A Note on Decidable Separability by Piecewise Testable Languages

Wojciech Czerwiński [1], Wim Martens [2],
Lorijn van Rooijen [3], and Marc Zeitoun [3]

[1] University of Warsaw
[2] University of Bayreuth
[3] Bordeaux University

Abstract. The separability problem for languages from a class $\mathcal{C}$ by languages of a class $\mathcal{S}$ asks, for two given word languages I and E from $\mathcal{C}$, whether there exists a language S from $\mathcal{S}$ which includes I and excludes E, that is, $I \subseteq S$ and $S \cap E = \emptyset$. It is known that separability for context-free languages by any class containing all definite languages (such as regular languages) is undecidable. We show that separability of context-free languages by piecewise testable languages is decidable. This contrasts the fact that testing if a context-free language is piecewise testable is undecidable. We generalize this decidability result by showing that, for every full trio (a class of languages that is closed under rather weak operations) which has decidable diagonal problem, separability with respect to piecewise testable languages is decidable. Examples of such classes are the class of languages defined by labeled vector addition systems and the class of languages accepted by higher order pushdown automata of order two. The proof goes through a result which is of independent interest and shows that, for *any* kind of languages I and E, separability can be decided by testing the existence of common patterns in I and E.

1 Introduction

We say that language I can be *separated* from E by a language S if S includes I and excludes E, that is, $I \subseteq S$ and $S \cap E = \emptyset$. In this case, we call S a *separator*. We study the *separability problem* of classes $\mathcal{C}$ by classes $\mathcal{S}$:

Given: Two languages I and E from a class $\mathcal{C}$.
Question: Can I and E be separated by some language from $\mathcal{S}$?

Separability is a classical problem in mathematics and computer science that recently found much new interest. For example, recent work investigated the separability problem of regular languages by piecewise testable languages [11,31], locally testable and locally threshold testable languages [30] or by first order definable languages [33]. Another recent example which uses separation and goes beyond regularity, is the proof of Leroux [21] for the decidability of reachability for vector addition systems or petri nets. This proof simplifies earlier proofs by Mayr [24] and Kosaraju [19].

In this paper we focus on the theoretical underpinnings of separating languages by piecewise testable languages. Our interest in piecewise testable languages is mainly because of the following two reasons. First, it was shown recently [11, 31] that separability of regular languages (given by their non-deterministic automata) by piecewise testable languages is in PTIME. We found the tractability of this problem to be rather surprising.

Second, piecewise testable languages are a very natural class of languages in the sense that they only reason about the *order* of symbols. More precisely, they are finite Boolean combinations of regular languages of the form $A^*a_1A^*a_2A^* \cdots A^*a_nA^*$ in which $a_i \in A$ for every $i = 1, \ldots, n$ [35]. We are investigating to which extent piecewise testable languages and fragments thereof can be used for computing simple explanations for the behavior of complex systems [17].

Separation and Characterization. For classes $\mathcal{C}$ effectively closed under complement, separation of $\mathcal{C}$ by $\mathcal{S}$ is a natural generalization of the *characterization problem* for $\mathcal{C}$ by $\mathcal{S}$, which is defined as follows. For a given language $L \in \mathcal{C}$ decide whether L is in $\mathcal{S}$. Indeed, L is in $\mathcal{S}$ if and only if L can be separated from its complement by a language from $\mathcal{S}$. The characterization problem is well studied. The starting points were famous works of Schützenberger [34] and Simon [35], who solved the characterization problem for the regular languages by the first-order definable languages and piecewise testable languages, respectively. There were many more results showing that, for regular languages and a subclass thereof (often characterised by a given logic), the problem is decidable, see for example [4, 27, 29, 32, 37, 39]. Similar problems have been considered for trees [3, 5, 7, 8].

Decidability. To the best of our knowledge, all the above work and in general all the decidable characterizations were obtained in cases where $\mathcal{C}$ is the class of regular languages, or a subclass of it. This could be due to several negative results which may seem to form a barrier for any non-trivial decidability beyond regular languages. For a context-free language (given by a grammar or a pushdown automaton) it is undecidable to determine whether it is a regular language, by Greibach's theorem [16]. Furthermore, it is also undecidable to determine whether a given context-free language is piecewise testable.[4]

Concerning context-free languages, there is a strong connection between the intersection emptiness problem and separability. Trivially, testing intersection emptiness of two given context-free languages is the same as deciding if they can be separated by some context-free language. However, in general, the negative result is even more overwhelming. Szymanski and Williams [38] proved that separability of context-free languages by regular languages is undecidable. This was then generalized by Hunt [18], who proved that separability of context-free languages by any class containing all the *definite languages* is undecidable. A language L is *definite* if it can be written as $L = F_1A^* \cup F_2$, where F_1 and F_2 are finite languages over alphabet A. As such, for definite languages, it can be

