Regular Separability of Parikh Automata

Lorenzo Clemente[1], Wojciech Czerwiński[1], Sławomir Lasota[1], and Charles Paperman[2]

[1] University of Warsaw
[2] University of Tübingen

Abstract. We investigate a subclass of languages recognized by vector addition systems, namely languages of nondeterministic Parikh automata. While the regularity problem (is the language of a given automaton regular?) is undecidable for this model, we show surprising decidability of the regular separability problem: given two Parikh automata, is there a regular language that contains one of them and is disjoint from the other?

1 Introduction

In this paper we investigate separability problems for languages of finite words. We say that a language U is *separated from* a language V by S if $U \subseteq S$ and $V \cap S = \emptyset$. In the sequel we also often say that U *and* V are *separated by* S. For two families of languages $\mathcal{F}$ and $\mathcal{G}$, the $\mathcal{F}$ *separability problem for* $\mathcal{G}$ asks, given two given languages $U, V \in \mathcal{G}$, whether U is separated from V by some language from $\mathcal{F}$. The same notion of separability makes clearly sense if $\mathcal{F}$ and $\mathcal{G}$ are classes of sets of vectors instead of classes of languages.

Concretely, in this paper we mainly consider $\mathcal{F}$ to be regular languages, and $\mathcal{G}$ to be the languages of Parikh automata; or $\mathcal{F}$ the unary sets, and $\mathcal{G}$ the semilinear sets.

Motivation. Separability is a classical problem in theoretical computer science. It was investigated most extensively in the area of formal languages, for $\mathcal{G}$ being the family of all regular word languages. Since regular languages are effectively closed under complement, the $\mathcal{F}$ separability problem is a generalization of the $\mathcal{F}$ characterization problem, which asks whether a given language belongs to $\mathcal{F}$. Indeed, $L \in \mathcal{F}$ if and only if L is separated from its complement by some language from $\mathcal{F}$. Separability problems for regular languages attracted recently a lot of attention, which resulted in establishing the decidability of $\mathcal{F}$ separability for the family $\mathcal{F}$ of separators being the piecewise testable languages [5,13] (recently generalized to finite ranked trees [7]), the locally and locally threshold testable languages [12], the languages definable in first order logic [15], and the languages of certain higher levels of the first order hierarchy [14], among others.

Separability of nonregular languages attracted little attention till now. The reasons for this may be twofold. First, for regular languages one can use standard algebraic tools, like syntactic monoids, and indeed most of the results have been obtained with the help of such techniques. Second, some strong intractability results have been known already since 70's, when Szymanski and Williams proved that regular separability of context-free languages is undecidable [16]. Later Hunt [8] generalized this result: he showed that $\mathcal{F}$-separability of context-free languages is undecidable for every class $\mathcal{F}$ which is closed under finite boolean combinations and contains all languages of the form $w\Sigma^*$ for $w \in \Sigma^*$. This is a very weak condition, so it seemed that nothing nontrivial can be done outside regular languages with respect to separability problems. Furthermore, Kopczyński has recently shown that regular separability is undecidable even for languages of visibly pushdown automata [11], thus strengthening the result by Szymanski and Williams. On the positive side, piecewise testable separability has been shown decidable for context-free languages, languages of vector addition systems (VAS languages), and some other classes of languages [6]. This inspired us to start a quest for decidable cases beyond regular languages.

In [4] we have shown decidability of *unary separability* of reachability sets of vector addition systems (VASes). By *unary sets* we mean Parikh images of commutative regular languages, and thus the latter problem is equivalent to commutative regular separability of (commutative closures of) VAS languages. The decidability status of the regular separability problem for the whole class of VAS languages remains open.

Our contribution. This paper is a continuation of the line of research trying to understand the regular separability problem for language classes beyond regular languages. We report a further progress towards solving the open problem mentioned above: we show decidability of the regular separability problem for the subclass of VAS languages where we allow negative counter values during a run. This class of languages is also known as languages of *integer VASSes*, and it admits many different characterizations; for instance, it coincides with languages of *one-way reversal-bounded counter machines* [9], *Parikh automata* [10] (cf. also [1, Proposition 11]), which in turn are equivalent to the very similar model of *constrained automata* [2]. In this paper, we present our results in terms of constrained automata, but given the similarity with Parikh automata (and in light of their equivalence), we overload the name Parikh automata for both models.

Notice that PA languages are not closed under complement, and thus our decidability result about regular separability does not imply decidability of the regularity problem (is the language of a given Parikh automaton regular?). Moreover, the regularity problem for PA languages is actually *undecidable* [1][3], which makes our decidability result a rare instance of a case where regularity is undecidable but regular separability is decidable. A result in a similar spirit is that piecewise testability of a context-free language is undecidable, while piecewise-testable separability of two context-free languages is decidable [6].

