Piecewise Testable Languages and Nondeterministic Automata*

Tomáš Masopust

Institute of Theoretical Computer Science and Center of Advancing Electronics Dresden (cfaed), TU Dresden, Dresden, Germany
tomas.masopust@tu-dresden.de

Abstract

A regular language is k-piecewise testable if it is a finite boolean combination of languages of the form $\Sigma^* a_1 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$, where $a_i \in \Sigma$ and $0 \le n \le k$. Given a DFA $\mathcal{A}$ and $k \ge 0$, it is an NL-complete problem to decide whether the language $L(\mathcal{A})$ is piecewise testable and, for $k \ge 4$, it is coNP-complete to decide whether the language $L(\mathcal{A})$ is k-piecewise testable. It is known that the depth of the minimal DFA serves as an upper bound on k. Namely, if $L(\mathcal{A})$ is piecewise testable, then it is k-piecewise testable for k equal to the depth of $\mathcal{A}$. In this paper, we show that some form of nondeterminism does not violate this upper bound result. Specifically, we define a class of NFAs, called ptNFAs, that recognize piecewise testable languages and show that the depth of a ptNFA provides an (up to exponentially better) upper bound on k than the minimal DFA. We provide an application of our result, discuss the relationship between k-piecewise testability and the depth of NFAs, and study the complexity of k-piecewise testability for ptNFAs.

1998 ACM Subject Classification F.1.1 Models of Computation, F.4.3 Formal Languages

Keywords and phrases Automata, Logics, Languages, k-piecewise testability, Nondeterminism

Digital Object Identifier 10.4230/LIPIcs.MFCS.2016.67

1 Introduction

A regular language L over an alphabet Σ is *piecewise testable* if it is a finite boolean combination of languages of the form

$$L_{a_1 a_2 \dots a_n} = \Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$$

where $a_i \in \Sigma$ and $n \ge 0$. If L is piecewise testable, then there exists a nonnegative integer k such that L is a finite boolean combination of languages L_u, where the length of $u \in \Sigma^*$ is at most k. In this case, the language L is called *k-piecewise testable*.

Piecewise testable languages are studied in semigroup theory [2, 3, 28] and in logic over words [10, 29] because of their close relation to first-order logic FO(<). They actually form the first level of the Straubing-Thérien hierarchy [27, 36]. This hierarchy is closely related to the dot-depth hierarchy [7], see more in [23]. They are indeed studied in formal languages and automata theory [20], recently mainly in the context of separation [29, 38]. Although the separability of context-free languages by regular languages is undecidable, separability by piecewise testable languages is decidable [9] (even for some non-context-free families). Piecewise testable languages form a strict subclass of star-free languages, that is, of the limit of the above-mentioned hierarchies or, in other words, of the languages definable by

* This work was supported by the by the German Research Foundation (DFG) in Emmy Noether grant KR 4381/1-1 (DIAMOND).

© Tomáš Masopust;
licensed under Creative Commons License CC-BY
41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016).
Editors: Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier; Article No. 67; pp. 67:1–67:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

LTL logic. They are investigated in natural language processing [11, 30], in cognitive and sub-regular complexity [31], in learning theory [12, 21], and in databases in the context of XML schema languages [8, 14, 15]. They have been extended from words to trees [4, 13].

Recently, the complexity of computing the minimal k and/or bounds on k for which a piecewise testable language is k-piecewise testable was studied in [14, 19, 20], motivated by applications in databases and in algebra and logic. However, the knowledge of such a k that is either minimal or of reasonable size is of interest in many other applications as well, see, e.g., [24]. The complexity to test whether a piecewise testable language is k-piecewise testable was shown to be coNP-complete for $k \geq 4$ if the language is given as a DFA [19] and PSPACE-complete if the language is given as an NFA [25]. The complexity for DFAs and $k < 4$ is discussed in detail in [25]. The best upper bound on k known so far is given by the depth of the minimal DFA [20].

In this paper, we define a class of NFAs, called ptNFAs, that characterizes piecewise testable languages. This characterization is based on purely structural properties, therefore it is NL-complete to check whether an NFA is a ptNFA (Theorem 5). We show that the depth of ptNFAs also provides an upper bound on k-piecewise testability (Theorem 8) and that this new bound is up to exponentially lower than the one given by minimal DFAs (Section 3 and Theorem 15). We further show that this property does not hold for general NFAs, and that the gap between k-piecewise testability and the depth of NFAs can be arbitrarily large (Lemma 12). The opposite implication of Theorem 8 does not hold and a brief discussion is provided. We give a non-trivial application of our result in Section 5, where we also provide more discussion. Finally, in Section 6, we study the complexity of k-piecewise testability for ptNFAs.

The paper is organized as follows. Section 2 presents basic notions and definitions, fixes the notation, and defines the ptNFAs. Section 3 motivates and demonstrates Theorem 8 on a simple example. Section 4 then proves Theorem 8 and the related results. Section 5 provides a non-trivial application and further discussion. Section 6 recalls the known complexity results and studies the complexity of the related problems for ptNFAs. Section 7 concludes the paper.

