On the Complexity of Intersection Non-emptiness for Star-Free Language Classes

Emmanuel Arrighi
University of Bergen, Norway

Henning Fernau
Fachbereich IV, Informatikwissenschaften, Universität Trier, Germany

Stefan Hoffmann
Fachbereich IV, Informatikwissenschaften, Universität Trier, Germany

Markus Holzer
Institut für Informatik, Universität Giessen, Germany

Ismaël Jecker
Institute of Science and Technology, Klosterneuburg, Austria

Mateus de Oliveira Oliveira
University of Bergen, Norway

Petra Wolf
Fachbereich IV, Informatikwissenschaften, Universität Trier, Germany

Abstract

In the Intersection Non-emptiness problem, we are given a list of finite automata $A_1, A_2, \ldots, A_m$ over a common alphabet Σ as input, and the goal is to determine whether some string $w \in \Sigma^*$ lies in the intersection of the languages accepted by the automata in the list. We analyze the complexity of the Intersection Non-emptiness problem under the promise that all input automata accept a language in some level of the dot-depth hierarchy, or some level of the Straubing-Thérien hierarchy. Automata accepting languages from the lowest levels of these hierarchies arise naturally in the context of model checking. We identify a dichotomy in the dot-depth hierarchy by showing that the problem is already NP-complete when all input automata accept languages of the levels $\mathcal{B}_0$ or $\mathcal{B}_{1/2}$ and already PSPACE-hard when all automata accept a language from the level $\mathcal{B}_1$. Conversely, we identify a tetrachotomy in the Straubing-Thérien hierarchy. More precisely, we show that the problem is in AC^0 when restricted to level $\mathcal{L}_0$; complete for L or NL, depending on the input representation, when restricted to languages in the level $\mathcal{L}_{1/2}$; NP-complete when the input is given as DFAs accepting a language in $\mathcal{L}_1$ or $\mathcal{L}_{3/2}$; and finally, PSPACE-complete when the input automata accept languages in level $\mathcal{L}_2$ or higher. Moreover, we show that the proof technique used to show containment in NP for DFAs accepting languages in $\mathcal{L}_1$ or $\mathcal{L}_{3/2}$ does not generalize to the context of NFAs. To prove this, we identify a family of languages that provide an exponential separation between the state complexity of general NFAs and that of partially ordered NFAs. To the best of our knowledge, this is the first superpolynomial separation between these two models of computation.

2012 ACM Subject Classification Theory of computation → Regular languages; Theory of computation → Problems, reductions and completeness

Keywords and phrases Intersection Non-emptiness Problem, Star-Free Languages, Straubing-Thérien Hierarchy, dot-depth Hierarchy, Commutative Languages, Complexity

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2021.34

Related Version *Full Version*: `http://arxiv.org/abs/2110.01279` [3]

Funding *Emmanuel Arrighi*: Research Council of Norway (no. 274526), IS-DAAD (no. 309319).
Henning Fernau: DAAD PPP (no. 57525246).
Ismaël Jecker: Marie Skłodowska-Curie Grant Agreement no. 754411.
Mateus de Oliveira Oliveira: Research Council of Norway (no. 288761), IS-DAAD (no. 309319).
Petra Wolf: DFG project FE 560/9-1, DAAD PPP (no. 57525246).

41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021).
Editors: Mikołaj Bojańczyk and Chandra Chekuri; Article No. 34; pp. 34:1–34:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements We like to thank Lukas Fleischer and Michael Wehar for our discussions. This work started at the Schloss Dagstuhl Event 20483 *Moderne Aspekte der Komplexitätstheorie in der Automatentheorie* `https://www.dagstuhl.de/20483`.

1 Introduction

The Intersection Non-emptiness problem for finite automata is one of the most fundamental and well studied problems in the interplay between algorithms, complexity theory, and automata theory [12, 20, 21, 24, 26, 43, 44, 45]. Given a list $A_1, A_2, \ldots, A_m$ of finite automata over a common alphabet Σ, the goal is to determine whether there is a string $w \in \Sigma^*$ that is accepted by each of the automata in the list. This problem is PSPACE-complete when no restrictions are imposed [24], and becomes NP-complete when the input automata accept unary languages (implicitly contained already in [38]) or finite languages [34].

In this work, we analyze the complexity of the Intersection Non-emptiness problem under the assumption that the languages accepted by the input automata belong to a given level of the Straubing-Thérien hierarchy [33, 39, 40, 42] or to some level of the Cohen-Brzozowski dot-depth hierarchy [6, 11, 33]. Somehow, these languages are severely restricted, in the sense that both hierarchies, which are infinite, are entirely contained in the class of star-free languages, a class of languages that can be represented by expressions that use union, concatenation, and complementation, but *no* Kleene star operation [6, 8, 33]. Yet, languages belonging to fixed levels of either hierarchy may already be very difficult to characterize, in the sense that the very problem of deciding whether the language accepted by a given finite automaton belongs to a given full level or half-level k of either hierarchy is open, except for a few values of k [2, 15, 16, 33]. It is worth noting that while the problem of determining whether a given automaton accepts a language in a certain level of either the dot-depth or of the Straubing-Thérien hierarchy is computationally hard (Theorem 1), automata accepting languages in lower levels of these hierarchies arise naturally in a variety of applications such as model checking where the Intersection Non-emptiness problem is of fundamental relevance [1, 4, 5].

An interesting question to consider is how the complexity of the Intersection Non-emptiness problem changes as we move up in the levels of the Straubing-Thérien hierarchy or in the levels of the dot-depth hierarchy. In particular, does the complexity of this problem changes gradually, as we increase the complexity of the input languages? In this work, we show that this is actually not the case, and that the complexity landscape for the Intersection Non-emptiness problem is already determined by the very first levels of either hierarchy (see Figure 1). Our first main result states that the Intersection Non-emptiness problem for NFAs and DFAs accepting languages from the level 1/2 of the Straubing-Thérien hierarchy are NL-complete and L-complete, respectively, under AC^0 reductions (Theorem 3). Additionally, this completeness result holds even in the case of unary languages. To prove hardness for NL and L, respectively, we will use a simple reduction from the reachability problem for DAGs and for directed trees, respectively. Nevertheless, the proof of containment in NL and in L, respectively, will require a new insight that may be of independent interest. More precisely, we will use a characterization of languages in the level 1/2 of the Straubing-Thérien hierarchy as shuffle ideals to show that the Intersection Non-emptiness problem can be reduced to concatenation non-emptiness (Lemma 5). This allows us to decide Intersection Non-emptiness by analyzing each finite automaton given at the input individually. It is worth mentioning that this result is optimal in the sense that the problem becomes NP-hard even if we allow a single DFA to accept a language from $\mathcal{L}_1$, and require all the others to accept languages from $\mathcal{L}_{1/2}$ (Theorem 8).

