# Unboundedness Problems for Formal Languages

**Georg Zetzsche**
Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany

**Abstract**

Informally, unboundedness problems are decision problems that ask about the existence of infinitely many words (satisfying certain properties) in a formal language. For example: Is a given language infinite? Or: Does a given language have super-polynomial growth? These came into focus in recent years because of their connections to downward closure computation and separability problems. Although unboundedness problems may seem difficult at first, it turns out that there are techniques that are at the same time conceptually very simple, but also apply to a surprisingly wide variety of language classes. The talk will survey recent results (and techniques) concerning unboundedness problems.

**2012 ACM Subject Classification** Theory of computation → Models of computation

**Keywords and phrases** Decidability, formal languages, unifying frameworks, downward closure, separability

**Digital Object Identifier** 10.4230/LIPIcs.FSTTCS.2025.2

**Category** Invited Talk

**Funding** Funded by the European Union (ERC, FINABIS, 101077902). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

## 1 Introduction

A classic theme of automata theory is to understand, given (i) a particular model of automata and (ii) a decision problem, whether this decision problem is decidable for this model. For example, the emptiness problem is decidable for the model of non-deterministic automata. Sometimes, it is even possible to obtain *general decidability results*: This means, they are of one of the following forms:

**Type I: Class of models** For a class of models, the result shows decidability of a particular problem.

**Type II: Class of problems** For a class of decision problems, the result shows decidability for a particular model of automata.

A well-known example of a result of type I is the fact that for well-structured transition systems (under mild assumptions), the coverability and termination problem are decidable [2, 34]. However, in this talk, we will focus on *unboundedness problems*: This is not a precise notion, but intuitively, these are problems concerning formal languages (represented by some automaton) that ask about the existence of *infinitely many words.*

The following are examples of unboundedness problems:

**Infinity** Given a language $L \subseteq \Sigma^*$ (represented by some automaton), does $L$ contain infinitely many words?

**Downward closure computation** A classic result of Higman [38] and Haines [37] is that the set $L\downarrow$ of (scattered) subwords of any language $L \subseteq \Sigma^*$ is regular. The problem of downward closure computation is, given a language $L \subseteq \Sigma^*$, compute an NFA for $L\downarrow$.

45th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2025).
Editors: C. Aiswarya, Ruta Mehta, and Subhajit Roy; Article No. 2; pp. 2:1–2:10
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

**Boundedness** A language $L \subseteq \Sigma^*$ is *bounded* if there are words $w_1, \ldots, w_n \in \Sigma^*$ with $L \subseteq w_1^* \cdots w_n^*$. The boundedness problem asks, given a language $L \subseteq \Sigma^*$, whether $L$ is bounded.

Here, downward closure computation is an unboundedness problem, because e.g. determining that $L\!\downarrow = a^*$ means in particular concluding that $L$ contains arbitrarily many occurrences of $a$. Moreover, concluding that a language $L$ is not bounded means proving in particular that $L$ is infinite.

We focus on unboundedness problems for two reasons. On the one hand, they have attracted significant attention in recent years: For example, computing downward closures is often useful for the verification for complex types of automata (see, e.g. [7,12,13,15,16,45,52]). Another application is separability problems, which have been studied intensively in recent years, either as separability by PTL [31,35] (see below for a definition), regular languages [8,14, 17,20,24,29,30,32,32,40–43,59], or other kinds of languages [1,18,21,23,54–56]. Separability problems are closely intertwined with unboundedness problems (see, e.g. [31]) and can be viewed as unboundedness problems themselves. The second reason is that unboundedness problems have turned out to be particularly well-suited for general results as above.

Here, we will briefly outline the results mentioned in the talk.

## 2 Downward closures, PTL separability, and simultaneous unboundedness

Our first result is about computing downward closures, and separability by piecewise testable languages (PTL). The former result is explained above. The PTL separability problem asks, given languages $K, L \subseteq \Sigma^*$ whether there exists a piecewise testable language $S$ such that $K \subseteq S$ and $L \cap S = \emptyset$. Here, a language $S \subseteq \Sigma^*$ is *piecewise testable* if it is a Boolean combination of languages of the form $\Sigma^* a_1 \Sigma^* \cdots a_n \Sigma^*$, where $n \geq 0$, $a_1, \ldots, a_n \in \Sigma$.