2 Preliminaries and Definitions

We assume that the reader is familiar with automata theory, see, e.g., [1]. The cardinality of a set A is denoted by $|A|$ and the power set of A by 2^A. An alphabet, Σ, is a finite nonempty set; the elements of an alphabet are called symbols or letters. The free monoid generated by Σ is denoted by Σ^*. A word over Σ is any element of Σ^*; the empty word is denoted by ε. For a word $w \in \Sigma^*$, $\mathrm{alph}(w) \subseteq \Sigma$ denotes the set of all letters occurring in w, and $|w|_a$ denotes the number of occurrences of letter a in w. A language over Σ is a subset of Σ^*. For a language L over Σ, let $\overline{L} = \Sigma^* \setminus L$ denote the complement of L.

A *nondeterministic finite automaton* (NFA) is a quintuple $\mathcal{A} = (Q, \Sigma, \cdot, I, F)$, where Q is a finite nonempty set of states, Σ is an input alphabet, $I \subseteq Q$ is a set of initial states, $F \subseteq Q$ is a set of accepting states, and $\cdot : Q \times \Sigma \to 2^Q$ is the transition function that can be extended to the domain $2^Q \times \Sigma^*$ by induction. The language *accepted* by $\mathcal{A}$ is the set $L(\mathcal{A}) = \{w \in \Sigma^* \mid I \cdot w \cap F \neq \emptyset\}$. In what follows, we usually omit $\cdot$ and write simply Iw instead of $I \cdot w$.

A *path* π from a state q_0 to a state q_n under a word $a_1 a_2 \cdots a_n$, for some $n \geq 0$, is a sequence of states and input symbols $q_0 a_1 q_1 a_2 \ldots q_{n-1} a_n q_n$ such that $q_{i+1} \in q_i \cdot a_{i+1}$, for all $i = 0, 1, \ldots, n-1$. The path π is *accepting* if $q_0 \in I$ and $q_n \in F$. We use the notation

$q_0 \xrightarrow{a_1 a_2 \cdots a_n} q_n$ to denote that there exists a path from q_0 to q_n under the word $a_1 a_2 \cdots a_n$. A path is *simple* if all states of the path are pairwise distinct. The number of states on the longest simple path of $\mathcal{A}$, starting in an initial state, decreased by one (i.e., the number of transitions on that path) is called the *depth* of the automaton $\mathcal{A}$, denoted by $depth(\mathcal{A})$.

The NFA $\mathcal{A}$ is *complete* if for every state q of $\mathcal{A}$ and every letter a in Σ, the set $q \cdot a$ is nonempty, that is, in every state, a transition under every letter is defined.

Let $\mathcal{A} = (Q, \Sigma, \cdot, I, F)$ be an NFA, and let p be a state of $\mathcal{A}$. The sub-automaton of $\mathcal{A}$ induced by state p is the automaton $\mathcal{A}_p = (reach(p), \Sigma, \cdot_p, p, F \cap reach(p))$ with state p being the sole initial state and with only those states of $\mathcal{A}$ that are reachable from p; formally, $reach(p)$ denotes the set of all states reachable from state p in $\mathcal{A}$ and $\cdot_p$ is a restriction of $\cdot$ to $reach(p) \times \Sigma$.

The NFA $\mathcal{A}$ is *deterministic* (DFA) if $|I| = 1$ and $|q \cdot a| = 1$ for every state q in Q and every letter a in Σ. Then the transition function $\cdot$ is a map from $Q \times \Sigma$ to Q that can be extended to the domain $Q \times \Sigma^*$ by induction. Two states of a DFA are *distinguishable* if there exists a word w that is accepted from one of them and rejected from the other. A DFA is *minimal* if all its states are reachable and pairwise distinguishable.

Let $\mathcal{A} = (Q, \Sigma, \cdot, I, F)$ be an NFA. The reachability relation $\leq$ on the set of states is defined by $p \leq q$ if there exists a word w in Σ^* such that $q \in p \cdot w$. The NFA $\mathcal{A}$ is *partially ordered* if the reachability relation $\leq$ is a partial order. In other words, the automaton is acyclic, but self-loops are allowed. Therefore, partially ordered automata are sometimes also called acyclic automata. For two states p and q of $\mathcal{A}$, we write $p < q$ if $p \leq q$ and $p \neq q$. A state p is *maximal* if there is no state q such that $p < q$.

An NFA $\mathcal{A} = (Q, \Sigma, \cdot, I, F)$ can be turned into a directed graph $G(\mathcal{A})$ with the set of vertices Q, where a pair (p, q) in $Q \times Q$ is an edge in $G(\mathcal{A})$ if there is a transition from p to q in $\mathcal{A}$. For $\Gamma \subseteq \Sigma$, we define the directed graph $G(\mathcal{A}, \Gamma)$ with the set of vertices Q by considering all those transitions that correspond to letters in Γ. For a state p, let $\Sigma(p) = \{a \in \Sigma \mid p \in p \cdot a\}$ denote the set of all letters under which the NFA $\mathcal{A}$ has a self-loop in state p. Let $\mathcal{A}$ be a partially ordered NFA. If for every state p of $\mathcal{A}$, state p is the unique maximal state of the connected component of $G(\mathcal{A}, \Sigma(p))$ containing p, then we say that the NFA satisfies the *unique maximal state (UMS) property.*

An equivalent notion to the UMS property for minimal DFAs has been introduced in the literature. A DFA $\mathcal{A}$ over Σ is *confluent* if, for every state q of $\mathcal{A}$ and every pair of letters a, b in Σ, there exists a word w in $\{a, b\}^*$ such that $(qa)w = (qb)w$.