[4] This is not an immediate consequence of Greibach's theorem as far as we know but can be proved similarly.

decided whether a given word w belongs to L by looking at the prefix of w of a given fixed length. (The same statement holds for *reverse definite* languages, in which we are looking at suffixes.) Containing all the definite, or reverse definite, languages is a very weak condition. Note that if a logic can test what is the i-th letter of a word and is closed under boolean combinations, it already can define all the definite languages. In his paper, Hunt makes an explicit link between intersection emptiness and separability. Hunt says: *"We show that separability is undecidable in general for the same reason that the emptiness-of-intersection problem is undecidable. Therefore, it is unlikely that separability can be used to circumvent the undecidability of the emptiness-of-intersection problem."*

Our Contribution. In this paper we show that the above mentioned quote does not apply for separability by piecewise testable languages (PTLs): we show that it can be decided whether two given context-free languages are separable by a PTL. This may come as a surprise in the light of the undecidability results we already discussed.

In fact, we prove a stronger result that implies that separability by PTLs is also decidable for some rather expressive classes such as Petri net languages (or, alternatively, labeled vector addition system languages). This result is a reduction from separability by PTLs to a problem that we call *diagonal problem*. The proof of the reduction consists of two parts. First we show that (arbitrary) languages I and E are not separable by PTL if and only if they possess a certain *common pattern*. Then, we use this fact to reduce to the diagonal problem.

A curiosity of this work is perhaps the absence of algebraic methods. Most decidability results we are aware of have considered syntactic monoids of regular languages and investigated properties thereof. The exceptions are the recent studies of separability of regular languages by piecewise testable languages (e.g., [11,31]). However, since the algebraic framework for regular languages is so rich, some may not find it clear whether [11,31] do or do not rely on algebraic methods; perhaps simply in a rephrased way. Here, the situation is different in the sense that for context-free languages the syntactic monoid is infinite and it is difficult to design any algebraic framework for them. So the work shows that it is not always necessary to use algebraic techniques to prove separability questions.

2 Preliminaries

The set of all integers and nonnegative integers are denoted by $\mathbb{Z}$ and $\mathbb{N}$ respectively. A *word* is a concatenation $w = a_1 \cdots a_n$ of symbols a_i that come from a finite alphabet A. The *length* of w is n, the number of its symbols. The *alphabet of* w is the set $\{a_1, \ldots, a_n\}$ and is denoted $\mathsf{alph}(w)$. For a subalphabet $B \subseteq A$, a word $v \in A^*$ is a B-*subsequence* of w, denoted $v \preceq_B w$, if $v = b_1 \cdots b_m$ and $w \in B^* b_1 B^* \cdots B^* b_m B^*$. (We do not require that $\{b_1, \ldots, b_m\} \subseteq B$ or $B \subseteq \{b_1, \ldots, b_m\}$.) We refer to the relation $\preceq_A$ as the *subsequence* relation and denote it by $\preceq$. A regular word language over alphabet A is a *piece language* if it is of the form $A^* a_1 A^* \cdots A^* a_n A^*$ for some $a_1, \ldots, a_n \in A$, that is, it is the set of words which have $a_1 \cdots a_n$ as a subsequence. A regular language is a *piecewise*

testable language if it is a (finite) boolean combination of piece languages. The class of all piecewise testable languages is denoted PTL.

Separability and Common Patterns. The first main result of the paper proves that two (not necessarily regular) languages are not separable by PTL if and only if they have a common subpattern. We now make this more precise.

A *factorization pattern* is an element of $(A^*)^{p+1} \times (2^A \setminus \emptyset)^p$ for some $p \geq 0$. In other terms, if $(\overrightarrow{u}, \overrightarrow{B})$ is such a factorization pattern, there exist words $u_0, \ldots, u_p \in A^*$ and non-empty alphabets $B_1, \ldots, B_p \subseteq A$ such that $\overrightarrow{u} = (u_0, \ldots, u_p)$ and $\overrightarrow{B} = (B_1, \ldots, B_p)$. For $B \subseteq A$, we denote by $B^{\circledast}$ the set of words with alphabet exactly B, that is, $B^{\circledast} = \{w \in B^* \mid \mathsf{alph}(w) = B\}$. Given a factorization pattern $(\overrightarrow{u}, \overrightarrow{B})$, with $\overrightarrow{u} = (u_0, \ldots, u_p)$ and $\overrightarrow{B} = (B_1, \ldots, B_p)$, let

$$\mathcal{L}(\overrightarrow{u}, \overrightarrow{B}, n) = u_0(B_1^{\circledast})^n u_1 \cdots u_{p-1}(B_p^{\circledast})^n u_p.$$