For many models of automata, their recognized class of languages is closed under *rational transductions*: Those are relations described by automata, called *transducers*, with one input tape and one output tape. A run of the transducer thus reads an input word and produces an output word. Applying the transducer to a language $L$ then yields a new language consisting of all ouput words of transducer runs that read a word from $L$.

In this result, the model of automata is represented by its *language class*, i.e. the collection of languages recognized by automata of our model. Such a language class $\mathcal{C}$ is a *full trio* if for every language $L$ from $\mathcal{C}$, applying a transducer to $L$ will again yield a language from $\mathcal{C}$.

Finally, our result involves the *simultaneous unbounded problem* (SUP), which asks, given a language $L \subseteq a_1^* \cdots a_n^*$, whether for every $k \geq 0$, there is a word $a_1^{x_1} \cdots a_n^{x_n} \in L$ such that $x_1, \ldots, x_n \geq k$.

We are ready to state the first result, which characterizes computability of downward closures and decidability of PTL separability, by the much simpler SUP. It is combination of a characterization of computable downward closures [60, Theorem 1] and a characterization of decidable PTL-separability due to Czerwiński, Martens, Rooijen, Zeitoun, and the author [31, Theorem 2.6]:

▶ **Theorem 2.1.** *For every full trio $\mathcal{C}$, the following are equivalent:*

1. *Downward closures are computable for $\mathcal{C}$.*
2. *PTL separability is decidable for $\mathcal{C}$.*
3. *The SUP is decidable for $\mathcal{C}$.*

Motivated by this result, the SUP was then shown to be decidable for indexed languages [60], more generally for higher-order pushdown automata [36], and even for more generally for higher-order recursion schemes [22] (see also [10, 53]). A similar approach was then also developed for computing tree downward closures of higher-order schemes [9].

## 3 Unboundedness predicates for VASS

The second result is about a specific model, and a class of decision problems. The model is the prominent model of *vector addition systems with states* (VASS), which are automata equipped with multiple counters that (i) hold natural numbers and (ii) cannot be tested for zero during the run. Hence, the counters can only be incremented and decremented. In the end, a run is *accepting* if all counters are zero.

The class of decision problems is the class of "unboundedness predicates": An *unboundedness predicate* is a predicate $\mathfrak{p}$ on languages such that

1. If $K \subseteq L$ and $\mathfrak{p}(K)$ holds, then $\mathfrak{p}(L)$ holds as well.
2. If $\mathfrak{p}(F(KL))$ holds, then $\mathfrak{p}(F(K))$ or $\mathfrak{p}(F(L))$ holds.
3. If $\mathfrak{p}(K \cup L)$ holds, then $\mathfrak{p}(K)$ or $\mathfrak{p}(L)$ holds.

Here, $F(L) := \{v \in \Sigma^* \mid \exists u, w \in \Sigma^* \colon uvw \in L\}$ denotes the set of *factors* of a language $L \subseteq \Sigma^*$. Examples of such unboundedness predicates are:

**Infinity** the predicate $\mathfrak{p}_\infty$, where $\mathfrak{p}_\infty(L)$ if and only if $L$ is infinite,

**Not being bounded** the predicate $\mathfrak{p}_{\neg\mathsf{b}}$, where $\mathfrak{p}_{\neg\mathsf{b}}(L)$ if and only if $L$ is not bounded,

**Factor universality** the predicate $\mathfrak{p}_{\mathsf{funi}}$, where $\mathfrak{p}_{\mathsf{funi}}(L)$ if and only if every word in $\Sigma^*$ appears as a factor in $L$

We say that an unboundedness predicate $\mathfrak{p}$ is *decidable* for a language class $\mathcal{C}$ if given a language $L$ from $\mathcal{C}$, we can decide whether $\mathfrak{p}(F(L))$ holds. Hence, note that unboundedness predicates are always checked for the set of factors of the input language.

The following was shown by Czerwiński, Hofman, and the author [28, Theorem 3.1]:

▶ **Theorem 3.1.** *Let $\mathfrak{p}$ be an unboundedness predicate. Then $\mathfrak{p}$ is decidable for languages of VASS if and only if $\mathfrak{p}$ is decidable for regular languages.*

In fact, [28, Theorem 3.1] is stronger, since it is about a more general notion of unboundedness predicate, which also generalizes the SUP. In [11, Theorem 2], the authors show a version of Theorem 3.1 with complexity guarantees for automata with integer counters (i.e. which can attain negative values) and with/without a pushdown store.