Parikh automata are finite nondeterministic automata where accepting runs are further restricted to satisfy a semilinear condition on the multiset of transi-

[3] Later shown decidable for unambiguous PA [2].

tions appearing in the run. Our decidability result is actually stated in the more general setting of $\mathcal{C}$*-Parikh automata*, where $\mathcal{C} \subseteq \bigcup_{d \in \mathbb{N}} \mathcal{P}(\mathbb{N}^d)$ is a class of sets of vectors used as an acceptance condition. We prove that the regular separability problem for languages of $\mathcal{C}$-Parikh automata reduces to the *unary* separability problem for the class $\mathcal{C}$ itself, provided that $\mathcal{C}$ is effectively closed under inverse images of affine functions. Two prototypical classes $\mathcal{C}$ satisfying the latter closure condition are semilinear sets and VAS reachability sets. Moreover, unary separability of semilinear set is known to be decidable [3], and as recalled before the same result has recently been extended to VAS reachability sets [4]. As a consequence of our reduction, we thus deduce decidability of regular separability of $\mathcal{C}$-Parikh automata languages where the acceptance condition $\mathcal{C}$ can be instantiated to either the semilinear sets, or the VAS reachability sets.

2 Preliminaries

Vectors sets. A set $S \subseteq \mathbb{N}^d$ is *linear* if there exist a *base* $b \in \mathbb{N}^d$ and *periods* $p_1, \ldots, p_k \in \mathbb{N}^d$ s.t. $S = \{b + n_1 p_1 + \ldots + n_k p_k \mid n_1, \ldots, n_k \in \mathbb{N}\}$, and it is *semilinear* if it is a finite union of linear sets. For a vector $v \in \mathbb{N}^d$ and $i \in \{1, \ldots, d\}$, let $v[i]$ denote its i-th coordinate. For $n \in \mathbb{N}$, we say that two vectors $x, y \in \mathbb{N}^d$ are *n-unary equivalent*, written $x \equiv_n y$, if for every coordinate $i \in \{1, \ldots, d\}$ it holds $x[i] \equiv y[i] \mod n$ and moreover $x[i] \leq n \iff y[i] \leq n$. A set $S \subseteq \mathbb{N}^d$ is *unary* if for some n, S is a union of equivalence classes of $\equiv_n$. Intuitively, to decide membership in a unary set S it is enough to count on every coordinate exactly up to some threshold n, and modulo n for values larger than n. Every unary set is in particular semilinear.

Let $\Sigma = \{a_1, \ldots, a_k\}$ be an ordered alphabet. For a word $w \in \Sigma^*$ and a letter $a_i \in \Sigma$, by $\#_{a_i}(w)$ we denote the number of letters a_i in w. The *Parikh image* of a word $w \in \Sigma^*$ is the vector $\Pi(w) = (\#_{a_1}(w), \ldots, \#_{a_k}(w)) \in \mathbb{N}^k$. The *Parikh image* of a language $L \subseteq \Sigma^*$ is $\Pi(L) = \{\Pi(w) \mid w \in L\}$, the set of Parikh images of all words belonging to L.

Parikh automata. A *nondeterministic finite automaton* (NFA) $\mathcal{A} = (Q, I, F, T)$ over a finite alphabet Σ consists of a finite set of states Q, distinguished subsets of initial and final states $I, F \subseteq Q$, and a set of transitions $T \subseteq Q \times \Sigma \times Q$. A *nodeterministic Parikh automaton*[4] is a pair $(\mathcal{A}, S)$ consisting of an NFA $\mathcal{A}$ and a semilinear set $S \subseteq \mathbb{N}^d$, for $d = |T|$. A *run* of a Parikh automaton over a word $w = a_1 \ldots a_n \in \Sigma^*$ is a sequence of transitions $\rho = t_1 \ldots t_n \in T^*$, where $t_i = (q_{i-1}, a_i, q_i)$, starting in an initial state q_0. A run ρ is *accepting* if its ending state q_n is final and $\Pi(\rho) \in S$. The language of a Parikh automaton, denoted $L(\mathcal{A}, S)$, contains all words w admitting an accepting run; it is thus a subset of the language $L(\mathcal{A})$ of the underlying NFA.

Remark 1. A more liberal definition for ε-Parikh automata can be given by allowing transitions to read ε's, i.e, $T \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$. However, allowing

[4] This is the same as *constrained automata* from [2]

ε-transitions does not increase the expressiveness of Parikh automata, which follows from closure under (possibly erasing) homomorphisms of the latter class [10, Property 4.(2)].

One can generalize Parikh automata by using some other family of vector sets in the place of semilinear sets. For a class $\mathcal{C} \subseteq \bigcup_{d \in \mathbb{N}} \mathcal{P}(\mathbb{N}^d)$ of vector sets, a *$\mathcal{C}$-Parikh automaton* is a pair $(\mathcal{A}, S)$, where $\mathcal{A}$ is an NFA and $S \in \mathcal{C}$. The language $L(\mathcal{A}, S)$ is then defined exactly as above.

A $\mathcal{C}$-Parikh automaton $(\mathcal{A}, S)$ is *deterministic* if the underlaying automaton $\mathcal{A}$ is so. The languages of (non)deterministic $\mathcal{C}$-Parikh automata are shortly called (non)deterministic $\mathcal{C}$-Parikh languages below.

3 Main result

A function $f : \mathbb{N}^k \to \mathbb{N}^\ell$ is called *affine* if it is of the form $f(v) = Mv + u$ for a matrix M of dimension $\ell \times k$ and a vector $u \in \mathbb{N}^\ell$. A class of vector sets $\mathcal{C} \subseteq \bigcup_{d \in \mathbb{N}} \mathcal{P}(\mathbb{N}^d)$ is called *robust* if it fulfills the following two conditions:

- $\mathcal{C}$ is effectively closed under inverse images of affine functions,
- the unary separability problem is decidable for $\mathcal{C}$.

