*Discrete Mathematics and Theoretical Computer Science* DMTCS vol. **19**:4, 2017, #1

# *A Characterization for Decidable Separability by Piecewise Testable Languages*[*]

Wojciech Czerwiński[1†] Wim Martens[2‡] Lorijn van Rooijen[3§]
Marc Zeitoun[4¶] Georg Zetzsche[5‖]

[1] *University of Warsaw*
[2] *University of Bayreuth*
[3] *Paderborn University*
[4] *LaBRI, University of Bordeaux*
[5] *LSV, CNRS & ENS Paris-Saclay*



*received 20$^{th}$ Nov. 2015, accepted 25$^{th}$ Sep. 2017.*

The separability problem for word languages of a class $\mathcal{C}$ by languages of a class $\mathcal{S}$ asks, for two given languages $I$ and $E$ from $\mathcal{C}$, whether there exists a language $S$ from $\mathcal{S}$ that includes $I$ and excludes $E$, that is, $I \subseteq S$ and $S \cap E = \emptyset$. In this work, we assume some mild closure properties for $\mathcal{C}$ and study for which such classes separability by a piecewise testable language (PTL) is decidable. We characterize these classes in terms of decidability of (two variants of) an unboundedness problem. From this, we deduce that separability by PTL is decidable for a number of language classes, such as the context-free languages and languages of labeled vector addition systems. Furthermore, it follows that separability by PTL is decidable if and only if one can compute for any language of the class its downward closure wrt. the scattered substring ordering (*i.e.*, if the set of scattered substrings of any language of the class is effectively regular).

The obtained decidability results contrast some undecidability results. In fact, for all (non-regular) language classes that we present as examples with decidable separability, it is undecidable whether a given language is a PTL itself.

Our characterization involves a result of independent interest, which states that for *any* kind of languages $I$ and $E$, non-separability by PTL is equivalent to the existence of common patterns in $I$ and $E$.

## 1 Introduction

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

[*]This article is an extended version of (Czerwiński et al., 2015).

[†]Supported by Poland's National Science Centre grant no. UMO-2013/11/D/ST6/03075.
[‡]Supported by DFG grant MA 4938/2-1.
[§]Supported by Agence Nationale de la Recherche ANR 2010 BLAN 0202 01 FREC.
[¶]Supported by Agence Nationale de la Recherche, ANR 2010 BLAN 0202 01 FREC and ANR-16-CE40-0007 DeLTA.
[‖]Supported by a fellowship within the PostDoc-Program of the German Academic Exchange Service (DAAD) and by Labex DigiCosme, Univ. Paris-Saclay, project VERICONISS.

ISSN 1365–8050 

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

When $\mathcal{C}$ is understood, we also say $\mathcal{S}$-separability for separability of $\mathcal{C}$ by $\mathcal{S}$.

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 (Place et al., 2013b; Czerwiński et al., 2013b), by several other levels in the quantifier alternation hierarchy of first-order logic (Place and Zeitoun, 2014a; Place, 2015b; Place and Zeitoun, 2017), by locally testable and locally threshold testable languages (Place et al., 2013a, 2014) and by first order definable languages (Place and Zeitoun, 2014b, 2016b). Another remarkable example goes beyond regular languages: the proof of Leroux (2010) for the decidability of reachability for vector addition systems or Petri nets greatly simplifies earlier proofs by Mayr (1981, 1984) and Kosaraju (1982), by establishing a crucial separation result: non-reachability of a marking "*goal*" from a marking "*start*" can be witnessed by a semilinear set containing all markings reachable from *start* and excluding *goal*.

In this paper, we focus on the theoretical underpinnings of separation by piecewise testable languages. Our interest in piecewise testable languages is mainly because of the following two reasons. First, piecewise testable languages form a natural class 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 $i = 1, \ldots, n$. This definition generalizes to tree languages, for which PTL-separability has already been solved by Goubault-Larrecq and Schmitz (2016). 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 (Hofman and Martens, 2015).

Second, it was shown recently (Place et al., 2013b; Czerwiński et al., 2013b) that separability of regular word languages (given by non-deterministic automata) by piecewise testable languages is in PTIME, a situation which is quite uncommon. The surprising tractability of this problem is another motivation for further investigating the class of piecewise testable languages.

**Separation and Characterization.** For classes $\mathcal{C}$ effectively closed under complement, separation of $\mathcal{C}$ by $\mathcal{S}$ is a generalization of the *characterization problem* of $\mathcal{C}$ by $\mathcal{S}$, often also called *membership problem*, which is defined as follows: for a given language $L$ of $\mathcal{C}$ 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 (1965) and Simon (1975), who solved it 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 subclasses thereof, often corresponding to a logical fragment, the problem is decidable (see for example Brzozowski and Simon, 1971; Zalcstein, 1972; McNaughton, 1974; Knast, 1983; Arfi, 1987; Pin and Weil, 1997; Straubing, 1988; Thérien and Wilke, 1998; Glaßer and Schmitz, 2000; Tesson and Thérien, 2002; Klíma, 2011; Czerwiński et al., 2013a; Place and Zeitoun, 2014a, 2017; Place, 2015b; Almeida et al., 2015). Similar problems have been considered for trees (Bojańczyk and Idziaszek, 2009; Benedikt and Segoufin, 2009; Bojańczyk et al., 2012; Antonopoulos et al., 2012; Place and Segoufin, 2015).

Obtaining a decidable characterization for a class is considered as a way to get a fine understanding of the class. In particular, solving a characterization problem requires a proof that a language satisfying some decidable property belongs to the class, which usually yields a canonical construction for the language. For instance, in the case of a logical class, this gives a canonical sentence defining any input language that fulfills the condition we want to prove as a decidable characterization. Recently, several generalizations of the characterization problem have been introduced as a means to obtain even more information about the class $\mathcal{S}$ under study. Such generalizations amount to *approximating* languages of $\mathcal{C}$ by languages of $\mathcal{S}$ (Place and Zeitoun, 2016a). In turn, such information may be exploited when studying more complex classes, *e.g.*, classes that are higher in the quantifier alternation hierarchy (Place and Zeitoun, 2014a). Separation is the simplest of such approximation problems: it asks to over-approximate the first language, $I$, by a language $S$ in $\mathcal{C}$, while the second language, $E$, serves as a quality measure of this approximation.

**Beyond regular languages.** 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 nontrivial decidability beyond regular languages. In this work, we consider language classes beyond the regular languages, but we assume them to be full trios (meaning they satisfy some mild closure properties). Beyond the regular languages, we quickly encounter undecidability of the problems above. For instance, for a context-free language (given by a grammar or a pushdown automaton) it is well known that it is undecidable to determine whether it is a regular language, by Greibach's theorem (1968). In the same way, one can show that for every full trio that contains the language $\{a^n b^n \mid n \geq 0\}$, it is undecidable to determine whether a given language of the full trio is piecewise testable (see Section 6).

In the case of 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 (1976) proved that separability of context-free languages by *regular* languages is undecidable. This was then generalized by Hunt III (1982), 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 $L = F_1 A^* \cup F_2$, where $F_1$ and $F_2$ are finite languages over alphabet $A$. As such, for definite languages, it can be 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 symbol of a word and is closed under Boolean combinations, it can already define all the definite languages. In his paper, Hunt III makes an explicit link between intersection emptiness and separability. Hunt III writes: *"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 a piecewise testable language (PTL): we characterize those full trios for which separability by a PTL is decidable. A *full trio* is a nonempty language class that is closed under rational transductions, or, equivalently by Nivat's theorem (see Nivat, 1968; Berstel, 1979), closed under direct and inverse homomorphic images of free monoids, and under intersections with regular languages.

Our characterization states that separability by PTL is decidable if and only if (one of two variants of) an unboundedness problem is decidable. This yields decidability for a range of language classes, such as those with effectively semilinear Parikh images and the languages of labeled vector addition systems. In particular, this means that separability by PTL is decidable for context-free languages, and, according to very recent results, also for the languages of higher-order pushdown automata (Hague et al., 2016) and even higher-order recursion schemes (Clemente et al., 2016).

The two variants of the unboundedness problem are called *diagonal problem* and *simultaneous unboundedness problem (SUP)*, of which the latter is ostensibly easier than the former. Our reduction of separability by PTL consists of three steps:

- In the first step, we show that (arbitrary) languages $I$ and $E$ are *not* separable by PTL if and only if they possess a certain *common pattern*.
- In the second step we use this fact to reduce the PTL-separation problem to the diagonal problem.
- In the last step, we employ ideal decompositions for downward closed languages to reduce the diagonal problem to the SUP.

For the converse direction, we directly reduce the SUP to separability by PTL.

Another consequence of our characterization is a connection to the problem of *computing downward closures*. It is well known that for any language $L$, its *downward closure*, *i.e.*, the set of scattered substrings of members of $L$, is a regular language (Haines, 1969). This is a straightforward consequence of Higman's Lemma (Higman, 1952). However, given a language $L$, it is not always possible to compute a finite automaton for the downward closure of $L$. Since the downward closure appears to be a useful abstraction, this raises the question of when we can compute it. Our characterization here implies that for each full trio $\mathcal{S}$, downward closures of languages of $\mathcal{S}$ are computable if and only if separability of languages of $\mathcal{S}$ by PTL is decidable.

A curiosity of this work is perhaps the absence of algebraic methods, as most decidability results we are aware of have considered syntactic monoids of regular languages and investigated properties thereof. In the setting of this paper, the situation is different since we want to separate input languages that are not regular (hence, whose syntactic monoid is infinite), such as context-free languages. This would make it difficult to design any algebraic framework for them. However, the main reason why we do not rely on syntactic monoids is that the characterization and separation problems are of very different nature. For characterization, whether the input language belongs to $\mathcal{C}$ indeed boils down to checking a property of the syntactic (ordered) monoid for the classes $\mathcal{C}$ that have been investigated so far, namely, (positive) varieties of regular languages. For separation however, the language we are looking for in the class $\mathcal{C}$ is *not* one of the inputs, so that a given language from $\mathcal{C}$ can serve as a separator for several unrelated instances of the $\mathcal{C}$-separation problem, and these instances may well have completely different syntactic properties.

**Structure of the paper** In Section 2 we introduce basic notions and notation, the diagonal and SUP problems, and state our main result, which connects these problems with separation by PTL. Section 3 gives a simple criterion by "common patterns" for two arbitrary languages to be separable by PTL. Section 4 is devoted to proving equivalent conditions for separability by PTL: the existence of common patterns, the diagonal problem, the SUP problem, and the computation of downward closures. In Section 5 we discuss applications of our result, in particular we show that it applies to context-free languages and languages of labeled vector addition systems (alternatively, languages of labeled Petri nets). Finally, in Section 6, we present some classes for which all these problems are undecidable.

# 2 Preliminaries and Main Results

We assume the reader to be familiar with regular expressions and regular languages. In this section, we first set the notation and state our main results.