## 4 Amalgamation

Our next result provides a sufficient condition for an automata model to have decidability of a particular set of decision problems. This sufficient condition is called *(concatenative) amalgamation*. The result is in the same spirit as the classical results about well-structured transition systems: In the latter, one assumes that the set of *configurations* in an automaton is equipped with a well-quasi ordering such that the reachability relation is (in some particular sense) compatible with the ordering. This leads to several decidability results [2, 30, 34].

In contrast, in *(concatenative) amalgamation systems*, one assumes a well-quasi ordering $\trianglelefteq$ on the set of *runs* of an automaton. Here if $\rho_0 \trianglelefteq \rho_1$, then the word read by $\rho_0$ must be a subword of the word read by $\rho_1$. Moreover, whenever one has runs $\rho_0, \rho_1, \rho_2$ such that $\rho_0 \trianglelefteq \rho_1$ and $\rho_0 \trianglelefteq \rho_2$, then there is a fourth run $\rho_3$ with $\rho_1 \trianglelefteq \rho_3$ and $\rho_2 \trianglelefteq \rho_3$ such that, roughly speaking, the word of $\rho_3$ is obtained by concatenating the gap words between $\rho_0$ and

$\rho_1$ and between $\rho_0$ and $\rho_2$. We say that a language class *has (concatenative) amalgamation* if each of its languages is recognized by a concatenative amalgamation system. For a detailed definition, see [5].

The idea for this notion comes from the theory of VASS: Jančar has introduced a well-quasi ordering on runs [39]. Based on ideas of Leroux [48], Leroux and Schmitz [50] observed that runs in VASS can be amalgamated as above. In [49], this was extended to pushdown VASS, which are VASS that also have access to a pushdown store. The amalgamation technique has then been used in structural [33, 49] and in algorithmic [21, 32] results on (pushdown) VASS.

**Amalgamation as an abstract notion.** The paper [5] proposes to view amalgamation as an abstract property that an automata model can satisfy. The paper shows that amalgamation is not only available in (pushdown) VASS, but in a hierarchy of infinite-state automata beyond pushdown VASS. In particular, all models in the unifying framework of valence automata [61] with decidable emptiness satisfy the amalgamation property.

Moreover, the paper [5] shows that many unboundedness problems can actually be shown decidable using only amalgamation, and with *surprisingly simple proofs*. The main algorithmic result of that paper [5, Main Theorem A] is the following:

▶ **Theorem 4.1.** *If $\mathcal{C}$ is a full trio with concatenative amalgamation, then the following are equivalent:*

1. *The SUP is decidable.*
2. *Downward closures are computable.*
3. *Separability by piecewise testable languages is decidable.*
4. *Boundedness is decidable.*
5. *Languages over one letter are effectively regular.*
6. *Priority downward closures are computable.*
7. *Emptiness is decidable.*

Here, priority downward closures are a regular overapproximation of languages that preserves more information than downward closures [6].

## 5 Bounded treewidth, multiple context-free languages, and downward closures

Finally, let us briefly mention some very recent work [3]. It concerns the notion of *bounded-treewidth systems*, which is a unifying framework for deciding *reachability* (and other properties) of infinite-state automata.

A seminal paper by Madhusudan and Parlato [51], together with a line of follow-up work [4, 26, 27, 46] revealed that many models of infinite-state automata (particularly those obtained as underapproximations of multi-pushdown automata) are *bounded-treewidth* [19, 44, 47, 57]. This means, the executions can be represented as graphs that have small tree decompositions. Therefore, decidability of these models can be uniformly explained by Courcelle's Theorem [25], which says that among structures of bounded treewidth, satisfiability of any property definable in monadic second-order logic (MSO) is decidable in polynomial time.

While this uniform framework is useful for yes-or-no queries like reachability, it yields little information about languages.

The results in [3] bring some light into this. It analyzes the languages in the slightly smaller class of *bounded-special-treewidth systems*: Bounded special treewidth is a somewhat stronger restriction that bounded treewidth, but coincides with bounded treewidth on all bounded-degree graphs. In fact, all bounded-treewidth systems studied in the literature have in fact bounded special treewidth.