As our main result we prove decidability of the regular separability problem for $\mathcal{C}$-Parikh automata.

Theorem 2. *The regular separability problem is decidable for $\mathcal{C}$-Parikh automata, for every robust class $\mathcal{C}$ of vector sets.*

The proof of Theorem 2 is split into two parts. In Section 4 we provide a reduction of the regular separability problem of *nondeterministic* $\mathcal{C}$-Parikh automata to the same problem of *deterministic* ones; this step is crucial for understanding how the regular separability problem differs from the regularity problem, which does not admit a similar reduction. Then in Section 5 we reduce the regular separability problem for deterministic $\mathcal{C}$-Parikh automata to the unary separability problem for vector sets in $\mathcal{C}$.

In Section 6 we consider two instantiations of the class $\mathcal{C}$. First, taking semilinear sets as $\mathcal{C}$ we derive decidability for plain Parikh automata. Second, we consider the class $\mathcal{C}_{\text{SEC-VAS}}$ of *sections of* reachability sets of VASes (detailed definitions are deferred to Section 6), which allows us to obtain decidability for $\mathcal{C}_{\text{SEC-VAS}}$-Parikh automata. Note that the latter model properly extends plain Parikh automata.

4 From nondeterministic to deterministic PA

The aim of this section is to prove the following lemma:

Lemma 3. *If $\mathcal{C}$ is closed under inverse images of linear mappings, then the regular separability problem of nondeterministic $\mathcal{C}$-Parikh automata effectively reduces to the same problem of deterministic ones.*

Before embarking on the proof, we need to state and prove a couple of auxiliary facts. In the rest of the section, we assume that the class $\mathcal{C}$ is closed under inverse images of linear mappings. Given two alphabets Σ and Γ, a *letter-to-letter homomorphism* is a function $h : \Sigma \to \Gamma$ which extends homomorphically to a function from Σ^* to Γ^*, and thus to languages.

Lemma 4. *Every nondeterministic $\mathcal{C}$-Parikh language is the image of a letter-to-letter homomorphism of a deterministic $\mathcal{C}$-Parikh language.*

Proof. Fix a nondeterministic $\mathcal{C}$-Parikh automaton $(\mathcal{A}, S)$ of maximal nondeterministic branching n recognizing the language $L(\mathcal{A}, S) \subseteq \Sigma^*$. Consider the extended alphabet $\Gamma = \Sigma \times \{1, \dots, n\}$ obtained by labelling each symbol from Σ with an index to resolve nondeterminism, and consider the letter-to-letter homomorphism $h : \Gamma \to \Sigma$ that maps (a, i) to a. Let $(\mathcal{B}, T)$ be the deterministic $\mathcal{C}$-Parikh automaton over Γ which is obtained from $\mathcal{A}$ by a relabelling in every state the i-th transition over a by (a, i). The acceptance condition $T \subseteq \mathbb{N}^{|\Sigma| \cdot n}$ is $T := \phi^{-1}(S)$, where $\phi : \mathbb{N}^{|\Sigma| \cdot n} \to \mathbb{N}^{|\Sigma|}$ is the linear mapping that sums up the entries $(a, 1), \dots, (a, n)$ corresponding to the original symbol a. One easily verifies that $L(\mathcal{A}, S) = h(L(\mathcal{B}, T))$, as required. $\square$

Lemma 5. *Deterministic $\mathcal{C}$-Parikh languages are effectively closed under inverse images of letter-to-letter homomorphisms.*

Proof. Given a deterministic $\mathcal{C}$-Parikh automaton $(\mathcal{A}, S)$ over Σ and a letter-to-letter homomorphism $h : \Gamma \to \Sigma$, one computes a deterministic $\mathcal{C}$-Parikh automaton $(\mathcal{B}, T)$ as follows. The automaton $\mathcal{B}$ is obtained by replacing every transition (p, a, q) in $\mathcal{A}$ by transitions (p, b, q), one for every $b \in h^{-1}(a)$. The constraint $T \in \mathcal{C}$ is the inverse image of S under the linear function that sums up values on all coordinates corresponding to letters in $h^{-1}(a) \subseteq \Gamma$ in order to compute the value on the coordinate corresponding to $a \in \Sigma$. Finally, the constraint T, and hence also the automaton $(\mathcal{B}, T)$ can be computed. $\square$

Lemma 6. *Nondeterministic $\mathcal{C}$-Parikh languages are effectively closed under inverse images of letter-to-letter homomorphisms.*

Proof. The construction is exactly the same as in the proof of Lemma 5 above, but the resulting automaton does not have to be deterministic. $\square$

The next lemma is the cornerstone of our reduction. It allows to make one automaton deterministic without introducing nondeterminism in the second one.

Lemma 7. *Languages $h(L)$ and K are regular separable if, and only if, L and $h^{-1}(K)$ are so.*

Proof. For the "only if" direction, if a regular language R separates $h(L)$ and K then the language $h^{-1}(R)$ separates L and $h^{-1}(K)$. Indeed, the inclusion $L \subseteq h^{-1}(R)$ follows from the inclusion $h(L) \subseteq R$ since $L \subseteq h^{-1}(h(L))$, and the disjointness of $h^{-1}(R)$ and $h^{-1}(K)$ follows from disjointness of R and K.

