i, \iota_i, f_i)$ with $i = 1, 2$ be two $\mathbb{Z}$-VASS. From $\mathcal{V}_1$ and $\mathcal{V}_2$ we will construct two $\mathbb{Z}$-VASS $\mathcal{V}_i' = (Q_i, \Gamma, T_i', \iota_i, f_i)$ such that $\mathcal{V}_1'$ is deterministic and we have

$$L(\mathcal{V}_1) \mid L(\mathcal{V}_2) \iff L(\mathcal{V}_1') \mid L(\mathcal{V}_2')\,.$$

We will obtain the determinism of $\mathcal{V}_1'$ by making each label of a transition in $\mathcal{V}_1$ unique. So, set $\Gamma = T_1$. T_1' is obtained from T_1 by replacing each transition

[4] A morphism $h\colon \Gamma^* \to \Sigma^*$ is *alphabetic* if $|h(a)| \le 1$ holds for each letter $a \in \Gamma$.

$t = (p, a, \boldsymbol{x}, q) \in T_1$ by the new transition $(p, t, \boldsymbol{x}, q)$. Using this translation we also obtain a morphism $h\colon \Gamma^* \to \Sigma^*$ with $h((p, a, \boldsymbol{x}, q)) = a$ for each transition $(p, a, \boldsymbol{x}, q) \in \Gamma = T_1$. Then we obtain $\mathcal{V}_2'$ from $\mathcal{V}_2$ with $L(\mathcal{V}_2') = h^{-1}(L(\mathcal{V}_2))$ by replacing each label $a \in \Sigma_\varepsilon$ of a transition in T_2' with all labels $t \in T_1$ with $h(t) = a$. Additionally, to each state of $\mathcal{V}_2$ we add loops labeled with $t \in T_1$ satisfying $h(t) = \varepsilon$. Formally, this is the following set of transitions:

$$\begin{aligned} T_2' = \quad & \{(p, t, \boldsymbol{x}, q) \mid t \in T_1, (p, h(t), \boldsymbol{x}, q) \in T_2\} \\ & \cup \{(p, t, \mathbf{0}, q) \mid p, q \in Q, t \in T_1, h(t) = \varepsilon\}\,. \end{aligned}$$

Note that this is a well known construction for the application of the inverse of an alphabetic morphism and, hence, we have $L(\mathcal{V}_2') = h^{-1}(L(\mathcal{V}_2))$.

Since each letter from Γ occurs in exactly one transition of $\mathcal{V}_1'$, this $\mathbb{Z}$-VASS is deterministic. Additionally, $\mathcal{V}_1'$ and $\mathcal{V}_2'$ can be constructed from $\mathcal{V}_1$ and $\mathcal{V}_2$ in polynomial time. It is also clear that the morphism h is alphabetical. We can also prove the following properties:

1. $L(\mathcal{V}_1') \subseteq h^{-1}(L(\mathcal{V}_1))$: Let $w \in L(\mathcal{V}_1')$. Then there is an accepting run $t_1' t_2' \cdots t_\ell'$ in $\mathcal{V}_1'$ with $t_i' = (q_{i-1}, t_i, \boldsymbol{x}_i, q_i) \in T_1'$ for each $1 \leq i \leq \ell$. In particular, we have $w = t_1 t_2 \cdots t_\ell \in T_1^*$. By definition of $\mathcal{V}_1'$ we have $t_i = (q_{i-1}, a_i, \boldsymbol{x}_i, q_i) \in T_1$ for an $a_i \in \Sigma_\varepsilon$. But this means that $w = t_1 t_2 \cdots t_\ell$ is an accepting run in $\mathcal{V}_1'$ labeled by $a_1 a_2 \cdots a_\ell$, i.e., $a_1 a_2 \cdots a_\ell \in L(\mathcal{V}_1)$. Moreover, we have $h(w) = h(t_1 t_2 \cdots t_\ell) = a_1 a_2 \cdots a_\ell$ implying $w \in h^{-1}(a_1 a_2 \cdots a_\ell) \subseteq h^{-1}(L(\mathcal{V}_1))$.
2. $h(L(\mathcal{V}_1')) = L(\mathcal{V}_1)$: A word $w \in \Sigma^*$ is in $h(L(\mathcal{V}_1'))$ if, and only if, there is a word $w' \in L(\mathcal{V}_1') \subseteq \Gamma^*$ with $w = h(w')$. This is exactly the case if there is an accepting run $t_1' t_2' \cdots t_\ell'$ in $\mathcal{V}_1'$ that is labeled with w', i.e., we have $t_i' = (q_{i-1}, t_i, \boldsymbol{x}_i, q_i) \in T_1'$ and $w' = t_1 t_2 \cdots t_\ell$. By construction this is equivalent to an accepting run $t_1 t_2 \cdots t_\ell$ in $\mathcal{V}_1$ that is labeled with $h(w') = w$. But this is exactly the definition of $w \in L(\mathcal{V}_1)$.

Now, we can apply Claim 6.2 and obtain

$$L(\mathcal{V}_1) \mid L(\mathcal{V}_2) \iff L(\mathcal{V}_1') \mid L(\mathcal{V}_2')\,.$$

In a final step, we can apply the same polynomial-time procedure to $\mathcal{V}_2'$ and $\mathcal{V}_1'$ to determinize $\mathcal{V}_2'$. The result are two $\mathbb{Z}$-VASS $\mathcal{V}_1''$ and $\mathcal{V}_2''$ with

$$L(\mathcal{V}_1) \mid L(\mathcal{V}_2) \iff L(\mathcal{V}_1') \mid L(\mathcal{V}_2') \iff L(\mathcal{V}_1'') \mid L(\mathcal{V}_2'')\,.$$

While $\mathcal{V}_2''$ is deterministic by construction, it is not clear that the same holds for $\mathcal{V}_1''$. However, due to the fact that $\mathcal{V}_1'$ and $\mathcal{V}_2'$ do not have any ε-transitions, our construction does not introduce any loops in $\mathcal{V}_1''$ compensating ε-transitions in $\mathcal{V}_2'$. Hence, $\mathcal{V}_1''$ is also deterministic. □

6.1.2. Unifying the automata

Next, we reduce regular separability for deterministic $\mathbb{Z}$-VASS to regular separability of two languages accepted by the same deterministic $\mathbb{Z}$-VASS, but with different sets of counters. To this end, given two d-dimensional $\mathbb{Z}$-VASS $\mathcal{V}_1$ and $\mathcal{V}_2$ we construct one $2d$-dimensional $\mathbb{Z}$-VASS $\mathcal{V}$ (using product construction) and two index sets $I_1, I_2 \subseteq [1, 2d]$ such that $L(\mathcal{V}_i) = L(\mathcal{V}, I_i)$.

Lemma 6.3 ([7, Proposition 1])**.** *Regular separability for deterministic $\mathbb{Z}$-VASS reduces in polynomial time to the following problem:*

Given: *A d-dimensional deterministic $\mathbb{Z}$-VASS $\mathcal{V}$ with two sets $I_1, I_2 \subseteq [1, d]$.*

Question: *Are the languages $L(\mathcal{V}, I_1)$ and $L(\mathcal{V}, I_2)$ regularly separable?*

Proof. Let $\mathcal{V}_i = (Q_i, \Sigma, T_i, \iota_i, f_i)$ be two deterministic d-dimensional $\mathbb{Z}$-VASS. We apply the product construction and obtain a new deterministic $2d$-dimensional $\mathbb{Z}$-VASS $\mathcal{V}_1 \times \mathcal{V}_2 = (Q_1 \times Q_2, \Sigma, T, (\iota_1, \iota_2), (f_1, f_2))$ with

$$T = \left\{ ((p_1, p_2), a, (\boldsymbol{v}_1, \boldsymbol{v}_2), (q_1, q_2)) \;\middle|\; \begin{array}{c} (p_i, a, \boldsymbol{v}_i, q_i) \in T_i \\ \text{for all } i = 1, 2 \end{array} \right\}.$$

We show now that $L(\mathcal{V}_1) \mid L(\mathcal{V}_2)$ holds if, and only if,

$$L(\mathcal{V}_1 \times \mathcal{V}_2, [1, d]) \mid L(\mathcal{V}_1 \times \mathcal{V}_2, [d+1, 2d]) \,.$$

Let $\mathcal{A}_i = (Q_i, \Sigma, \Delta_i, \iota_i, \{f_i\})$ with $\Delta_i = \{(p, a, q) \mid \exists \boldsymbol{v} \in \mathbb{Z}^d \colon (p, a, \boldsymbol{v}, q) \in T_i\}$ be the DFA obtained from $\mathcal{V}_i$ (for $i = 1, 2$) by removing all counter updates from the transitions. Then we can observe that $L(\mathcal{V}_1 \times \mathcal{V}_2, [1, d]) = L(\mathcal{V}_1) \cap L(\mathcal{A}_2)$ and $L(\mathcal{V}_1 \times \mathcal{V}_2, [d+1, 2d]) = L(\mathcal{V}_2) \cap L(\mathcal{A}_1)$ holds.