It is shown in [3] that the languages of bounded-special-treewidth systems are exactly those generated by *multiple context-free grammars* (MCFG), a formalism originally developed in computational linguistics [58]. Intuitively, in an derivation of an MCFG, a non-terminal generates a tuple of words rather than a single word as in a context-free grammar.

In addition, it is shown in [3] that the downward closure of MCFG is computable in doubly exponential time (and that this is asymptotically optimal). Hence, this yields a downward closure computation algorithm for all bounded-special-treewidth systems.

## References

1 Parosh Aziz Abdulla, Mohamed Faouzi Atig, Vrunda Dave, and Shankara Narayanan Krishna. On the separability problem of string constraints. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPIcs*, pages 16:1–16:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.CONCUR.2020.16.

2 Parosh Aziz Abdulla, Kārlis Čerāns, Bengt Jonsson, and Yih-Kuen Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Inf. Comput.*, 160(1-2):109–127, 2000. doi:10.1006/inco.1999.2843.

3 C. Aiswarya, Pascal Baumann, Prakash Saivasan, Lia Schütze, and Georg Zetzsche. Bounded treewidth, multiple context-free grammars, and downward closures, 2026. To appear.

4 C. Aiswarya, Paul Gastin, and K. Narayan Kumar. Verifying communicating multi-pushdown systems via split-width. In Franck Cassez and Jean-François Raskin, editors, *Automated Technology for Verification and Analysis - 12th International Symposium, ATVA 2014, Sydney, NSW, Australia, November 3-7, 2014, Proceedings*, volume 8837 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2014. doi:10.1007/978-3-319-11936-6_1.

5 Ashwani Anand, Sylvain Schmitz, Lia Schütze, and Georg Zetzsche. Verifying unboundedness via amalgamation. In Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 4:1–4:15. ACM, 2024. doi:10.1145/3661814.3662133.

6 Ashwani Anand and Georg Zetzsche. Priority downward closures. In Guillermo A. Pérez and Jean-François Raskin, editors, *34th International Conference on Concurrency Theory, CONCUR 2023, September 18-23, 2023, Antwerp, Belgium*, volume 279 of *LIPIcs*, pages 39:1–39:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.CONCUR.2023.39.

7 Mohamed Faouzi Atig, Ahmed Bouajjani, and Shaz Qadeer. Context-bounded analysis for concurrent programs with dynamic creation of threads. *Log. Methods Comput. Sci.*, 7(4), 2011. doi:10.2168/LMCS-7(4:4)2011.

8 Pablo Barceló, Diego Figueira, and Rémi Morvan. Separating automatic relations. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023, August 28 to September 1, 2023, Bordeaux, France*, volume 272 of *LIPIcs*, pages 17:1–17:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.MFCS.2023.17.

9 David Barozzini, Lorenzo Clemente, Thomas Colcombet, and Pawel Parys. Cost automata, safe schemes, and downward closures. *Fundam. Informaticae*, 188(3):127–178, 2022. doi:10.3233/FI-222145.

10 David Barozzini, Pawel Parys, and Jan Wroblewski. Unboundedness for recursion schemes: A simpler type system. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 112:1–112:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.112.

11 Pascal Baumann, Flavio D'Alessandro, Moses Ganardi, Oscar H. Ibarra, Ian McQuillan, Lia Schütze, and Georg Zetzsche. Unboundedness problems for machines with reversal-bounded

counters. In Orna Kupferman and Pawel Sobocinski, editors, *Foundations of Software Science and Computation Structures - 26th International Conference, FoSSaCS 2023, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2023, Paris, France, April 22-27, 2023, Proceedings*, volume 13992 of *Lecture Notes in Computer Science*, pages 240–264. Springer, 2023. doi:10.1007/978-3-031-30829-1_12.

12 Pascal Baumann, Moses Ganardi, Rupak Majumdar, Ramanathan S. Thinniyam, and Georg Zetzsche. Checking refinement of asynchronous programs against context-free specifications. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 110:1–110:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ICALP.2023.110.