We adopt the notation $L_{a_1 a_2 \cdots a_n} = \Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$ from [20]. For two words $v = a_1 a_2 \cdots a_n$ and $w \in L_v$, we say that v is a *subsequence* of w or that v can be *embedded* into w, denoted by $v \preccurlyeq w$. For $k \geq 0$, let $sub_k(v) = \{u \in \Sigma^* \mid u \preccurlyeq v, |u| \leq k\}$. For two words w_1, w_2, we define $w_1 \sim_k w_2$ if and only if $sub_k(w_1) = sub_k(w_2)$. Note that $\sim_k$ is a congruence with finite index.

The following is well known.

▶ **Fact 1** ([33]). *Let L be a regular language, and let $\sim_L$ denote the Myhill congruence [26]. A language L is k-piecewise testable if and only if $\sim_k \subseteq \sim_L$. Moreover, L is a finite union of $\sim_k$ classes.*

In what follows, we will use this fact in several proofs in the form that if L is not k-piecewise testable, then there exist two words u and v such that $u \sim_k v$ and $|L \cap \{u, v\}| = 1$.

▶ **Fact 2.** *Let L be a language recognized by the minimal DFA $\mathcal{A}$. The following is equivalent.*

1. *The language L is piecewise testable.*

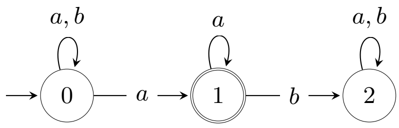

Figure 1 Confluent automaton accepting a non-piecewise testable language.

2. *The minimal DFA $\mathcal{A}$ is partially ordered and confluent [20].*
3. *The minimal DFA $\mathcal{A}$ is partially ordered and satisfies the UMS property [37].*

We now define a special class of nondeterministic automata called ptNFAs. The name comes from piecewise testable, since, as we show below, they characterize piecewise testable languages. And indeed include all minimal DFAs recognizing piecewise testable languages.

▶ **Definition 3.** An NFA $\mathcal{A}$ is called a *ptNFA* if it is partially ordered, complete, and satisfies the UMS property.

The reason why we use the UMS property in the definition of ptNFAs rather than confluence is simply because confluence does not naturally generalize to NFAs as shown in Example 4 below. Moreover, it is known that partially ordered NFAs characterize the level $\frac{3}{2}$ of the Straubing-Thérien hierarchy [32] and that partially ordered NFAs satisfying that $q \in q \cdot a$ implies $q \cdot a = \{q\}$ characterize $\mathcal{R}$-trivial languages [5, 22]. It can be shown that adding confluence and completeness on top of these properties results in ptNFAs.

▶ **Example 4.** Consider the automaton depicted in Figure 1. The notion of confluence is not clear for NFAs. If we consider the point of view that whenever the computation is split, a common state can be reached under a word over the splitting alphabet, then this automaton is confluent. However, it does not satisfy the UMS property and its language is not piecewise testable; there is an infinite sequence $a, ab, aba, abab, \ldots$ that alternates between accepted and non-accepted words, which implies that there is a non-trivial cycle in the corresponding minimal DFA and, thus, it proves non-piecewise testability by Fact 2.

Note that to check whether an NFA is a ptNFA requires to check whether the automaton is partially ordered, complete and satisfies the UMS property. The violation of these properties can be tested by several reachability tests, hence its complexity belongs to coNL=NL. On the other hand, to check the properties is NL-hard even for minimal DFAs [6]. Thus, we have the following.

▶ **Theorem 5.** *It is NL-complete to check whether an NFA is a ptNFA.*

3 Motivation and an Example

Considering applications, such as XML, where the alphabet can hardly be considered as fixed, the results of [19] (cf. Theorem 18 below) say that it is intractable to compute the minimal k for which a piecewise testable language is k-piecewise testable, unless P=NP. This leads to the investigation of reasonably small upper bounds. The result of [20] says that k is bounded by the depth of the minimal DFA. However, applications usually require to work with NFAs, which motivates the research of this paper. Another motivation comes from a simple observation that, given several DFAs, a result of an operation can lead to an NFA that in some sense still has the DFA-like properties, see more discussion below. Moreover, it seems to be a human nature to use a kind of nondeterminism, for instance, to reuse already defined parts as demonstrated here on a very simple example.