The set of all integers and nonnegative integers are denoted by $\mathbb{Z}$ and $\mathbb{N}$ respectively. An *alphabet* is a finite nonempty set, which we usually denote with $A$, $B$, $C$, ... or indexed versions thereof. We refer to elements of $A$ as *symbols*.

A *word* is a (possibly empty) concatenation $w = a_1 \cdots a_n$ of symbols $a_i$ that come from an alphabet $A$. A *substring* of $w$ is a sequence $a_j a_{j+1} \cdots a_{j+k}$ of consecutive symbols of $w$. The *length* of a word $w = a_1 \cdots a_n$ is $n$, that is, the number of its symbols. The *alphabet of a word* $w = a_1 \cdots a_n$ is the set $\{a_1, \dots, a_n\}$ and is denoted $\mathrm{Alph}(w)$. The empty word, of length 0 and of alphabet $\emptyset$, is denoted by $\varepsilon$.

For a subalphabet $B \subseteq A$, a word $v \in A^*$ is a $B$-*scattered substring* of $w$, denoted $v \preceq_B w$, if $v$ can be obtained from $w$ by removing symbols from $B$, that is, if $v = b_1 \cdots b_m$ and $w \in B^* b_1 B^* \cdots B^* b_m B^*$. Notice that we do not require that $\{b_1, \dots, b_m\} \subseteq B$ or $B \subseteq \{b_1, \dots, b_m\}$. We simply refer to $A$-scattered substrings as *scattered substrings* and refer to the relation $\preceq_A$ as the *scattered substring* relation, denoted 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 $a_1, \dots, a_n \in A$, that is, it is the set of words having $a_1 \cdots a_n$ as a scattered substring. A regular language is a *piecewise testable* language (PTL) if it is a finite Boolean combination of piece languages. We denote the class of all piecewise testable languages also by PTL.

## 2.1 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 pattern. 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 $(\vec{u}, \vec{B})$ is such a factorization pattern, there exist words $u_0, \dots, u_p \in A^*$ and nonempty alphabets $B_1, \dots, B_p \subseteq A$ such that $\vec{u} = (u_0, \dots, u_p)$ and $\vec{B} = (B_1, \dots, B_p)$. For $B \subseteq A$, we denote by $B^{\circledast}$ the set of words with alphabet exactly $B$, that is,

$$B^{\circledast} = \{w \in B^* \mid \mathrm{Alph}(w) = B\}.$$

Given a factorization pattern $(\vec{u}, \vec{B})$ with $\vec{u} = (u_0, \dots, u_p)$ and $\vec{B} = (B_1, \dots, B_p)$, define

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

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

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

**Example 2.1** As an example, consider the factorization pattern $(\vec{u}, \vec{B})$ with $\vec{u} = (\varepsilon, c, \varepsilon)$ and $\vec{B} = (\{a, b\}, \{a\})$. Then, $\mathcal{L}(\vec{u}, \vec{B}, n) = (\{a, b\}^{\circledast})^n c (\{a\}^{\circledast})^n$. A sequence of words $w_1, w_2, \dots$ is $(\vec{u}, \vec{B})$-adequate if first of all for every $n \in \mathbb{N}$, there is a single symbol $c$ in word $w_n$. Second of all, after this symbol $c$ there are only symbols $a$ and, moreover, at least $n$ of them. Finally, the prefix of $w_n$ before the distinguished symbol $c$ contains only symbols $a$ and $b$ and can be split into at least $n$ words such that all of them contain at least one symbol $a$ and at least one symbol $b$.

Finally, we say that 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 can now formally state our first main theorem which we will prove in Section 3:

**Theorem 2.2** *Two word languages $I$ and $E$ are not separable by* PTL *if and only if they contain a common pattern* $(\overrightarrow{u}, \overrightarrow{B})$.

## 2.2 Characterizations for Decidable Separability

The second main result is a set of characterizations that say, for *full trios*, when separability by piecewise testable languages is decidable. Full trios, also called *cones*, are language classes that are closed under three operations that we recall next (Berstel, 1979; Ginsburg and Greibach, 1967).

Fix a language $L$ over alphabet $A$. For an alphabet $B$, the $B$*-projection* of a word is its longest scattered substring 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 that have a $B$-scattered substring in $L$, *i.e.*,

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

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

A *language class* is a collection of languages that contains at least one nonempty language. A language class $\mathcal{C}$ is *closed* under an operation $\textsc{op}$ if $L \in \mathcal{C}$ implies that $\textsc{op}(L) \in \mathcal{C}$. We use the term *effectively closed* if, furthermore, the representation of $\textsc{op}(L)$ 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, either through closures under direct and inverse images of morphisms and intersection with regular languages, or through closure under rational transductions (Ginsburg and Greibach, 1967; Berstel, 1979). However, we use the above mentioned properties in the proofs, which are easily seen to be equivalent.

We will now introduce three problems whose decidability or computability will be equivalent to decidability of separability by piecewise testable languages: the *diagonal problem*, the *simultaneous unboundedness problem*, and the *downward closure problem*.

**Downward Closure Problem.** For a language $L \subseteq A^*$, its *downward closure* $L\!\downarrow$ is defined as the set of all scattered substrings of words in $L$, that is,

$$L\!\downarrow = \{u \in A^* \mid \exists v \in L\colon u \preceq v\}.$$

We say that $L$ is downward closed if $L = L\!\downarrow$. It is well-known that the downward closure of any language $L$ is regular (Haines, 1969). This is a direct consequence of Higman's Lemma (Higman, 1952), which states that $\preceq$ is a well quasi ordering[1] (WQO) on $A^*$, *i.e.*, that any infinite sequence of words admits

[1] Actually, the relation $\preceq$ that we defined over $A^*$ is even a well *ordering*.

an infinite $\preceq$-increasing subsequence. Indeed, the complement of a downward closure is closed under $A$-upward closure. It follows from Higman's Lemma that this complement has a finite number of $\preceq$-minimal elements and is therefore a finite union of piece languages.

Downward closed languages play an important role in the theory of lossy channel systems (Finkel, 1987; Abdulla and Jonsson, 1996; Finkel and Schnoebelen, 2001), which feature communication channels where messages can be dropped arbitrarily. These have been investigated extensively, because the WQO property of the subword relation yields decidability results that fail in the non-lossy case. However, the complexity of such decidable problems may be huge (Schnoebelen, 2002; Schmitz and Schnoebelen, 2013).

Moreover, the downward closure is a useful abstraction of languages: Suppose a language $L$ describes the set of action sequences of a system modeled, for example, by a vector addition system or a pushdown automaton. If this system is observed via a lossy channel, then $L\downarrow$ is the set of sequences seen by the observer (Habermehl et al., 2010). Furthermore, computing a regular representation of $L\downarrow$ for a given language $L$ is in many situations sufficient for safety verification of parametrized asynchronous shared-memory systems (La Torre et al., 2015).

Starting from a regular language given, *e.g.*, by a finite automaton, one can easily compute a representation of its downward closure (one can even obtain precise upper and lower bounds in terms of the size of an automaton recognizing it (Karandikar et al., 2015)). However, despite the fact that the downward closure of a language is always a simple regular language (the complement of a union of piece languages), it is not always possible to effectively compute a finite automaton for $L\downarrow$ given a description of a (possibly nonregular) language $L$. Such negative results are known, for example, for Church-Rosser languages (Gruber et al., 2007) and reachability sets of lossy channel systems (Mayr, 2003). We say that *downward closures are computable for* $\mathcal{C}$ if a finite automaton for $L\downarrow$ can be algorithmically computed for every language $L$ in $\mathcal{C}$.

**Unboundedness Problems.** Let $A = \{a_1, \ldots, a_n\}$ be ordered with $a_1 < \cdots < a_n$. 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 $i = 1, \ldots, n$.

**Definition 2.3** *The* diagonal problem for $\mathcal{C}$ *is the following decision problem:*

INPUT. *A language $L \subseteq A^*$ from $\mathcal{C}$.*

QUESTION. *Is each tuple $(m, \ldots, m) \in \mathbb{N}^n$ dominated by some tuple in the Parikh image of $L$?*

Equivalently, the diagonal problem for $\mathcal{C}$ asks whether there are *infinitely many* tuples $(m, \ldots, m)$ dominated by some tuple in the Parikh image of $L$.

The *simultaneous unboundedness problem (SUP)* is a restricted version of the diagonal problem where the input is a language in which the symbols are grouped together:

**Definition 2.4** *The* simultaneous unboundedness problem (SUP) for $\mathcal{C}$ *is the following decision problem:*

INPUT. *A language $L \subseteq a_1^* \cdots a_n^*$ from $\mathcal{C}$, for some ordering $a_1, \ldots, a_n$ of $A$.*

QUESTION. *Is each tuple $(m, \ldots, m) \in \mathbb{N}^n$ dominated by some tuple in the Parikh image of $L$?*

In other words, the SUP asks whether $L\downarrow = a_1^* \cdots a_n^*$.

**Characterizations.** Perhaps surprisingly, our second main result here implies that in a full trio, separability by PTL is decidable if and only if downward closures are computable. Our proof relies on the following characterization of computability of downward closures:

**Theorem 2.5 (Zetzsche, 2015a)** *Let $\mathcal{C}$ be a full trio. Then downward closures are computable for $\mathcal{C}$ if and only if the SUP is decidable for $\mathcal{C}$.*

We are now ready to state the second main result, which is an extension of Theorem 2.5 and connects it to separability:

**Theorem 2.6** *For each full trio $\mathcal{C}$, the following are equivalent:*

*(1) Separability of $\mathcal{C}$ by* PTL *is decidable.*

*(2) The diagonal problem for $\mathcal{C}$ is decidable.*

*(3) The SUP for $\mathcal{C}$ is decidable.*

*(4) Downward closures are computable for $\mathcal{C}$.*

Theorem 2.5 provides the equivalence between (3) and (4). We prove the remaining equivalences in Section 4. In particular, 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 theorem to Vector Addition System languages, which do not have a semilinear Parikh image.

# 3 Common Patterns

In this section we prove Theorem 2.2. We say that an infinite sequence is *adequate* if it is $(\overrightarrow{u}, \overrightarrow{B})$-adequate for some factorization pattern $(\overrightarrow{u}, \overrightarrow{B})$. We will show the following combinatorial statement using Simon's Factorization Forest Theorem (Simon, 1990).

**Lemma 3.1** *Every infinite sequence $(w_n)_n$ of words admits an adequate subsequence.*

Before proving it, we note that Lemma 3.1 gives us an alternative proof of Higman's Lemma.