13 Pascal Baumann, Moses Ganardi, Rupak Majumdar, Ramanathan S. Thinniyam, and Georg Zetzsche. Context-bounded analysis of concurrent programs (invited talk). In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 3:1–3:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ICALP.2023.3.

14 Pascal Baumann, Eren Keskin, Roland Meyer, and Georg Zetzsche. Separability in Büchi VASS and singly non-linear systems of inequalities. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, July 8-12, 2024, Tallinn, Estonia*, volume 297 of *LIPIcs*, pages 126:1–126:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ICALP.2024.126.

15 Pascal Baumann, Rupak Majumdar, Ramanathan S. Thinniyam, and Georg Zetzsche. The complexity of bounded context switching with dynamic thread creation. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 111:1–111:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.ICALP.2020.111.

16 Pascal Baumann, Rupak Majumdar, Ramanathan S. Thinniyam, and Georg Zetzsche. Context-bounded verification of thread pools. *Proc. ACM Program. Lang.*, 6(POPL):1–28, 2022. doi:10.1145/3498678.

17 Pascal Baumann, Roland Meyer, and Georg Zetzsche. Regular separability in Büchi VASS. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 9:1–9:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.STACS.2023.9.

18 Mikolaj Bojanczyk. It is undecidable if two regular tree languages can be separated by a deterministic tree-walking automaton. *Fundam. Informaticae*, 154(1-4):37–46, 2017. doi:10.3233/FI-2017-1551.

19 Luca Breveglieri, Alessandra Cherubini, Claudio Citrini, and Stefano Crespi-Reghizzi. Multi-push-down languages and grammars. *Int. J. Found. Comput. Sci.*, 7(3):253–292, 1996. doi:10.1142/S0129054196000191.

20 Lorenzo Clemente, Wojciech Czerwiński, Slawomir Lasota, and Charles Paperman. Regular Separability of Parikh Automata. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 117:1–117:13, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2017.117.

21 Lorenzo Clemente, Wojciech Czerwiński, Slawomir Lasota, and Charles Paperman. Separability of reachability sets of vector addition systems. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017,*

*Hannover, Germany*, volume 66 of *LIPIcs*, pages 24:1–24:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. `doi:10.4230/LIPICS.STACS.2017.24`.

22 Lorenzo Clemente, Pawel Parys, Sylvain Salvati, and Igor Walukiewicz. The diagonal problem for higher-order recursion schemes is decidable. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 96–105. ACM, 2016. `doi:10.1145/2933575.2934527`.

23 Lorenzo Clemente and Michal Skrzypczak. Deterministic and game separability for regular languages of infinite trees. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 126:1–126:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPICS.ICALP.2021.126`.

24 Elias Rojas Collins, Chris Köcher, and Georg Zetzsche. The complexity of separability for semilinear sets and parikh automata. In Pawel Gawrychowski, Filip Mazowiecki, and Michal Skrzypczak, editors, *50th International Symposium on Mathematical Foundations of Computer Science, MFCS 2025, August 25-29, 2025, Warsaw, Poland*, volume 345 of *LIPIcs*, pages 38:1–38:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. `doi:10.4230/LIPICS.MFCS.2025.38`.

25 Bruno Courcelle. The monadic second-order logic of graphs, II: infinite graphs of bounded width. *Math. Syst. Theory*, 21(4):187–221, 1989. `doi:10.1007/BF02088013`.

26 Aiswarya Cyriac. *Verification of communicating recursive programs via split-width. (Vérification de programmes récursifs et communicants via split-width)*. PhD thesis, École normale supérieure de Cachan, France, 2014. URL: `https://tel.archives-ouvertes.fr/tel-01015561`.

27 Aiswarya Cyriac, Paul Gastin, and K. Narayan Kumar. MSO decidability of multi-pushdown systems via split-width. In Maciej Koutny and Irek Ulidowski, editors, *CONCUR 2012 - Concurrency Theory - 23rd International Conference, CONCUR 2012, Newcastle upon Tyne, UK, September 4-7, 2012. Proceedings*, volume 7454 of *Lecture Notes in Computer Science*, pages 547–561. Springer, 2012. `doi:10.1007/978-3-642-32940-1_38`.

28 Wojciech Czerwinski, Piotr Hofman, and Georg Zetzsche. Unboundedness problems for languages of vector addition systems. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 119:1–119:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. `doi:10.4230/LIPICS.ICALP.2018.119`.