Assume that $L(\mathcal{V}_1) \mid L(\mathcal{V}_2)$ holds. Then there is a regular separator $R \subseteq \Sigma^*$ with $L(\mathcal{V}_1) \subseteq R$ and $L(\mathcal{V}_2) \cap R = \emptyset$. Since $L(\mathcal{V}_1 \times \mathcal{V}_2, [1, d]) = L(\mathcal{V}_1) \cap L(\mathcal{A}_2) \subseteq L(\mathcal{V}_1)$ and, similarly, $L(\mathcal{V}_1 \times \mathcal{V}_2, [d+1, 2d]) \subseteq L(\mathcal{V}_2)$ holds, the regular language R is also a separator of $L(\mathcal{V}_1 \times \mathcal{V}_2, [1, d])$ and $L(\mathcal{V}_1 \times \mathcal{V}_2, [d+1, 2d])$.

Conversely, let $R \subseteq \Sigma^*$ be a regular separator of $L(\mathcal{V}_1 \times \mathcal{V}_2, [1, d])$ and $L(\mathcal{V}_1 \times \mathcal{V}_2, [d+1, 2d])$. Set $R' = (R \cap L(\mathcal{A}_1)) \cup (\Sigma^* \setminus L(\mathcal{A}_2))$. Clearly the language R' is regular. We also have

$$\begin{aligned} L(\mathcal{V}_1) &= (L(\mathcal{V}_1) \cap L(\mathcal{A}_2)) \cup (L(\mathcal{V}_1) \cap \Sigma^* \setminus L(\mathcal{A}_2)) \\ &= (L(\mathcal{V}_1) \cap L(\mathcal{A}_2) \cap L(\mathcal{A}_1)) \cup (L(\mathcal{V}_1) \cap \Sigma^* \setminus L(\mathcal{A}_2)) && (\text{by } L(\mathcal{V}_1) \subseteq L(\mathcal{A}_1)) \\ &\subseteq (R \cap L(\mathcal{A}_1)) \cup (L(\mathcal{V}_1) \cap \Sigma^* \setminus L(\mathcal{A}_2)) && (R \text{ is a separator}) \\ &\subseteq (R \cap L(\mathcal{A}_1)) \cup (\Sigma^* \setminus L(\mathcal{A}_2)) \\ &= R' \,. \end{aligned}$$

Additionally, by $L(\mathcal{V}_2) \subseteq L(\mathcal{A}_2)$ we have $L(\mathcal{V}_2) \cap (\Sigma^* \setminus L(\mathcal{A}_2)) = \emptyset$ and

$$(R \cap L(\mathcal{A}_1)) \cap L(\mathcal{V}_2) = R \cap L(\mathcal{V}_1 \times \mathcal{V}_2, [d+1, 2d]) = \emptyset$$

implying $L(\mathcal{V}_2) \cap R' = \emptyset$. Hence, R' is a regular separator of $L(\mathcal{V}_1)$ and $L(\mathcal{V}_2)$. ◻

Therefore, we now fix a $\mathbb{Z}$-VASS $\mathcal{V} = (Q, \Sigma, T, \iota, f)$.

6.1.3. Skeletons

Now, we want to further simplify the regular separability problem. Concretely, we want to consider only runs in $\mathcal{V}$ that are in some sense similar. We consider some base paths—so called *skeletons*—in $\mathcal{V}$. Two runs in $\mathcal{V}$ are similar if they follow the same base path and only differ in the order and repetition of some cycles. We define the function skel: $T^* \to T^*$ such that $\mathrm{skel}(r) = \rho$ for a path $r \in T^*$ in $\mathcal{V}$ such that ρ is a sub-path of the original path r in which we keep the same set of visited states while removing all cycles that do not increase the set of visited states. Here, ρ is called the *skeleton* of r.

Let $t_1 \cdots t_\ell \in T^*$ be a path in $\mathcal{V}$, i.e., we have $t_i = (q_{i-1}, a_i, \boldsymbol{x}_i, q_i) \in T$ for each $1 \le i \le \ell$. The map skel is defined inductively as follows: $\mathrm{skel}(\varepsilon) = \varepsilon$ and $\mathrm{skel}(t_1) = t_1$. For $1 \le i < \ell$ assume that $\mathrm{skel}(t_1 \cdots t_i) = s_1 \cdots s_j$ is already constructed and that $s_1 \cdots s_j$ is a path ending in q_i. Now we consider the transition t_{i+1}. If there is no transition s_k (with $0 \le k \le j$) with target state q_{i+1}, we set $\mathrm{skel}(t_1 \cdots t_i t_{i+1}) = s_1 \cdots s_j t_{i+1}$. Note that $s_1 \cdots s_j t_{i+1}$ is a path ending in the state q_{i+1}.

Otherwise, let $0 \le k \le j$ be maximal such that s_k ends in q_{i+1}. Then $s_{k+1} \cdots s_j t_{i+1}$ is a cycle in $\mathcal{V}$ (note that s_{k+1} starts with q_{i+1} since $s_1 \cdots s_j$ is a path). If all states occurring in the cycle $s_{k+1} \cdots s_j t_{i+1}$ also occur in the path $s_1 \cdots s_k$, then we set $\mathrm{skel}(t_1 \cdots t_i t_{i+1}) = s_1 \cdots s_k$, i.e., we omit the cycle $s_{k+1} \cdots s_j t_{i+1}$ in the skeleton. Note that the skeleton $s_1 \cdots s_k$ is a path ending in q_{i+1}. Otherwise at least one state in the cycle does not occur in the path $s_1 \cdots s_k$. In this case, we simply add t_{i+1} resulting in $\mathrm{skel}(t_1 \cdots t_i t_{i+1}) = s_1 \cdots s_j t_{i+1}$ where $s_1 \cdots s_j t_{i+1}$ is also a path ending in q_{i+1}. Note that any skeleton of $\mathcal{V}$ has length at most quadratic in the number of transitions $|T|$ as shown in [7, Lemma 10].

Let ρ be a skeleton. A *ρ-cycle* is a cycle that only visits states occurring in ρ; a *ρ-run* is a run $r \in T^*$ with skeleton $\mathrm{skel}(r) = \rho$ (i.e., r is obtained from ρ by inserting ρ-cycles). We write $L(\mathcal{V}, I, \rho)$ for the set of all words in $L(\mathcal{V}, I)$ accepted via ρ-runs.

Lemma 6.4 ([7, Lemma 11]). *We have $L(\mathcal{V}, I_1) \mid L(\mathcal{V}, I_2)$ if, and only if, $L(\mathcal{V}, I_1, \rho) \mid L(\mathcal{V}, I_2, \rho)$ holds for every skeleton ρ.*

Although this was essentially shown in [7, Lemma 11], our setting is strictly speaking slightly different (e.g. we have all short rather than only simple cycles), so we include a detailed proof below.

Proof. First, note that there are only finitely many skeletons: Clemente et al. proved in [7, page 9] that each skeleton has length at most $|Q|^2$. Hence, there are at most $|T|^{|Q|^2}$ many skeletons in $\mathcal{V}$. It is also clear that $L(\mathcal{V}, I) = \bigcup_{\text{skeleton } \rho \text{ of } \mathcal{V}} L(\mathcal{V}, I, \rho)$ holds.

Let ρ be a skeleton of $\mathcal{V}$. There is also a regular language $K_\rho \subseteq \Sigma^*$ such that $L(\mathcal{V}, I, \rho) = L(\mathcal{V}, I) \cap K_\rho$ holds: we can obtain a finite automaton accepting K_ρ from $\mathcal{V}$ and ρ by removing the counters and all edges and states that do not belong to the skeleton ρ.