**Corollary 3.2 (Higman, 1952)** *The scattered subword ordering over $A^*$ is well.*

**Proof:** We want to show that for any sequence of words $(w_n)_{n\in\mathbb{N}}$, there exist two indices $i, j$ such that $i < j$ and $w_i \preceq w_j$, where $\preceq$ denotes the scattered substring relation. Taking a subsequence of $(w_n)_{n\in\mathbb{N}}$ if necessary, we may assume by Lemma 3.1 that $(w_n)_{n\in\mathbb{N}}$ is adequate. This means that there exists a factorization pattern $(\overrightarrow{u}, \overrightarrow{B})$, where $\overrightarrow{u} = (u_0, \dots, u_p)$ and $\overrightarrow{B} = (B_1, \dots, B_p)$ such that $(w_n)_{n\in\mathbb{N}}$ is $(\overrightarrow{u}, \overrightarrow{B})$-adequate, that is,

$$\forall n \in \mathbb{N}, \quad w_n \in u_0({B_1}^{\circledast})^n u_1 \cdots u_{p-1}({B_p}^{\circledast})^n u_p. \tag{3.1}$$

In particular for $n = 0$, there exist $v_0, \dots, v_p$ such that

$$w_0 = u_0 v_0 u_1 \dots u_{p-1} v_p u_p \tag{3.2}$$

and

$$v_i \in {B_i}^{\circledast}. \tag{3.3}$$

Note that by definition of the $^{\circledast}$ operation, if $v \in B^{\circledast}$, then for all $v' \in (B^{\circledast})^{|v|}$, we have $v \preceq v'$. Together with (3.1), (3.2) and (3.3), this entails that $w_0 \preceq w_n$ for all $n \geq n_0 = \max\{|v_i| \mid 0 \leq i \leq p\}$. It then suffices to choose $i = 0$ and $j = n_0$. □

**Proof of Lemma 3.1:** We use Simon's Factorization Forest Theorem, which we recall. See (Simon, 1990; Kufleitner, 2008; Bojańczyk, 2009; Colcombet, 2010, 2017) for proofs and extensions of this theorem. A *factorization tree* of a nonempty word $x$ is a finite ordered unranked tree $T(x)$ whose nodes are labeled by nonempty words, and such that:

- all leaves of $T(x)$ are labeled by symbols,
- all internal nodes of $T(x)$ have at least 2 children, and
- if a node labeled $y$ has $k$ children labeled $y_1, \dots, y_k$ from left to right, then $y = y_1 \cdots y_k$.

Given a semigroup morphism $\varphi : A^+ \to S$ into a finite semigroup $S$, such a factorization tree is called *$\varphi$-Ramseyan* if every internal node has either 2 children, or $k$ children labeled $y_1, \dots, y_k$, in which case $\varphi$ maps all words $y_1, \dots, y_k$ to the same idempotent of $S$, *i.e.*, to an element $e$ such that $ee = e$. Simon's Factorization Forest Theorem states that every word has a $\varphi$-Ramseyan factorization tree of height at most $3|S|$.

Let $(w_n)_n$ be an infinite sequence of words. We use Simon's Factorization Forest Theorem with the morphism $\mathrm{Alph} : A^+ \to 2^A$. Recall that Alph maps a word $w$ to the set of symbols used in $w$.

Consider a sequence $(T(w_n))_n$, where $T(w_n)$ is an Alph-Ramseyan tree of $w_n$, given by the Factorization Forest Theorem. In particular, one may choose $T(w_n)$ of height at most $3 \cdot 2^{|A|}$. Therefore, extracting a subsequence if necessary, one may assume that the sequence of heights of the trees $T(w_n)$ is a constant $H$. We argue by induction on $H$. If $H = 0$, then every $w_n$ is a symbol. Hence, one may extract from $(w_n)_n$ a constant subsequence, say $(a)_{n \in \mathbb{N}}$, which is therefore $((a), ())$-adequate, that is, adequate for the factorization pattern consisting of $(a)$ and the empty tuple $()$. Hence, it is adequate, which concludes the case $H = 0$.

Assume now that $H > 0$. We denote the arity of the root of $T(w_n)$ by $\mathrm{arity}(w_n)$ and we call it the arity of $w_n$. We distinguish two cases:

CASE 1. One can extract from $(w_n)_n$ a subsequence of bounded arity. Therefore, one may extract a subsequence of constant arity, say $K$, from $w_n$, and replacing $(w_n)_n$ by such a subsequence, one may assume that $(w_n)_n$ itself has this property. This implies that each $w_n$ can be written as a concatenation of $K$ words

$$w_n = w_{n,1} \cdots w_{n,K},$$

where $w_{n,i}$ is the label of the $i$-th child of the root in $T(w_n)$. Therefore, the Alph-Ramseyan subtree of each $w_{n,i}$ is of height at most $H - 1$. By induction, one can extract from every $(w_{n,i})_n$ an adequate subsequence. Proceeding iteratively for $i = 1, 2, \dots, K$, one extracts from $(w_n)_n$ a subsequence $(w_{\sigma(n)})_n$ such that every $(w_{\sigma(n),i})_n$ is adequate. But a finite concatenation of adequate sequences is obviously adequate. Therefore, the subsequence $(w_{\sigma(n)})_n$ of $(w_n)_n$ is also adequate.

CASE 2. The arity of $w_n$ grows to infinity. Therefore, extracting if necessary, one can assume for every $n$, that arity$(w_n) \geq \max(n, 3)$. Since each factorization tree is Alph-Ramseyan and since the arity of the root of each tree is at least 3, all children of the root of the $n$-th tree map to the same idempotent in $2^A$, say $B_n \subseteq A$. Since $2^A$ is finite, one can further extract a subsequence, say $w_{\sigma(n)}$, such that $B_{\sigma(n)}$ is constant, equal to some $B \subseteq A$. To sum up, each word of the subsequence is of the form

$$w_{\sigma(n)} = w_{n,1} \cdots w_{n,K_n},$$

with $K_n \geq n$ and, where the alphabet of $w_{n,i}$ is $B$. Therefore, $w_{\sigma(n)} \in (B^{\circledast})^{K_n} \subseteq (B^{\circledast})^n$, which means that $(w_{\sigma(n)})_n$ is $((\varepsilon, \varepsilon), (B))$-adequate, hence it is adequate. $\square$

For a nonempty word $w$, denote its first (resp., last) symbol by first$(w)$, (resp., last$(w)$). A factorization pattern $(\overrightarrow{u}, \overrightarrow{B}) = ((u_0, \ldots, u_p), (B_1, \ldots, B_p))$ is said to be *proper* if

1. for all $i = 0, \ldots, p-1$, we have $u_i = \varepsilon$ or last$(u_i) \notin B_{i+1}$,
2. for all $i = 1, \ldots, p$, we have $u_i = \varepsilon$ or first$(u_i) \notin B_i$, and
3. for all $i = 1, \ldots, p-1$, if $u_i = \varepsilon$, then we have $\big(B_i \nsubseteq B_{i+1}$ and $B_{i+1} \nsubseteq B_i\big)$.

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:

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

The following lemma gives a condition under which two sequences share a factorization pattern and is very similar to (Almeida, 1994, Theorem 8.2.6). In its statement, we write $v \sim_n w$ for two words $v$ and $w$ if they have the same scattered substrings up to length $n$, that is, if for every word $u$ of length at most $n$, we have $u \preceq v$ if and only if $u \preceq w$. Notice that $\sim_n$ is an equivalence relation for every $n \in \mathbb{N}$.

**Lemma 3.3** *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, and*
- $v_n \sim_n w_n$ *for every* $n \geq 0$.

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

**Proof:** For a factorization pattern $(\overrightarrow{u}, \overrightarrow{B})$, we define

$$\|(\overrightarrow{u}, \overrightarrow{B})\| = (\sum_{i=0}^{\ell} |u_i|) + \ell,$$

where $\ell + 1$ is the number of components in the vector $\overrightarrow{u} = (u_0, \ldots, u_\ell)$. Let

$$k = \max(\|(\overrightarrow{u}, \overrightarrow{B})\|, \|(\overrightarrow{t}, \overrightarrow{C})\|) + 1.$$

Consider the second word of the sequence $(v_n)_n$, *i.e.*, $v_1 = u_0 r_1 u_1 \cdots r_\ell u_\ell$, where $\mathrm{Alph}(r_i) = B_i$. Define

$$v_1^{(k)} = u_0 r_1^k u_1 \ldots r_\ell^k u_\ell. \tag{3.4}$$

Recall that $(v_n)_n$ being a $(\overrightarrow{u}, \overrightarrow{B})$-adequate sequence means that

$$\forall n \in \mathbb{N}, \quad v_n \in u_0 (B_1{}^{\circledast})^n u_1 \cdots u_{\ell-1} (B_\ell{}^{\circledast})^n u_\ell. \tag{3.5}$$

Let $N = k \cdot \max(|r_1|, \ldots, |r_n|)$. By (3.4) and (3.5), we get $v_1^{(k)} \preceq v_N$. Let $M \geq \max(N, |v_1^{(k)}|)$, so that $v_1^{(k)}$ is a scattered substring of $v_M$ of length at most $M$. Since $v_M \sim_M w_M$, this gives that

$$v_1^{(k)} \preceq w_M. \tag{3.6}$$

To show that $(\overrightarrow{u}, \overrightarrow{B}) = (\overrightarrow{t}, \overrightarrow{C})$, we first define a bijection between elements of the sequence of indexed alphabets in $\overrightarrow{B}$ and elements of the sequence of those in $\overrightarrow{C}$. For this, we embed $v_1^{(k)}$ in $w_M$ in the 'leftmost' way. Let us make this precise: let $x, y \in A^*$ with $x = x_1 \cdots x_{|x|}$ and $y = y_1 \cdots y_{|y|}$, where $x_i, y_i \in A$. If $x \preceq y$, by definition there exists an increasing mapping $h : \{1, \ldots, |x|\} \to \{1, \ldots, |y|\}$ such that $x_i = y_{h(i)}$ holds for all $i$. We say that such a mapping is an *embedding* of $x$ in $y$. The *leftmost embedding* $h_{\text{left}}$ of $x$ in $y$ is such that additionally, for any embedding $h$, we have $h_{\text{left}}(i) \leq h(i)$ for all $i$.

To define the bijection, it is convenient to tag the subalphabets $B_i, C_j$, as some of them may be equal. Therefore, we set $\mathbf{B} = \{(B_1, 1), \ldots, (B_\ell, \ell)\}$ and $\mathbf{C} = \{(C_1, 1), \ldots, (C_m, m)\}$, where $\ell$ (resp. $m$) denotes the length of the vector $\overrightarrow{B}$ (resp. $\overrightarrow{C}$). We define a function $f : \mathbf{B} \to \mathbf{C}$ as follows. Consider the leftmost embedding $h_{\text{left}}$ of $v_1^{(k)}$ in $w_M$, which exists by (3.6). Since $(w_n)_n$ is a $(\overrightarrow{t}, \overrightarrow{C})$-adequate sequence, one may write

$$w_M = t_0 s_1 t_1 \cdots t_{m-1} s_m t_m \quad \text{with } s_j \in (C_j{}^{\circledast})^M \text{ for all } j.$$

For $i$ in $\{1, \ldots, \ell\}$, consider the substring $r_i^k$ of $v_1^{(k)}$. By definition of $k > \|(\overrightarrow{t}, \overrightarrow{C})\|$ and since $|r_i| > 0$, the pigeonhole principle implies that all positions of one of these $k$ copies of $r_i$ must be mapped by $h_{\text{left}}$ to positions of some substring $s_j \in (C_j{}^{\circledast})^M$. We define $f(B_i, i) = (C_j, j)$ for the smallest such index $j$.