In other terms, in a word of $\mathcal{L}(\overrightarrow{u}, \overrightarrow{B}, n)$, the infix between u_{k-1} and u_k is required to be the concatenation of n words, each containing *all* letters of B_k (for each $1 \leq k \leq p$). A sequence $(w_n)_n$ is said $(\overrightarrow{u}, \overrightarrow{B})$-*adequate* if

$$\forall n \in \mathbb{N},\ w_n \in \mathcal{L}(\overrightarrow{u}, \overrightarrow{B}, n).$$

Finally, language L *contains the pattern* $(\overrightarrow{u}, \overrightarrow{B})$ if there exists an infinite sequence of words $(w_n)_n$ in L that is $(\overrightarrow{u}, \overrightarrow{B})$-adequate. We prove the following Theorem in Section 3:

Theorem 1. *Two word languages I and E are not separable by* PTL *if and only if they contain a common pattern $(\overrightarrow{u}, \overrightarrow{B})$.*

A Characterization for Decidable Separability. The second main result is an algorithm that decides separability for *full trios* that have a decidable *diagonal* problem. Full trios, also called *cones*, are language classes that are closed under rather weak operations.

Fix a language L over alphabet A. For an alphabet B, the *B-projection* of a word is its longest subsequence consisting of symbols from B. The *B-projection* of a language L is the set of all B-projections of words belonging to L. Therefore, the B-projection of L is a language over alphabet $A \cap B$. The *B-upward closure* of a language L is the set of all words which have a B-subsequence in L, i.e.,

$$\{w \in (A \cup B)^* \mid \exists v \in L \text{ such that } v \preceq_B w\}.$$

In other words, the B-upward closure of L consists of all words that can be obtained from taking a word in L and padding it with symbols from B.

A class of languages $\mathcal{C}$ is *closed* under an operation OP if $L \in \mathcal{C}$ implies that $\text{OP}(L) \in \mathcal{C}$. We use term *effectively closed* if, furthermore, the representation of $\text{OP}(L)$ can be effectively computed from the representation of L.

A class $\mathcal{C}$ of languages is a *full trio* if it is effectively closed under:

1. B-projection for every finite alphabet B,
2. B-upward closure for every finite alphabet B, and
3. intersection with regular languages.

We note that full trios are usually defined differently (through closures under homomorphisms or rational transductions [6,13]) but we use the above mentioned properties in the proofs.

The problem that we will require to be decidable is the *diagonal problem*, which we explain next. Let $A = \{a_1, \ldots, a_n\}$. For a symbol $a \in A$ and a word $w \in A^*$, let $\#_a(w)$ denote the number of occurrences of a in w. The *Parikh image* of a word w is the n-tuple $(\#_{a_1}(w), \ldots, \#_{a_n}(w))$. The *Parikh image* of a language L is the set of all Parikh images of words from L. A tuple $(m_1, \ldots, m_n) \in \mathbb{N}^n$ is *dominated* by a tuple $(d_1, \ldots, d_n) \in \mathbb{N}^n$ if $d_i \geq m_i$ for every $i = 1, \ldots, n$. The *diagonal problem* for language L asks whether there exist infinitely many $m \in \mathbb{N}$ such that the tuple $(m, \ldots, m)$ is dominated by some tuple in the Parikh image of L. We are now ready to state the second main result:

Theorem 2. *For each full trio $\mathcal{C}$, the diagonal problem for $\mathcal{C}$ is decidable if and only if separability of $\mathcal{C}$ by PTL is decidable.*

In Section 4 we present an algorithm to decide separability for full trios that have a decidable diagonal problem, showing one direction of the equivalence. The algorithm does not rely on semilinearity of Parikh images. For example, in Section 5 we apply the lemma to Vector Addition System languages, which do not have a semilinear Parikh image. Very recently, we were informed by Georg Zetzsche [**?**] that the other implication also holds and that, therefore, there actually is an equivalence.

3 Common Patterns

In this section we prove Theorem 1. We say that a sequence is *adequate* if it is $(\overrightarrow{u}, \overrightarrow{B})$-adequate for some factorization pattern.

Lemma 3. *Every sequence $(w_n)_n$ of words admits an adequate subsequence.*

For a word w, denote its first (resp., last) letter by $\text{first}(w)$, (resp., $\text{last}(w)$). We call a factorization pattern $(\overrightarrow{u}, \overrightarrow{B})$ *proper* if (i) for all i, $\text{last}(u_i) \notin B_{i+1}$ and $\text{first}(u_i) \notin B_i$, and (ii) for all i, $u_i = \varepsilon \Rightarrow \big(B_i \not\subseteq B_{i+1}$ and $B_{i+1} \not\subseteq B_i\big)$.