Subsequently, we analyze the complexity of Intersection Non-emptiness when all input automata are assumed to accept languages from one of the levels of $\mathcal{B}_0$ or $\mathcal{B}_{1/2}$ of the dot-depth hierarchy, or from the levels $\mathcal{L}_1$ or $\mathcal{L}_{3/2}$ of the Straubing-Thérien hierarchy. It is worth noting that NP-hardness follows straightforwardly from the fact that Intersection Non-emptiness for DFAs accepting finite languages is already NP-hard [34]. Containment in NP, on the other hand, is a more delicate issue, and here the representation of the input automaton plays an important role. A characterization of languages in $\mathcal{L}_{3/2}$ in terms of languages accepted by partially ordered NFAs [37] is crucial for us, combined with the fact that Intersection Non-emptiness when the input is given by such automata is NP-complete [29]. Intuitively, the proof in [29] follows by showing that the minimum length of a word in the intersection of languages in the level 3/2 of the Straubing-Thérien hierarchy is bounded by a polynomial on the sizes of the minimum partially ordered NFAs accepting these languages. To prove that Intersection Non-emptiness is in NP when the input automata are given as DFAs, we prove a new result establishing that the number of Myhill-Nerode equivalence classes in a language in the level $\mathcal{L}_{3/2}$ is at least as large as the number of states in a minimum partially ordered automaton representing the same language (Lemma 12).

Interestingly, we show that the proof technique used to prove this last result does not generalize to the context of NFAs. To prove this, we carefully design a sequence $(L_n)_{n\in\mathbb{N}_{\geq 1}}$ of languages over a binary alphabet such that for every $n \in \mathbb{N}_{\geq 1}$, the language L_n can be accepted by an NFA of size n, but any partially ordered NFA accepting L_n has size $2^{\Omega(\sqrt{n})}$. This lower bound is ensured by the fact that the syntactic monoid of L_n has many $\mathcal{J}$-factors. Our construction is inspired by a technique introduced by Klein and Zimmermann, in a completely different context, to prove lower bounds on the amount of look-ahead necessary to win infinite games with delay [22]. To the best of our knowledge, this is the first exponential separation between the state complexity of general NFAs and that of partially ordered NFAs. While this result does not exclude the possibility that Intersection Non-emptiness for languages in $\mathcal{L}_{3/2}$ represented by general NFAs is in NP, it gives some indication that proving such a containment requires substantially new techniques.

Finally, we show that Intersection Non-emptiness for both DFAs and for NFAs is already PSPACE-complete if all accepting languages are from the level $\mathcal{B}_1$ of the dot-depth hierarchy or from the level $\mathcal{L}_2$ of the Straubing-Thérien hierarchy. We can adapt Kozen's classical PSPACE-completeness proof by using the complement of languages introduced in [28] in the study of partially ordered automata. Since the languages in [28] belong to $\mathcal{L}_{3/2}$, their complement belong to $\mathcal{L}_2$ (and to $\mathcal{B}_1$), and therefore, the proof follows.

Due to space constraints, many details of the paper can be found in the long version [3].

2 Preliminaries

We let $\mathbb{N}_{\geq k}$ denote the set of natural numbers greater or equal than k.

We assume the reader to be familiar with the basics in computational complexity theory [31]. In particular, we recall the inclusion chain: $\mathsf{AC}^0 \subset \mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE}$. Let AC^0 (NC^1, respectively) refer to the class of problems accepted by Turing machines with a bounded (unbounded, respectively) number of alternations in logarithmic time; alternatively one can define these classes by uniform Boolean circuits. Here, L (NL, respectively) refers to the class of problems that are accepted by deterministic (nondeterministic, respectively) Turing machines with logarithmic space, P (NP, respectively) denotes the class of problems solvable by deterministic (nondeterministic, respectively) Turing machines in polynomial time, and PSPACE refers to the class of languages accepted by deterministic or

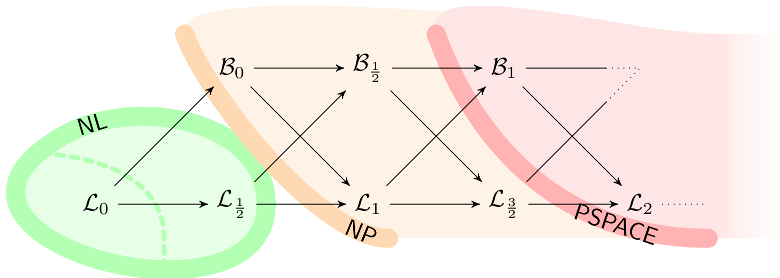

Figure 1 Straubing-Thérien and dot-depth hierarchies: the INTERSECTION NON-EMPTINESS status.

nondeterministic Turing machines in polynomial space [35]. Completeness and hardness are always meant with respect to deterministic logspace many-one reductions unless otherwise stated. We will also consider the parameterized class XP of problems that can be solved in time $n^{f(k)}$, where n is the size of the input, k is a parameter, and f is a computable function [13].

We mostly consider *nondeterministic finite automata* (NFAs). An NFA A is a tuple $A = (Q, \Sigma, \delta, q_0, F)$, where Q is the finite *state set* with the *start state* $q_0 \in Q$, the *alphabet* Σ is a finite set of input symbols, and $F \subseteq Q$ is the *final state set*. The *transition function* $\delta : Q \times \Sigma \to 2^Q$ extends to words from Σ^* as usual. Here, 2^Q denotes the powerset of Q. By $L(A) = \{\, w \in \Sigma^* \mid \delta(q_0, w) \cap F \neq \emptyset \,\}$, we denote the *language accepted by* A. The NFA A is a *deterministic finite automaton* (DFA) if $|\delta(q, a)| = 1$ for every $q \in Q$ and $a \in \Sigma$. Then, we simply write $\delta(q, a) = p$ instead of $\delta(q, a) = \{p\}$. If $|\Sigma| = 1$, we call A a *unary* automaton.

We study INTERSECTION NON-EMPTINESS problems and their complexity. For finite automata, this problem is defined as follows:

- *Input*: Finite automata $A_i = (Q_i, \Sigma, \delta_i, q_{(0,i)}, F_i)$, for $1 \leq i \leq m$.
- *Question*: Is there a word w that is accepted by all A_i, i.e., is $\bigcap_{i=1}^{m} L(A_i) \neq \emptyset$?

Observe that the automata have a common input alphabet. Note that the complexity of the non-emptiness problem for finite automata of a certain type is a lower bound for the INTERSECTION NON-EMPTINESS for this particular type of automata. Throughout the paper we are mostly interested in the complexity of the INTERSECTION NON-EMPTINESS problem for finite state devices whose languages are contained in a particular language class.

We study the computational complexity of the intersection non-emptiness for languages from the classes of the Straubing-Thérien [39, 42] and Cohen-Brzozowski's dot-depth hierarchy [11]. Both hierarchies are concatenation hierarchies that are defined by alternating the use of polynomial and Boolean closures. Let's be more specific. Let Σ be a finite alphabet. A language $L \subseteq \Sigma^*$ is a *marked product* of the languages $L_0, L_1, \ldots, L_k$, if $L = L_0 a_1 L_1 \cdots a_k L_k$, where the a_i's are letters. For a class of languages $\mathcal{M}$, the *polynomial closure* of $\mathcal{M}$ is the set of languages that are finite unions of marked product of languages from $\mathcal{M}$.

The concatenation hierarchy of basis $\mathcal{M}$ (a class of languages) is defined as follows (also refer to [32]): Level 0 is $\mathcal{M}$, i.e., $\mathcal{M}_0 = \mathcal{M}$ and, for each $n \geq 0$,

1. $\mathcal{M}_{n+1/2}$, that is, level $n + 1/2$, is the polynomial closure of level n and
2. $\mathcal{M}_{n+1}$, that is, level $n + 1$, is the Boolean closure of level $n + 1/2$.