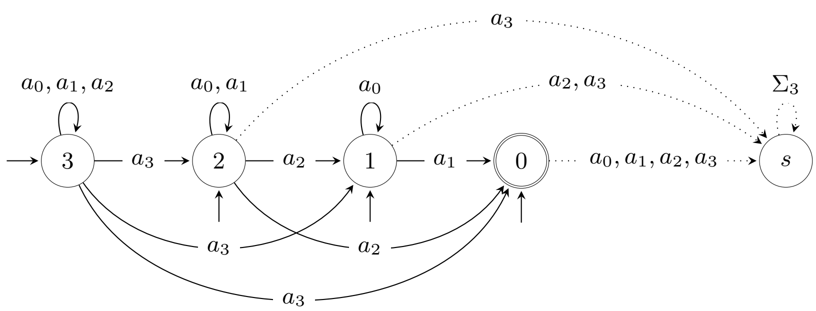

Figure 2 Automaton $\mathcal{A}_3$; the dotted transitions depict the completion of $\mathcal{A}_3$.

Let $L_0 = \{\varepsilon\}$ be a language over the alphabet $\Sigma_0 = \{a_0\}$. Assume that the language L_i over Σ_i is defined, and let $L_{i+1} = L_i \cup \Sigma_i^* a_{i+1} L_i$ over $\Sigma_{i+1} = \Sigma_i \cup \{a_{i+1}\}$, where a_{i+1} is a new symbol not in Σ_i. We now construct the NFAs for the languages L_i,

$$\mathcal{A}_i = (\{0, 1, \ldots, i\}, \{a_0, a_1, \ldots, a_i\}, \cdot, \{0, 1, \ldots, i\}, \{0\})$$

where $\ell \cdot a_j = \ell$ if $i \geq \ell > j \geq 0$ and $\ell \cdot a_\ell = \{0, 1, \ldots, \ell - 1\}$ if $i \geq \ell \geq 1$. The automaton $\mathcal{A}_3$ is depicted in Figure 2. The dotted transitions are to complete the NFA in the meaning that $\ell \cdot a \neq \emptyset$ for any state ℓ and letter a.

Although the example is very simple, the reader can see the point of the construction in nondeterministically reusing the existing parts.

Now, to decide whether the language is piecewise testable and, if so, to obtain an upper bound on its k-piecewise testability, a naive application of the known results for DFAs requires to compute the minimal DFA. Doing so shows that L_i is piecewise testable. However, the minimal DFA for the language L_i is of exponential size and its depth is $2^{i+1} - 1$, cf. [25], which implies that L_i is $(2^{i+1} - 1)$-piecewise testable. Another way is to use the PSPACE algorithm of [25] to compute the minimal k. Both approaches are basically of the same complexity.

This is the place, where our result comes into the picture. According to Theorem 8 proved in the next section, the easily testable structural properties say that the language L_i is $(i + 1)$-piecewise testable. This provides an exponentially better upper bound for every language L_i than the technique based on minimal DFAs. Finally, we note that it can be shown that L_i is not i-piecewise testable, so the bound is tight for L_i.

4 Piecewise Testability and Nondeterminism

In this section, we establish a relation between piecewise testable languages and nondeterministic automata and generalize the bound given by the depth of DFAs to ptNFAs. We first recall the known result for DFAs.

▶ **Theorem 6** ([20]). *Let $\mathcal{A}$ be a partially ordered and confluent DFA. If the depth of $\mathcal{A}$ is k, then the language $L(\mathcal{A})$ is k-piecewise testable.*

This result is currently the best known structural upper bound on k-piecewise testability. The opposite implication of the theorem does not hold and we have shown in [25] (see also Section 3) that this bound can be exponentially far from the minimal value of k.

This observation has motivated our investigation of the relationship between piecewise testability and the depth of NFAs. We have already generalized a structural automata characterization for piecewise testability from DFAs to NFAs as follows.

▶ **Theorem 7** ([25]). *A regular language is piecewise testable if and only if it is recognized by a ptNFA.*

We now generalize Theorem 6 to ptNFAs and discuss the relation between the depth of NFAs and k-piecewise testability in more detail. An informal idea behind the proof is that every ptNFA can be "decomposed" into a finite number of partially ordered and confluent DFAs. We now formally prove the theorem by generalizing the proof of Theorem 6 given in [20].

▶ **Theorem 8.** *If the depth of a ptNFA $\mathcal{A}$ is k, then the language $L(\mathcal{A})$ is k-piecewise testable.*

The proof of Theorem 8 follows directly from Lemmas 9 and 11 proved below.

▶ **Lemma 9.** *Let $\mathcal{A}$ be a ptNFA with I denoting the set of initial states. Then the language $L(\mathcal{A}) = \bigcup_{i\in I} L(\mathcal{A}_i)$, where every sub-automaton $\mathcal{A}_i$ is a ptNFA.*

Based on the previous lemma, it is sufficient to show the theorem for ptNFAs with a single initial state. We make use of the following lemma.