The function $g : \mathbf{C} \to \mathbf{B}$ is defined symmetrically, by exchanging the roles of $(v_n)_n$ and $(w_n)_n$. Moreover, since $\mathrm{Alph}(r_i) = B_i$ and $r_i \preceq s_j \in (C_j{}^{\circledast})^M$, we obtain that $f(B_i, i) = (C_j, j)$ entails $B_i \subseteq C_j$. Similarly, if $g(C_j, j) = (B_i, i)$, then $C_j \subseteq B_i$. If we show that $f$ and $g$ define a bijective correspondence between $\mathbf{B}$ and $\mathbf{C}$, then $\ell = m$. The above observations would then imply that $B_i = C_i$, for every $i$.

To establish that $f$ and $g$ are inverses of one another, we apply Lemma 8.2.5 from (Almeida, 1994), which we shall first repeat:

**Lemma 3.4 (Almeida, 1994, Lemma 8.2.5)** *Let $X$ and $Y$ be finite totally ordered sets and let $P$ be a partially ordered set. Let $f : X \to Y, g : Y \to X, p : X \to P$ and $q : Y \to P$ be functions such that*

1. *$f$ and $g$ are nondecreasing.*

2. *for any* $x \in X, p(x) \leq q(f(x))$,
3. *for any* $y \in Y, q(y) \leq p(g(y))$,
4. *if* $x_1, x_2 \in X, f(x_1) = f(x_2)$ *and* $p(x_1) = q(f(x_1))$, *then* $x_1 = x_2$,
5. *if* $y_1, y_2 \in Y, g(y_1) = g(y_2)$ *and* $q(y_1) = p(g(y_1))$, *then* $y_1 = y_2$.

*Then* $f$ *and* $g$ *are mutually inverse functions and* $p = q \circ f$ *and* $q = p \circ g$.

We prove Lemma 3.4 for the sake of completeness, and because there is a minor mistake in (Almeida, 1994): the original statement lacks the hypothesis that $f, g$ are nondecreasing, and this hypothesis can easily be seen to be necessary.

**Proof of Lemma 3.4:** By symmetry, it suffices to prove that for all $x \in X$, we have $g(f(x)) = x$. Note that this equality together with Item 2 and Item 3 applied to $y = f(x)$ then entails that $p = q \circ f$.

Let $x \in X$, we have to show that $g(f(x)) = x$. We define inductively two sequences, by $x_0 = x$, $y_n = f(x_n) \in Y$ and $x_{n+1} = g(y_n) \in X$. Assume by contradiction that $x_0 \neq x_1$. By symmetry, we may assume that $x_0 < x_1$. We claim that the sequence $(x_n)_n$ is strictly increasing, which immediately gives a contradiction since $X$ is finite. We prove this claim by induction. We already know that $x_0 < x_1$.

Assume now that $x_{n-1} < x_n$, we show that $x_n < x_{n+1}$. Since $f$ is nondecreasing, we have $y_{n-1} \leq y_n$. We want to show that $y_{n-1} < y_n$. If on the contrary $y_{n-1} = y_n$, then $f(x_{n-1}) = f(x_n)$ and by Items 2 and 3, we would have,

$$p(x_n) \leq q(f(x_n)) = q(y_n) = q(y_{n-1}) \leq p(g(y_{n-1})) = p(x_n),$$

so that $p(x_n) = q(f(x_n))$. Item 4 applied to $x_n$ and $x_{n-1}$ would imply that $x_{n-1} = x_n$, contradicting our induction hypothesis. Therefore, $y_{n-1} < y_n$. With a similar argument, from $y_{n-1} < y_n$ we obtain $x_n < x_{n+1}$, as desired. □

To apply Lemma 3.4, let $X = \mathbf{B}$ be ordered according to the second component, that is, $(B_i, i) < (B_j, j)$ when $i < j$. Similarly, let $Y = \mathbf{C}$ be ordered according to the second component. Observe that the functions $f : \mathbf{B} \to \mathbf{C}$ and $g : \mathbf{C} \to \mathbf{B}$ defined above are nondecreasing by construction. Finally, let $P$ be the set of subalphabets of $A$ partially ordered by inclusion, and let $p$ and $q$ be the projections onto the first coordinate.

Let us verify that $f$ and $g$ fulfill all conditions of Lemma 3.4. Item 2 holds since for all $i$ and $j$, $f(B_i, i) = (C_j, j)$ implies that $B_i \subseteq C_j$. Item 3 holds symmetrically.

For Item 4, suppose that $f(B_{i_1}, i_1) = f(B_{i_2}, i_2) = (C_j, j)$ and that $p(B_{i_1}, i_1) = q(f(B_{i_1}, i_1))$, that is, $B_{i_1} = C_j$. The first condition, $f(B_{i_1}, i_1) = f(B_{i_2}, i_2) = (C_j, j)$, means that a substring $r_{i_1}$ and a substring $r_{i_2}$ of $v_1^{(k)}$ have all their positions mapped by $h_{\text{left}}$ to positions of $s_j$ in $w_M$. Therefore, all intermediate positions are also mapped to positions of $s_j$, which implies, using the second condition, $B_{i_1} = C_j$, that $\text{Alph}(r_{i_1} u_{i_1} \cdots r_{i_2}) \subseteq \text{Alph}(s_j) = C_j = B_{i_1} = \text{Alph}(r_{i_1})$. But we assumed that $(\overrightarrow{u}, \overrightarrow{B})$ is a *proper* factorization pattern, so $i_1$ must be equal to $i_2$. This shows that Item 4 holds. Item 5 is dual.

It follows that indeed $f$ and $g$ define a bijective correspondence between $\mathbf{B}$ and $\mathbf{C}$, thus $\ell = m$ and $B_i = C_i$, for every $i$. Since we are dealing with proper factorization patterns, $v_1^{(k)} \preceq w_M$ now implies that $u_i \preceq t_i$ for every $i$. Symmetrically, $t_i \preceq u_i$ for every $i$. Thus, for every $i$, $u_i = t_i$. □

The proof of Theorem 2.2 relies on Lemmas 3.1 and 3.3, as well as on the following characterization of PTL-(non)-separation.

**Lemma 3.5** *Let $I$ and $E$ be two languages. Then, $I$ and $E$ are* not PTL*-separable if and only if for every $n \in \mathbb{N}$, there exist $v_n \in I$ and $w_n \in E$ such that $v_n \sim_n w_n$.*

**Proof:** We first prove the implication from left to right by contraposition: suppose that there exists $n \in \mathbb{N}$ such that $(I \times E) \cap \sim_n = \emptyset$. We have to prove that $I$ and $E$ are PTL-separable. The assumption implies that $[I]_n := \{v \mid \exists w \in I \text{ such that } v \sim_n w\}$ separates $I$ from $E$. It remains to verify that it is a PTL. Observe that $\sim_n$ has finite index, because an $\sim_n$-class is defined by a subset of words of length at most $n$. Therefore, $[I]_n$ is a finite union of $\sim_n$-classes. To show that $[I]_n$ is a PTL, it is therefore enough to check that a single $\sim_n$-class is a PTL, which is the case since the $\sim_n$-class of $u$ is

$$\bigcap_{v \preceq u,\ |v| \leq n} v\!\uparrow \cap \bigcap_{v \not\preceq u,\ |v| \leq n} A^* \setminus (v\!\uparrow)$$

where $v\!\uparrow$ is the $A$-upward closure of $\{v\}$, *i.e.*, the piece language $A^*a_1A^* \cdots A^*a_kA^*$, if $v = a_1 \cdots a_k$.

For the other direction, we first note that a piecewise testable language is a union of $\sim_n$-equivalence classes for some $n$. Indeed, notice that by definition, a piecewise testable language is a finite Boolean combination of piece languages. Let $n$ be the maximal length of the pieces defining these piece languages. Any piece language $A^*a_1A^* \cdots A^*a_kA^*$, may, for $v = a_1 \cdots a_k$, be written as $\bigcup_{v \preceq w} [w]_{\sim_k}$. This is a finite union, as $\sim_k$ has finite index. For every $k$, the equivalence relation $\sim_{k+1}$ is a refinement of $\sim_k$. Therefore, $A^*a_1A^* \cdots A^*a_kA^*$ may also be written as a finite union of $\sim_n$-equivalence classes. And, clearly, a finite Boolean combination of finite unions of $\sim_n$-equivalence classes is again a finite union of $\sim_n$-equivalence classes. Now suppose that for every $n \in \mathbb{N}$, there are words in $I$ (respectively, $E$) that are $\sim_n$-equivalent. Since a piecewise testable language is a union of $\sim_n$-equivalence classes for some $n$, any piecewise testable language containing $I$ will, in this case, have nonempty intersection with $E$. □

We are now able to prove Theorem 2.2 using Lemmas 3.1, 3.3 and 3.5.

**Proof of Theorem 2.2.** *Two word languages $I$ and $E$ are not separable by* PTL *if and only if they contain a common pattern $(\overrightarrow{u}, \overrightarrow{B})$.*

**Proof:** To prove the "if"-direction of the theorem, we show that for every $n$, there exist words in $I$ and in $E$ that are $\sim_n$-equivalent. By hypothesis, $I$ and $E$ contain a common factorization pattern $(\overrightarrow{u}, \overrightarrow{B})$. By definition, this gives us two sequences of words $(v_n)_n$ and $(w_n)_n$ such that for all $n$,

$$v_n \in I \cap u_0({B_1}^{\circledast})^n u_1 \cdots u_{p-1}({B_p}^{\circledast})^n u_p,$$
$$w_n \in E \cap u_0({B_1}^{\circledast})^n u_1 \cdots u_{p-1}({B_p}^{\circledast})^n u_p.$$

Observe that for all $i$, $({B_i}^{\circledast})^n$ contains precisely all words from $B_i^{\leq n}$ as scattered substrings of size up to $n$. It follows that $v_n \sim_n w_n$.

We now prove the "only-if" direction. The existence of $v_n \sim_n w_n$ for every $n \in \mathbb{N}$ 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 $\sim_n$-equivalence.