The basis of the dot-depth hierarchy is the class of all finite and co-finite languages[1] and their classes are referred to as $\mathcal{B}_n$ ($\mathcal{B}_{n+1/2}$, respectively), while the basis of the Straubing-Thérien hierarchy is the class of languages that contains only the empty set and Σ^* and their classes are denoted by $\mathcal{L}_n$ ($\mathcal{L}_{n+1/2}$, respectively). Their inclusion relation is given by

[1] The dot-depth hierarchy, apart level $\mathcal{B}_0$, coincides with the concatenation hierarchy starting with the language class $\{\emptyset, \{\lambda\}, \Sigma^+, \Sigma^*\}$.

$$\mathcal{B}_{n+1/2} \subseteq \mathcal{B}_{n+1} \subseteq \mathcal{B}_{n+3/2} \quad \text{and} \quad \mathcal{L}_{n+1/2} \subseteq \mathcal{L}_{n+1} \subseteq \mathcal{L}_{n+3/2},$$

for $n \geq 0$, and

$$\mathcal{L}_{n-1/2} \subseteq \mathcal{B}_{n-1/2} \subseteq \mathcal{L}_{n+1/2} \quad \text{and} \quad \mathcal{L}_n \subseteq \mathcal{B}_n \subseteq \mathcal{L}_{n+1},$$

for $n \geq 1$. In particular, $\mathcal{L}_0 \subseteq \mathcal{B}_0$, $\mathcal{B}_0 \subseteq \mathcal{B}_{1/2}$, and $\mathcal{L}_0 \subseteq \mathcal{L}_{1/2}$. Both hierarchies are infinite for alphabets of at least two letters and completely exhaust the class of star-free languages, which can be described by expressions that use union, concatenation, and complementation, but *no* Kleene star operation. For singleton letter alphabets, both hierarchies collapse to $\mathcal{B}_0$ and $\mathcal{L}_1$, respectively. Next, we describe the first few levels of each of these hierarchies:

Straubing-Thérien hierarchy: A language of Σ^* is of level 0 if and only if it is empty or equal to Σ^*. The languages of level 1/2 are exactly those languages that are a finite (possibly empty) union of languages of the form $\Sigma^* a_1 \Sigma^* a_2 \cdots a_k \Sigma^*$, where the a_i's are letters from Σ. The languages of level 1 are finite Boolean combinations of languages of the form $\Sigma^* a_1 \Sigma^* a_2 \cdots a_k \Sigma^*$, where the a_i's are letters. These languages are also called *piecewise* testable languages. In particular, all finite and co-finite languages are of level 1. Finally, the languages of level 3/2 of Σ^* are the finite unions of languages of the form $\Sigma_0^* a_1 \Sigma_1^* a_2 \cdots a_k \Sigma_k^*$, where the a_i's are letters from Σ and the Σ_i are subsets of Σ.

Dot-depth hierarchy: A language of Σ^* is of dot-depth (level) 0 if and only if it is finite or co-finite. The languages of dot-depth 1/2 are exactly those languages that are a finite union of languages of the form $u_0 \Sigma^* u_1 \Sigma^* u_2 \cdots u_{k-1} \Sigma^* u_k$, where $k \geq 0$ and the u_i's are words from Σ^*. The languages of dot-depth 1 are finite Boolean combinations of languages of the form $u_0 \Sigma^* u_1 \Sigma^* u_2 \cdots u_{k-1} \Sigma^* u_k$, where $k \geq 0$ and the u_i's are words from Σ^*.

It is worth mentioning that in [37] it was shown that partially ordered NFAs (with multiple initial states) characterize the class $\mathcal{L}_{3/2}$, while partially ordered DFAs characterize the class of $\mathcal{R}$-trivial languages [7], a class that is strictly in between $\mathcal{L}_1$ and $\mathcal{L}_{3/2}$. For an automaton A with input alphabet Σ, a state q is reachable from a state p, written $p \leq q$, if there is a word $w \in \Sigma^*$ such that $q \in \delta(p, w)$. An automaton is partially ordered if $\leq$ is a partial order. Partially ordered automata are sometimes also called acyclic or weakly acyclic automata. We refer to a partially ordered NFA (DFA, respectively) as poNFA (poDFA, respectively).

The fact that some of our results have a promise looks a bit technical, but the following result implies that we cannot get rid of this condition in general. To this end, we study, for a language class $\mathcal{L}$, the following question of $\mathcal{L}$-Membership.

- *Input*: A finite automaton A.
- *Question*: Is $L(A) \in \mathcal{L}$?

▶ **Theorem 1.** *For each level $\mathcal{L}$ of the Straubing-Thérien or the dot-depth hierarchies, the $\mathcal{L}$-Membership problem for NFAs is* PSPACE*-hard, even when restricted to binary alphabets.*

Proof. For the PSPACE-hardness, note that each of the classes contains $\{0,1\}^*$ and is closed under quotients, since each class is a positive variety. As Non-universality is PSPACE-hard for NFAs, we can apply Theorem 3.1.1 of [19], first reducing regular expressions to NFAs. ◀

For some of the lower levels of the hierarchies, we also have containment in PSPACE, but in general, this is unknown, as it connects to the famous open problem if, for instance, $\mathcal{L}$-Membership is decidable for $\mathcal{L} = \mathcal{L}_3$; see [27, 33] for an overview on the decidability status of these questions. Checking for $\mathcal{L}_0$ up to $\mathcal{L}_2$ and $\mathcal{B}_0$ up to $\mathcal{B}_1$ containment for DFAs can be done in NL and is also complete for this class by ideas similar to the ones used in [9].

3 Inside Logspace

A language of Σ^* belongs to level 0 of the Straubing-Thérien hierarchy if and only if it is empty or Σ^*. The Intersection Non-emptiness problem for language from this language family is not entirely trivial, because we have to check for emptiness. Since by our problem definition the property of a language being a member of level 0 is a promise, we can do the emptiness check within AC^0, since we only have to verify whether the empty word belongs to the language L specified by the automaton. In case $\varepsilon \in L$, then $L = \Sigma^*$; otherwise $L = \emptyset$. Since in the definition of finite state devices we do not allow for ε-transitions, we thus only have to check whether the initial state is also an accepting one. Therefore, we obtain:

▶ **Theorem 2.** *The Intersection Non-emptiness problem for DFAs or NFAs accepting languages from $\mathcal{L}_0$ belongs to AC^0.*

For the languages of level $\mathcal{L}_{1/2}$ we find the following completeness result.

▶ **Theorem 3.** *The Intersection Non-emptiness problem for NFAs accepting languages from $\mathcal{L}_{1/2}$ is NL-complete. Moreover, the problem remains NL-hard even if we restrict the input to NFAs over a unary alphabet. If the input instance contains only DFAs, the problem becomes L-complete (under weak reductions[2]).*

Hardness is shown by standard reductions from variants of graph accessibility [17, 41].