Finally, we use the following well known fact:

Claim 6.5. *Let $K_1, \ldots, K_n \subseteq \Sigma^*$ be regular languages partitioning Σ^* and $L_1, L_2 \subseteq \Sigma^*$ be two languages. Then we have $L_1 \mid L_2$ if, and only if, $L_1 \cap K_i \mid L_2 \cap K_i$ holds for each $1 \leq i \leq n$.*

Now, if the languages K_i are the regular languages K_ρ for any skeleton ρ and $L_i = L(\mathcal{V}, I_i)$ for $i = 1, 2$ we obtain that $L(\mathcal{V}, I_1) \mid L(\mathcal{V}, I_2)$ holds if, and only if, $L(\mathcal{V}, I_1, \rho) = L(\mathcal{V}, I_1) \cap K_\rho$ is regular separable from $L(\mathcal{V}, I_2) \cap K_\rho = L(\mathcal{V}, I_2, \rho)$. □

Thus, it suffices to show that for a given skeleton ρ, one can decide regular inseparability of $L(\mathcal{V}, I_1, \rho)$ and $L(\mathcal{V}, I_2, \rho)$ in NP. So, from now on, we fix a skeleton ρ and simply write $L(I_i)$ for $L(\mathcal{V}, I_i, \rho)$. Since we only consider runs that visit states that occur in ρ, we may also assume that $\mathcal{V}$ consists only of the states occurring on ρ. In particular, we only say *cycle* instead of "ρ-cycle".

6.1.4. Counting cycles

We now phrase a characterization of regular separability from [7] in our setting. It says that regular separability of the languages $L(I_1)$ and $L(I_2)$ is equivalent to recognizable separability of vectors that count cycles. Here, we only count *short* cycles of length at most $|Q|$. This is possible since each cycle can be decomposed into short cycles. In the following, we fix the set $S \subseteq T^{\leq |Q|}$ of all *short* cycles in $\mathcal{V}$.[5]

For $I \subseteq [1, d]$, we define: if $t = (p, a, \boldsymbol{x}, q) \in T$ is a transition then the *effect* $\Delta_I(t)$ of t to the components in I is $\Delta_I(t) = \pi_I(\boldsymbol{x})$, i.e. the projection of the counter update $\boldsymbol{x}$ to I. If $r = t_1 t_2 \cdots t_\ell \in T^*$ is a path, then the *effect* $\Delta_I(r)$ of r to the components in I is the sum of the effects of all transitions on this path, i.e. $\Delta_I(r) = \sum_{i=1}^{\ell} \Delta_I(t_i)$. Now, let $\boldsymbol{u} \in \mathbb{N}^S$ be a multiset of short cycles. Then the *effect* of $\boldsymbol{u}$ to the components in I is $\Delta_I(\boldsymbol{u}) = \sum_{c \in S} \boldsymbol{u}[c] \cdot \Delta_I(c)$. If $\boldsymbol{v} \in \mathbb{N}^T$ is a multiset of transitions, then the *effect* of $\boldsymbol{v}$ to the components in I is $\Delta_I(\boldsymbol{v}) = \sum_{t \in T} \boldsymbol{v}[t] \cdot \Delta_I(t)$. In case of $I = [1, d]$ we will also write Δ instead of Δ_I. Finally, we define

$$M(I) = \left\{ \boldsymbol{u} \in \mathbb{N}^S \;\middle|\; \Delta_I(\rho) + \Delta_I(\boldsymbol{u}) = \mathbf{0} \right\} .$$

Hence, $M(I)$ is the set of multisets of short cycles such that inserting them into ρ would lead to an accepting run with acceptance condition $I \subseteq [1, d]$. Since $M(I)$ is the solution set of linear Diophantine equations, it is hyperlinear.

Observation 6.6. *Let $I \subseteq [1, d]$. Then $M(I)$ is hyperlinear, i.e., $M(I) = B + V^*$ for two finite sets $B, V \subseteq \mathbb{N}^S$.*

Proof. The equation $\Delta_{I_i}(\rho) + \Delta_{I_i}(\boldsymbol{u}) = \mathbf{0}$ is a system of linear equations (over $\mathbb{N}^S$) and $M(I)$ is the set of solutions of this equation system. Since the equations

[5] Although Lemmas 6.1, 6.3, 6.4 and 6.7 are essentially the same as in [7], we are working with *short cycles*, whereas [7] uses *simple cycles*. This will be crucial later, because short cycles can be guessed on-the-fly in a finite automaton without storing the whole cycle.

are expressible in Presburger arithmetic, we obtain that $M(I)$ is semilinear [46]. Hence, we have $M(I) = \bigcup_{1\leq i\leq k} \boldsymbol{u}_i + V_i^*$ (where $\boldsymbol{u}_i \in \mathbb{N}^S$ and $V_i \subseteq \mathbb{N}^S$ are finite). We can see that the vectors in V_i are solutions of the homogeneous linear equation system $\Delta_{I_i}(\boldsymbol{v}) = \mathbf{0}$ and the vectors $\boldsymbol{u}_j$ satisfy the inhomogeneous system $\Delta_{I_i}(\boldsymbol{u}_j) = -\Delta_{I_i}(\rho)$. Therefore, we have $\boldsymbol{u}_i + \boldsymbol{v} \in M(I)$ for each $1 \leq i \leq k$ and $\boldsymbol{v} \in \bigcup_{1\leq j\leq k} V_j^*$. According to this we can write the solution set $M(I)$ also as $B + V^*$ where $B = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k\}$ and $V = \bigcup_{1\leq i\leq k} V_i$. In other words, the set $M(I)$ is even hyperlinear. □

The following equivalence between regular separability of the languages $L(I_i)$ and recognizable separability of the (hyperlinear) sets $M(I_i)$ was shown in [7, Lemma 12]. It is straightforward to adapt it to our situation.

Lemma 6.7. *We have $L(I_1) \mid L(I_2)$ if, and only if, $M(I_1) \mid M(I_2)$.*

Proof. Before we prove the equivalence, let us introduce a map $\mathrm{cycles}\colon T^* \to \mathbb{N}^S$ such that for each ρ-run $r \in T^*$ we have $\mathrm{cycles}(r) = \boldsymbol{v} \in \mathbb{N}^S$ if r contains each ρ-cycle $c \in S$ exactly $\boldsymbol{v}[c]$ times.

Now, assume that $L(I_1)|L(I_2)$ holds, i.e., there is a regular separator $R \subseteq \Sigma^*$ with $L(I_1) \subseteq R$ and $R \cap L(I_2) = \emptyset$. We will use Lemma 4.4 to show that $M(I_1)$ and $M(I_2)$ are separable by a recognizable set. To this end, we will give a number $k \in \mathbb{N} \setminus \{0\}$ such that $\boldsymbol{v}_1 \not\sim_k \boldsymbol{v}_2$ holds for each $\boldsymbol{v}_i \in M(I_i)$ implying the separability of $M(I_1)$ and $M(I_2)$.

For two words $w_1, w_2 \in \Sigma^*$ write $w_1 \equiv_R w_2$ if $xw_1y \in R \iff xw_2y \in R$ for all $x, y \in \Sigma^*$ (i.e., $\equiv_R$ is the *syntactic* or *Myhill congruence* of R). Since R is regular, the index of $\equiv_R$ is finite and, hence, there is a number $k \in \mathbb{N} \setminus \{0\}$ such that

$$w^k \equiv_R w^{2k} \quad \text{for each } w \in \Sigma^*. \tag{4}$$

We show now $\boldsymbol{v}_1 \not\sim_k \boldsymbol{v}_2$ for each $\boldsymbol{v}_i \in M(I_i)$. Towards a contradiction, assume there are $\boldsymbol{v}_i \in M(I_i)$ (for $i = 1, 2$) with $\boldsymbol{v}_1 \sim_k \boldsymbol{v}_2$. We construct runs $r_i \in T^*$ such that $\mathrm{skel}(r_i) = \rho$ and $\mathrm{cycles}(r_i) = \boldsymbol{v}_i$ hold. For a short ρ-cycle $c \in S$ choose a prefix x_c of ρ such that $\mathrm{skel}(x_c c) = x_c$ (note that for each cycle $c \in S$ such an x_c exists). Let $c_1, \ldots, c_n$ be an enumeration of S such that $|x_{c_1}| \leq |x_{c_2}| \leq \cdots \leq |x_{c_n}|$ holds. In the following we will write x_i instead of x_{c_i}. Let $z_1, \ldots, z_{n+1} \in T^*$ such that $z_1 = x_1$, $x_i z_{i+1} = x_{i+1}$ for each $1 \leq i < n$, and $x_n z_{n+1} = \rho$, i.e., we have $\rho = z_1 z_2 \cdots z_{n+1}$. Set

$$r_i := z_1 c_1^{\boldsymbol{v}_i[c_1]} z_2 c_2^{\boldsymbol{v}_i[c_2]} \cdots z_n c_n^{\boldsymbol{v}_i[c_n]} z_{n+1}\,.$$

Clearly we have $\mathrm{skel}(r_i) = \rho$ and $\mathrm{cycles}(r_i) = \boldsymbol{v}_i$ hold for $i = 1, 2$. We can also show that the labels $w_1, w_2 \in \Sigma^*$ of the paths r_1 resp. r_2 satisfy $w_1 \equiv_R w_2$ using $\boldsymbol{v}_1 \sim_k \boldsymbol{v}_2$ and repeated usage of the equation (4). However, $\boldsymbol{v}_i \in M(I_i)$ implies $w_i \in L(I_i)$. Since $w_1 \in L(I_1) \subseteq R$ we also have $w_2 \in R$ (by $w_1 \equiv_R w_2$). Hence, we have $w_2 \in R \cap L(I_2) = \emptyset$—a contradiction.