Note that if a sequence $(w_n)_n$ is adequate, then there exists a proper factorization pattern $(\overrightarrow{u}, \overrightarrow{B})$ such that $(w_n)_n$ is $(\overrightarrow{u}, \overrightarrow{B})$-adequate. This is easily seen from the following observations and their symmetric counterparts:

$$u = a_1 \cdots a_k \text{ and } a_k \in B \;\Rightarrow\; a_1 \cdots a_k (B^{\circledast})^n \subseteq a_1 \cdots a_{k-1} (B^{\circledast})^n,$$
$$B_{i-1} \subseteq B_i \;\Rightarrow\; ({B_{i-1}}^{\circledast})^n ({B_i}^{\circledast})^n \subseteq ({B_i}^{\circledast})^n.$$

The following lemma gives a condition under which two sequences share a factorization pattern and is very similar to [2, Theorem 8.2.6]. In its statement, we write $v \sim_n w$ for two words v and w if they have the same subsequences up to length n, that is, for every word u of length at most n, $u \preceq v$ iff $u \preceq w$.

Lemma 4. *Let $(\overrightarrow{u}, \overrightarrow{B})$ and $(\overrightarrow{t}, \overrightarrow{C})$ be proper factorization patterns. Let $(v_n)_n$ and $(w_n)_n$ be two sequences of words such that*

- *$(v_n)_n$ is $(\overrightarrow{u}, \overrightarrow{B})$-adequate*
- *$(w_n)_n$ is $(\overrightarrow{t}, \overrightarrow{C})$-adequate*
- *$v_n \sim_n w_n$ for every $n \geq 0$.*

Then, $\overrightarrow{u} = \overrightarrow{t}$ *and* $\overrightarrow{B} = \overrightarrow{C}$.

Now we are equipped to prove Theorem 1. We only show the "only if" direction here, due to space restrictions.

Proof (of Theorem 1, "only-if"). By hypothesis, for every $n \in \mathbb{N}$, there exist $v_n \in I$ and $w_n \in E$ such that

$$v_n \sim_n w_n. \tag{1}$$

This defines an infinite sequence of pairs $(v_n, w_n)_n$, from which we will iteratively extract infinite subsequences to obtain additional properties, while keeping (1).

By Lemma 4, one can extract from $(v_n, w_n)_n$ a subsequence whose first component forms an adequate sequence. From this subsequence of pairs, using Lemma 4 again, we extract a subsequence whose second component is also adequate (note that the first component remains adequate). Therefore, one can assume that both $(v_n)_n$ and $(w_n)_n$ are themselves adequate. This means there exist proper factorization patterns for which $(v_n)_n$ resp. $(w_n)_n$ are adequate. Lemma 5 shows that one can choose the *same* proper factorization pattern $(\overrightarrow{u}, \overrightarrow{B})$ such that both $(v_n)_n$ and $(w_n)_n$ are $(\overrightarrow{u}, \overrightarrow{B})$-adequate. This means that I and E contain a common pattern $(\overrightarrow{u}, \overrightarrow{B})$. □

4 The Algorithm for Separability

We prove one direction of Theorem 3 by showing that, for full trios with decidable diagonal problem, we can decide separability by PTL. Fix two languages I and E from a full trio $\mathcal{C}$ which has decidable diagonal problem.

To test whether I is separable from E by a piecewise testable language S, we run two semi-procedures in parallel. The *positive* one looks for a witness that I and E are separable by PTL, whereas the *negative* one looks for a witness that they are *not* separable by a PTL. Since one of the semi-procedures always terminates, we have an effective algorithm that decides separability. It remains to describe the two semi-procedures.

Positive semi-procedure. We first note that, when a full trio has decidable diagonal problem, it also has decidable emptiness.[5] The positive semi-procedure enumerates all PTLs over the union of the alphabets of I and E. For every PTL S it checks whether S is a separator, so if $I \subseteq S$ and $E \cap S = \emptyset$. The first test is equivalent to $I \cap (A^* \setminus S) = \emptyset$. Thus both tests boil down to checking whether the intersection of a language from the class $\mathcal{C}$ (I or E, respectively) and a regular language (S and $A^* \setminus S$, respectively) is empty. This is decidable, as $\mathcal{C}$ is effectively closed under taking intersections with regular languages and has decidable emptiness.

[5] Emptiness of L over alphabet A can be decided by taking the $\{x\}$-upward closure of L, where $x \notin A$, intersecting the resulting language with the regular language $(A \cup \{x\})^* A (A \cup \{x\})^*$, and then taking the $\{x\}$-projection. In the resulting language, the diagonal problem returns true iff L is non-empty.

Negative semi-procedure. Theorem 1 shows that there is always a finite witness for inseparability: a pattern $(\overrightarrow{u}, \overrightarrow{B})$. The negative semi-procedure enumerates all possible patterns and for every one checks the condition of Theorem 1. We now show how to test this condition, i.e., for a pattern $(\overrightarrow{u}, \overrightarrow{B})$ test whether for all $n \in \mathbb{N}$ the intersection of $\mathcal{L}(\overrightarrow{u}, \overrightarrow{B}, n)$ with both I and E is nonempty.