▶ **Lemma 4.** *The Intersection Non-emptiness problem for NFAs over unary alphabet accepting languages from $\mathcal{L}_{1/2}$ is NL-hard. If the input instance contains only DFAs, the problem becomes L-hard under weak reductions.*

It remains to show containment in logspace. To this end, we utilize an alternative characterization of the languages of level 1/2 of the Straubing-Thérien hierarchy as exactly those languages that are shuffle ideals. A language L is a *shuffle ideal* if, for every word $w \in L$ and $v \in \Sigma^*$, the set $w \sqcup\!\sqcup v$ is contained in L, where $w \sqcup\!\sqcup v := \{\, w_0v_0w_1v_1 \ldots w_kv_k \mid w = w_0w_1 \ldots w_k \text{ and } v = v_0v_1 \ldots v_k \text{ with } w_i, v_i \in \Sigma^*, \text{ for } 0 \leq i \leq k \,\}$. The operation $\sqcup\!\sqcup$ naturally generalizes to sets. For the level $\mathcal{L}_{1/2}$, we find the following situation.

▶ **Lemma 5.** *Let $m \geq 1$ and languages $L_i \subseteq \Sigma^*$, for $1 \leq i \leq m$, be shuffle ideals, i.e., they belong to $\mathcal{L}_{1/2}$. Then, $\bigcap_{i=1}^m L_i \neq \emptyset$ iff the shuffle ideal $L_1L_2 \cdots L_m \neq \emptyset$ iff $L_i \neq \emptyset$ for every i with $1 \leq i \leq m$. Finally, $L_i \neq \emptyset$, for $1 \leq i \leq m$, iff $(a_1a_2 \ldots a_k)^{\ell_i} \in L_i$, where $\Sigma = \{a_1, a_2, \ldots a_k\}$ and the shortest word in L_i is of length ℓ_i.*

Now, we are ready to prove containment in logspace.

▶ **Lemma 6.** *The Intersection Non-emptiness problem for NFAs accepting languages from $\mathcal{L}_{1/2}$ belongs to NL. If the input instance contains only DFAs, the problem is solvable in L.*

Proof. In order to solve the Intersection Non-emptiness problem for given finite automata $A_1, A_2, \ldots, A_m$ with a common input alphabet Σ, regardless of whether they are deterministic or nondeterministic, it suffices to check non-emptiness for all languages $L(A_i)$, for $1 \leq i \leq m$, in sequence, because of Lemma 5. To this end, membership of the words $(a_1a_2 \ldots a_k)^{\ell_i}$ in L_i has to be tested, where ℓ_i is the length of the shortest word in L_i. Obviously, all ℓ_i are linearly bounded in the number of states of the appropriate finite automaton

[2] Some form of AC^0 reducibility can be employed.

that accepts L_i. Hence, for NFAs as input instance, the test can be done on a nondeterministic logspace-bounded Turing machine, guessing the computations in the individual NFAs on the input word $(a_1a_2\ldots a_k)^{\ell_i}$. For DFAs as input instance, nondeterminism is not needed, so that the procedure can be implemented on a deterministic Turing machine. ◀

4 NP-Completeness

In contrast to the Straubing-Thérien hierarchy, the Intersection Non-emptiness problem for languages from the dot-depth hierarchy is already NP-hard in the lowest level $\mathcal{B}_0$. More precisely, Intersection Non-emptiness for finite languages is NP-hard [34, Theorem 1] and $\mathcal{B}_0$ already contains all finite languages. Hence, the Intersection Non-emptiness problem for languages from the Straubing-Thérien hierarchy of level $\mathcal{L}_1$ and above is NP-hard, too. For the levels $\mathcal{B}_0$, $\mathcal{B}_{1/2}$, $\mathcal{L}_1$, or $\mathcal{L}_{3/2}$, we give matching complexity upper bounds if the input are DFAs, yielding the first main result of this section proven in Subsection 4.1.

▶ **Theorem 7.** *The Intersection Non-emptiness problem for DFAs accepting languages from either $\mathcal{B}_0$, $\mathcal{B}_{1/2}$, $\mathcal{L}_1$, or $\mathcal{L}_{3/2}$ is NP-complete. The same holds for poNFAs instead of DFAs. The results hold even for a binary alphabet.*

For the level $\mathcal{L}_1$ of the Straubing-Thérien hierarchy, we obtain with the next main theorem a stronger result. Recall that if all input DFAs accept languages from $\mathcal{L}_{1/2}$, the Intersection Non-emptiness problem is L-complete due to Lemmata 4 and 6.

▶ **Theorem 8.** *The Intersection Non-emptiness problem for DFAs is NP-complete even if only one DFA accepts a language from $\mathcal{L}_1$ and all other DFAs accept languages from $\mathcal{L}_{1/2}$ and the alphabet is binary.*

The proof of this theorem will be given in Subsection 4.2.

For the level $\mathcal{B}_0$, we obtain a complete picture of the complexity of the Intersection Non-emptiness problem, independent of structural properties of the input finite automata, i.e., we show that here the problem is NP-complete for general NFAs.

For the level $\mathcal{L}_{3/2}$, if the input NFA are from the class of poNFA, which characterize level $\mathcal{L}_{3/2}$, then the Intersection Non-emptiness problem is known to be NP-complete [28]. Recall that $\mathcal{L}_{3/2}$ contains the levels $\mathcal{B}_{1/2}$, and $\mathcal{L}_1$ and hence also languages from these classes can be represented by poNFAs. But if the input automata are given as NFAs without any structural property, then the precise complexity of Intersection Non-emptiness for $\mathcal{B}_{1/2}$, $\mathcal{L}_1$, and $\mathcal{L}_{3/2}$ is an open problem and narrowed by NP-hardness and membership in PSPACE. We present a "No-Go-Theorem" by proving that for an NFA accepting a co-finite language, the smallest equivalent poNFA is exponentially larger in Subsection 4.3.

▶ **Theorem 9.** *For every $n \in \mathbb{N}_{\geq 1}$, there exists a language $L_n \in \mathcal{B}_0$ on a binary alphabet such that L_n is recognized by an NFA of size $O(n^2)$, but the minimal poNFA recognizing L_n has more than 2^{n-1} states.*

While for NFAs the precise complexity for Intersection Non-emptiness of languages from $\mathcal{L}_1$ remains open, we can tackle this gap by narrowing the considered language class to *commutative* languages in level $\mathcal{L}_1$; recall that a language $L \subseteq \Sigma^*$ is *commutative* if, for any $a, b \in \Sigma$ and words $u, v \in \Sigma^*$, we have that $uabv \in L$ implies $ubav \in L$. We show that for DFAs, this restricted Intersection Non-emptiness problem remains NP-hard, in case the alphabet is unbounded. Concerning membership in NP, we show that even for NFAs, the Intersection Non-emptiness problem for *commutative* languages is contained in NP in general and in particular for commutative languages on each level. This generalizes the case of unary NFAs. Note that for commutative languages, the Straubing-Thérien hierarchy collapses at level $\mathcal{L}_{3/2}$. See Subsection 4.4 for the proofs.

▶ **Theorem 10.** *The* Intersection Non-emptiness *problem*
- *is* NP*-hard for DFAs accepting* commutative *languages in* $\mathcal{L}_1$*, but*
- *is contained in* NP *for NFAs accepting* commutative *languages that might not be star-free.*

The proof of NP-hardness for commutative star-free languages in $\mathcal{L}_1$ requires an arbitrary alphabet. However, we show that Intersection Non-emptiness is contained in XP for specific forms of NFAs such as poNFAs or DFAs accepting commutative languages, with the size of the alphabet as the parameter, i.e., for fixed input alphabets, our problem is solvable in polynomial time.