By Lemma 3.1, 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 3.1 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. Since $v_n \sim_n w_n$, Lemma 3.3 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 Characterization for Separability

In this section we prove our main characterization:

**Theorem 2.6.** *For each full trio $\mathcal{C}$, the following are equivalent:*

*(1) Separability of $\mathcal{C}$ by* PTL *is decidable.*

*(2) The diagonal problem for $\mathcal{C}$ is decidable.*

*(3) The SUP for $\mathcal{C}$ is decidable.*

*(4) Downward closures are computable for $\mathcal{C}$.*

The equivalence between (3) and (4) is immediate from Theorem 2.5. The implication "$(2) \Rightarrow (3)$" is trivial because the SUP is a special case of the diagonal problem. We prove the other direction "$(3) \Rightarrow (2)$" and the implication "$(1) \Rightarrow (3)$" in Section 4.1. Finally, we prove the implication "$(2) \Rightarrow (1)$" in Section 4.2, by giving an algorithm for separability if the diagonal problem is decidable.

## 4.1 Algorithms for the Diagonal Problem and the SUP

In this section, we reduce the diagonal problem to the simultaneous unboundedness problem and to separability by PTL. It should be noted that the implication "$(3) \Rightarrow (2)$" already follows easily from the equivalence between (3) and (4) (shown in (Zetzsche, 2015a)): when one can compute downward closures, the diagonal problem reduces to the case of regular languages. However, the algorithm for downward closure computation in (Zetzsche, 2015a) is not accompanied by any complexity bounds. Therefore, we mention here a reduction that can be carried out in non-deterministic polynomial time if we assume that the full trio operations require only polynomial time (Lemma 4.2). This reduction is based on the fact that downward closed languages can be written as finite unions of ideals, which is also a central ingredient in (Zetzsche, 2015a). An *ideal* is a language of the form $B_0^*\{b_1, \varepsilon\}B_1^* \cdots \{b_m, \varepsilon\}B_m^*$, where $m \geq 0$, $b_1, \ldots, b_m$ are symbols, and $B_0, \ldots, B_m$ are alphabets. Clearly, every ideal is downward closed. We use the following result by Jullien, which has later been rediscovered in (Abdulla et al., 2004).

**Theorem 4.1 (Jullien, 1969)** *Every downward closed language is a finite union of ideals.*

In fact, there is a notion of ideal in well quasi orderings for which this result holds in general. The property of the theorem is even equivalent to the fact that the underlying quasi ordering has no infinite antichains, see (Bonnet, 1975; Fraïssé, 2000) and (Erdős and Tarski, 1943). See also (Finkel and Goubault-Larrecq, 2009; Leroux and Schmitz, 2015) for other perspectives.

We now prove the implication "$(3) \Rightarrow (2)$" of Theorem 2.6. In the following lemma, we use the definition of full trios as being closed under rational transductions. A subset $T \subseteq A^* \times B^*$ for alphabets $A, B$ is called a *rational transduction* if there is an alphabet $C$, a regular language $R \subseteq C^*$, and morphisms

$\alpha\colon C^* \to A^*$, $\beta\colon C^* \to B^*$ such that we have $T = \{(\alpha(w), \beta(w)) \mid w \in R\}$. For a language $L \subseteq A^*$ and a rational transduction $T \subseteq A^* \times B^*$, we define

$$TL = \{w \in B^* \mid \exists v \in L\colon (v, w) \in T\}.$$

It is well-known that a nonempty class of languages $\mathcal{C}$ is an effective full trio if and only if for each $L \in \mathcal{C}$ and each rational transduction $T$, we have effectively $TL \in \mathcal{C}$, see (Berstel, 1979) for instance.

Observe that the set

$$D = \{(u, v) \in A^* \times A^* \mid v \preceq u\}$$

is a rational transduction, meaning that for every member $L$ of a full trio $\mathcal{C}$, the language $L\!\downarrow = DL$ is effectively contained in $\mathcal{C}$ as well.

**Lemma 4.2** *Let $\mathcal{C}$ be a full trio. If the SUP is decidable for $\mathcal{C}$, then so is the diagonal problem for $\mathcal{C}$.*

**Proof:** Let $L \subseteq A^*$ with $A = \{a_1, \ldots, a_n\}$. We say that $L$ satisfies the *diagonal property* if for each $m \in \mathbb{N}$, the vector $(m, \ldots, m)$ (with $|A|$ entries) is dominated by some vector in the Parikh image of $L$. We claim that $L$ satisfies the diagonal property if and only if we can order the symbols of $A$ as $A = \{b_1 \leq \cdots \leq b_n\}$ such that $b_1^* \cdots b_n^* \subseteq L\!\downarrow$. The "if" direction is immediate, so suppose $L$ satisfies the diagonal property.

Since $L$ satisfies the diagonal property, so does $L\!\downarrow$. According to Theorem 4.1, we can write $L\!\downarrow$ as a finite union of ideals. This means that at least one of these ideals has to satisfy the diagonal property. Let $I = C_0^*\{c_1, \varepsilon\}C_1^* \cdots \{c_m, \varepsilon\}C_m^*$ be one such ideal. Then we have $C_0 \cup \cdots \cup C_m = A$, since any element of $A \setminus (C_0 \cup \cdots \cup C_m)$ can occur at most $m$ times in words in $I$. Hence, if we order the elements of $A$ linearly by picking first the elements of $C_0$, then those of $C_1 \setminus C_0$, and so forth, the resulting ordering $A = \{b_1 \leq \cdots \leq b_n\}$ clearly satisfies $b_1^* \cdots b_n^* \subseteq I \subseteq L\!\downarrow$. This proves our claim.

Assume that $\mathcal{C}$ is a full trio for which the SUP is decidable. We show that the diagonal problem is decidable for $\mathcal{C}$ as well. Let $L \subseteq A^*$ be a language. .

We guess a linear ordering $A = \{b_1 \leq \cdots \leq b_n\}$. Notice that since $\mathcal{C}$ is closed under intersection with regular sets, the language $K = L\!\downarrow \cap\, b_1^* \cdots b_n^*$ is effectively contained in $\mathcal{C}$. Then, $K$ is an instance of the SUP. By our claim, $L$ satisfies the diagonal property if and only if there exists a linear ordering $A = \{b_1 \leq \cdots \leq b_n\}$ such that $K = L\!\downarrow \cap\, b_1^* \cdots b_n^*$ is a positive instance of the SUP. □

This concludes the proof of "(3) $\Rightarrow$ (2)" of Theorem 2.6, so that we now have established that Items (2), (3) and (4) are equivalent. It remains to connect these problems with the separation problem.

The correctness proof of our reduction of the SUP to the separability problem employs a characterization of separability by PTL from (Czerwiński et al., 2013b). For languages $K, L \subseteq A^*$, a $(K, L)$*-zigzag* is an infinite sequence of words $(w_i)_{i\in\mathbb{N}}$ such that

(i) $w_i \preceq w_{i+1}$ for every $i \in \mathbb{N}$,
(ii) $w_i \in K$ for every even $i$ and
(iii) $w_i \in L$ for every odd $i$.

The characterization for separability by PTL is then as follows.

**Theorem 4.3 (Czerwiński et al., 2013b)** *Let $K, L \subseteq A^*$ be languages. Then $L$ and $K$ are separable by* PTL *if and only if there is no $(K, L)$-zigzag.*

**Lemma 4.4** *Let $\mathcal{C}$ be a full trio. If separability by* PTL *is decidable for $\mathcal{C}$, then so is the SUP for $\mathcal{C}$.*

**Proof:** Let $A = \{a_1, \ldots, a_n\}$ and let $L \subseteq a_1^* \cdots a_n^*$ be a language in $\mathcal{C}$. Let $T \subseteq A^* \times A^*$ be the rational transduction

$$T = \{(a_1^{k_1} \cdots a_n^{k_n}, a_1^{2k_1} \cdots a_n^{2k_n}) \mid k_1, \ldots, k_n \geq 0\}.$$

Moreover, consider the regular language

$$K = \{a_1^{2k_1+1} \cdots a_n^{2k_n+1} \mid k_1, \ldots, k_n \geq 0\}.$$

We claim that $T(L\downarrow)$ and $K$ are inseparable by PTL if and only if $L\downarrow = a_1^* \cdots a_n^*$. We first show that this claim implies the lemma. Indeed, since $\mathcal{C}$ is a full trio, the language $T(L\downarrow)$ is effectively contained in $\mathcal{C}$. Since $K$ is a member of $\mathcal{C}$ (since every full trio contains the family of regular languages), the claim clearly implies the lemma.

We now prove the claim. Suppose that $L\downarrow = a_1^* \cdots a_n^*$. Then we have $T(L\downarrow) = (a_1a_1)^* \cdots (a_na_n)^*$ and the sequence $(w_i)_{i\in\mathbb{N}}$ with $w_i = a_1^i \cdots a_n^i$ is a $(T(L\downarrow), K)$-zigzag, meaning that $T(L\downarrow)$ and $K$ are inseparable by PTL, by Theorem 4.3.

Now suppose $T(L\downarrow)$ and $K$ are inseparable by PTL. Then again by Theorem 4.3, there is a $(T(L\downarrow), K)$-zigzag $(w_i)_{i\in\mathbb{N}}$. This means in particular that, for all $i$, we have $w_i \preceq w_{i+1}$, so that $|w_i|_a \leq |w_{i+1}|_a$ for $a \in A$. By construction of $T$ and $K$, the numbers $|w_i|_a$ and $|w_{i+1}|_a$ are incongruent modulo 2, which implies $|w_i|_a < |w_{i+1}|_a$ and therefore $|w_i|_a \geq i$. Moreover, since the sequence is a zigzag, we have $\{w_{2i} \mid i \geq 0\} \subseteq T(L\downarrow)$ and thus $L\downarrow = a_1^* \cdots a_n^*$. This completes the proof of our lemma. □

## 4.2 The Algorithm for Separability

It only remains to prove the implication "(2) $\Rightarrow$ (1)" to finish the proof of Theorem 2.6. In this section we show 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. Indeed, emptiness of a language $L \subseteq A^*$ can be decided by taking the $A$-upward closure of $L$ (which can be effectively computed from $L$, since it can be implemented by a rational transduction). In the resulting language, the diagonal problem returns true if and only if $L$ is nonempty.

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 problem.

**Negative semi-procedure.** Theorem 2.2 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 each one, checks the condition of Theorem 2.2. 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. Note that the difficulty for testing this condition comes from the universal quantification over $n$.

*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 "full 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 full trio $\mathcal{C}$ we will ensure that the investigated language still belongs to $\mathcal{C}$.