▶ **Lemma 10** ([20]). *Let $\ell \geq 1$, and let $u, v \in \Sigma^*$ be such that $u \sim_\ell v$. Let $u = u'au''$ and $v = v'av''$ such that $a \notin \mathrm{alph}(u'v')$. Then $u'' \sim_{\ell-1} v''$.*

▶ **Lemma 11.** *Let $\mathcal{A}$ be a ptNFA with a single initial state and depth k. Then the language $L(\mathcal{A})$ is k-piecewise testable.*

Proof. Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$. If the depth of $\mathcal{A}$ is 0, then $L(\mathcal{A})$ is either $\emptyset$ or Σ^*, which are both 0-piecewise testable by definition. Thus, assume that the depth of $\mathcal{A}$ is $\ell \geq 1$ and that the claim holds for ptNFAs of depth less than ℓ. Let $u, v \in \Sigma^*$ be such that $u \sim_\ell v$. We prove that u is accepted by $\mathcal{A}$ if and only if v is accepted by $\mathcal{A}$.

Assume that u is accepted by $\mathcal{A}$ and fix an accepting path of u in $\mathcal{A}$. If $\mathrm{alph}(u) \subseteq \Sigma(i)$, then the UMS property of $\mathcal{A}$ implies that $i \in F$. Therefore, v is also accepted in i. If $\mathrm{alph}(u) \not\subseteq \Sigma(i)$, then $u = u'au''$ and $v = v'bv''$, where $u', v' \in \Sigma(i)^*$, $a, b \in \Sigma \setminus \Sigma(i)$, and $u'', v'' \in \Sigma^*$. Let $p \in i \cdot a$ be a state on the fixed accepting path of u. Let $\mathcal{A}_p = (\mathit{reach}(p), \Sigma, \cdot_p, p, F \cap \mathit{reach}(p))$ be a sub-automaton of $\mathcal{A}$ induced by state p. Note that $\mathcal{A}_p$ is a ptNFA. By assumption, $\mathcal{A}_p$ accepts u'' and the depth of $\mathcal{A}_p$ is at most $\ell - 1$.

If $a = b$, Lemma 10 implies that $u'' \sim_{\ell-1} v''$. By the induction hypothesis, u'' is accepted by $\mathcal{A}_p$ if and only if v'' is accepted by $\mathcal{A}_p$. Hence, $v = v'av''$ is accepted by $\mathcal{A}$.

If $a \neq b$, then $u = u'au_0''bu_1''$ and $v = v'bv_0''av_1''$, where $b \notin \mathrm{alph}(u'au_0'')$ and $a \notin \mathrm{alph}(v'bv_0'')$. Then

$$u'' = u_0''bu_1'' \sim_{\ell-1} v_0''av_1'' = v''$$

because, by Lemma 10,

$$\mathit{sub}_{\ell-1}(u_0''bu_1'') = \mathit{sub}_{\ell-1}(v_1'') \subseteq \mathit{sub}_{\ell-1}(v_0''av_1'') = \mathit{sub}_{\ell-1}(u_1'') \subseteq \mathit{sub}_{\ell-1}(u_0''bu_1'')\,. \qquad (*)$$

If $p \in i \cdot b$, the induction hypothesis implies that v'' is accepted by $\mathcal{A}_p$, hence $v = v'bv''$ is accepted by $\mathcal{A}$.

If $p \notin i \cdot b$, let $q \in i \cdot b$. By the UMS property of $\mathcal{A}$, there exists a word $w \in \{a, b\}^*$ such that $pw = qw = r$, for some state r with $a, b \in \Sigma(r)$. Indeed, there exists w_1 and a unique maximal state r with respect to $\{a,b\}$ such that $pw_1 = \{r\}$ and $a,b \in \Sigma(r)$. By the UMS property, there exists w_2 over $\{a,b\}$ such that $qw_1w_2 = \{r\}$. Let $w = w_1w_2$. We now show that $wu'' \sim_{\ell-1} u''$ by induction on the length of w. There is nothing to show for $w = \varepsilon$. Thus, assume that $w = xw'$, for $x \in \{a,b\}$, and that $w'u'' \sim_{\ell-1} u''$. Notice that (*) shows that $u'' \sim_{\ell-1} v_1'' \sim_{\ell-1} v'' \sim_{\ell-1} u_1''$. This implies that $sub_{\ell-1}(v_1'') \subseteq sub_{\ell-1}(av_1'') \subseteq sub_{\ell-1}(v_0''av_1'') = sub_{\ell-1}(v'') = sub_{\ell-1}(v_1'')$, which shows that $av_1'' \sim_{\ell-1} v_1''$. Analogously, we can show that $bu_1'' \sim_{\ell-1} u_1''$. If $x = a$, then $w'u'' \sim_{\ell-1} u'' \sim_{\ell-1} v_1''$ implies that $aw'u'' \sim_{\ell-1} av_1'' \sim_{\ell-1} v_1'' \sim_{\ell-1} u''$. If $x = b$, then $w'u'' \sim_{\ell-1} u'' \sim_{\ell-1} u_1''$ implies that $bw'u'' \sim_{\ell-1} bu_1'' \sim_{\ell-1} u_1'' \sim_{\ell-1} u''$. Therefore, $wu'' \sim_{\ell-1} u''$; similarly, $wv'' \sim_{\ell-1} v''$.