For the "if" direction, if a regular language R separates L and $h^{-1}(K)$ then the language $h(R)$ separates the languages $h(L)$ and K. The inclusion $h(L) \subseteq h(R)$ follows by the inclusion $L \subseteq R$, and the disjointness of $h(R)$ and K follows from disjointness of R and $h^{-1}(K)$ since $h(h^{-1}(K)) \subseteq K$. $\square$

Proof of Lemma 3. Let L, K be two nondeterministic $\mathcal{C}$-PA languages. By Lemma 4, we may assume that L is the image $h(L_1)$ of a deterministic language L_1. By Lemma 7, regular separability for $h(L_1), K$ is the same as for $L_1, h^{-1}(K)$. By Lemma 6, $h^{-1}(K)$ is a nondeterministic language itself, so by Lemma 4 it equals the image $g(K_1)$ of a deterministic language K_1. We have thus reduced to regular separability for $L_1, g(K_1)$, where now both L_1 and K_1 are deterministic languages. Since regular separability is symmetric, regular separability for $L_1, g(K_1)$ is the same for $g(K_1), L_1$. Applying once more Lemma 7, the latter statement is equivalent to regular separability for $K_1, g^{-1}(L_1)$. By Lemma 5, $g^{-1}(L_1)$ is a deterministic language. Since every step was effective, this concludes the proof. $\square$

5 Regular separability reduces to unary separability

In this section we reduce regular separability of deterministic $\mathcal{C}$-Parikh languages to unary separability of vector sets in $\mathcal{C}$.

Lemma 8. *Let $\mathcal{C}$ be a class of vectors closed under inverse images of affine mappings. The regular separability problem for deterministic $\mathcal{C}$-Parikh automata reduces to the unary separability problem for vector sets in $\mathcal{C}$.*

The rest of this section is devoted to the proof of this lemma. Let $L_1, L_2 \subseteq \Sigma^*$ be languages of deterministic $\mathcal{C}$-Parikh automata $(\mathcal{A}_1, S_1)$ and $(\mathcal{A}_2, S_2)$, respectively. The proof comprises three steps:

1. As the first step, we show that w.l.o.g. we may assume $\mathcal{A}_1 = \mathcal{A}_2$.
2. In the second step, we partition Σ^* into finitely many regular languages $K_1, \ldots, K_m$ and we reduce regular separability of L_1 and L_2 to regular separability of $L_1 \cap K_i$ and $L_2 \cap K_i$ for every $i \in \{1, \ldots, m\}$. These subproblems turn out to be easier that the general one, due to the additional structural information encoded in the languages K_i's.
3. In the last step, we reduce separability of $L_1 \cap K_i$ and $L_2 \cap K_i$ to unary separability of vector sets in $\mathcal{C}$.

Step 1: Unifying the underlying automaton. As the input languages are subsets of regular languages recognised by their underlying finite automata, $L_1 = L(\mathcal{A}_1, S_1) \subseteq L(\mathcal{A}_1)$ and $L_2 = L(\mathcal{A}_2, S_2) \subseteq L(\mathcal{A}_2)$, it is enough to consider separability of L_1 and L_2 *inside* the intersection of $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$:

Proposition 9. *The languages L_1 and L_2 are regular separable if, and only if, the languages $L_1 \cap L(\mathcal{A}_2)$ and $L_2 \cap L(\mathcal{A}_1)$ are so.*

Proof. The "only if" direction is trivial as every language separating L_1 and L_2 separates $L_1 \cap L(\mathcal{A}_2)$ and $L_2 \cap L(\mathcal{A}_1)$ as well. For the opposite direction, we observe that if a regular language S separates $L_1 \cap L(\mathcal{A}_2)$ and $L_2 \cap L(\mathcal{A}_1)$, then $S' = S \cup \overline{L(\mathcal{A}_2)}$ is a regular language separating L_1 and L_2. □

Let $\mathcal{A}$ be the product automaton of $\mathcal{A}_1$ and $\mathcal{A}_2$, and thus $L(\mathcal{A}) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$. It is deterministic since both $\mathcal{A}_1$ and $\mathcal{A}_2$ are so. We claim that one can compute sets $U_1, U_2 \in \mathcal{C}$ such that $L_1 \cap L(\mathcal{A}_2) = L(\mathcal{A}, U_1)$ and $L_2 \cap L(\mathcal{A}_1) = L(\mathcal{A}, U_2)$. The set T of transitions of $\mathcal{A}$ is a subset of the product $T_1 \times T_2$ of transitions of $\mathcal{A}_1$ and $\mathcal{A}_2$, and thus there are obvious projections functions $\pi_1 : T \to T_1$ and $\pi_2 : T \to T_2$. If we enumerate the transition sets, say $T_1 = \{t_1^1, \ldots, t_1^m\}$, $T_2 = \{t_2^1, \ldots, t_2^n\}$, and $T = \{t_1, \ldots, t_\ell\}$ with $\ell \leq m \cdot n$, we obtain $\pi_1 : \{1, \ldots, \ell\} \to \{1, \ldots, m\}$ and $\pi_2 : \{1, \ldots, \ell\} \to \{1, \ldots, n\}$. We use these projections to define two linear (and in particular, affine) functions $\psi_1 : \mathbb{N}^\ell \to \mathbb{N}^m$ and $\psi_2 : \mathbb{N}^\ell \to \mathbb{N}^n$ which instead of counting transitions in T, count the corresponding transitions in T_1 or in T_2, respectively; formally,

$$\psi_1(v)[j] = \sum_{i:\pi_1(i)=j} v[i] \qquad \psi_2(v)[j] = \sum_{i:\pi_2(i)=j} v[i].$$