Checking the condition. Here we show for an arbitrary language from $\mathcal{C}$ how to check whether for all $n \in \mathbb{N}$ its intersection with the language $\mathcal{L}(\overrightarrow{u}, \overrightarrow{B}, n)$ is nonempty. Fix $L \in \mathcal{C}$ over an alphabet A and a pattern $(\overrightarrow{u}, \overrightarrow{B})$, where $\overrightarrow{u} = (u_0, \ldots, u_k)$ and $\overrightarrow{B} = (B_1, \ldots, B_k)$. Intuitively, we just consider a diagonal problem with some artifacts: we are counting the number of occurrences of alphabets B_i and checking whether those numbers can simultaneously become arbitrarily big.

We show decidability of the non-separability problem by a formal reduction to the diagonal problem. We perform a sequence of steps. In every step we will slightly modify the considered language L and appropriately customize the condition to be checked. Using the closure properties of the class $\mathcal{C}$ we will assure that the investigated language still belongs to $\mathcal{C}$.

First we add special letters $\$_i$, for $i \in \{1, \ldots, k\}$, which do not occur in A. These letters are meant to count how many times alphabet B_i is occurring in the word. Then we will assure that words are of the form

$$u_0 \, (B_1 \cup \{\$_1\})^* \, u_1 \cdots u_{k-1} \, (B_k \cup \{\$_k\})^* \, u_k,$$

which already is close to what we need for the pattern. Then we will check that between every two letters $\$_i$ (with the same i), every letter from B_i occurs, so that the $\$_i$ are indeed counting the number of iterations through the entire alphabet B_i. Finally we will remove all the letters except those from $\{\$_1, \ldots, \$_k\}$. The resulting language will contain only words of the form $\$_1^*\$_2^* \cdots \$_k^*$ and the condition to be checked will be exactly the diagonal problem.

More formally, let $L_0 := L$. We consider a sequence of modifications starting from L_0, resulting in L_1, L_2, L_3, and L_4. Each of them will be in $\mathcal{C}$ and we describe them next.

Language L_1 is the $\{\$_1, \ldots, \$_k\}$-upward closure of L_0. The idea is that L_1 contains, in particular, all words where the $\$_i$ are placed "correctly", that is, in between two $\$_i$ letters the whole alphabet B_i should occur. However at this moment we do not check it. By the closure under B-upward closures, language L_1 belongs to $\mathcal{C}$.

Note that L_1 also contains words in which the $\$_i$ letters are placed totally arbitrary. In particular, they can occur in the wrong order. The idea behind L_2 is to consider only those words, in which the $\$_i$ were guessed at least in the good areas. Concretely, L_2 is an intersection of L_1 with the language

$$u_0 \, (B_1 \cup \{\$_1\})^* \, u_1 \cdots u_{k-1} \, (B_k \cup \{\$_k\})^* \, u_k.$$

By the closure under intersection with regular languages L_2 belongs to $\mathcal{C}$.

Language L_2 still may contain words, such that in between two $\$_i$ letters not all the letters from B_i occur, we get rid of these by intersecting L_2 with the regular language

$$u_0(\$_1 B_1^{\circledast})^*\$_1 u_1 \cdots u_{k-1}(\$_k B_k^{\circledast})^*\$_k u_k.$$

As such, we obtain L_3 which, again by closure under intersection with regular languages, belongs to $\mathcal{C}$.[6]

Note that intersection of $L = L_0$ with the language $\mathcal{L}(\overrightarrow{u}, \overrightarrow{B}, n)$ is nonempty if and only if L_3 contains a word with precisely $n+1$ letters $\$_i$ for every $i \in \{1, \ldots, k\}$. Indeed, L_3 just contains the (slightly modified versions of) words from L_0 which fit into the pattern and in which the letters $\$_i$ "count" occurrences of $B_i^{\circledast}$. Furthermore, for every word in L_3, the word obtained by removing some occurrences of some $\$_i$ is in L_3 as well. It is thus enough to focus on the $\$_i$ letters. Language L_4 is therefore the $\{\$_1, \ldots, \$_k\}$-projection of L_3. By the closure under B-projections, for every $B \subseteq A$, language L_4 belongs to $\mathcal{C}$.

The words contained in L_4 are therefore of the form

$$\$_1^{a_1} \cdots \$_k^{a_k},$$

such that there exists $w \in L$ with at least $a_i - 1$ occurrences of $B_i^{\circledast}$. Therefore intersection of L with $\mathcal{L}(\overrightarrow{u}, \overrightarrow{B}, n)$ is nonempty for all $n \geq 0$ if and only if the tuple $(n, \ldots, n)$ belongs to the Parikh image of L_4 for infinitely many $n \geq 0$. This is precisely the diagonal problem, which we know to be decidable for $\mathcal{C}$.