Conversely, assume that $M(I_1) \mid M(I_2)$ holds. Hence, there is a recognizable set $X \subseteq \mathbb{N}^S$ such that $M(I_1) \subseteq X$ and $X \cap M(I_2) = \emptyset$. Let $R \subseteq \Sigma^*$ be the set of all labels of ρ-runs $r \in T^*$ such that $\mathrm{skel}(r) = \rho$ with $\mathrm{cycles}(r) \in X$. We show that R is a regular separator of $L(I_1)$ and $L(I_2)$. We have $L(I_1) \subseteq R$: let $w \in L(I_1)$. Then w is the label of a ρ-run $r \in T^*$ with $\text{skel}(r) = \rho$. But then we know $\text{cycles}(r) \in M(I_1) \subseteq X$ implying $w \in R$.

Now, suppose there is a word $w \in L(I_2) \cap R$. Then w is the label of runs $r_1, r_2 \in T^*$ with $\text{skel}(r_i) = \rho$, $\text{cycles}(r_1) \in M(I_2)$ and $\text{cycles}(r_2) \in X$. Since $\mathcal{V}$ is deterministic, we know that $r_1 = r_2$ implying $\text{cycles}(r_1) = \text{cycles}(r_2) \in M(I_2) \cap X = \emptyset$—a contradiction. Hence, we have $L(I_2) \cap R = \emptyset$.

Finally, we have to show that R is regular. To this end, we construct a nondeterministic finite automaton that simulates ρ-runs by storing the image of the map skel and cycles in its state. While the set of all skeletons is finite, the set of vectors appearing in the image of cycles is not necessarily bounded. However, since X is recognizable and, hence, semilinear we can evaluate the condition $\text{cycles}(r) \in X$ for a path $r \in T^*$ using only a finite memory. Concretely we guess a linear set $\boldsymbol{u} + P^* \subseteq X$ where $\boldsymbol{u} \in \mathbb{N}^S$ and $P \subseteq \mathbb{N}^S$ finite (recall that X is a finite union of such linear sets). Additionally, let $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$. The NFA stores in its memory vectors $\boldsymbol{u}', \boldsymbol{p}'_1, \ldots, \boldsymbol{p}'_n$ with $\boldsymbol{u}' \leq \boldsymbol{u}$ and $\boldsymbol{p}'_i \leq \boldsymbol{p}_i$ for all $1 \leq i \leq n$. Whenever the simulation of skel detects a ρ-cycle, we increase one of the vectors $\boldsymbol{u}', \boldsymbol{p}'_1, \ldots, \boldsymbol{p}'_n$. If we reach one of the vectors $\boldsymbol{p}_i$ due to this detection procedure, we reset this vector to $\mathbf{0}$. The NFA accepts if its memory contains the skeleton ρ and the (bounded) counter values $\boldsymbol{u}, \mathbf{0}, \ldots, \mathbf{0}$. Clearly, this NFA accepts the language R. Hence, R is a regular separator of $L(I_1)$ and $L(I_2)$. $\square$

6.2. Reducing inseparability to intersection

At this point, our proof deviates from the approach of [7]. According to Lemma 6.7, it remains to decide whether $M(I_1) | M(I_2)$, where $M(I_1)$ and $M(I_2)$ are sets of vectors of dimension $|S|$, which is exponential. In Theorem 4.6, we saw that recognizable separability of vector sets $A + U^*$ and $B + V^*$ reduces to intersection emptiness of $A + U^* + V_J^*$ and $B + V^* + U_J^*$, where J is a subset of the twin-unbounded components. However, the exponential dimension of $M(I_1), M(I_2)$ means a direct translation into existential Presburger arithmetic would incur an exponential blowup.

Instead, our key observation is that one can reduce inseparability to *intersection emptiness of $\mathbb{Z}$-VASS*: The idea is to encode the intersecting vectors $\boldsymbol{u} \in (A + U^* + V_J^*) \cap (B + V^* + U_J^*)$, where $M(I_1) = A + U^*$, $M(I_2) = B + V^*$, as *words containing the participating cycles*. Thus, we guess a subset J of the twin-unbounded components, and then construct in polynomial time two $\mathbb{Z}$-VASS $\mathcal{W}_1$ and $\mathcal{W}_2$ such that

$$L(\mathcal{W}_1) = \{\#c_1\#c_2 \cdots \#c_m \mid m \in \mathbb{N},\ c_1, \ldots, c_m \in S,\ \Phi(c_1, \ldots, c_m) \in A + U^* + V_J^*\}, \tag{5}$$

$$L(\mathcal{W}_2) = \{\#c_1\#c_2 \cdots \#c_m \mid m \in \mathbb{N},\ c_1, \ldots, c_m \in S,\ \Phi(c_1, \ldots, c_m) \in B + V^* + U_J^*\}, \tag{6}$$

where for cycles $c_1, \ldots, c_m \in S$, the so-called *Parikh vector* $\Phi(c_1, \ldots, c_m) \in \mathbb{N}^S$ counts how many times each short cycle occurs in $c_1, \ldots, c_m$: If $c \in S$, then

$\Phi(c_1,\ldots,c_m)[c]$ is the number of indices $i \in [1,m]$ with $c_i = c$. Note that then clearly, $(A+U^*+V_J^*) \cap (B+V^*+U_J^*) \neq \emptyset$ if and only if $L(\mathcal{W}_1) \cap L(\mathcal{W}_2) \neq \emptyset$.

The main challenge in constructing $\mathcal{W}_1$ and $\mathcal{W}_2$ is to guess a subset J of twin-unbounded components, and for the $\mathbb{Z}$-VASS to verify that a given cycle belongs to J, without being able to store an entire cycle in its state. To solve this, we we will characterize the twin-unbounded cycles in terms of its set of occurring transitions.

6.2.1. Characterizing twin-unbounded cycles

We define for any $\hat{T} \subseteq T$ the set

$$S[\hat{T}] = \left\{ c \in \hat{T}^{\leq |Q|} \;\middle|\; c \text{ is a cycle} \right\}.$$

Thus, $S[\hat{T}] \subseteq S$ is the set of all short cycles that consist solely of transitions from $\hat{T}$.

Our characterization uses an adaptation of the notion of "cancelable productions" in $\mathbb{Z}$-grammars used in [16]. We define the homomorphism $\partial\colon \mathbb{N}^T \to \mathbb{Z}^Q$ as follows: for each transition $t = (p, a, \boldsymbol{x}, q) \in T$ we set $\partial(\boldsymbol{e}_t) = \boldsymbol{e}_q - \boldsymbol{e}_p$, where $\boldsymbol{e}_t \in \mathbb{N}^T$ and $\boldsymbol{e}_p, \boldsymbol{e}_q \in \mathbb{N}^Q$ are unit vectors. Thus, $\partial(\boldsymbol{u})[q]$ is the number of incoming transitions to q, minus the number of outgoing edges from q, weighted by the coefficients in $\boldsymbol{u}$. A *flow* is a vector $\boldsymbol{f} \in \mathbb{N}^T$ with $\partial(\boldsymbol{f}) = \boldsymbol{0}$. The following is a standard fact in graph theory. For a proof that even applies to context-free grammars (rather than automata), see [52, Theorem 3.1].