29 Wojciech Czerwiński and Slawomir Lasota. Regular separability of one counter automata. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017. `doi:10.1109/LICS.2017.8005079`.

30 Wojciech Czerwinski, Slawomir Lasota, Roland Meyer, Sebastian Muskalla, K. Narayan Kumar, and Prakash Saivasan. Regular separability of well-structured transition systems. In Sven Schewe and Lijun Zhang, editors, *29th International Conference on Concurrency Theory, CONCUR 2018, September 4-7, 2018, Beijing, China*, volume 118 of *LIPIcs*, pages 35:1–35:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. `doi:10.4230/LIPICS.CONCUR.2018.35`.

31 Wojciech Czerwinski, Wim Martens, Lorijn van Rooijen, Marc Zeitoun, and Georg Zetzsche. A characterization for decidable separability by piecewise testable languages. *Discret. Math. Theor. Comput. Sci.*, 19(4), 2017. `doi:10.23638/DMTCS-19-4-1`.

32 Wojciech Czerwinski and Georg Zetzsche. An approach to regular separability in vector addition systems. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 341–354. ACM, 2020. `doi:10.1145/3373718.3394776`.

33 Alain Finkel, Shankara Narayanan Krishna, Khushraj Madnani, Rupak Majumdar, and Georg Zetzsche. Counter machines with infrequent reversals. In Patricia Bouyer and Srikanth Srinivasan, editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2023, December 18-20, 2023, IIIT Hyderabad, Telangana, India*, volume 284 of *LIPIcs*, pages 42:1–42:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.FSTTCS.2023.42.

34 Alain Finkel and Ph Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1-2):63–92, 2001. doi:10.1016/S0304-3975(00)00102-X.

35 Jean Goubault-Larrecq and Sylvain Schmitz. Deciding piecewise testable separability for regular tree languages. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 97:1–97:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.ICALP.2016.97.

36 Matthew Hague, Jonathan Kochems, and C.-H. Luke Ong. Unboundedness and downward closures of higher-order pushdown automata. In *POPL 2016: Principles of Programming Languages*, pages 151–163. ACM, 2016. doi:10.1145/2837614.2837627.

37 Leonard H Haines. On free monoids partially ordered by embedding. *Journal of Combinatorial Theory*, 6(1):94–98, 1969. doi:10.1016/S0021-9800(69)80111-0.

38 Graham Higman. Ordering by divisibility in abstract algebras. *Proceedings of The London Mathematical Society*, pages 326–336, 1952. doi:10.1112/PLMS/S3-2.1.326.

39 Petr Jancar. Decidability of a temporal logic problem for Petri nets. *Theor. Comput. Sci.*, 74(1):71–93, 1990. doi:10.1016/0304-3975(90)90006-4.

40 Eren Keskin and Roland Meyer. Separability and non-determinizability of WSTS. In Guillermo A. Pérez and Jean-François Raskin, editors, *34th International Conference on Concurrency Theory, CONCUR 2023, September 18-23, 2023, Antwerp, Belgium*, volume 279 of *LIPIcs*, pages 8:1–8:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.CONCUR.2023.8.

41 Eren Keskin and Roland Meyer. On the separability problem of VASS reachability languages. In Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 49:1–49:14. ACM, 2024. doi:10.1145/3661814.3662116.

42 Chris Köcher and Georg Zetzsche. Regular separators for VASS coverability languages. In Patricia Bouyer and Srikanth Srinivasan, editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2023, December 18-20, 2023, IIIT Hyderabad, Telangana, India*, volume 284 of *LIPIcs*, pages 15:1–15:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.FSTTCS.2023.15.

43 Eryk Kopczynski. Invisible pushdown languages. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 867–872. ACM, 2016. doi:10.1145/2933575.2933579.

44 Salvatore La Torre, Parthasarathy Madhusudan, and Gennaro Parlato. A robust class of context-sensitive languages. In *22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10-12 July 2007, Wroclaw, Poland, Proceedings*, pages 161–170. IEEE Computer Society, 2007. doi:10.1109/LICS.2007.9.