5 Decidable Classes

In this section we show that separability by piecewise testable languages is decidable for a wide range of classes, by proving that they meet the conditions of Theorem 3. In particular, we show this for context-free languages, languages of labeled vector addition systems (which are the same as languages of labeled Petri nets). We comment also on other natural classes of languages containing all the regular languages.

Theorem 5. *Separability by piecewise testable languages is decidable for*

1. *context-free languages; and for*
2. *languages of labeled vector addition systems.*

Our approach also allows to mix the above scenarios. That is, separability of a context-free language from the language of a labeled vector addition system is also decidable. In the remainder of this section, we prove the theorem.

Context-Free Languages. Context-free languages are well-known to be a full trio. The only nontrivial condition is deciding the diagonal problem. A set $S \subseteq \mathbb{N}^k$ is *linear* if it is of the form

$$S = \{v + n_1 v_1 + \ldots + n_m v_m \mid n_1, \ldots, n_m \in \mathbb{N}\}$$

for some *base* vector $v \in \mathbb{N}^k$ and *period* vectors $v_1, \ldots, v_m \in \mathbb{N}^k$. A *semilinear* set is a finite union of linear sets. Parikh's theorem [28] states that the Parikh image of a context-free language is semilinear, moreover the computation of its description as a (finite) union of linear sets if effective.[7] It is enough to check

[6] Of course, one could also immediately obtain L_3 from L_1 by performing a single intersection with a regular language.

[7] A simple proof of this fact can be found in [12].

whether for infinitely many $n \geq 0$ the mentioned semilinear set contains a tuple that dominates $(n, \ldots, n)$.

Semilinear sets are exactly these, which can be defined by Presburger logic. Moreover, the translation can be done effectively. Assume that $|A| = k$, so the Parikh image P of the considered language is a subset of $\mathbb{N}^k$ and ϕ is a Presburger formula describing P having exactly k free variables. Then

$$\psi = \forall_{n \in \mathbb{N}} \exists_{x_1, x_2, \ldots, x_k} \left(\bigwedge_{i \in \{1,\ldots,k\}} (x_i \geq n) \right) \wedge \phi(x_1, x_2, \ldots, x_k)$$

is true if and only if diagonal problem for the considered language is answered positively. Decidability of the Presburger logic finishes the proof of decidability of diagonal problem for context-free languages. We refer for the details of semilinear sets and Presburger logic to [14]. Finally, by Theorem 3, separability for context-free languages by piecewise testable languages is decidable.

Languages of Labeled Vector Addition Systems and Petri Nets. A k-dimensional *labeled vector addition system*, or *labeled VAS* $M = (A, T, \ell, s, t)$ over alphabet A consists of a set of *transitions* $T \subseteq \mathbb{Z}^k$, a labeling $\ell : T \to A \cup \{\varepsilon\}$, where ε stands for the empty word and *source* and *target* vectors $s, t \in \mathbb{N}^k$. A labeled VAS defines a transition relation on the set $\mathbb{N}^k$ of *markings*. For two markings $u, v \in \mathbb{N}^k$ we write $u \xrightarrow{a} v$ if there is $r \in T$ such that $u + r = v$ and $\ell(r) = a$, where the addition of vectors is defined as an addition on every coordinate. For two markings $u, v \in \mathbb{N}^k$ we say that u *reaches* v *via a word* w if there is a sequence of markings $u_0 = u, u_1, \ldots, u_{n-1}, u_n = v$ such that $u_i \xrightarrow{a_i} u_{i+1}$ for all $i \in \{0, \ldots, n-1\}$ and $w = a_0 \cdots a_{n-1}$. For a given labeled VAS M the *language* of M, denoted $L(M)$, is the set of all words $w \in A^*$ such that source reaches target via w. We note that languages of labeled VASs are the same as languages of labeled Petri nets.

Since labeled VAS languages are known to be a full trio [**?**], we only need to prove decidability of the diagonal problem. First we will show that it is enough to consider VASs in which the target marking equals $(0, \ldots, 0)$. To this end, let M be a k-dimensional labeled VAS with source vector $s = (s_1, \ldots, s_k)$ and target vector $t = (t_1, \ldots, t_k)$. We transform M to a new VAS M' in which we add two auxiliary coordinates, called *life* coordinates. The source coordinate is enriched by 0 on one life coordinate and by 1 on the other one, so it is $s' = (s_1, \ldots, s_k, 0, 1) \in \mathbb{N}^{k+2}$. Every original transition has two copies. One of these transitions subtracts one from the first life coordinate and adds one to the second life coordinate, the second transition does the opposite. Note that nonemptiness of life coordinates serve just as a necessary condition for firing any transition, as every transition subtracts one from one of these coordinates. Therefore, the original source marking s reaches the original target marking t via the same set of words by which the new source marking s' reaches either $(t_1, \ldots, t_k, 0, 1)$ or $(t_1, \ldots, t_k, 1, 0)$. We add also two *final* transitions, which subtract the original target vector, subtract one from one of the life coordinates and are labeled by ε. Note that, therefore, s can reach t by a word w in M if and only if s' can reach 0^{k+2} by w in M'. Indeed, the implication from left to right is immediate. On the other hand, in order to reach the marking 0^{k+2} in M', the last transition has to be the final transition, so implication from right to left also holds. Thus it is

enough to solve the diagonal problem for VASs in which the target marking is $(0, \ldots, 0)$.