Finally, using the induction hypothesis (of the main statement) on $\mathcal{A}_p$, we get that u'' is accepted by $\mathcal{A}_p$ if and only if wu'' is accepted by $\mathcal{A}_p$, which is if and only if u'' is accepted by $\mathcal{A}_r$. Since $u'' \sim_{\ell-1} v''$, the induction hypothesis applied on $\mathcal{A}_r$ gives that u'' is accepted by $\mathcal{A}_r$ if and only if v'' is accepted by $\mathcal{A}_r$. However, this is if and only if wv'' is accepted by $\mathcal{A}_q$. Using the induction hypothesis on $\mathcal{A}_q$, we obtain that wv'' is accepted by $\mathcal{A}_q$ if and only if v'' is accepted by $\mathcal{A}_q$. Together, the assumption that u'' is accepted by $\mathcal{A}_p$ implies that v'' is accepted by $\mathcal{A}_q$. Hence $v = v'bv''$ is accepted by $\mathcal{A}$, which completes the proof. ◀

In other words, the previous theorem says that if k is the minimum number for which a piecewise testable language L is k-piecewise testable, then the depth of any ptNFA recognizing L is at least k.

It is natural to ask whether this property holds for any NFA recognizing the language L. The following result shows that it is not the case. Actually, for any natural number ℓ, there exists a piecewise testable language such that the difference between its k-piecewise testability and the depth of an NFA is at least ℓ.

▶ **Lemma 12.** *For every $k \geq 3$, there exists a k-piecewise testable language that is recognized by an NFA of depth at most $\lfloor \frac{k}{2} \rfloor$.*

Proof. For every $i \geq 1$, let $L_i = a^i + a^{2i+1} \cdot a^*$. We show that the language L_i is $(2i+1)$-piecewise testable and that there exists an NFA of depth at most i recognizing it.

The minimal DFA for L_i consists of $2i+1$ states $\{0,1,\ldots,2i+1\}$, where 0 is the initial state, i and $2i+1$ are accepting, $p \cdot a = p+1$ for $p < 2i+1$, and $(2i+1) \cdot a = 2i+1$. The depth is $2i+1$, which shows that L_i is $(2i+1)$-piecewise testable. Notice that $a^{2i} \sim_{2i} a^{2i+1}$, but a^{2i} does not belong to L_i, hence L_i is not $2i$-piecewise testable.

The NFA for L_i consists of two cycles of length $i+1$, the structure is depicted in Figure 3. The initial state is state 0 and the solely accepting state is state i. The automaton accepts L_i. Indeed, it accepts a^i and no shorter word. After reading a^i, the automaton is in state i or i'. In both cases, the shortest nonempty path to the single accepting state i is of length $i+1$. Thus, the automaton accepts a^{2i+1}, but nothing between a^i and a^{2i+1}. Finally, using the self-loop in state i', the automaton accepts $a^ia^*a^{i+1} = a^{2i+1}a^*$. The depth of the automaton is i. ◀

Piecewise testability and the depth of NFAs. Theorem 8 gives rise to a question whether the opposite implication holds true. This is not the case. Notice that although the depth of ptNFAs is more suitable to provide bounds on k-piecewise testability, the depth is significantly influenced by the size of the input alphabet. For instance, for an alphabet Σ, the language $L = \bigcap_{a\in\Sigma} L_a$ of all words containing all letters of Σ is a 1-piecewise testable language such that any NFA recognizing it requires at least $2^{|\Sigma|}$ states and is of depth $|\Sigma|$, cf. [25]. The

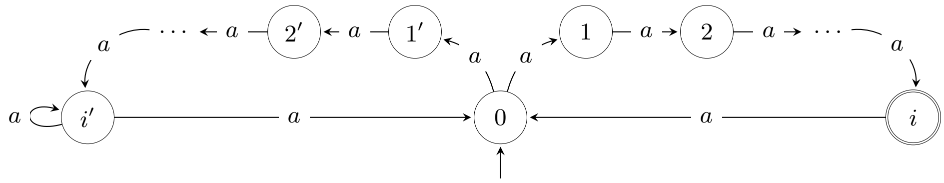

Figure 3 The NFA of depth i recognizing L_i.

depth follows from the fact that the shortest accepted word is of length $|\Sigma|$, hence any path from an initial state to an accepting state must be of length at least $|\Sigma|$.

The dependence on the alphabet is even stronger as shown below.