Lemma 6.8. *A vector $\boldsymbol{f} \in \mathbb{N}^T$ is a flow if and only if it is a sum of (the Parikh vectors of) cycles.*

The following notion will be key in characterizing which cycles are twin-unbounded for $M(I_1)$ and $M(I_2)$. A transition $t \in T$ is *bi-cancelable* if there exist flows $\boldsymbol{f}_1, \boldsymbol{f}_2 \in \mathbb{N}^T$ such that (i) $\Delta_{I_1}(\boldsymbol{f}_1) = \boldsymbol{0}$ and $\Delta_{I_2}(\boldsymbol{f}_2) = \boldsymbol{0}$, (ii) t occurs in both $\boldsymbol{f}_1$ and in $\boldsymbol{f}_2$, and (iii) $\mathrm{supp}(\boldsymbol{f}_1) = \mathrm{supp}(\boldsymbol{f}_2)$. In other words, t is bi-cancelable if it is part of two flows $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ with the same support and with effect zero (wrt. the components I_1 resp. I_2).

Lemma 6.9. *A cycle $c \in S$ is twin-unbounded for $M(I_1)$ and $M(I_2)$ if, and only if, every transition in c is bi-cancelable.*

Proof. For the "only if" direction, suppose that c is twin-unbounded for $M(I_1)$ and $M(I_2)$. Then by definition there exist sums of period vectors $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{N}^S$ of $M(I_1)$ resp. $M(I_2)$ with $c \in \mathrm{supp}(\boldsymbol{u}_1) = \mathrm{supp}(\boldsymbol{u}_2)$. Define $\boldsymbol{f}_i = \tau(\boldsymbol{u}_i) \in \mathbb{N}^T$, where $\tau\colon \mathbb{N}^S \to \mathbb{N}^T$ maps cycles to the number of occurrences of each transition in these cycles. Then clearly $\boldsymbol{f}_i$ are flows with $\Delta_{I_i}(\boldsymbol{f}_i) = \Delta_{I_i}(\boldsymbol{u}_i) = \boldsymbol{0}$, c occurs in both $\boldsymbol{f}_1$ and in $\boldsymbol{f}_2$, and $\mathrm{supp}(\boldsymbol{f}_1) = \mathrm{supp}(\boldsymbol{f}_2)$. Hence, all transitions in c are bi-cancelable.

For the "if" direction, suppose a cycle $c \in S$ only contains bi-cancelable transitions and write $c = t_1 \cdots t_n$ for $t_1, \ldots, t_n \in T$. For each t_i, there are flows $\boldsymbol{f}_{i,1}$ and $\boldsymbol{f}_{i,2}$ witnessing that t_i is bi-cancelable. Notice that $\boldsymbol{f}_1 := \boldsymbol{f}_{1,1} + \cdots + \boldsymbol{f}_{n,1}$

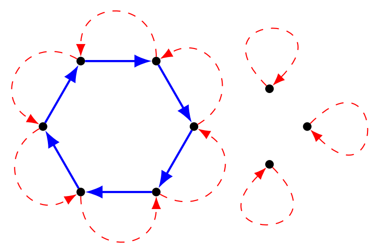

Figure 1: The flow $\tau(\boldsymbol{e}_u) + (\boldsymbol{f}_i - \tau(\boldsymbol{e}_u))$ where the cycle u is depicted in bold blue and the cycles of the flow $\boldsymbol{f}_i - \tau(\boldsymbol{e}_u)$ are depicted in red. Note that the new flower shaped cycle is not necessarily short, but can be easily split into short cycles.

and $\boldsymbol{f}_2 = \boldsymbol{f}_{1,2} + \cdots + \boldsymbol{f}_{n,2}$ are flows as well and they have $\text{supp}(\boldsymbol{f}_1) = \text{supp}(\boldsymbol{f}_2)$. As flows, both $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ can be written as a sum of cycles: There are $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{N}^S$ with $\tau(\boldsymbol{u}_1) = \boldsymbol{f}_1$ and $\tau(\boldsymbol{u}_2) = \boldsymbol{f}_2$. Observe that $\Delta_{I_1}(\boldsymbol{u}_1) = \Delta_{I_2}(\boldsymbol{u}_2) = \boldsymbol{0}$, meaning $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are sums of period vectors of $M(I_1)$ and $M(I_2)$, respectively. If we knew that c occurs in both $\boldsymbol{u}_1$ and in $\boldsymbol{u}_2$, and $\boldsymbol{u}_1, \boldsymbol{u}_2$ had the same support, we could conclude twin-unboundedness of c. Since $\boldsymbol{u}_1, \boldsymbol{u}_2$ may not have these properties, we will now modify them. Consider the set $S' = S[\text{supp}(\boldsymbol{f}_1)] = S[\text{supp}(\boldsymbol{f}_2)]$; hence S' is the set of short cycles $u \in T^*$ such that $\text{supp}(u) \subseteq \text{supp}(\boldsymbol{f}_1) = \text{supp}(\boldsymbol{f}_2)$. By the choice of $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$, we know $c \in S'$. For each cycle $u \in S'$, the vectors $\boldsymbol{f}_1 - \tau(\boldsymbol{e}_u)$ and $\boldsymbol{f}_2 - \tau(\boldsymbol{e}_u)$ are again flows, because $\tau(\boldsymbol{e}_u)$ is a flow. Now observe

$$\sum_{u \in S'} \tau(\boldsymbol{e}_u) + (\boldsymbol{f}_i - \tau(\boldsymbol{e}_u)) = |S'| \cdot \boldsymbol{f}_i$$

for $i = 1, 2$ (cf. Fig. 1). Hence, the flow $|S'| \cdot \boldsymbol{f}_i$ can be written as a sum of cycles in which each cycle from S' occurs. Moreover, in this sum, every occurring cycle belongs to S'. This means, $\boldsymbol{u}'_1, \boldsymbol{u}'_2$ have the same support S', which includes c. Moreover, since $\tau(\boldsymbol{u}'_i) = |S'| \cdot \boldsymbol{f}_i$, we know that $\Delta_{I_i}(\boldsymbol{u}'_i) = \boldsymbol{0}$, meaning $\boldsymbol{u}'_i$ is a sum of period vectors of $M(I_i)$, for $i = 1, 2$. This means, c is indeed twin-unbounded for $M(I_1)$ and $M(I_2)$. ◻

To construct our $\mathbb{Z}$-VASS $\mathcal{W}_1$ and $\mathcal{W}_2$, we first guess a set of transitions and then verify that all of them are bi-cancelable. For the verification, we translate the definition of bi-cancelability into an existential Presburger formula φ_t which is satisfiable if, and only if, t is bi-cancelable.

Lemma 6.10. *Given a transition $t \in T$, we can decide in* NP *whether it is bi-cancelable.*

Proof. We construct an existential Presburger formula φ_t which is satisfiable if, and only if, t is bi-cancelable. Recall that t is bi-cancelable if, and only if, there exist two flows $\boldsymbol{f}_1, \boldsymbol{f}_2 \in \mathbb{N}^T$ such that the properties (i)–(iii) on page 23 hold.

We express in the following these three properties as quantifier-free Presburger formulas using the variables $x_{t'}$ and $y_{t'}$ for each transition.

(i) $\psi_1 = \bigwedge_{i\in[1,d]} \sum_{t'=(p,a,\boldsymbol{v},q)\in T} \boldsymbol{v}[i]\cdot x_{t'} = 0 \wedge \sum_{t'=(p,a,\boldsymbol{v},q)\in T} \boldsymbol{v}[i]\cdot y_{t'} = 0$

(ii) $\psi_{2,t} = x_t > 0 \wedge y_t > 0$

(iii) $\psi_3 = \bigwedge_{t'\in T}(x_{t'} > 0 \longleftrightarrow y_{t'} > 0)$

Additionally, we have to express that $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are flows. This is possible with the following formula:

$$\psi_0 = \bigwedge_{q\in Q} \sum_{t'=(p,a,\boldsymbol{v},q)\in T} x_{t'} = \sum_{t'=(q,a,\boldsymbol{v},p)\in T} x_{t'} \wedge \sum_{t'=(p,a,\boldsymbol{v},q)\in T} y_{t'} = \sum_{t'=(q,a,\boldsymbol{v},p)\in T} y_{t'}\,.$$

Set $\varphi_t = \exists \boldsymbol{x},\boldsymbol{y}\colon \psi_0 \wedge \psi_1 \wedge \psi_{2,t} \wedge \psi_3$ where $\boldsymbol{x} = (x_{t'})_{t'\in T}$ and $\boldsymbol{y} = (y_{t'})_{t'\in T}$ are T-vectors of variables. Clearly, φ_t is satisfiable if, and only if, t is bi-cancelable. □