Finally, we set $U_1 := \psi_1^{-1}(S_1)$ and $U_2 := \psi_2^{-1}(S_2)$. Intuitively, U_1 and U_2 are as S_1 and S_2, except that instead of single transitions of $\mathcal{A}_1$ or $\mathcal{A}_2$ they are seeing pairs of transitions, and simply ignore one of them. Since $\mathcal{C}$ is closed under inverse images of affine mappings by assumption, $U_1, U_2 \in \mathcal{C}$. For the rest of the proof we may thus assume that the input automata are $(\mathcal{A}, U_1)$ and $(\mathcal{A}, U_2)$.

Step 2: Regular partition using skeletons. We now define a partition of Σ^* into finitely many parts, such that words belonging to the same part behave similarly with respect to automaton $\mathcal{A}$.

We use the notion of *skeleton* of a run, defined already in [2], where it was used to solve the regularity problem of *unambiguous* Parikh automata. Consider a run $\rho = t_1 \cdots t_k \in T^*$. The idea of skeleton is to traverse ρ from left to right and remove loops, but only if such removal does not decrease the set of states visited so far. Formally, the skeleton is a function from runs to runs defined by induction. We set $\textsc{skel}(\varepsilon) = \varepsilon$. For the induction step, suppose that $\textsc{skel}(t_1 \ldots t_{k-1}) = u_1 \ldots u_\ell \in T^*$ is already defined, and let q be the ending state of the new transition t_k. If q does not appear in the run $u_1 \ldots u_\ell$, we put $\textsc{skel}(t_1 \ldots t_k) = u_1 \ldots u_\ell t_k$. Otherwise, let u_m, for $m < \ell$, be the last transition that ends in state q. If all states visited by $u_{m+1} \ldots u_\ell$ are also visited by $u_1 \ldots u_m$, we put $\textsc{skel}(t_1 \ldots t_k) = u_1 \ldots u_m$ thus removing the loop; otherwise, we put $\textsc{skel}(t_1 \ldots t_k) = u_1 \ldots u_\ell t_k$.

The so defined skeleton $\textsc{skel}(\rho)$ of a run ρ has two properties: 1) $\textsc{skel}(\rho)$ visits the same states as ρ, 2) the length of $\textsc{skel}(\rho)$ is at most n^2, where n is the number of states in the automaton $\mathcal{A}$. The first point is clear by definition. In order to see the second point, assume towards a contradiction that the length of the skeleton is longer than n^2. By the pigeonhole principle, some state is thus visited more than n times, so there are at least n loops in between two consecutive occurrences of this state in the skeleton. Therefore it is impossible that each loop contains some new state not present in all the previous loops, and thus one of these loops should be removed during the process of creating the skeleton, a contradiction.

We abusively call a run ρ a *skeleton* if $\textsc{skel}(\rho) = \rho$. Because of the bound n^2 on the length of a skeleton, if d is the total number of transitions of $\mathcal{A}$, then there are at most d^{n^2} skeleton runs. Let $\rho_1, \ldots, \rho_m$ be all the skeletons, with $m \leq k^{n^2}$. We define K_i to be the set of all words w having an accepting run ρ in automaton $\mathcal{A}$ with $\textsc{skel}(\rho) = \rho_i$. Since $\mathcal{A}$ is deterministic we know that $K_i \cap K_j = \emptyset$ for $i \neq j$. Therefore $K_1, \ldots, K_m$ and $K_{m+1} = \Sigma^* \setminus (\bigcup_{1 \leq i \leq m} K_i)$ form a partition of Σ^*. All languages K_i are necessarily regular, since the skeleton can be computed by a finite automaton.

We state the following lemma, which can be seen as generalization of Proposition 9.

Lemma 10. *Let $K_1, \ldots, K_k$ be regular languages forming a partition of Σ^*. Two languages $L_1, L_2 \subseteq \Sigma^*$ are regular separable if, and only if, $L_1 \cap K_i$ and $L_2 \cap K_i$ are regular separable for all $i \in \{1, \ldots, k\}$.*

Proof. The "only if" direction is trivial, since every language separating L_1 and L_2 separates $L_1 \cap K_i$ and $L_2 \cap K_i$ as well. For the opposite direction, we observe that if for every i the languages $L_1 \cap K_i$ and $L_2 \cap K_i$ are separable by a regular language S_i, then L_1 and L_2 are separable by the regular language $S = \bigcup_{1 \leq i \leq k} (S_i \cap K_i)$. $\square$

Therefore, it only remains to decide regular separability for the languages $L(\mathcal{A}, U_1) \cap K_i$ and $L(\mathcal{A}, U_2) \cap K_i$.