▶ **Lemma 13.** *For any alphabet of cardinality $n > 1$, there exists an n^2-piecewise testable language such that any NFA recognizing it is of depth at least n^{n-1}.*

Proof. Let $L_n(k)$ denote the maximal length of the shortest representatives of the $\sim_k$-classes over an n-element alphabet. It was shown in [18] that $(L_n(k)+1)\log n > (\frac{k}{n})^{n-1}\log(\frac{k}{n})$. Setting $k = n^2$ then gives that $L_n(n^2) \geq n^{n-1}$. Let L be a language defined by a single $\sim_{n^2}$-class with shortest representatives of length $L_n(n^2)$. Then L is n^2-piecewise testable, since it is defined as a union of $\sim_{n^2}$ classes. Consider any NFA recognizing L. Since the shortest word of L is of length $L_n(n^2)$, any path from an initial state to an accepting state must be of length at least $L_n(n^2)$. ◀

Recall that it was independently shown in [19, 25] that, given a k-piecewise testable language over an n-letter alphabet, the tight upper bound on the depth of the minimal DFA recognizing it is $\binom{k+n}{k} - 1$. In other words, this formula gives the tight upper bound on the depth of the $\sim_k$-canonical DFA [25] over an n element alphabet. A related question on the size of this DFA is still open, see [18] for more details.

▶ **Theorem 14** ([19, 25]). *For any natural numbers k and n, the depth of the minimal DFA recognizing a k-piecewise testable language over an n-letter alphabet is at most $\binom{k+n}{k} - 1$. The bound is tight for any k and n.*

The lower bound for NFAs and ptNFAs remains open.

5 Application and Discussion

The reader might have noticed that the reverse of the automaton $\mathcal{A}_i$ constructed in Section 3 is deterministic and, when made complete, it satisfies the conditions of Fact 2. Since, by definition, a language is k-piecewise testable if and only if its reverse is k-piecewise testable, this observation provides the same upper bound $i+1$ on k-piecewise testability of the language $L(\mathcal{A}_i)$. However, this is just a coincidence and it is not difficult to find an example of a ptNFA whose reverse is not deterministic.

Since both the minimal DFA for L and the minimal DFA for L^R provide an upper bound on k, it could seem reasonable to compute both DFAs in parallel with the hope that (at least) one of them will be computed in a reasonable (polynomial) time. Although this may work for many cases (including the case of Section 3), we now show that there are cases where both the DFAs are of exponential size.

▶ **Theorem 15.** *For every $n \geq 0$, there exists a $(2n+1)$-state ptNFA $\mathcal{B}$ such that the depth of both the minimal DFA for $L(\mathcal{B})$ and the minimal DFA for $L(\mathcal{B})^R$ are exponential with respect to n.*

Proof sketch. The idea of the proof is to make use of the automaton $\mathcal{A}_i$ constructed in Section 3 to build a ptNFA $\mathcal{B}_i$ such that $L(\mathcal{B}_i) = L(\mathcal{A}_i) \cdot L(\mathcal{A}_i)^R$. Then $L(\mathcal{B}_i) = L(\mathcal{B}_i)^R$ and it can be shown that the minimal DFA recognizing the language $L(\mathcal{B}_i)$ requires an exponential number of states compared to $\mathcal{B}_i$. Namely, the depth of both the minimal DFA for $L(\mathcal{B}_i)$ and the minimal DFA for $L(\mathcal{B}_i)^R$ are of length at least $2^{i+1} - 1$. ◀

The previous proof provides another motivation to investigate nondeterministic automata for piecewise testable languages. Given several DFAs, the result of a sequence of operations may result in an NFA that preserves some good properties. Namely, the language $L(\mathcal{B}_i)$ from the previous proof is a result of the operation concatenation of a language L^R with L, where L is a piecewise testable language given as a DFA.

It immediately follows from Theorem 8 that the language $L(\mathcal{B}_i)$ is $(2i+1)$-piecewise testable. This result is not easily derivable from known results, which are either in PSPACE or require to compute an exponentially larger minimal DFA, which provides only the information that the language $L(\mathcal{B}_i)$ is k-piecewise testable for some $k \geq 2^{i+1} - 1$.

Even the information that the language $L(\mathcal{B}_i)$ is of the form $L^R \cdot L$, for a piecewise testable language L, does not seem very helpful, since piecewise testable languages are not closed under concatenation, even with its own reverse, as we show in the example below.

▶ **Example 16.** Let L be the language over the alphabet $\{a, b, c\}$ defined by the regular expression $ab^* + c(a+b)^*$. The reader can construct the minimal DFA for L and check that the properties of Fact 2 are satisfied. In addition, the depth of the minimal DFA is two, hence the language is 2-piecewise testable. Since the properties of Theorem 19 (see below) are not satisfied, the language L is not 1-piecewise testable.

On the other hand, the reader can notice that the sequence $ca, cab, caba, cabab, cababa, \ldots$ is an infinite sequence where every word on the odd position belongs to $L \cdot L^R$, whereas every word on the even position does not. This means that there exists a cycle in the minimal DFA recognizing $L \cdot L^R$, which shows that $L \cdot L^R$ is not a piecewise testable language according to Fact 2. The reader can also directly compute the minimal DFA for $L \cdot L^R$ and notice a non-trivial cycle in it.

To complete this part, we show that the language $L(\mathcal{B}_i)$ is not $(2i)$-piecewise testable. Thus, there are no ptNFAs recognizing the language $L(\mathcal{B}_i)$ with depth less than $2i + 1$.

▶ **Lemma 17.** *For every $i \geq 0$, the language $L(\mathcal{B}_i)$ is not $2i$-piecewise testable.*

6 Complexity

In this section, we first give an overview of known complexity results and characterization theorems for DFAs and then discuss the related complexity for ptNFAs.