6.2.2. Constructing the $\mathbb{Z}$-VASS

Lemma 6.11. *There are $\mathbb{Z}$-VASS $\mathcal{W}_1$ and $\mathcal{W}_2$ with $L(\mathcal{W}_1)\cap L(\mathcal{W}_2) = \emptyset$ if and only if $M(I_1)\mid M(I_2)$ holds. $\mathcal{W}_1$ and $\mathcal{W}_2$ can be constructed from $\mathcal{V}$, I_1, and I_2 in nondeterministically polynomial time.*

Let us now describe how the $\mathbb{Z}$-VASS $\mathcal{W}_1$ and $\mathcal{W}_2$ are constructed. Concretely, we build two $\mathbb{Z}$-VASS that satisfy Eqs. (5) and (6). But instead of literally guessing the whole set J of twin-unbounded cycles (which could require exponentially many bits), we guess a set $\hat{T}\subseteq T$ of transitions in $\mathcal{V}$ and then verify in NP that they are all bi-cancelable using Lemma 6.10. This means, we will have

$$L(\mathcal{W}_1) = \{\#c_1\#c_2\cdots\#c_m \mid m\in\mathbb{N}, c_1,\ldots,c_m\in S, \Phi(c_1,\ldots,c_m)\in A+U^*+V^*_{S[\hat{T}]}\} \quad (7)$$

$$L(\mathcal{W}_2) = \{\#c_1\#c_2\cdots\#c_m \mid m\in\mathbb{N}, c_1,\ldots,c_m\in S, \Phi(c_1,\ldots,c_m)\in B+V^*+U^*_{S[\hat{T}]}\} \quad (8)$$

and from now on, we will also write $J = S[\hat{T}]$. Note that the result of our algorithm is correct, even when the guess for $\hat{T}$ is not the *entire* set of bi-cancelable transitions: when $L(\mathcal{W}_1)$ intersects $L(\mathcal{W}_2)$ for some choice of $\hat{T}$, it will do so for any larger choice of $\hat{T}$.

Ensuring membership in $\boldsymbol{A+U^*}$ The idea for constructing $\mathcal{W}_1$ (and analogously $\mathcal{W}_2$) is simple. For each cycle in the input, it guesses whether it belongs to $A+U^*$ or to $V^*_{S[\hat{T}]}$. Let $\boldsymbol{u}_0\in\mathbb{N}^S$ and $\boldsymbol{u}_1\in\mathbb{N}^S$ be the collection of cycles guessed to be in $A+U^*$ and in $V^*_{S[\hat{T}]}$, respectively. To make sure that $\boldsymbol{u}_0\in A+U^*$, we note that $\boldsymbol{u}_0\in A+U^*$ is equivalent to $\Delta_{I_1}(\boldsymbol{u}_0)+\Delta_{I_1}(\rho)=\mathbf{0}$, where ρ is the skeleton guessed earlier in the algorithm. Thus, we can use $|I_1|$ counters to sum up the effect of the cycles $\boldsymbol{u}_0$ and add $\Delta_{I_1}(\rho)$ once in the end. Hence, these counters being zero in the end is equivalent to $\boldsymbol{u}_0\in A+U^*$.

Ensuring membership in $V^*_{S[\hat{T}]}$ To make sure that $\boldsymbol{u}_1 \in V^*_{S[\hat{T}]}$, we note that this is equivalent to $\Delta_{I_2}(\boldsymbol{u}_1) = \mathbf{0}$ and $\mathrm{supp}(\boldsymbol{u}_1) \subseteq S[\hat{T}]$. Thus, our $\mathbb{Z}$-VASS has a separate set of $|I_2|$ counters that carry the total effect of all the cycles in $\boldsymbol{u}_1$. Moreover, it is easy to check that all cycles in $\boldsymbol{u}_1$ only use transitions in $\hat{T}$.

Note that membership in $B + V^*$ and in $U^*_{S[\hat{T}]}$ are checked similarly.

Polynomial time construction Finally, we have to show that the construction of $\mathcal{W}_1$ (and $\mathcal{W}_2$) is possible in polynomial time. To this end, let $\mathcal{V} = (Q, \Sigma, T, \iota, f)$ be the components of $\mathcal{V}$ and let ρ be a skeleton from ι to f visiting all states in Q. We construct a $|I_1| + |I_2|$-dimensional $\mathbb{Z}$-VASS $\mathcal{W}_1 = (Q', \Gamma, T', \iota, f)$ over the input alphabet $\Gamma = T \cup \{\#\}$. The set of states Q' contains (among others) the states $\{\iota, f\}$. We have a transition from ι to f labeled with ε and adding $(\Delta_{I_1}(\rho), \mathbf{0})$ to the counters (note that since the skeleton ρ is fixed for our construction, we can simulate it in one step). For simulating cycles we then guess whether we simulate one in $A + U^*$ or one in $V^*_{S[\hat{T}]}$. For both cases we construct a gadget $\mathcal{G}$ which is the following automaton:

- The states of $\mathcal{G}$ consist of two states from Q and a bounded counter with values in $[1, |Q|]$, i.e., $\{(p, q, j) \mid p, q \in Q, 1 \leq j \leq |Q|\}$ is the set of states in $\mathcal{G}$. Here, the state (p, q, j) has the following meaning: the simulation of the cycle started in state p, we are currently in state q, and we can simulate at most j more steps until finishing the cycle.

- There are transitions from ι to each state $(q, q, |Q|)$ with label $\#$ and counter update $(\mathbf{0}, \mathbf{0})$.

- For each $1 < j \leq |Q|$ we have a transition from (p, q, j) to $(p, q', j-1)$ if $\mathcal{V}$ has a transition $t = (q, a, \boldsymbol{x}, q') \in T$. The label of the new transition is t and the counter update depends on the decision made at the beginning of the simulation: if we are simulating a cycle in $A + U^*$, the counter update is $(\pi_{I_1}(\boldsymbol{x}), \mathbf{0})$. Otherwise it is $(\mathbf{0}, \pi_{I_2}(\boldsymbol{x}))$. In the latter case we also have to ensure that $t \in \hat{T}$ holds.

- We also have transitions from (p, q, j) back to ι if $\mathcal{V}$ has a transition $t = (q, a, \boldsymbol{x}, p) \in T$. The label and the counter update are defined as above.

In other words, the gadget $\mathcal{G}$ is actually the computation graph that is truncated to runs of length $\leq |Q|$. Note that each of the two gadgets has at most $|Q|^3$ many nodes implying that $\mathcal{W}$ has polynomial size (in $|Q|$).

With this polynomial-time construction of $\mathcal{W}_1$ and $\mathcal{W}_2$, we are ready to prove Theorem 3.6:

Proof of Theorem 3.6. We give an NP algorithm for regular inseparability of two $\mathbb{Z}$-VASS (which can be obtained from Parikh automata in logarithmic space [16, Corollary 1]).

Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be two d-dimensional $\mathbb{Z}$-VASS. From $\mathcal{V}_1$ and $\mathcal{V}_2$ we can compute a single $2d$-dimensional deterministic $\mathbb{Z}$-VASS $\mathcal{V}$ and two sets $I_1, I_2 \subseteq [1, 2d]$ in polynomial time such that $L(\mathcal{V}_1) | L(\mathcal{V}_2)$ holds if, and only if, $L(\mathcal{V}, I_1) | L(\mathcal{V}, I_2)$ (Lemmas 6.1 and 6.3). According to Lemma 6.4 we have $L(\mathcal{V}, I_1) \mid L(\mathcal{V}, I_2)$ if, and only if, $L(\mathcal{V}, I_1, \rho) \mid L(\mathcal{V}, I_2, \rho)$ for each skeleton ρ in $\mathcal{V}$ holds. So, we guess a skeleton ρ and check regular inseparability of $L(\mathcal{V}, I_1, \rho)$ and $L(\mathcal{V}, I_2, \rho)$ certifying regular inseparability of $L(\mathcal{V}, I_1)$ and $L(\mathcal{V}, I_2)$.