Step 3: Reduction to unary separability in $\mathcal{C}$. Fix a skeleton ρ_i. Let $c_1, \ldots, c_m$ be all the simple cycles in the automaton $\mathcal{A}$ which visit only states visited by ρ_i. Since a cycle cannot visit the same state twice (except the initial state), it has length at most n, and thus the number of simple cycles is $m \leq d^n$, where d is the number of transitions of the automaton. Notice that any run ρ with $\textsc{skel}(\rho) = \rho_i$ decomposes into the skeleton ρ_i and a bunch of simple cycles from $\{c_1, \ldots, c_m\}$. Let $T = \{t_1, \ldots, t_d\}$, thus $\rho \in T^*$. Let $\mu : \mathbb{N}^m \to \mathbb{N}^d$ be the affine function that transforms counting cycles into counting transitions, which is defined as

$$\mu(x_1, \ldots, x_m) = \Pi(\rho_i) + \sum_{1 \leq i \leq m} \Pi(c_i) \cdot x_i.$$

(Notice that the function above is affine, and not linear, since it requires to take into account the initial cost of the skeleton $\Pi(\rho_i)$.) In other words, $\mu(x_1, \ldots, x_m)$ returns Parikh image of a run which decomposes into the skeleton ρ_i and x_i cycles c_i, for every i. Let $V_1 = \mu^{-1}(U_1)$ and $V_2 = \mu^{-1}(U_2)$ be the corresponding sets counting cycles instead of transitions. Since $\mathcal{C}$ is closed under the inverse image of affine mappings, $V_1, V_2 \in \mathcal{C}$.

Lemma 11. *The following two conditions are equivalent:*

1. *The two languages $L(\mathcal{A}, U_1) \cap K_i, L(\mathcal{A}, U_2) \cap K_i \subseteq \Sigma^*$ are regular separable.*
2. *The two sets of vectors $V_1, V_2 \subseteq \mathbb{N}^m$ are unary separable.*

Proof. For the implication 1) $\Rightarrow$ 2), suppose R is a regular language separating $L(\mathcal{A}, U_1) \cap K_i$ and $L(\mathcal{A}, U_2) \cap K_i$. Fix $\omega \in \mathbb{N}$ such that for all words $x, y, z \in \Sigma^*$,

$$xy^{\omega}z \in R \iff xy^{2\omega}z \in R. \tag{1}$$

It is easy to see that for every regular language R such ω exists. The simplest way of showing this is to consider the syntactic monoid M of R and to let ω be its idempotent power, i.e., a number such that $m^{\omega} = (m^{\omega})^2$ for every $m \in M$.

Recall n-unary equivalence: $u \equiv_n v$ if for every coordinate $1 \leq i \leq m$ we have $u[i] \equiv v[i] \mod n$ and moreover $u[i] \leq n \iff v[i] \leq n$. It is enough to show that for all $v_1 \in V_1, v_2 \in V_2$ it holds $v_1 \not\equiv_{\omega} v_2$. Indeed, if this is the case, the unary set $S = \{v \in \mathbb{N}^m \mid \exists_{v_1 \in V_1} v \equiv_{\omega} v_1\}$ separates V_1 and V_2.

Suppose, towards a contradiction, that there are some $v_1 \in V_1, v_2 \in V_2$ such that $v_1 \equiv_{\omega} v_2$. Recall that $c_1, \ldots, c_m$ are all the simple cycles in automaton $\mathcal{A}$ visiting only states visited by the skeleton ρ_i. For every cycle c_j, let's arbitrarily choose a state on it, and let's call it the *fixing state* of c_j. Let $w_1, \ldots, w_m$ be words labeling the cycles $c_1, \ldots, c_m$, resp., when reading from its fixing state, and let w be the word labeling skeleton ρ_i. Consider a partition $w = s_0 \ldots s_k$ and let q_d, for $d \in \{0, \ldots, k-1\}$, be the state, which is reached in $\mathcal{A}$ after reading $s_0 \ldots s_d$. This partition of w is chosen such that among q_d are all fixing states of cycles $c_1, \ldots, c_m$, every one exactly ones. For every $v \in \mathbb{N}^m$ we define a *canonical word* w_v for v as the word obtained from pasting into w, in places between some s_d and s_{d+1}, words $w_1^{v[1]}, \ldots, w_m^{v[m]}$ in such a way that every $w_j^{v[j]}$ is pasted into the place where its fixing state equals q_d and words pasted into the same place are sorted according to indices of the corresponding cycles.

Notice an important fact: if $v \in V_1$ then $w_v \in L(\mathcal{A}, U_1) \cap K_i$, and likewise for V_2. Consider words w_{v_1} and w_{v_2}. One can see that by repeated application of equation (1) we can obtain that $w_{v_1} \in R \iff w_{v_2} \in R$. But R was supposed to separate $L(\mathcal{A}, U_1) \cap K_i$ and $L(\mathcal{A}, U_2) \cap K_i$, a contradiction.

For proving the implication 2) $\Rightarrow$ 1), suppose that a unary set S separates V_1 and V_2. We claim that the language $R = L(\mathcal{A}, \mu(S)) \cap K_i$ is regular and separates $L(\mathcal{A}, U_1) \cap K_i$ and $L(\mathcal{A}, U_2) \cap K_i$.