4.1 NP-Membership

Next, we focus on the NP-membership part of Theorem 7 and begin by proving that for $\mathcal{B}_0$, regardless of whether the input automata are NFAs or DFAs, the Intersection Non-emptiness problem is contained in NP and therefore NP-complete in combination with [34].

▶ **Lemma 11.** *The* Intersection Non-emptiness *problem for DFAs or NFAs all accepting languages from* $\mathcal{B}_0$ *is contained in* NP*.*

Proof. Let $A_1, A_2, \ldots, A_m$ be NFAs accepting languages from $\mathcal{B}_0$. If all NFAs accept co-finite languages, which can be verified in deterministic polynomial time, the intersection $\bigcap_{i=1}^m L(A_i)$ is non-empty. Otherwise, there is at least one NFA accepting a finite language, where the longest word is bounded by the number of states of this device. Hence, if $\bigcap_{i=1}^m L(A_i) \neq \emptyset$, there is a word w of length polynomial in the length of the input that witnesses this fact. Such a w can be nondeterministically guessed by a Turing machine checking membership of w in $L(A_i)$, for all NFAs A_i, in sequence. This shows containment in NP as desired. ◀

Notice that Masopust and Krötzsch have shown in [28] that Intersection Non-emptiness for poDFAs and for poNFAs is NP-complete. Also the unary case is discussed there, which can be solved in polynomial time. We cannot directly make use of these results, as we consider arbitrary NFAs or DFAs as inputs, only with the promise that they accept languages from a certain level of the studied hierarchies. In order to prove that for the levels $\mathcal{B}_0$, $\mathcal{B}_{1/2}$, $\mathcal{L}_1$, and $\mathcal{L}_{3/2}$, the Intersection Non-emptiness problem for DFAs is contained in NP, it is sufficient to prove the claim for $\mathcal{L}_{3/2}$ as all other stated levels are contained in $\mathcal{L}_{3/2}$. We prove the latter statement by obtaining a bound, polynomial in the size of the largest DFA, on the length of a shortest word accepted by all DFAs. Therefore, we show that for a minimal poNFA A, the size of an equivalent DFA is lower-bounded by the size of A and use a result of [28] for poNFAs. They have shown that given poNFAs $A_1, A_2, \ldots, A_m$, if the intersection of these automata is non-empty, then there exists a word of size at most $\sum_{i\in\{1,\ldots,m\}} d_i$, where d_i is the *depth* of A_i [28, Theorem 3.3]. Here, the depth of A_i is the length of the longest path (without self-loops) in the state graph of A_i. This result implies that the Intersection Non-emptiness problem for poNFAs accepting languages from $\mathcal{L}_{3/2}$ is contained in NP. We will further use this result to show that the Intersection Non-emptiness problem for DFAs accepting languages from $\mathcal{L}_{3/2}$ is NP-complete. First, we show that the number of states in a minimal poNFA is at most the number of classes in the Myhill-Nerode equivalence relation.

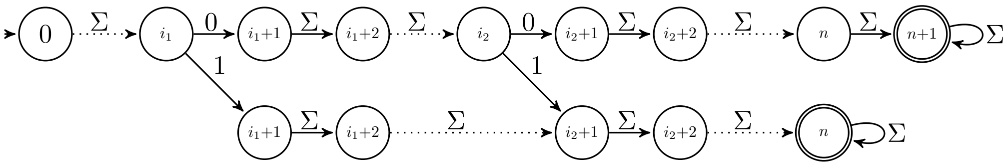

Figure 2 DFA A_{e_i} with $L(A_{e_i}) = \Sigma^{i_1} \cdot 1 \cdot \Sigma^{n-i_1-1} \cup \Sigma^{i_2} \cdot 1 \cdot \Sigma^{n-i_2-1} \cup \Sigma^{\geq n+1}$. A dotted arrow between some states j and j' represents a chain of length $j' - j$ with the same transition labels.

▶ **Lemma 12.** *Let $A = (Q, \Sigma, \delta, q_0, F)$ be a minimal poNFA. Then, $L({}_{q_1}A) \neq L({}_{q_2}A)$ for all states $q_1, q_2 \in Q$, where ${}_qA$ is defined as $(Q, \Sigma, \delta, q, F)$.*

Now, we can use the result from Masopust and Krötzsch to prove that the Intersection Non-emptiness problem for DFAs accepting languages in $\mathcal{L}_{3/2}$ is in NP.

▶ **Lemma 13.** *The Intersection Non-emptiness problem for DFAs accepting languages from $\mathcal{L}_{3/2}$ belongs to* NP*.*

Proof. By Lemma 12, we have that the number of states in a minimal poNFA is at most the number of classes of the Myhill-Nerode equivalence relation. Hence, given a DFA accepting a language $L \in \mathcal{L}_{3/2}$, there exists a smaller poNFA that recognizes L. By [28, Theorem 3.3], if the intersection is not empty, then there is a certificate of polynomial size. ◀

4.2 NP-Hardness

Recall that by [34, Theorem 1] Intersection Non-emptiness for finite languages accepted by DFAs is already NP-complete. As the level $\mathcal{B}_0$ of the dot-depth hierarchy contains all finite language, the NP-hardness part of Theorem 7 follows directly from inclusion of language classes. Combining Lemma 13, and [28, Theorem 3.3] with the inclusion between levels in the Straubing-Thérien and the dot-depth hierarchy, we conclude the proof of Theorem 7.

▶ Remark 14. Recall that the dot-depth hierarchy, apart form $\mathcal{B}_0$, coincides with the concatenation hierarchy starting with the language class $\{\emptyset, \{\lambda\}, \Sigma^+, \Sigma^*\}$. The Intersection Non-emptiness problem for DFAs or NFAs accepting only languages from $\{\emptyset, \{\lambda\}, \Sigma^+, \Sigma^*\}$ belongs to AC^0, by similar arguments as in the proof of Theorem 2.

We showed in Section 3 that Intersection Non-emptiness for DFAs, all accepting languages from $\mathcal{L}_{1/2}$, belongs to L. If we allow only one DFA to accept a language from $\mathcal{L}_1$, the problem becomes NP-hard. The statement also holds if the common alphabet is binary.

▶ **Theorem 8.** *The Intersection Non-emptiness problem for DFAs is* NP*-complete even if only one DFA accepts a language from $\mathcal{L}_1$ and all other DFAs accept languages from $\mathcal{L}_{1/2}$ and the alphabet is binary.*

Proof sketch. The reduction is from Vertex Cover. Let $k \in \mathbb{N}_{\geq 0}$ and let $G = (V, E)$ be a graph with vertex set $V = \{v_0, v_1, \ldots, v_{n-1}\}$ and edge set $E = \{e_0, e_1, \ldots, e_{m-1}\}$. The only words $w = a_0 a_1 \ldots a_\ell$ accepted by all DFAs will be of length exactly $n = \ell + 1$ and encode a vertex cover by: v_j is in the vertex cover if and only if $a_j = 1$. Therefore, we construct for each edge $e_i = \{v_{i_1}, v_{i_2}\} \in E$, with $i_1 < i_2$, a DFA A_{e_i}, as depicted in Figure 2, that accepts the language $L(A_{e_i}) = \Sigma^{i_1} \cdot 1 \cdot \Sigma^{n-i_1-1} \cup \Sigma^{i_2} \cdot 1 \cdot \Sigma^{n-i_2-1} \cup \Sigma^{\geq n+1}$. We show that $L(A_{e_i})$ is from $\mathcal{L}_{1/2}$, as it also accepts all words of length at least $n + 1$. We further construct a DFA $A_{=n,\leq k}$ that accepts all words of length exactly n that contain at most k letters 1. The finite language $L(A_{=n,\leq k})$ is the only language from $\mathcal{L}_1$ in the instance. ◀

4.3 Large Partially Ordered NFAs

The results obtained in the last subsection left the precise complexity membership of Intersection Non-emptiness in the case of input automata being NFAs without any structural properties for the levels $\mathcal{B}_{1/2}$, $\mathcal{L}_1$, and $\mathcal{L}_{3/2}$ open. We devote this subsection to the proof of Theorem 9, showing that already for languages of $\mathcal{B}_0$ being accepted by an NFA, the size of an equivalent minimal poNFA can be exponential in the size of the NFA.