First we add special symbols $\$_i$, for $i \in \{1, \ldots, k\}$, which do not occur in $A$. These symbols are meant to count how many times alphabet $B_i$ is "fully 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 symbols $\$_i$ (with the same $i$), every symbol 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 symbols 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 modify iteratively $L_0$, 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 $L_0$. Thus, $L_1$ contains, in particular, all words where the $\$_i$ are placed "correctly", *i.e.*, in between two $\$_i$-symbols the whole alphabet $B_i$ should occur. However at this moment we do not check it. By closure under $B$-upward closures, which can be implemented by rational transductions, language $L_1$ belongs to $\mathcal{C}$.

Note that $L_1$ also contains words in which the $\$_i$-symbols 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 $\$_i$-symbols are placed at least in the "right areas". Concretely, $L_2$ is the intersection of $L_1$ with the language

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

Since $\mathcal{C}$ is a full trio, it is closed under intersection with regular languages, whence $L_2$ belongs to $\mathcal{C}$.

Language $L_2$ may still contain words, such that in between two $\$_i$-symbols not *all* the symbols 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 $\mathcal{C}$.[2]

Note that the 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 $n + 1$ symbols $\$_i$ 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 symbols $\$_i$

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

"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 $\$_i$-symbols. Language $L_4$ is therefore the $\{\$_1, \ldots, \$_k\}$-projection of $L_3$. By closure under projections, 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 substrings of $B_i^{\circledast}$ between $u_{i-1}$ and $u_i$. Therefore, $L \cap \mathcal{L}(\overrightarrow{u}, \overrightarrow{B}, n)$ is nonempty for all $n \geq 0$ if and only if the tuple $(n, \ldots, n)$ is dominated by an element of 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 2.6, in particular, for context-free languages and languages of labeled vector addition systems (which are the same as languages of labeled Petri nets).

**Effectively semilinear Parikh images.** A number of language classes is known to exhibit effectively semilinear Parikh images. 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. We say that a full trio $\mathcal{C}$ exhibits *effectively semilinear Parikh images* if every language in $\mathcal{C}$ has a semilinear Parikh image and one can compute a representation as a (finite) union of linear sets.

Clearly, one can decide the diagonal problem for language classes with effectively semilinear Parikh images. This amounts to checking whether in a representation of the Parikh image of the given language, there is some linear set in which each of the symbols occurs in some period vector.

Another option is to use Presburger logic. Semilinear sets are exactly those definable by Presburger logic. Moreover, the translation is effective. Assume that $|A| = k$, so the Parikh image $P$ of the considered language is a subset of $\mathbb{N}^k$ and $\phi$ is a Presburger formula describing $P$ having $k$ free variables. Then

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

is true if and only if the diagonal problem for the considered language is answered positively. Decidability of the Presburger logic finishes the proof of decidability of the diagonal problem. We refer for the details of semilinear sets and Presburger logic to (Ginsburg and Spanier, 1966).

Examples of full trios with effectively semilinear Parikh images are *context-free languages* (Parikh, 1966), *multiple context-free languages* (Seki et al., 1991), languages of *reversal-bounded counter automata* (Ibarra, 1978), *stacked counter automata* (Zetzsche, 2015b) and *finite index matrix languages* (Dassow and Păun, 1989).

**Higher-Order Pushdown Automata and Recursion Schemes.** Very recently, it was shown that the diagonal problem is decidable for higher-order pushdown automata (Hague et al., 2016) and even higher-order recursion schemes (Clemente et al., 2016). Both of these methods use an inductive approach: The diagonal problem for order-$(n+1)$ pushdown automata (resp. schemes) is decided by constructing an order-$n$ pushdown automaton (resp. scheme) that is equivalent with respect to the diagonal problem. Hence, the algorithm arrives at order 0, for which the diagonal problem is easy to solve.

**Languages of Labeled Vector Addition Systems and Petri Nets.** A $k$-dimensional *(labeled) vector addition system*, or *(labeled) VAS* $M = (A, T, \delta_0, \delta_1, \ell, s, t)$ over alphabet $A$ consists of a finite set of *transitions* $T$, a labeling $\ell : T \to A \cup \{\varepsilon\}$, mappings $\delta_0, \delta_1 : T \to \mathbb{N}^k$, 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 an $r \in T$ such that $\delta_0(r) \leq u$ and $v = u - \delta_0(r) + \delta_1(r)$ and $\ell(r) = a$, where the addition and comparison of vectors is defined coordinate-wise. For two markings $u, v \in \mathbb{N}^k$ we say that $u$ *reaches* $v$ *(via the 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 (Jantzen, 1979), we only need to prove decidability of the diagonal problem. For instance, this can be done by using the computability of downward closures for VAS languages (Habermehl et al., 2010).

Here, we present an alternative approach to this problem. We show that the *diagonal problem for VAS languages* is decidable by reduction to the *place-boundedness problem for VASs with one zero test*, which has been shown decidable (Bonnet et al., 2010, 2012). A $k$-dimensional *(labeled) vector addition system with one zero-test*, or *(labeled) VAS*$_z$ over the alphabet $A$ is a tuple $M = (A, T, r_z, \delta_0, \delta_1, \ell, s, t)$ such that $(A, T, \delta_0, \delta_1, \ell, s, t)$ is a VAS and additionally $r_z \in T$ is a distinguished *zero-test transition*. The semantics of a VAS$_z$ differs slightly from that of VASs. For two markings $u, v \in \mathbb{N}^k$ in a VAS$_z$ with $u = (u_1, \ldots, u_k)$, we write $u \xrightarrow{a} v$ if there is $r \in T$ such that (i) $\delta_0(r) \leq u$, (ii) $v = u - \delta_0(r) + \delta_1(r)$, (iii) if $r = r_z$, then $u_1 = 0$, and (iv) $\ell(r) = a$. Then, reaching another marking via a word is defined accordingly as in a VAS. In (Bonnet et al., 2010, 2012), it was shown that given a $k$-dimensional VAS$_z$ and an $i \in \{1, \ldots, k\}$, it is decidable whether for every $n \in \mathbb{N}$, there is a reachable marking $u = (u_1, \ldots, u_k)$ with $u_i \geq n$. This is known as the *place-boundedness problem*.

To set up our reduction, we start from a VAS $M$. Our objective is to build from $M$ a VAS$_z$ $M_z$, such that $M_z$ is unbounded in some distinguished coordinate if and only if $L(M)$ has the diagonal property. To build $M_z$ from $M$, we proceed in two steps.

STEP 1. The first step consists in modifying the VAS $M$ to get another VAS $M'$, in the following way.

- First, it is well-known that any VAS can be turned into one generating the same language and having $(0, \ldots, 0) \in \mathbb{N}^k$ as target marking (Hack, 1976), meaning we may assume that the target is zero.
- Second, we introduce a new *sum coordinate*, say as coordinate 1: it is easy to modify the VAS so that in the new first coordinate, one always has the sum of all original coordinates of $M$.
- Finally, we add for each $a \in A$ a *symbol coordinate*, which counts how many times we read symbol $a$. That is, for every transition which is labeled by $a \in A$, we add 1 in the symbol-coordinate corresponding to $a$ and 0 in the symbol-coordinates corresponding to other symbols. Hence, the set of symbol-coordinates always contains the Parikh image of the read prefix.

STEP 2. In the second step, we turn the VAS $M'$ into a VAS$_z$ $M_z$. For this, we add,

– two *mode coordinates*: the VAS$_z$ $M_z$ will be able to switch from the first mode to the second mode (but not the other way around), and
– a *minimum coordinate*, counting the minimum of all *original* coordinates. The intention is that this specific coordinate will be unbounded if and only if the original VAS $M$ has the diagonal property.

More precisely:

1. The initial marking is enriched by 1 on the first mode coordinate and by 0 in the second one.
2. All transitions of $M'$ are changed so that they subtract 1 from the first mode coordinate and add 1 to it as well (*i.e.*, on the first mode coordinate, $\delta_0(r)$ is $-1$ and $\delta_1(r)$ is 1). This means that the resulting transitions do not modify the mode coordinates, but can be fired only in the first mode.
3. The new *zero-test transition* $r_z$ tests to 0 the sum coordinate that was introduced in Step 1. Moreover, it only subtracts 1 from the first mode coordinate and adds 1 to the second mode coordinate. In other words, it is applicable only in the first mode, and switches to the second mode.
4. Finally, we add a new transition $\bar{r}$, labeled by $\varepsilon$, which subtracts 1 from each symbol coordinate and adds 1 to the minimum coordinate. Moreover, $\bar{r}$ subtracts 1 from the second mode coordinate and adds 1 to it (in other words, it does not affect the mode, but applies only in the second one).

Let us argue that the resulting VAS$_z$ $M_z$ is unbounded in the minimum coordinate if and only if $L(M)$ has the diagonal property. If $L(M)$ has the diagonal property, then for each $n$, we find a run in $M$ reading a word $w$ whose Parikh image dominates $(n, \ldots, n)$. Hence, we can take the corresponding run in the VAS$_z$, then fire $r_z$ once and then fire $\bar{r}$ exactly $n$ times to get the minimum coordinate to $n$.

On the other hand, if the minimum coordinate is unbounded in the VAS$_z$ $M_z$, then for each $n \geq 0$, there is a run in which the minimum coordinate holds at least $n$. The minimum coordinate can only have a nonzero content if the zero-test transition had been applied beforehand, so that this run has to correspond to a run in $M$ that arrives at $(0, \ldots, 0) \in \mathbb{N}^k$ (since the zero-test is performed on the sum coordinate), and reads a word whose Parikh image dominates $(n, \ldots, n)$. We have therefore reduced the diagonal problem for VAS languages to the place-unboundedness problem for VAS$_z$. The former is therefore decidable.

**Effective Parikh equivalence.** Let us mention another obvious consequence of our characterization. Assume separability by PTL is decidable for a full trio $\mathcal{D}$. Furthermore, suppose that for each given language $L$ in a full trio $\mathcal{C}$, one can effectively produce a Parikh equivalent language in $\mathcal{D}$. Then, separability by PTL is also decidable for $\mathcal{C}$. For example, this means that separability by PTL is decidable for *matrix languages* (Dassow and Păun, 1989), a natural language class that generalizes VAS languages and context-free languages. Indeed, it is well-known that given a matrix language, one can construct a Parikh equivalent VAS language (Dassow and Păun, 1989).

**Mixed instances.** Our result further yields that if separability by PTL is decidable for each of the full trios $\mathcal{C}$ and $\mathcal{D}$, then it is also decidable whether given $K \in \mathcal{C}$ and $L \in \mathcal{D}$ are separable by a PTL. Indeed, if the diagonal problem is decidable for $\mathcal{C}$ and for $\mathcal{D}$, then combining the respective algorithms yields decidability of the diagonal problem for the full trio $\mathcal{C} \cup \mathcal{D}$. For instance, separability of a context-free language from the language of a labeled vector addition system is decidable.