Additionally, we will guess a set $\hat{T} \subseteq T$ of transitions and verify in NP that all of them are bi-cancelable (Lemma 6.10). Then we can construct in polynomial time two $\mathbb{Z}$-VASS $\mathcal{W}_1$ and $\mathcal{W}_2$ such that (7) and (8) hold (Lemma 6.11). If $L(\mathcal{W}_1) \cap L(\mathcal{W}_2) \neq \emptyset$, the algorithm reports "inseparable". For this, it uses a simple product construction to obtain a $\mathbb{Z}$-VASS $\mathcal{W}$ for the intersection $L(\mathcal{W}_1) \cap L(\mathcal{W}_2)$, and decide in NP whether an accepting configuration can be reached in $\mathcal{W}$.

This is sound: We have $L(\mathcal{W}_1) \cap L(\mathcal{W}_2) \neq \emptyset$ if and only if $(A + U^* + V_J^*) \cap (B + V^* + U_J^*) \neq \emptyset$ for $J = S[\hat{T}]$; and by Lemma 6.7, we know that the latter rules out $M(I_1) \mid M(I_2)$. For completeness, note that if $M(I_1) \mid M(I_2)$ does not hold, then there exists a choice for $\hat{T}$ such that $L(\mathcal{W}_1) \cap L(\mathcal{W}_2) \neq \emptyset$: Take the set of all bi-cancelable transitions. □

References

[1] T. G. Szymanski, J. H. Williams, Noncanonical extensions of bottom-up parsing techniques, SIAM Journal on Computing 5 (2) (1976). doi:10.1137/0205019.

[2] H. B. Hunt III, On the Decidability of Grammar Problems, Journal of the ACM 29 (2) (1982) 429–447. doi:10.1145/322307.322317.

[3] W. Czerwiński, S. Lasota, R. Meyer, S. Muskalla, K. N. Kumar, P. Saivasan, Regular separability of well-structured transition systems, in: S. Schewe, L. Zhang (Eds.), 29th International Conference on Concurrency Theory, CONCUR 2018, September 4-7, 2018, Beijing, China, Vol. 118 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 35:1–35:18. doi:10.4230/LIPICS.CONCUR.2018.35.

[4] E. Keskin, R. Meyer, Separability and non-determinizability of WSTS, in: G. A. Pérez, J. Raskin (Eds.), 34th International Conference on Concurrency Theory, CONCUR 2023, September 18-23, 2023, Antwerp, Belgium, Vol. 279 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 8:1–8:17. doi:10.4230/LIPICS.CONCUR.2023.8.

[5] C. Köcher, G. Zetzsche, Regular separators for VASS coverability languages, in: P. Bouyer, S. Srinivasan (Eds.), 43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2023, December 18-20, 2023, IIIT Hyderabad, Telangana, India,

Vol. 284 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 15:1–15:19. doi:10.4230/LIPICS.FSTTCS.2023.15.

[6] W. Czerwiński, S. Lasota, Regular separability of one counter automata, in: 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017, IEEE Computer Society, 2017, pp. 1–12. doi:10.1109/LICS.2017.8005079.

[7] L. Clemente, W. Czerwiński, S. Lasota, C. Paperman, Regular Separability of Parikh Automata, in: I. Chatzigiannakis, P. Indyk, F. Kuhn, A. Muscholl (Eds.), 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017), Vol. 80 of Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2017, pp. 117:1–117:13. doi:10.4230/LIPIcs.ICALP.2017.117.
URL http://drops.dagstuhl.de/opus/volltexte/2017/7497

[8] L. Clemente, W. Czerwiński, S. Lasota, C. Paperman, Separability of reachability sets of vector addition systems, in: H. Vollmer, B. Vallée (Eds.), 34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany, Vol. 66 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 24:1–24:14. doi:10.4230/LIPICS.STACS.2017.24.

[9] R. S. Thinniyam, G. Zetzsche, Regular separability and intersection emptiness are independent problems, in: A. Chattopadhyay, P. Gastin (Eds.), 39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019, December 11-13, 2019, Bombay, India, Vol. 150 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 51:1–51:15. doi:10.4230/LIPICS.FSTTCS.2019.51.

[10] P. Baumann, E. Keskin, R. Meyer, G. Zetzsche, Separability in Büchi VASS and singly non-linear systems of inequalities, in: K. Bringmann, M. Grohe, G. Puppis, O. Svensson (Eds.), 51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, July 8-12, 2024, Tallinn, Estonia, Vol. 297 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, pp. 126:1–126:19. doi:10.4230/LIPICS.ICALP.2024.126.

[11] P. Baumann, R. Meyer, G. Zetzsche, Regular separability in Büchi VASS, in: P. Berenbrink, P. Bouyer, A. Dawar, M. M. Kanté (Eds.), 40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany, Vol. 254 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 9:1–9:19. doi:10.4230/LIPICS.STACS.2023.9.

[12] W. Czerwiński, G. Zetzsche, An approach to regular separability in vector addition systems, in: Proceedings of the 35th Annual ACM/IEEE

Symposium on Logic in Computer Science, LICS '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 341–354. doi:10.1145/3373718.3394776.
URL https://doi.org/10.1145/3373718.3394776

[13] E. Keskin, R. Meyer, On the separability problem of VASS reachability languages, in: P. Sobocinski, U. D. Lago, J. Esparza (Eds.), Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024, ACM, 2024, pp. 49:1–49:14. doi:10.1145/3661814.3662116.

[14] O. H. Ibarra, Reversal-bounded multicounter machines and their decision problems, Journal of the ACM (JACM) 25 (1) (1978) 116–133. doi:10.1145/322047.322058.

[15] S. A. Greibach, Remarks on blind and partially blind one-way multicounter machines, Theoretical Computer Science 7 (3) (1978) 311–324. doi:10.1016/0304-3975(78)90020-8.

[16] P. Baumann, F. D'Alessandro, M. Ganardi, O. Ibarra, I. McQuillan, L. Schütze, G. Zetzsche, Unboundedness problems for machines with reversal-bounded counters, in: O. Kupferman, P. Sobocinski (Eds.), Foundations of Software Science and Computation Structures, Springer Nature Switzerland, Cham, 2023, pp. 240–264. doi:10.1007/978-3-031-30829-1_12.

[17] M. Jantzen, A. Kurganskyy, Refining the hierarchy of blind multicounter languages and twist-closed trios, Inf. Comput. 185 (2) (2003) 159–181. doi:10.1016/S0890-5401(03)00087-7.

[18] M. Cadilhac, A. Finkel, P. McKenzie, Affine Parikh automata, RAIRO Theor. Informatics Appl. 46 (4) (2012) 511–545. doi:10.1051/ITA/2012013.

[19] A. Bostan, A. Carayol, F. Koechlin, C. Nicaud, Weakly-unambiguous Parikh automata and their link to holonomic series, in: A. Czumaj, A. Dawar, E. Merelli (Eds.), 47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference), Vol. 168 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 114:1–114:16. doi:10.4230/LIPICS.ICALP.2020.114.

[20] M. Cadilhac, A. Finkel, P. McKenzie, Unambiguous constrained automata, Int. J. Found. Comput. Sci. 24 (7) (2013) 1099–1116. doi:10.1142/S0129054113400339.

[21] A. Finkel, A. Sangnier, Reversal-bounded counter machines revisited, in: E. Ochmanski, J. Tyszkiewicz (Eds.), Mathematical Foundations of Computer Science 2008, 33rd International Symposium, MFCS 2008,

Torun, Poland, August 25-29, 2008, Proceedings, Vol. 5162 of Lecture Notes in Computer Science, Springer, 2008, pp. 323–334. doi:10.1007/978-3-540-85238-4_26.

[22] M. M. Bersani, S. Demri, The complexity of reversal-bounded model-checking, in: C. Tinelli, V. Sofronie-Stokkermans (Eds.), Frontiers of Combining Systems, 8th International Symposium, FroCoS 2011, Saarbrücken, Germany, October 5-7, 2011. Proceedings, Vol. 6989 of Lecture Notes in Computer Science, Springer, 2011, pp. 71–86. doi:10.1007/978-3-642-24364-6_6.

[23] G. Zetzsche, Silent transitions in automata with storage, in: F. V. Fomin, R. Freivalds, M. Z. Kwiatkowska, D. Peleg (Eds.), Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II, Vol. 7966 of Lecture Notes in Computer Science, Springer, 2013, pp. 434–445. doi:10.1007/978-3-642-39212-2_39.

[24] M. Cadilhac, A. Krebs, P. McKenzie, The algebraic theory of Parikh automata, Theory Comput. Syst. 62 (5) (2018) 1241–1268. doi:10.1007/S00224-017-9817-2.