▶ **Theorem 9.** *For every $n \in \mathbb{N}_{\geq 1}$, there exists a language $L_n \in \mathcal{B}_0$ on a binary alphabet such that L_n is recognized by an NFA of size $O(n^2)$, but the minimal poNFA recognizing L_n has more than 2^{n-1} states.*

Proof. While the statement requires languages over a binary alphabet, we begin by constructing an auxiliary family $(M_n)_{n\in\mathbb{N}_{\geq 1}}$ of languages over an unbounded alphabet. For all $n \in \mathbb{N}_{\geq 1}$ we then define L_n by encoding M_n with a binary alphabet, and we prove three properties of these languages that directly imply the statement of the Theorem.

For every $n \in \mathbb{N}_{\geq 1}$, we define the languages M'_n and M''_n over the alphabet $\{1, 2, \ldots, n\}$ as follows. The language M'_n contains all the words of odd length, and M''_n contains all the words in which there are two occurrences of some letter $i \in \{1, 2, \ldots, n\}$ with only letters smaller than i appearing in between.[3] Formally,

$$M'_n = \{\, x \in \{1, 2, \ldots, n\}^* \mid |x| \text{ is odd} \,\},$$
$$M''_n = \{\, xiyiz \in \{1, 2, \ldots, n\}^* \mid i \in \{1, 2, \ldots, n\}, y \in \{1, 2, \ldots, i-1\}^* \,\}.$$

We then define M_n as the union $M'_n \cup M''_n$. Moreover, we define L_n by encoding M_n with the binary alphabet $\{a, b\}$: Let us consider the function $\phi_n : \{1, 2, \ldots, n\}^* \to \{a, b\}^*$ defined by $\phi(i_1 i_2 \ldots i_m) = a^{i_1} b^{n-i_1} a^{i_2} b^{n-i_2} \ldots a^{i_m} b^{n-i_m}$. We set $L_n \subseteq \{a, b\}^*$ as the union of $\phi_n(M_n)$ with the language $\{a, b\}^* \setminus \phi(\{1, 2, \ldots, n\}^*)$ containing all the words that are not a proper encoding of some word in $\{1, 2, \ldots, n\}^*$.

The statement of the theorem immediately follows from the following claim

▷ Claim 15.
1. The languages M_n and L_n are cofinite, thus they are in $\mathcal{B}_0$.
2. The languages M_n and L_n are recognized by NFAs of size $n + 4$, resp. $O(n^2)$.
3. Every poNFA recognizing either M_n or L_n has a size greater than 2^{n-1}.

The formal proof of this claim is presented in the long version [3]. ◀

4.4 Commutative Star-Free Languages

In the case of commutative languages, we have a complete picture of the complexities for both hierarchies, even for arbitrary input NFAs. Observe, that commutative languages generalize unary languages, where it is known that for unary star-free languages both hierarchies collapse. For commutative star-free languages, a similar result holds, employing [18, Prop. 30].

▶ **Theorem 16.** *For* commutative star-free languages *the levels $\mathcal{L}_n$ of the Straubing-Thérien and $\mathcal{B}_n$ of the dot-depth hierarchy coincide for all full and half levels, except for $\mathcal{L}_0$ and $\mathcal{B}_0$. Moreover, the hierarchy collapses at level one.*

Next we will give the results, summarized in Theorem 10, for the case of the commutative (star-free) languages. The NP-hardness follows by a reduction from 3-CNF-SAT.

---

[3] The languages $(M''_n)_{n\in\mathbb{N}_{\geq 1}}$ were previously studied in [22] with a game-theoretic background. We also refer to [30] for similar “fractal languages.”

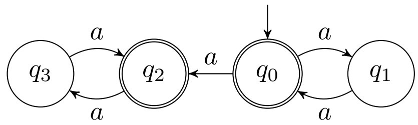

Figure 3 An example of a non-totally star-free NFA that accepts a star-free language.

▶ **Lemma 17.** *The* Intersection Non-emptiness *problem is* NP*-hard for DFAs accepting* commutative *languages in* $\mathcal{L}_1$*.*

The upper bound shown next also holds for arbitrary commutative languages.

▶ **Theorem 18.** *The* Intersection Non-emptiness *problem for NFAs accepting arbitrary, i.e., not necessarily star-free,* commutative *languages is in* NP*.*

Proof. It was shown in [38] that Intersection Non-emptiness is NP-complete for unary NFAs as input. Fix some order $\Sigma = \{a_1, a_2, \ldots, a_r\}$ of the input alphabet. Let $A_1, A_2, \ldots, A_m$ be the NFAs accepting commutative languages with $A_i = (Q_i, \Sigma, \delta_i, q_{0,i}, F_i)$ for $1 \leq i \leq m$. Without loss of generality, we may assume that every F_i is a singleton set, namely $F_i = \{q_{f,i}\}$. For each $1 \leq i \leq m$ and $1 \leq j \leq r$, let $B_{i,j}$ be the automaton over the unary alphabet $\{a_j\}$ obtained from A_i by deleting all transitions labeled with letters different from a_j and only retaining those labeled with a_j. Each $B_{i,j}$ will have one initial and one final state. Let $\vec{q_0} = (q_{0,1}, q_{0,2}, \ldots, q_{0,m})$ be the tuple of initial states of the NFAs; they are the initial states of $B_{1,1}, B_{2,1}, \ldots, B_{m,1}$, respectively. Then, nondeterministically guess further tuples $\vec{q_j}$ from $Q_1 \times Q_2 \times \ldots \times Q_m$ for $1 \leq j \leq r-1$. The jth tuple is considered as collecting the final states of the $B_{i,j}$ but also as the start states for the $B_{i,j+1}$. Finally, let $\vec{q_f} = (q_{f,1}, q_{f,2}, \ldots, q_{f,m})$ and consider this as the final states of $B_{1,r}, B_{2,r}, \ldots, B_{m,r}$. Then, for each $1 \leq j \leq r$ solve Intersection Non-emptiness for the unary automata $B_{1,j}, B_{2,j}, \ldots, B_{m,j}$. If there exist words w_j in the intersection of $L(B_{1,j}), L(B_{2,j}), \ldots, L(B_{m,j})$, for each $1 \leq j \leq r$, then, by commutativity, there exists one in $a_1^* a_2^* \cdots a_r^*$, namely, $w_1 w_2 \cdots w_m$, and so the above procedure finds it. Conversely, if the above procedure finds a word, this is contained in the intersection of the languages induced by the A_i's. ◀

For fixed alphabets, we have a polynomial-time algorithm, showing that the problem is in XP for alphabet size as a parameter, for a class of NFAs generalizing, among others, poNFAs and DFAs (accepting star-free languages). This is in contrast to the other results on the Intersection Non-emptiness problem in this paper. We say that an NFA $A = (Q, \Sigma, \delta, q_0, F)$ is *totally star-free*, if the language accepted by ${}_qA_p = (Q, \Sigma, \delta, q, \{p\})$ is star-free for any states $q, p \in Q$. For instance, partially ordered NFAs are totally star-free.