We first verify that R separates the languages. Clearly, $U_1 \subseteq \mu(V_1) \subseteq \mu(S)$, so $L(\mathcal{A}, U_1) \cap K_i \subseteq L(\mathcal{A}, \mu(S)) \cap K_i = R$. The disjointness of $L(\mathcal{A}, U_2) \cap K_i$ and R is shown by contradiction. Suppose that there is a word $w \in K_i$ belonging both to $L(\mathcal{A}, \mu(S))$ and to $L(\mathcal{A}, U_2)$, let ρ we run of $\mathcal{A}$ over w and let $v = \Pi(\rho)$. We have $v \in \mu(S) \cap U_2$, which implies $v = \mu(s)$ for some $s \in S \cap \mu^{-1}(U_2) = S \cap V_2$. In consequence $S \cap V_2$ is nonempty, thus contradicting the assumption that S separates V_1 and V_2.

In order to prove that R is regular it suffices to prove that $L(\mathcal{A}, \mu(S))$ is regular. The finite nondeterministic automaton recognizing this language simulates a run $\rho = t_{i_1} \ldots t_{i_\ell}$ of $\mathcal{A}$, and accepts when $\Pi(\rho) \in \mu(S)$. Since S is unary, the automaton can evaluate this condition using finite memory. For every cycle c_j, the automaton would store a vector $x_j < \Pi(c_j)$, and a number n_j up to the unary equivalence $\equiv_n$, with the following meaning: the vector $\Pi(c_i)$ has been already executed n_j times, and x_i is the current "remainder". Additionally, the automaton stores a vector $x \leq \Pi(\rho_i)$ which is counting those transitions on the skeleton which have not been counted as cycles. At every input letter the automaton guesses nondeterministically one of cycles c_i or the skeleton and updates x_j, n_j and x accordingly. The automaton accepts when $x = \Pi(\rho_i)$, $x_j = 0$ for all j, and $(n_1, \ldots, n_m) \in S$. $\square$

6 Applications

We now derive two direct corollaries of Theorem 2. In this section by a *projection* we mean a function $\pi_{k,I} : \mathbb{N}^k \to \mathbb{N}^{|I|}$, for $I \subseteq \{1 \ldots k\}$, that drops coordinates not in I. We start with a simple but useful lemma:

Lemma 12. *If a class $\mathcal{C} \subseteq \bigcup_{d \in \mathbb{N}} \mathcal{P}(\mathbb{N}^d)$ contains all semilinear sets and is effectively closed under intersections, projections, and inverse images of projections, then it is effectively closed under inverse images of affine maps.*

Proof. Let S be a set in $\mathcal{C}$ and $f : \mathbb{N}^k \to \mathbb{N}^\ell$ be an affine map defined by $f(u) = Mu + v$ for $M = (m_{i,j})$ a matrix of dimension $\ell \times k$ and v a vector of dimension ℓ. Let $e_j \in \mathbb{N}^k$ be the vector s.t. $e_j[j] = 1$ and 0 otherwise, and let $m_j = (m_{1,j}, m_{2,j}, \ldots, m_{\ell,j})$ be the (transpose of) the j-th column of M. First remark that the set

$$E_1 = \{(x, f(x)) \mid x \in \mathbb{N}^k\} \subseteq \mathbb{N}^{k+\ell}$$

is linear with base $(0^k, v)$ and periods $\{p_1, \ldots, p_k\}$, where $p_j = (e_j, m_j) \in \mathbb{N}^{k+\ell}$. Thus, $E_1 \in \mathcal{C}$. Therefore the set $E_2 = E_1 \cap \pi^{-1}_{k+\ell,I}(S)$ is also in $\mathcal{C}$, for $I = \{k+1, \ldots, k+\ell\}$. Finally, we conclude since $\pi_{k+\ell,J}(E_2) = f^{-1}(S)$ with $J = \{1, \ldots, k\}$. $\square$

Corollary 13. *The regular separability problem is decidable for nondeterministic Parikh automata.*

Proof. In order to apply Theorem 2 for $\mathcal{C}$ being semilinear sets, we need to know that the class of semilinear sets is robust. First, Lemma 12 yields effective closure under inverse images of affine maps, as semilinear sets are effectively closed under boolean combinations, images, and inverse images of projections. Second, decidability of the unary separability problem for semilinear sets is a corollary of the main result in [3]. This theorem states that separability of rational relations in $\Sigma^* \times \mathbb{N}^m$ by recognizable relations is decidable. If we ignore the Σ^* component we get the same result for rational and recognizable relations in $\mathbb{N}^m$, which are exactly semilinear sets and unary sets, respectively. $\square$

For the second corollary we have to introduce vector addition systems (VASes) and sections thereof.

A d-dimensional *vector addition system* (VAS) is a pair $V = (s, T)$, where $s \in \mathbb{N}^d$ is a *source* configuration and $T \subseteq_{\text{FIN}} \mathbb{Z}^d$ is a finite set of *transitions*. A *run* of a VAS $V = (s, T)$ is a sequence

$$(v_0, t_0, v_1), (v_1, t_1, v_2), \ldots, (v_{n-1}, t_{n-1}, v_n) \in \mathbb{N}^d \times T \times \mathbb{N}^d$$

such that for all $i \in \{0, \ldots, n-1\}$ we have $v_i + t_i = v_{i+1}$ and $v_0 = s$. The *target* of this run is the configuration v_n. The *reachability set* of a VAS V is the set of targets of all its runs.