We will show that this diagonal problem is decidable by a reduction to the place-boundedness problem for VASs with one zero test, which is decidable due to Bonnet et al. [9]. We modify the considered VAS in the following way. For every letter $a \in A$ we add a new *letter-coordinate*, which is counting how many times we read the letter a, that is, for every transition which is labeled by $a \in A$ we write 1 in the letter-coordinate corresponding to a and 0 in the letter-coordinates corresponding to other letters. The set of letter-coordinates computes the Parikh image of a word. We also add one new *minimum-coordinate* and a new transition, which subtracts one from all the letter-coordinates and adds one to the minimum-coordinate. It is easy to see that minimum-coordinate can maximally reach the minimum number from the Parikh image tuple. Additionally, for every letter-coordinate we add a transition, labeled by ε, which can decrease this coordinate by one. The diagonal problem for the original VAS is equivalent to the question whether for infinitely many $n \geq 0$ the source marking, enriched by zeros in the new coordinates, reaches a marking $(0, \ldots, 0, n)$, with zeros everywhere beside the minimum-coordinate with number n. This can be easily reduced to the place-boundedness for a VAS with one zero test. We do not show the details. Intuitively, the zero test checks whether there are zeros everywhere else than the minimum-coordinate and we check whether under this condition the minimum-coordinate can get unbounded. This finishes the proof of decidability of the diagonal problem for labeled VASs.

Other classes. Among another natural language classes extending regular languages one can think about context-sensitive languages or languages of process rewrite systems (defined by Mayr [25]), which are a common generalization of pushdown automata and VASs. Another interesting class is the class of languages defined by lossy counter machines [26].

Unfortunately context-sensitive languages do not meet the conditions of Theorem 3, as they are no full trio, nor is their emptiness problem decidable.

Very recently, Zetzsche [41] showed that *indexed languages* [1] or, equivalently, languages accepted by higher-order pushdown automata of order two [23] fulfill the conditions of Theorem 3 and therefore have decidable separability by PTL. His proof showing that indexed languages have a decidable diagonal problem is much more involved than the one for context-free languages we showed here. This shows that separability of languages definable by pushdown automata of order two by PTL is decidable as well. It would be interesting to know if it is decidable for pushdown automata for even higher order as well.

6 Concluding Remarks

Since the decidability results we presented seem to be in strong contrast with the remark of Hunt in the introduction, we briefly comment on this. What we essentially do is show that undecidable emptiness-of-intersection for a class $\mathcal{C}$ does not always imply undecidability for separability of $\mathcal{C}$ with respect to some non-trivial

class of languages. In the case of separability with respect to piecewise testable languages, the main reason is basically that we only need to construct intersections of languages from $\mathcal{C}$ with languages that are regular (or even piecewise or co-piecewise testable). Here, the fact that such intersections can be effectively constructed, together with decidable emptiness and diagonal problems seem to be sufficient for decidability.

Regarding future work, we see many interesting directions and new questions. Which language classes have a decidable diagonal problem? Which other characterizations are there for decidable separability by PTL? Can Theorem 3 be extended to also give complexity guarantees? Can we find similar characterizations for separability by subclasses of PTL, as considered in [17]?

Acknowledgments We would like to thank Tomáš Masopust for pointing us to [18] and Thomas Place for pointing out to us that determining if a given context-free language is piecewise testable is undecidable. We are also grateful to the anonymous reviewers for many helpful remarks that simplified proofs. We are much indebted to Georg Zetzsche for many useful remarks and most of all for sending us a simple proof that showed that, for full trios, separability by PTL implies decidability of the diagonal problem, thereby turning Theorem 3 into an equivalence. We plan to incorporate his proof in the full version of this paper.