45 Salvatore La Torre, Anca Muscholl, and Igor Walukiewicz. Safety of parametrized asynchronous shared-memory systems is almost always decidable. In Luca Aceto and David de Frutos-Escrig, editors, *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, volume 42 of *LIPIcs*, pages 72–84. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICS.CONCUR.2015.72.

46 Salvatore La Torre, Margherita Napoli, and Gennaro Parlato. A unifying approach for multistack pushdown automata. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 - 39th International*

*Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, volume 8634 of *Lecture Notes in Computer Science*, pages 377–389. Springer, 2014. doi:10.1007/978-3-662-44522-8_32.

47 Salvatore La Torre, Margherita Napoli, and Gennaro Parlato. Reachability of scope-bounded multistack pushdown systems. *Inf. Comput.*, 275:104588, 2020. doi:10.1016/J.IC.2020.104588.

48 Jérôme Leroux. Vector addition system reachability problem: a short self-contained proof. In Thomas Ball and Mooly Sagiv, editors, *Proceedings of the 38th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2011, Austin, TX, USA, January 26-28, 2011*, pages 307–316. ACM, 2011. doi:10.1145/1926385.1926421.

49 Jérôme Leroux, M. Praveen, Philippe Schnoebelen, and Grégoire Sutre. On functions weakly computable by pushdown Petri nets and related systems. *Log. Methods Comput. Sci.*, 15(4), 2019. doi:10.23638/LMCS-15(4:15)2019.

50 Jérôme Leroux and Sylvain Schmitz. Demystifying reachability in vector addition systems. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 56–67. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.16.

51 P. Madhusudan and Gennaro Parlato. The tree width of auxiliary storage. *SIGPLAN Not.*, 46(1):283–294, 2011. doi:10.1145/1925844.1926419.

52 Rupak Majumdar, Ramanathan S. Thinniyam, and Georg Zetzsche. General decidability results for asynchronous shared-memory programs: Higher-order and beyond. *Log. Methods Comput. Sci.*, 18(4), 2022. doi:10.46298/LMCS-18(4:2)2022.

53 Pawel Parys. The complexity of the diagonal problem for recursion schemes. In Satya V. Lokam and R. Ramanujam, editors, *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2017, December 11-15, 2017, Kanpur, India*, volume 93 of *LIPIcs*, pages 45:1–45:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.FSTTCS.2017.45.

54 Thomas Place and Marc Zeitoun. Separation for dot-depth two. *Log. Methods Comput. Sci.*, 17(3), 2021. doi:10.46298/LMCS-17(3:24)2021.

55 Thomas Place and Marc Zeitoun. A generic polynomial time approach to separation by first-order logic without quantifier alternation. In Anuj Dawar and Venkatesan Guruswami, editors, *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022, December 18-20, 2022, IIT Madras, Chennai, India*, volume 250 of *LIPIcs*, pages 43:1–43:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.FSTTCS.2022.43.

56 Thomas Place and Marc Zeitoun. Group separation strikes back. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–13. IEEE, 2023. doi:10.1109/LICS56636.2023.10175683.

57 Shaz Qadeer and Jakob Rehof. Context-bounded model checking of concurrent software. In Nicolas Halbwachs and Lenore D. Zuck, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 11th International Conference, TACAS 2005, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2005, Edinburgh, UK, April 4-8, 2005, Proceedings*, volume 3440 of *Lecture Notes in Computer Science*, pages 93–107. Springer, 2005. doi:10.1007/978-3-540-31980-1_7.

58 Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context-free grammars. *Theor. Comput. Sci.*, 88(2):191–229, 1991. doi:10.1016/0304-3975(91)90374-B.

59 Ramanathan S. Thinniyam and Georg Zetzsche. Regular separability and intersection emptiness are independent problems. In Arkadev Chattopadhyay and Paul Gastin, editors, *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019, December 11-13, 2019, Bombay, India*, volume 150 of *LIPIcs*, pages 51:1–51:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICS.FSTTCS.2019.51.

60 Georg Zetzsche. An approach to computing downward closures. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 440–451. Springer, 2015. doi:10.1007/978-3-662-47666-6_35.

61 Georg Zetzsche. *Monoids as Storage Mechanisms*. PhD thesis, Kaiserslautern University of Technology, Germany, 2016. URL: https://nbn-resolving.org/urn:nbn:de:hbz:386-kluedo-44003.