In order to ensure robustness, we slightly enlarge the family of VAS reachability sets to *sections* thereof. The intuition about a section is that we fix values on a subset of coordinates in vectors, and collect all the values that can occur on the other coordinates. For a subset $I \subseteq \{1, \ldots, d\}$, the projection $\pi_{d,I}$ extends element-wise to sets of vectors $S \subseteq \mathbb{N}^d$, denoted $\pi_{d,I}(S)$. For a vector $u \in \mathbb{N}^{d-|I|}$, the *section* of S w.r.t. I and u is the set

$$\pi_{d,I}(\{v \in S \mid \pi_{d,\{1,\ldots,d\}\setminus I}(v) = u\}) \subseteq \mathbb{N}^{|I|} \ .$$

We denote by $\mathcal{C}_{\text{SEC-VAS}}$ the family of all sections of VAS reachability sets.

Corollary 14. *The regular separability problem is decidable for nondeterministic $\mathcal{C}_{\text{SEC-VAS}}$-Parikh automata.*

Proof. We apply Theorem 2 for $\mathcal{C} = \mathcal{C}_{\text{SEC-VAS}}$; we thus need to show that class $\mathcal{C}_{\text{SEC-VAS}}$ is robust. Decidability of unary separability of sets from $\mathcal{C}_{\text{SEC-VAS}}$ is shown in Theorem 9 in [4]. Effective closure of $\mathcal{C}$ under inverse images of affine functions will follow by Lemma 12 once we prove all its assumptions.

First, $\mathcal{C}_{\text{SEC-VAS}}$ contains all semilinear sets. Effective closure under intersections is shown in Proposition 7 in [4]. Effective closure under inverse images of projections is easy: extend the VAS with additional coordinates, and allow it to arbitrarily increase these coordinates.

Finally, to see that $\mathcal{C}_{\text{SEC-VAS}}$ is effectively closed under projections consider a section $S \subseteq \mathbb{N}^d$ of the reachability set of a VAS V, and a subset of coordinates $I \subseteq \{1, \ldots, d\}$. We construct a VAS V' which is like V, but additionally allows to decrease every coordinate from $\{1, \ldots, d\} \setminus I$. Projection $\pi_{d,I}(S)$ of S onto I is a section of the reachability set of V' defined similarly as S, but with an additional requirement that all coordinates from $\{1, \ldots, d\} \setminus I$ have value 0. $\square$

References

1. Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. On the expressiveness of parikh automata and related models. In *3rd Workshop on Non-Classical Models for Automata and Applications - NCMA'11*, pages 103–119, 2011.
2. Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Unambiguous constrained automata. *Int. J. Found. Comput. Sci.*, 24(7):1099–1116, 2013.

3. Christian Choffrut and Serge Grigorieff. Separability of rational relations in $A^* \times \mathbb{N}^m$ by recognizable relations is decidable. *Inf. Process. Lett.*, 99(1):27–32, 2006.
4. Lorenzo Clemente, Wojciech Czerwiński, Sławomir Lasota, and Charles Paperman. Separability of reachability sets of vector addition systems. *CoRR*, abs/1609.00214, 2016.
5. Wojciech Czerwiński, Wim Martens, and Tomás Masopust. Efficient separability of regular languages by subsequences and suffixes. In *ICALP'13*, pages 150–161, 2013.
6. Wojciech Czerwiński, Wim Martens, Lorijn van Rooijen, and Marc Zeitoun. A note on decidable separability by piecewise testable languages. In *FCT'15*, pages 173–185, 2015.
7. Jean Goubault-Larrecq and Sylvain Schmitz. Deciding piecewise testable separability for regular tree languages. In *ICALP'16*, pages 97:1–97:15, 2016.
8. Harry B. Hunt III. On the decidability of grammar problems. *Journal of the ACM*, 29(2):429–447, 1982.
9. Oscar H. Ibarra. Reversal-bounded multicounter machines and their decision problems. *J. ACM*, 25(1):116–133, 1978.
10. Felix Klaedtke and Harald Rueß. Monadic second-order logics with cardinalities. In *ICALP'03*, pages 681–696, 2003.
11. Eryk Kopczyński. Invisible pushdown languages. *CoRR*, abs/1511.00289, 2015.
12. Thomas Place, Lorijn van Rooijen, and Marc Zeitoun. Separating regular languages by locally testable and locally threshold testable languages. In *FSTTCS'13*, pages 363–375, 2013.
13. Thomas Place, Lorijn van Rooijen, and Marc Zeitoun. Separating regular languages by piecewise testable and unambiguous languages. In *MFCS'13*, pages 729–740, 2013.
14. Thomas Place and Marc Zeitoun. Going higher in the first-order quantifier alternation hierarchy on words. In *ICALP'14*, pages 342–353, 2014.
15. Thomas Place and Marc Zeitoun. Separating regular languages with first-order logic. *Logical Methods in Computer Science*, 12(1), 2016.
16. Thomas G. Szymanski and John H. Williams. Noncanonical extensions of bottom-up parsing techniques. *SIAM Journal on Computing*, 5(2):231–250, 1976.