**Corollary 5.1** *Let $\mathcal{C}$ and $\mathcal{D}$ be full trios. If separability by* PTL *is decidable for $\mathcal{C}$ and for $\mathcal{D}$, then it is also decidable for $\mathcal{C} \cup \mathcal{D}$.*

# 6 Undecidable Classes

Given the fact that separability by PTL is decidable for context-free languages, the question arises whether the same is true of the context-sensitive languages. Here we show that this is *not* the case, because of the following reasons. First of all, context-sensitive languages are closed under complementation. Therefore, decidable separability of context-sensitive languages by PTL would imply that it is decidable to test if a given context-sensitive language is a PTL. The latter question, however, is already undecidable for context-free languages (Corollary 6.3). In this section we provide these observations with more detail and also exhibit a more general technique to show undecidable separability by PTL.

Thomas Place pointed out to us that, by a proof that follows similar lines to the proof of Greibach's Theorem (Greibach, 1968), one can show that testing whether a given context-free language is piecewise testable is undecidable (Place, 2015a). We give the proof below (Proposition 6.2).

We noticed that the proof he gave us even applies to all full trios that contain the language,

$$L_{ab} = \{a^n b^n \mid n \geq 0\}.$$

Note that this means that for *all the language classes mentioned in Section 5*, for which PTL-*separability* is *decidable*, it is *undecidable* whether a given language *is* a PTL.

We introduce some additional material to allow for this generalization. Given a language $L$, the *full trio generated by* $L$ is the smallest full trio containing $L$, which we denote by $\mathcal{T}(L)$. Since the intersection of full trios is a full trio and every full trio includes the regular languages, such a class indeed exists. Observe that if $L \neq \emptyset$, then $\mathcal{T}(L)$ consists of precisely those languages of the form $RL$, where $R$ is a rational transduction. In particular, a language $RL$ in $\mathcal{T}(L)$ can be denoted using a representation for $R$. Furthermore, note that $\mathcal{T}(L)$ is always closed under union, because $RL \cup SL = (R \cup S)L$ and the union of rational transductions is again a rational transduction.

**Lemma 6.1** *The universality problem is undecidable for* $\mathcal{T}(L_{ab})$*.*

**Proof:** We reduce Post's Correspondence Problem (PCP) to the (non-)universality problem of $\mathcal{T}(L_{ab})$[3]. An instance of PCP consists in an alphabet $A$, with $A \cap \{0,1\} = \emptyset$, together with two morphisms $\alpha, \beta \colon A^* \to \{0,1\}^*$. The problem asks whether there is a word $w \in A^+$ such that $\alpha(w) = \beta(w)$.

Suppose we are given a PCP instance. From $\alpha$ and $\beta$, we build a language $M_{\alpha,\beta} \in \mathcal{T}(L_{ab})$ over some alphabet $B$ such that $M_{\alpha,\beta} = B^*$ iff there exists no $w \in A^+$ with $\alpha(w) = \beta(w)$. Our language $M_{\alpha,\beta}$ will be of the form $R_\alpha L_{ab} \cup R_\beta L_{ab} \cup K$, where $R_\alpha, R_\beta$ are rational transductions and $K$ is regular. Since $\mathcal{T}(L_{ab})$ is effectively closed under union and includes the regular languages, it contains $M_{\alpha,\beta}$.

Given a morphism $\gamma \colon A^* \to \{0,1\}^*$, we first describe the construction of $R_\gamma$ (which will be instantiated for $\gamma = \alpha, \beta$ to get $R_\alpha, R_\beta$). We want to construct a rational transduction $R_\gamma$ such that

$$R_\gamma L_{ab} = \{uv \mid u \in A^+,\ v \in \{0,1\}^*, v \neq \gamma(u)\}. \tag{6.1}$$

Assume first that we are able to construct such a rational transduction $R_\gamma$ from $\gamma$. We argue that this provides the desired reduction. Indeed, let $B = A \cup \{0,1\}$ and let $K = B^* \setminus (A^+\{0,1\}^*)$. As explained above, $\mathcal{T}(L_{ab})$ effectively contains $M_{\alpha,\beta} = R_\alpha L_{ab} \cup R_\beta L_{ab} \cup K$. Now, in view of the values of $R_\alpha L_{ab}$ and $R_\beta L_{ab}$ given by (6.1), there is no $w \in A^+$ with $\alpha(w) = \beta(w)$ if and only if $M_{\alpha,\beta} = B^*$.

[3] It is worth comparing this proof to that of (Berstel, 1979, Lemma 8.3)

It remains to construct a rational transduction $R_\gamma$ satisfying (6.1). The intuition is that one uses the input word $a^n b^n$ of $L_{ab}$ to produce all words $uv$ where the $n$-th position of $v$ differs from the $n$-th position of $\gamma(u)$. Observe that $\gamma(u) \neq v$ means that

1. either $|\gamma(u)| > |v|$,
2. or $|\gamma(u)| < |v|$,
3. or $\gamma(u)$ and $v$ differ at some common position.

Thus, the rational transduction $R_\gamma$ is the union of three rational transductions. We illustrate its construction for the third case. As it is well-known that rational transductions are realized by finite transducers, *i.e.*, finite automata reading an input and producing an output along transitions (Berstel, 1979), we describe the third part of $R_\gamma$ as such a transducer. For our purpose, it suffices to describe its behavior on inputs of the form $a^n b^n \in L_{ab}$. On such an input, the transducer first outputs the prefix $u = xyz$ of $uv \in R_\gamma L_{ab}$ in three steps:

- It outputs some word $x \in A^*$ while, at the same time, it reads $a^{|\gamma(x)|}$.
- It outputs a symbol $y \in A$ and remembers in its state a position $k$ in $\gamma(y)$ and the symbol $i$ in $\gamma(y)$ at position $k$.
- It outputs some word $z \in A^*$.

The transducer then produces the suffix $v = rst$ of $uv \in R_\gamma L_{ab}$, again in three steps:

- It outputs some word $r \in \{0,1\}^*$ while, at the same time, it reads $b^{|r|}$.
- It outputs a word $s$ that, at position $k$, has a symbol that differs from $i$.
- It outputs some word $t \in \{0,1\}^*$.

Since the input is of the form $a^n b^n$, the above procedure ensures that $|\gamma(x)| = |r|$ and therefore produces precisely the words $uv$ such that $v$ and $\gamma(u)$ differ at some common position. Treating the other two cases accordingly yields a construction of $R_\gamma$ from $\gamma$, which concludes the proof. □

**Proposition 6.2 (Place, 2015a)** *Let $\mathcal{C}$ be a full trio that contains the language $L_{ab}$. Then, testing whether a given language from $\mathcal{C}$ is piecewise testable is undecidable.*

**Proof:** Since $\mathcal{T}(L_{ab})$ is effectively contained in $\mathcal{C}$, it clearly suffices to show that piecewise testability is undecidable for $\mathcal{T}(L_{ab})$.

Let $L$ be a language from $\mathcal{T}(L_{ab})$ over an alphabet $A$. We show that, if we can decide whether $L$ is piecewise testable, then we can also decide whether $L$ is universal. Since universality is undecidable for $\mathcal{T}(L_{ab})$ by Lemma 6.1, we have a contradiction.

Fix $K$ as any language in $\mathcal{T}(L_{ab})$ that is not piecewise testable and $\#$ as a symbol that is not in $A$. We claim that the language $L' = K\#A^* \cup A^*\#L$ is piecewise testable if and only if $L$ is universal. Note that $L'$ belongs to $\mathcal{T}(L_{ab})$ since the latter is effectively closed under union.

If $L$ is universal, then $L' = A^*\#A^*$, which is trivially piecewise testable.

If $L$ is not universal, assume by contradiction that $L'$ is piecewise testable. By hypothesis there is a word $w \notin L$. It is then immediate that $K = L'(\#w)^{-1}$ is also piecewise testable (as piecewise testable languages are closed under residuals) which is a contradiction by choice of $K$. □

Since the context-free languages are a full trio and contain $L_{ab}$, we have the following as one example. However, note that this is also true for all language classes mentioned in Section 5.

**Corollary 6.3** *Testing if a context-free language is piecewise testable is undecidable.*

# 7 Concluding Remarks

Since the decidability results we presented are in strong contrast with the remark of Hunt III quoted in the introduction, we briefly comment on this. What we essentially show is that undecidable emptiness-of-intersection for a class $\mathcal{C}$ does not always imply undecidability for separability of $\mathcal{C}$ with respect to some nontrivial 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 testable). Here, the fact that such intersections can be effectively constructed are, together with a decidable diagonal problem and some mild closure properties, sufficient for decidability.

In terms of future work, we see many interesting directions and new questions. A first set of problems concerns separability by PTL and very related topics. Which language classes have a decidable diagonal problem? Can Theorem 2.6 be extended to also give complexity guarantees? In particular, although we now know that separability of context-free languages by PTL is decidable, we do not know what the complexity of the problem is. For example, it is not known if this problem is elementary or not. Finally, can we find similar characterizations for separability by subclasses of PTL, as considered in (Hofman and Martens, 2015)?

Another interesting set of questions concerns possible generalizations of our result to broader classes of separators. Recently, it was shown that PTL is not the only nontrivial class for which separability is decidable beyond regular languages. In Clemente et al. (2017) and Czerwiński and Lasota (2017) decidability of separability by *regular languages* was proven for classes of integer VASS languages and one counter net languages, respectively. Both these language classes are full trios. Can one formulate natural equivalent conditions (in the style of Theorem 2.6) not only for PTL separability, but for regular separability or separability by other classes? For example, all the cases of which we are aware seem to confirm the following conjecture: under mild robustness conditions, *full trios have decidable regular separability if and only if they have decidable intersection emptiness problem.* Such a result would again be close to the remark of Hunt III that we quoted in the introduction and would make it more precise.

**Acknowledgments** We would like to thank Tomáš Masopust for pointing us to (Hunt III, 1982) 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.

# References

P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.

P. A. Abdulla, A. Collomb-Annichini, A. Bouajjani, and B. Jonsson. Using forward reachability analysis for verification of lossy channel systems. *Formal Methods in System Design*, 25(1):39–65, 2004.

J. Almeida. *Finite semigroups and universal algebra*, volume 3 of *Series in Algebra*. World Scientific Publishing Co., 1994.

J. Almeida, J. Bartoňová, O. Klíma, and M. Kunc. On decidability of intermediate levels of concatenation hierarchies. In *Proceedings of the 19th International Conference on Developments in Language Theory (DLT)*, Lecture Notes in Computer Science, pages 58–70. Springer, 2015.

T. Antonopoulos, D. Hovland, W. Martens, and F. Neven. Deciding twig-definability of node selecting tree automata. In *International Conference on Database Theory (ICDT)*, pages 61–73, 2012.

M. Arfi. Polynomial operations on rational languages. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 198–206, 1987.

M. Benedikt and L. Segoufin. Regular tree languages definable in FO and in $\mathrm{FO}_{\mathrm{mod}}$. *ACM Transactions on Computational Logic*, 11(1):4:1–4:32, 2009.