An example of a non-totally star-free NFA accepting a star-free language is given next. Consider the following NFA $A = (\{q_0, q_1, q_2, q_3\}, \delta, q_0, \{q_0, q_2\})$ with $\delta(q_0, a) = \{q_1, q_2\}$, $\delta(q_1, a) = \{q_0\}$, $\delta(q_2, a) = \{q_3\}$, and $\delta(q_3, a) = \{q_2\}$ that accepts the language $\{a\}^*$. The automaton is depicted in Figure 3. Yet, neither $L({}_{q_0}A_{q_0}) = \{aa\}^*$ nor $L({}_{q_0}A_{q_2}) = \{a\}\{aa\}^* \cup \{\varepsilon\}$ are star-free.
The proof of the following theorem uses classical results of Chrobak and Schützenberger [10,36].

▶ **Theorem 19.** *The* Intersection Non-emptiness *problem for totally star-free NFAs accepting* star-free commutative *languages, i.e., commutative languages in* $\mathcal{L}_{3/2}$*, is contained in* XP *(with the size of the alphabet as the parameter).*

▶ Remark 20. Note that Theorem 19 does not hold for arbitrary commutative languages concerning a fixed alphabet, but only for star-free commutative languages, since in the general case, the problem is NP-complete even for languages over a common unary alphabet [38].

5 PSPACE-Completeness

Here, we prove that even when restricted to languages from $\mathcal{B}_1$ or $\mathcal{L}_2$, Intersection Non-emptiness is PSPACE-complete, as it is for unrestricted DFAs or NFAs. We will profit from the close relations of Intersection Non-emptiness to the Non-universality problem for NFAs: Given an NFA A with input alphabet Σ, decide if $L(A) \neq \Sigma^*$. Conversely, we can also observe that Non-universality for NFAs is PSPACE-complete for languages from $\mathcal{B}_1$.

▶ **Theorem 21.** *The Intersection Non-emptiness problem for DFAs or NFAs accepting languages from $\mathcal{B}_1$ or $\mathcal{L}_2$ is PSPACE-complete, even for binary input alphabets.*

As $\mathcal{B}_1 \subseteq \mathcal{L}_2$, it is sufficient to show that the problem is PSPACE-hard for $\mathcal{B}_1$. While without paying attention to the size of the input alphabet, this result can be readily obtained by re-analyzing Kozen's original proof in [24], the restriction to binary input alphabets needs some more care. Details can be found in the long version [3]. We modify the proof of Theorem 3 in [25] that showed PSPACE-completeness for Non-universality for poNFAs (that characterize the level 3/2 of the Straubing-Thérien hierarchy). Also, it can be observed that the languages involved in the intersection are actually locally testable languages. Without giving details of definitions, we can therefore formulate:

▶ **Corollary 22.** *The Intersection Non-emptiness problem for DFAs or NFAs accepting locally testable languages is PSPACE-complete, even for binary input alphabets.*

By the proof of Theorem 3 in [25], also $\bigcup_i L_i$ belongs to $\mathcal{B}_1$, so that we can conclude:

▶ **Corollary 23.** *The Non-universality problem for NFAs accepting languages from $\mathcal{B}_1$ is PSPACE-complete, even for binary input alphabets.*

6 Conclusion and Open Problems

We have investigated how the increase in complexity within the dot-depth and the Straubing-Thérien hierarchies is reflected in the complexity of the Intersection Non-emptiness problem. We have shown the complexity of this problem is already completely determined by the very first levels of either hierarchy.

Our work leaves open some very interesting questions and directions of research. First, we were not able to prove containment in NP for the Intersection Non-emptiness problem when the input automata are allowed to be NFAs accepting a language in the level 3/2 or in the level 1 of the Straubing-Thérien hierarchy. Interestingly, we have shown that such containment holds in the case of DFAs, but have shown that the technique we have used to prove this containment does not carry over to the context of NFAs. In particular, to show this we have provided the first exponential separation between the state complexity of general NFAs and partially ordered NFAs. The most immediate open question is if Intersection Non-emptiness for NFAs accepting languages in $\mathcal{B}_{1/2}$, $\mathcal{L}_1$, or $\mathcal{L}_{3/2}$ is complete for some level higher up in the polynomial-time hierarchy (PH), or if this case is already PSPACE-complete. Another tantalizing open question is whether one can capture the levels of PH in terms of the Intersection Non-emptiness problem when the input automata are assumed to accept languages belonging to levels of a sub-hierarchy of $\mathcal{L}_2$. Such sub-hierarchies have been considered for instance in [23].

It would also be interesting to have a systematic study of these two well-known subregular hierarchies for related problems like Non-universality for NFAs or Union Non-universality for DFAs. Notice the technicality that Union Non-universality (similar to Intersection Non-emptiness) has an implicit Boolean operation (now union instead of intersection) within the problem statement, while Non-universality lacks this implicit Boolean operation. This might lead to a small "shift" in the discussions of the hierarchy levels that involve Boolean closure. Another interesting hierarchy is the group hierarchy [32], where we start with the group languages, i.e., languages acceptable by automata in which every letter induces a permutation of the state set, at level 0. Note that for group languages, Intersection Non-emptiness is NP-complete even for a unary alphabet [38]. As Σ^* is a group language, the Straubing-Thérien hierarchy is contained in the corresponding levels of the group hierarchy, and hence, we get PSPACE-hardness for level 2 and above in this hierarchy. However, we do not know what happens in the levels in between.

References

1 Parosh Aziz Abdulla. Regular model checking. *International Journal on Software Tools for Technology Transfer*, 14(2):109–118, 2012.

2 Jorge Almeida and Ondrej Klíma. New decidable upper bound of the second level in the Straubing-Thérien concatenation hierarchy of star-free languages. *Discrete Mathematics & Theoretical Computer Science*, 12(4):41–58, 2010.

3 Emmanuel Arrighi, Henning Fernau, Stefan Hoffmann, Markus Holzer, Ismaël Jecker, Mateus de Oliveira Oliveira, and Petra Wolf. On the Complexity of Intersection Non-emptiness for Star-Free Language Classes. *CoRR*, abs/2110.01279, 2021. URL: `http://arxiv.org/abs/2110.01279`, `arXiv:2110.01279`.

4 Ahmed Bouajjani, Bengt Jonsson, Marcus Nilsson, and Tayssir Touili. Regular model checking. In E. Allen Emerson and A. Prasad Sistla, editors, *Computer Aided Verification, 12th International Conference, CAV*, volume 1855 of *Lecture Notes in Computer Science*, pages 403–418. Springer, 2000.

5 Ahmed Bouajjani, Anca Muscholl, and Tayssir Touili. Permutation rewriting and algorithmic verification. *Information and Computation*, 205:199–224, 2007.