References

1. A. V. Aho. Indexed grammars — an extension of context-free grammars. *J. ACM*, 15(4):647–671, 1968.
2. J. Almeida. *Finite semigroups and universal algebra*, volume 3 of *Series in Algebra*. World Scientific Publishing Co., 1994.
3. T. Antonopoulos, D. Hovland, W. Martens, and F. Neven. Deciding twig-definability of node selecting tree automata. In *ICDT*, pages 61–73, 2012.
4. M. Arfi. Polynomial operations on rational languages. In *STACS*, pages 198–206, 1987.
5. M. Benedikt and L. Segoufin. Regular tree languages definable in FO and in $\mathrm{FO}_{\mathrm{mod}}$. *ACM Trans. Comput. Logic*, 11(1):4:1–4:32, Nov. 2009.
6. J. Berstel. *Transductions and context-free languages*. Teubner Stuttgart, 1979.
7. M. Bojanczyk and T. Idziaszek. Algebra for infinite forests with an application to the temporal logic EF. In *CONCUR*, pages 131–145, 2009.
8. M. Bojanczyk, L. Segoufin, and H. Straubing. Piecewise testable tree languages. *LMCS*, 8(3), 2012.
9. R. Bonnet, A. Finkel, J. Leroux, and M. Zeitoun. Model checking vector addition systems with one zero-test. *LMCS*, 8(2), 2012.
10. T. Colcombet. Factorization forests for infinite words and applications to countable scattered linear orderings. *Theor. Comput. Sci.*, 411(4-5):751–764, 2010.
11. W. Czerwinski, W. Martens, and T. Masopust. Efficient separability of regular languages by subsequences and suffixes. In *ICALP*, pages 150–161, 2013.
12. J. Esparza, P. Ganty, S. Kiefer, and M. Luttenberger. Parikh's theorem: A simple and direct automaton construction. *Inf. Process. Lett.*, 111(12):614–619, 2011.
13. S. Ginsburg and S. A. Greibach. Abstract families of languages. In *SWAT / FOCS*, pages 128–139, 1967.

14. S. Ginsburg and E. H. Spanier. Semigroups, Presburger formulas, and languages. *Pacific J. of Math.*, 16:285–296, 1966.
15. S. Greibach. A note on undecidable properties of formal languages. *Math. Systems Theory*, 2(1):1–6, 1968.
16. P. Hofman and W. Martens. Separability by short subsequences and subwords. In *ICDT*, pages 230–246, 2015.
17. H. B. Hunt III. On the decidability of grammar problems. *J. ACM*, 29(2):429–447, 1982.
18. M. Jantzen. On the hierarchy of Petri net languages. *RAIRO Informatique Théorique*, 13(1):19–30, 1979.
19. S. R. Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *STOC*, pages 267–281, 1982.
20. M. Kufleitner. The height of factorization forests. In *MFCS*, pages 443–454, 2008.
21. J. Leroux. The general vector addition system reachability problem by Presburger inductive invariants. *LMCS*, 6(3), 2010.
22. A. N. Maslov. Multilevel stack automata. *Problems of Information Transmission*, 12(1):38–42, 1976.
23. E. W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM J. Comput.*, 13(3):441–460, 1984.
24. R. Mayr. Process rewrite systems. *Electr. Notes Theor. Comput. Sci.*, 7:185–205, 1997.
25. R. Mayr. Undecidable problems in unreliable computations. *Theor. Comput. Sci.*, 297(1-3):337–354, 2003.
26. R. McNaughton. Algebraic decision procedures for local testability. *Math. Syst. Theory*, 8(1):60–76, 1974.
27. R. Parikh. On context-free languages. *J. ACM*, 13(4):570–581, 1966.
28. J.-E. Pin and P. Weil. Polynominal closure and unambiguous product. *Theory Comput. Syst.*, 30(4):383–422, 1997.
29. T. Place, L. van Rooijen, and M. Zeitoun. Separating regular languages by locally testable and locally threshold testable languages. In *FSTTCS*, pages 363–375, 2013.
30. T. Place, L. van Rooijen, and M. Zeitoun. Separating regular languages by piecewise testable and unambiguous languages. In *MFCS*, pages 729–740, 2013.
31. T. Place and M. Zeitoun. Going higher in the first-order quantifier alternation hierarchy on words. In *ICALP*, pages 342–353, 2014.
32. T. Place and M. Zeitoun. Separating regular languages with first-order logic. In *CSL-LICS*, pages 75:1–75:10. ACM, 2014.
33. M. P. Schützenberger. On finite monoids having only trivial subgroups. *Inform. Control*, 8(2):190–194, 1965.
34. I. Simon. Piecewise testable events. In *ICALP*, pages 214–222. Springer, 1975.
35. I. Simon. Factorization forests of finite height. *Theor. Comput. Sci.*, 72(1):65–94, 1990.
36. H. Straubing. Semigroups and languages of dot-depth two. *Theor. Comput. Sci.*, 58:361–378, 1988.
37. T. Szymanski and J. Williams. Noncanonical extensions of bottom-up parsing techniques. *SIAM J. Comput.*, 5(2), 1976.
38. Y. Zalcstein. Locally testable languages. *J. Comput. Syst. Sci.*, 6(2):151–167, 1972.
39. G. Zetzsche. Personal communication.
40. G. Zetzsche. An approach to computing downward closures. In *ICALP*, 2015. To appear.