J. Berstel. *Transductions and context-free languages*. Teubner Stuttgart, 1979.

M. Bojańczyk. Factorization forests. In *International Conference on Developments in Language Theory (DLT)*, pages 1–17, 2009.

M. Bojańczyk and T. Idziaszek. Algebra for infinite forests with an application to the temporal logic EF. In *International Conference on Concurrency Theory (CONCUR)*, pages 131–145, 2009.

M. Bojańczyk, L. Segoufin, and H. Straubing. Piecewise testable tree languages. *Logical Methods in Computer Science*, 8(3), 2012.

R. Bonnet. On the cardinality of the set of initial intervals of a partially ordered set. In *Infinite sets, Keszthely, 1973*, volume 3, pages 189–198. North-Holland, 1975.

R. Bonnet, A. Finkel, J. Leroux, and M. Zeitoun. Place-boundedness for vector addition systems with one zero-test. In K. Lodaya and M. Mahajan, editors, *FSTTCS 2010*, volume 8 of *LIPIcs*, pages 192–203. Leibniz-Zentrum für Informatik, 2010.

R. Bonnet, A. Finkel, J. Leroux, and M. Zeitoun. Model checking vector addition systems with one zero-test. *Logical Methods in Computer Science*, 8(2), 2012.

J. A. Brzozowski and I. Simon. Characterizations of locally testable events. In *Symposium on Switching and Automata Theory (SWAT)*, 1971.

L. Clemente, P. Parys, S. Salvati, and I. Walukiewicz. The diagonal problem for higher-order recursion schemes is decidable. In *Symposium on Logic in Computer Science (LICS)*, pages 96–105. ACM, 2016.

L. Clemente, W. Czerwiński, S. Lasota, and C. Paperman. Regular separability of Parikh automata. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 117:1–117:13, 2017.

T. Colcombet. Factorization forests for infinite words and applications to countable scattered linear orderings. *Theoretical Computer Science*, 411(4-5):751–764, 2010.

T. Colcombet. The factorisation forest theorem. To appear in the handbook "Automata: from Mathematics to Applications", 2017.

W. Czerwiński and S. Lasota. Regular separability of one counter automata. In *Symposium on Logic in Computer Science (LICS)*, pages 1–12, 2017.

W. Czerwiński, C. David, K. Losemann, and W. Martens. Deciding definability by deterministic regular expressions. In *Foundations of Software Science and Computation Structures (FOSSACS)*, pages 289–304, 2013a.

W. Czerwiński, W. Martens, and T. Masopust. Efficient separability of regular languages by subsequences and suffixes. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 150–161, 2013b.

W. Czerwiński, W. Martens, L. van Rooijen, and M. Zeitoun. A note on decidable separability by piecewise testable languages. In *Fundamentals of Computation Theory (FCT)*, pages 173–185, 2015.

J. Dassow and G. Păun. *Regulated rewriting in formal language theory*. Springer, Berlin, 1989.

P. Erdős and A. Tarski. Families of mutually exclusive sets. *Annals of Mathematics*, 44(2):315–329, 1943.

A. Finkel. A generalization of the procedure of Karp and Miller to well structured transition systems. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 499–508, 1987.

A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part II: Complete WSTS. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 188–199, 2009.

A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1-2):63–92, 2001.

R. Fraïssé. *Theory of Relations*, volume 145 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2000.

S. Ginsburg and S. A. Greibach. Abstract families of languages. In *Symposium on Switching and Automata Theory (SWAT/FOCS)*, pages 128–139, 1967.

S. Ginsburg and E. H. Spanier. Semigroups, Presburger formulas, and languages. *Pacific Journal of Mathematics*, 16:285–296, 1966.

C. Glaßer and H. Schmitz. Languages of dot-depth 3/2. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 555–566, 2000.

J. Goubault-Larrecq and S. Schmitz. Deciding piecewise testable separability for regular tree languages. In *43rd International Colloquium on Automata, Languages, and Programming, (ICALP)*, volume 55 of *LIPIcs*, pages 97:1–97:15. Leibniz-Zentrum fuer Informatik, 2016.

S. Greibach. A note on undecidable properties of formal languages. *Math. Systems Theory*, 2(1):1–6, 1968.

H. Gruber, M. Holzer, and M. Kutrib. The size of Higman-Haines sets. *Theoretical Computer Science*, 387(2):167–176, 2007.

P. Habermehl, R. Meyer, and H. Wimmel. The downward-closure of Petri net languages. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 466–477, 2010.

M. Hack. Petri net languages. Technical report, Massachusetts Institute of Technology, 1976. Computation Structures Group Memo 124.

M. Hague, J. Kochems, and C.-H. L. Ong. Unboundedness and downward closures of higher-order pushdown automata. In *Proc. of the Symposium on Principles of Programming Languages (POPL)*, pages 151–163, 2016.

L. H. Haines. On free monoids partially ordered by embedding. *Journal of Combinatorial Theory*, 6(1): 94–98, 1969.

G. Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, s3-2(1):326–336, 1952.

P. Hofman and W. Martens. Separability by short subsequences and subwords. In *International Conference on Database Theory (ICDT)*, pages 230–246, 2015.

H. B. Hunt III. On the decidability of grammar problems. *Journal of the ACM*, 29(2):429–447, 1982.

O. H. Ibarra. Reversal-bounded multi counter machines and their decision problems. *Journal of the ACM*, 25(1):116–133, 1978.

M. Jantzen. On the hierarchy of Petri net languages. *RAIRO Informatique Théorique*, 13(1), 1979.

P. Jullien. *Contribution à l'étude des types d'ordres dispersés*. PhD thesis, Université de Marseille, 1969.

P. Karandikar, M. Niewerth, and Ph. Schnoebelen. On the state complexity of closures and interiors of regular languages with subwords and superwords. *Theoretical Computer Science*, 2015.

O. Klíma. Piecewise testable languages via combinatorics on words. *Discrete Mathematics*, 311(20): 2124–2127, 2011.

R. Knast. A semigroup characterization of dot-depth one languages. *RAIRO - Theoretical Informatics and Applications*, 17(4):321–330, 1983.

S. R. Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *ACM Symposium on Theory of Computing (STOC)*, pages 267–281, 1982.

M. Kufleitner. The height of factorization forests. In *Mathematical Foundations of Computer Science (MFCS)*, pages 443–454, 2008.

S. La Torre, A. Muscholl, and I. Walukiewicz. Safety of parametrized asynchronous shared-memory systems is almost always decidable. In *International Conference on Concurrency Theory (CONCUR)*, volume 42, pages 72–84, 2015.

J. Leroux. The general vector addition system reachability problem by Presburger inductive invariants. *Logical Methods in Computer Science*, 6(3), 2010.

J. Leroux and S. Schmitz. Demystifying reachability in vector addition systems. In *Symposium on Logic in Computer Science (LICS)*, 2015.

E. W. Mayr. An algorithm for the general Petri net reachability problem. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 238–246. ACM, 1981.

E. W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM Journal on Computing*, 13(3):441–460, 1984.

R. Mayr. Undecidable problems in unreliable computations. *Theoretical Computer Science*, 297(1-3): 337–354, 2003.

R. McNaughton. Algebraic decision procedures for local testability. *Mathematical Systems Theory*, 8(1): 60–76, 1974.

M. Nivat. Transductions des langages de Chomsky. *Annales de l'institut Fourier*, 18(1):339–455, 1968.

R. Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966.

J.-E. Pin and P. Weil. Ponynominal closure and unambiguous product. *Theory Comput. Syst.*, 30(4): 383–422, 1997.

T. Place. Personal communication, 2015a.

T. Place. Separating regular languages with two quantifier alternations. In *Symposium on Logic in Computer Science (LICS)*, pages 202–213. IEEE, 2015b.

T. Place and L. Segoufin. Deciding definability in $\mathrm{FO}^2(<_v, <_h)$ on trees. *Logical Methods in Computer Science (LMCS)*, 11(3), 2015.

T. Place and M. Zeitoun. Going higher in the first-order quantifier alternation hierarchy on words. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 342–353, 2014a.

T. Place and M. Zeitoun. Separating regular languages with first-order logic. In *Joint Meeting of the Conference on Computer Science Logic (CSL) and the Symposium on Logic in Computer Science (LICS)*, pages 75:1–75:10. ACM, 2014b.

T. Place and M. Zeitoun. The covering problem: a unified approach for investigating the expressive power of logics. In *Mathematical Foundations of Computer Science (MFCS)*, volume 58 of *LIPIcs*, pages 78:1–78:14, 2016a.

T. Place and M. Zeitoun. Separating regular languages with first-order logic. *Logical Methods in Computer Science*, 12(1), 2016b.

T. Place and M. Zeitoun. Separation for dot-depth two. In *Symposium on Logic in Computer Science (LICS)*. IEEE, 2017.

T. Place, L. van Rooijen, and M. Zeitoun. Separating regular languages by locally testable and locally threshold testable languages. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 363–375, 2013a.

T. Place, L. van Rooijen, and M. Zeitoun. Separating regular languages by piecewise testable and unambiguous languages. In *Mathematical Foundations of Computer Science (MFCS)*, pages 729–740, 2013b.

T. Place, L. van Rooijen, and M. Zeitoun. On separation by locally testable and locally threshold testable languages. *Logical Methods in Computer Science*, 10(3:24):1–28, 2014.

S. Schmitz and Ph. Schnoebelen. The power of well-structured systems. In *International Conference on Concurrency Theory (CONCUR)*, pages 5–24, 2013.

Ph. Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letters*, 83(5):251–261, 2002.

M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2): 190–194, 1965.

H. Seki, T. Matsumura, M. Fujii, and T. Kasami. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):191–229, 1991.

I. Simon. Piecewise testable events. In *Automata Theory and Formal Languages*, pages 214–222, 1975.

I. Simon. Factorization forests of finite height. *Theoretical Computer Science*, 72(1):65–94, 1990.

H. Straubing. Semigroups and languages of dot-depth two. *Theoretical Computer Science*, 58:361–378, 1988.

T. G. Szymanski and J. H. Williams. Noncanonical extensions of bottom-up parsing techniques. *SIAM Journal on Computing*, 5(2), 1976.

P. Tesson and D. Thérien. Diamonds are forever: The variety DA. In *Semigroups, Algorithms, Automata and Languages*, pages 475–500, 2002.

D. Thérien and T. Wilke. Over words, two variables are as powerful as one quantifier alternation. In *Symposium on Theory of Computing (STOC)*, pages 234–240, 1998.

Y. Zalcstein. Locally testable languages. *Journal of Computer and System Sciences*, 6(2):151–167, 1972.

G. Zetzsche. An approach to computing downward closures. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 440–451, 2015a.

G. Zetzsche. Computing downward closures for stacked counter automata. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 30, pages 743–756, 2015b.