6 Janusz A. Brzozowski. Hierarchies of aperiodic languages. *RAIRO Informatique théorique et Applications/Theoretical Informatics and Applications*, 10(2):33–49, 1976.

7 Janusz A. Brzozowski and Faith E. Fich. Languages of $\mathcal{R}$-trivial monoids. *Journal of Computer and System Sciences*, 20(1):32–49, February 1980.

8 Janusz A. Brzozowski and Robert Knast. The dot-depth hierarchy of star-free languages is infinite. *Journal of Computer and System Sciences*, 16(1):37–55, 1978.

9 Sang Cho and Dung T. Huynh. Finite-automaton aperiodicity is PSPACE-complete. *Theoretical Computer Science*, 88(1):99–116, September 1991.

10 Marek Chrobak. Finite automata and unary languages. *Theoretical Computer Science*, 47(3):149–158, 1986.

11 Rina S. Cohen and Janusz A. Brzozowski. Dot-depth of star-free events. *Journal of Computer and System Sciences*, 5(1):1–16, 1971.

12 Henning Fernau and Andreas Krebs. Problems on finite automata and the exponential time hypothesis. *Algorithms*, 10(1):24, 2017.

13 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Springer, 2006.

14 Pawel Gawrychowski. Chrobak normal form revisited, with applications. In Béatrice Bouchou-Markhoff, Pascal Caron, Jean-Marc Champarnaud, and Denis Maurel, editors, *Implementation and Application of Automata - 16th International Conference, CIAA*, volume 6807 of *Lecture Notes in Computer Science*, pages 142–153. Springer, 2011.

15 Christian Glaßer and Heinz Schmitz. Decidable hierarchies of starfree languages. In Sanjiv Kapoor and Sanjiva Prasad, editors, *Foundations of Software Technology and Theoretical Computer Science, 20th Conference, FST TCS*, volume 1974 of *Lecture Notes in Computer Science*, pages 503–515. Springer, 2000.

16 Christian Glaßer and Heinz Schmitz. Level 5/2 of the Straubing-Thérien hierarchy for two-letter alphabets. In Werner Kuich, Grzegorz Rozenberg, and Arto Salomaa, editors, *Developments in Language Theory, 5th International Conference, DLT*, volume 2295 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.

17 Juris Hartmanis, Neil Immerman, and Stephen R. Mahaney. One-way log-tape reductions. In *19th Annual Symposium on Foundations of Computer Science, FOCS*, pages 65–72. IEEE Computer Society, 1978.

18 Stefan Hoffmann. Regularity conditions for iterated shuffle on commutative regular languages. *accepted at CIAA*, 2021.

19 Harry B. Hunt III and Daniel J. Rosenkrantz. Computational parallels between the regular and context-free languages. *SIAM Journal on Computing*, 7(1):99–114, 1978.

20 George Karakostas, Richard J. Lipton, and Anastasios Viglas. On the complexity of intersecting finite state automata and NL versus NP. *Theoretical Computer Science*, 302(1):257–274, 2003.

21 Takumi Kasai and Shigeki Iwata. Gradually intractable problems and nondeterministic log-space lower bounds. *Mathematical Systems Theory*, 18(1):153–170, 1985.

22 Felix Klein and Martin Zimmermann. How much lookahead is needed to win infinite games? *Logical Methods in Computer Science*, 12(3), 2016. doi:10.2168/LMCS-12(3:4)2016.

23 Ondrej Klíma and Libor Polák. Subhierarchies of the second level in the straubing-thérien hierarchy. *International Journal of Algebra and Computation*, 21(7):1195–1215, 2011.

24 Dexter Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science, FOCS*, pages 254–266. IEEE Computer Society, 1977.

25 Markus Krötsch, Tomás Masopust, and Michaël Thomazo. Complexity of universality and related problems for partially ordered NFAs. *Information and Computation*, 255:177–192, 2017.

26 Klaus-Jörn Lange and Peter Rossmanith. The emptiness problem for intersections of regular languages. In Ivan M. Havel and Václav Koubek, editors, *Mathematical Foundations of Computer Science 1992, 17th International Symposium, MFCS*, volume 629 of *Lecture Notes in Computer Science*, pages 346–354. Springer, 1992.

27 Tomás Masopust. Separability by piecewise testable languages is PTime-complete. *Theoretical Computer Science*, 711:109–114, 2018.

28 Tomás Masopust and Markus Krötzsch. Partially ordered automata and piecewise testability. *CoRR*, abs/1907.13115, 2019. arXiv:1907.13115.

29 Tomás Masopust and Michaël Thomazo. On the complexity of k-piecewise testability and the depth of automata. In Igor Potapov, editor, *Developments in Language Theory - 19th International Conference, DLT*, number 9168 in Lecture Notes in Computer Science, pages 364–376. Springer, 2015.

30 Mike Naylor. Abacaba! – using a mathematical pattern to connect art, music, poetry and literature. *Bridges*, pages 89–96, 2011.

31 Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

32 Jean-Éric Pin. Bridges for concatenation hierarchies. In Kim Guldstrand Larsen, Sven Skyum, and Glynn Winskel, editors, *Automata, Languages and Programming, 25th International Colloquium, ICALP*, volume 1443 of *Lecture Notes in Computer Science*, pages 431–442. Springer, 1998.

33 Thomas Place and Marc Zeitoun. Generic results for concatenation hierarchies. *Theory of Computing Systems*, 63(4):849–901, 2019.

34 Narad Rampersad and Jeffrey Shallit. Detecting patterns in finite regular and context-free languages. *Information Processing Letters*, 110(3):108–112, 2010.

35 Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

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

37 Thomas Schwentick, Denis Thérien, and Heribert Vollmer. Partially-ordered two-way automata: A new characterization of DA. In Werner Kuich, Grzegorz Rozenberg, and Arto Salomaa, editors, *Developments in Language Theory, 5th International Conference, DLT*, volume 2295 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2001.

38 Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time: Preliminary report. In Alfred V. Aho, Allan Borodin, Robert L. Constable, Robert W. Floyd, Michael A. Harrison, Richard M. Karp, and H. Raymond Strong, editors, *5th Annual Symposium on Theory of Computing, STOC*, pages 1–9. ACM, 1973.

39 Howard Straubing. A generalization of the Schützenberger product of finite monoids. *Theoretical Computer Science*, 13:137–150, 1981.

40 Howard Straubing. Finite semigroup varieties of the form V^*D. *Journal of Pure and Applied Algebra*, 36:53–94, 1985.

41 Ivan Hal Sudborough. On tape-bounded complexity classes and multihead finite automata. *Journal of Computer and System Sciences*, 10(1):62–76, February 1975.

42 Denis Thérien. Classification of finite monoids: the language approach. *Theoretical Computer Science*, 14(2):195–208, 1981.

43 Todd Wareham. The parameterized complexity of intersection and composition operations on sets of finite-state automata. In Sheng Yu and Andrei Paun, editors, *Implementation and Application of Automata, 5th International Conference, CIAA*, volume 2088 of *Lecture Notes in Computer Science*, pages 302–310. Springer, 2000.

44 Michael Wehar. Hardness results for intersection non-emptiness. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP, Part II*, volume 8573 of *Lecture Notes in Computer Science*, pages 354–362. Springer, 2014.

45 Michael Wehar. *On the Complexity of Intersection Non-Emptiness Problems*. PhD thesis, University at Buffalo, 2016.