[25] G. Zetzsche, The complexity of downward closure comparisons, in: I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, D. Sangiorgi (Eds.), 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy, Vol. 55 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, pp. 123:1–123:14. doi:10.4230/LIPICS.ICALP.2016.123.

[26] E. Erlich, S. Guha, I. Jecker, K. Lehtinen, M. Zimmermann, History-deterministic Parikh automata, in: G. A. Pérez, J. Raskin (Eds.), 34th International Conference on Concurrency Theory, CONCUR 2023, September 18-23, 2023, Antwerp, Belgium, Vol. 279 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 31:1–31:16. doi:10.4230/LIPICS.CONCUR.2023.31.

[27] S. Guha, I. Jecker, K. Lehtinen, M. Zimmermann, Parikh automata over infinite words, in: A. Dawar, V. Guruswami (Eds.), 42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022, December 18-20, 2022, IIT Madras, Chennai, India, Vol. 250 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, pp. 40:1–40:20. doi:10.4230/LIPICS.FSTTCS.2022.40.

[28] E. Filiot, S. Guha, N. Mazzocchi, Two-way Parikh automata, in: A. Chattopadhyay, P. Gastin (Eds.), 39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019, December 11-13, 2019, Bombay, India, Vol. 150 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 40:1–40:14. doi:10.4230/LIPICS.FSTTCS.2019.40.

[29] M. Grobler, L. Sabellek, S. Siebertz, Remarks on Parikh-recognizable omega-languages, in: A. Murano, A. Silva (Eds.), 32nd EACSL Annual Conference on Computer Science Logic, CSL 2024, February 19-23, 2024, Naples, Italy, Vol. 288 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, pp. 31:1–31:21. doi:10.4230/LIPICS.CSL.2024.31.

[30] M. Cadilhac, A. Ghosh, G. A. Pérez, R. Raha, Parikh one-counter automata, in: J. Leroux, S. Lombardy, D. Peleg (Eds.), 48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023, August 28 to September 1, 2023, Bordeaux, France, Vol. 272 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 30:1–30:15. doi:10.4230/LIPICS.MFCS.2023.30.

[31] Y. Shakiba, H. Sinclair-Banks, G. Zetzsche, A complexity dichotomy for semilinear target sets in automata with one counter, to appear in Proc. of LICS 2025 (2025). arXiv:2505.13749, doi:10.48550/ARXIV.2505.13749.

[32] F. Klaedtke, H. Rueß, Monadic second-order logics with cardinalities, in: J. C. M. Baeten, J. K. Lenstra, J. Parrow, G. J. Woeginger (Eds.), Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003. Proceedings, Vol. 2719 of Lecture Notes in Computer Science, Springer, 2003, pp. 681–696. doi:10.1007/3-540-45061-0_54.

[33] S. Halfon, P. Schnoebelen, G. Zetzsche, Decidability, complexity, and expressiveness of first-order logic over the subword ordering, in: 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017, IEEE Computer Society, 2017, pp. 1–12. doi:10.1109/LICS.2017.8005141.

[34] A. Bouajjani, P. Habermehl, Symbolic reachability analysis of FIFO-channel systems with nonregular sets of configurations, Theor. Comput. Sci. 221 (1-2) (1999) 211–250. doi:10.1016/S0304-3975(99)00033-X.

[35] C. Haase, S. Halfon, Integer vector addition systems with states, in: J. Ouaknine, I. Potapov, J. Worrell (Eds.), Reachability Problems - 8th International Workshop, RP 2014, Oxford, UK, September 22-24, 2014. Proceedings, Vol. 8762 of Lecture Notes in Computer Science, Springer, 2014, pp. 112–124. doi:10.1007/978-3-319-11439-2_9.

[36] K. L. McMillan, Interpolation and SAT-based model checking, in: W. A. H. Jr., F. Somenzi (Eds.), Computer Aided Verification, 15th International Conference, CAV 2003, Boulder, CO, USA, July 8-12, 2003, Proceedings, Vol. 2725 of Lecture Notes in Computer Science, Springer, 2003, pp. 1–13. doi:10.1007/978-3-540-45069-6_1.

[37] Y. Vizel, G. Weissenbacher, S. Malik, Boolean satisfiability solvers and their applications in model checking, Proc. IEEE 103 (11) (2015) 2021–2035. doi:10.1109/JPROC.2015.2455034.

[38] M. Veanes, N. S. Bjørner, L. Nachmanson, S. Bereg, Monadic decomposition, J. ACM 64 (2) (2017) 14:1–14:28. doi:10.1145/3040488.

[39] P. Bergsträßer, M. Ganardi, A. W. Lin, G. Zetzsche, Ramsey quantifiers in linear arithmetics, Proc. ACM Program. Lang. 8 (POPL) (2024) 1–32. doi:10.1145/3632843.

[40] M. Hague, A. W. Lin, P. Rümmer, Z. Wu, Monadic decomposition in integer linear arithmetic, in: N. Peltier, V. Sofronie-Stokkermans (Eds.), Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part I, Vol. 12166 of Lecture Notes in Computer Science, Springer, 2020, pp. 122–140. doi:10.1007/978-3-030-51074-9_8.

[41] C. Haase, S. N. Krishna, K. Madnani, O. S. Mishra, G. Zetzsche, An efficient quantifier elimination procedure for Presburger arithmetic, in: K. Bringmann, M. Grohe, G. Puppis, O. Svensson (Eds.), 51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, July 8-12, 2024, Tallinn, Estonia, Vol. 297 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, pp. 142:1–142:17. doi:10.4230/LIPICS.ICALP.2024.142.

[42] S. Grumbach, P. Rigaux, L. Segoufin, Spatio-temporal data handling with constraints, in: R. Laurini, K. Makki, N. Pissinou (Eds.), ACM-GIS '98, Proceedings of the 6th international symposium on Advances in Geographic Information Systems, November 6-7, 1998, Washington, DC, USA, ACM, 1998, pp. 106–111. doi:10.1145/288692.288712.

[43] G. Kuper, L. Libkin, J. Paredaens, Constraint databases, Springer Science & Business Media, 2013. doi:10.1007/978-3-662-04031-7.

[44] C. Choffrut, S. Grigorieff, Separability of rational relations in $A^* \times \mathbb{N}^m$ by recognizable relations is decidable, Information Processing Letters 99 (1) (2006) 27–32. doi:10.1016/j.ipl.2005.09.018.

[45] D. Chistikov, C. Haase, A. Mansutti, Quantifier elimination for counting extensions of presburger arithmetic, in: P. Bouyer, L. Schröder (Eds.), Foundations of Software Science and Computation Structures - 25th International Conference, FOSSACS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2-7, 2022, Proceedings, Vol. 13242 of Lecture Notes in Computer Science, Springer, 2022, pp. 225–243. doi:10.1007/978-3-030-99253-8_12.
URL https://doi.org/10.1007/978-3-030-99253-8_12

[46] S. Ginsburg, E. H. Spanier, Semigroups, Presburger formulas, and languages, Pacific Journal of Mathematics 16 (2) (1966) 285–296.

[47] J. Berstel, Transductions and Context-Free Languages, Teubner, 1979. doi:10.1007/978-3-663-09367-1.

[48] J. Sakarovitch, Elements of Automata Theory, Cambridge University Press, Cambridge, 2009. doi:10.1017/CBO9781139195218.

[49] O. H. Ibarra, B. Ravikumar, On the Parikh Membership Problem for FAs, PDAs, and CMs, in: A.-H. Dediu, C. Martín-Vide, J.-L. Sierra-Rodríguez, B. Truthe (Eds.), Language and Automata Theory and Applications, Springer International Publishing, Cham, 2014, pp. 14–31. doi:10.1007/978-3-319-04921-2_2.

[50] H. Seidl, T. Schwentick, A. Muscholl, P. Habermehl, Counting in trees for free, in: J. Díaz, J. Karhumäki, A. Lepistö, D. Sannella (Eds.), Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings, Vol. 3142 of Lecture Notes in Computer Science, Springer, 2004, pp. 1136–1149. doi:10.1007/978-3-540-27836-8_94.

[51] S. Ginsburg, E. H. Spanier, Bounded regular sets, Proceedings of the American Mathematical Society 17 (5) (1966) 1043–1049. doi:10.1090/S0002-9939-1966-0201310-3.

[52] J. Esparza, Petri nets, commutative context-free grammars, and basic parallel processes, Fundam. Informaticae 31 (1) (1997) 13–25. doi:10.3233/FI-1997-3112.