Simon [33] proved that piecewise testable languages are exactly those regular languages whose syntactic monoid is $\mathcal{J}$-trivial, which shows decidability of the problem whether a regular language is piecewise testable. Later, Stern proved that the problem is decidable in polynomial time for languages represented as minimal DFAs [34], and Cho and Huynh [6] showed that it is NL-complete for DFAs. Trahtman [37] improved Stern's result by giving an algorithm quadratic in the number of states of the minimal DFA, and Klíma and Polák [20] presented an algorithm quadratic in the size of the alphabet of the minimal DFA. If the language is represented as an NFA, the problem is PSPACE-complete [16] (see more details below).

By definition, a regular language is piecewise testable if there exists $k \geq 0$ such that it is k-piecewise testable. It gives rise to a question to find such a minimal k. The *k-piecewise testability problem* asks, given an automaton, whether it recognizes a k-piecewise testable language. The problem is trivially decidable because there are only finitely many k-piecewise testable languages over a fixed alphabet. The coNP upper bound on k-piecewise testability for DFAs was independently shown in [14, 25].[1] The coNP-completeness for $k \geq 4$ was recently shown in [19]. The complexity holds even if k is given as part of the input. The complexity analysis of the problem for $k < 4$ is provided in [25]. We recall the results we need later.

▶ **Theorem 18** ([19])**.** *For $k \geq 4$, to decide whether a DFA represents a k-piecewise testable language is coNP-complete. It remains coNP-complete even if the parameter $k \geq 4$ is given as part of the input. For a fixed alphabet, the problem is decidable in polynomial time.*

It is not difficult to see that, given a minimal DFA, it is decidable in constant time whether its language is 0-piecewise testable, since it is either empty or Σ^*.

▶ **Theorem 19** (1-piecewise testability DFAs, [25])**.** *Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal DFA. Then $L(\mathcal{A})$ is 1-piecewise testable if and only if* (i) *for every $p \in Q$ and $a \in \Sigma$, $paa = pa$ and* (ii) *for every $p \in Q$ and $a, b \in \Sigma$, $pab = pba$. The problem is in AC^0.*

It is not hard to see that this result does not hold for ptNFAs. Indeed, one can simply consider a minimal DFA satisfying the properties and add a nondeterministic transition that violates them, but not the properties of ptNFAs. On the other hand, the conditions are still sufficient.

▶ **Lemma 20** (1-piecewise testability ptNFAs)**.** *Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a complete NFA. If* (i) *for every $p \in Q$ and $a \in \Sigma$, $paa = pa$ and* (ii) *for every $p \in Q$ and $a, b \in \Sigma$, $pab = pba$, then the language $L(\mathcal{A})$ is 1-piecewise testable.*

Note that any ptNFA $\mathcal{A}$ satisfying (i) must have $|pa| = 1$ for every state p and letter a. If $pa = \{r_1, r_2, \ldots, r_m\}$ with $r_1 < r_2 < \ldots < r_m$, then $paa = pa$ implies that $\{r_1, \ldots, r_m\}a = \{r_1, \ldots, r_m\}$. Then $r_1 \in r_1a$ and the UMS property says that $r_1a = \{r_1\}$. By induction, we can show hat $r_ia = \{r_i\}$. Consider the component of $G(\mathcal{A}, \Sigma(r_1))$ containing r_1. Then $r_1, \ldots, r_m$ all belong to this component. Since r_1 is maximal, r_1 is reachable from every r_i under $\Sigma(r_1) \supseteq \{a\}$. However, the partial order $r_1 < \ldots < r_m$ implies that r_1 is reachable from r_i only if $r_i = r_1$. Thus, $|pa| = 1$. However, $\mathcal{A}$ can still have many initial states, which can be seen as a finite union of piecewise testable languages rather then a nondeterminism.

The 2-piecewise testability characterization for DFAs is as follows.

▶ **Theorem 21** (2-piecewise testability DFAs, [25])**.** *Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal partially ordered and confluent DFA. The language $L(\mathcal{A})$ is 2-piecewise testable if and only if for every $a \in \Sigma$ and every state s such that $iw = s$ for some $w \in \Sigma^*$ with $|w|_a \geq 1$, $sba = saba$ for every $b \in \Sigma \cup \{\varepsilon\}$. The problem is NL-complete.*

It is again sufficient for ptNFAs.

[1] Actually, [14] gives the bound NEXPTIME for the problem for NFAs where k is part of the input. The coNP bound for DFAs can be derived from the proof omitted in the conference version. The problem is formulated in terms of separability, hence it requires the NFA for the language and for its complement.

▶ **Lemma 22** (2-piecewise testability ptNFAs). *Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a ptNFA. If for every $a \in \Sigma$ and every state s such that $iw = s$ for some $w \in \Sigma^*$ with $|w|_a \geq 1$, $sba = saba$ for every $b \in \Sigma \cup \{\varepsilon\}$, then the language $L(\mathcal{A})$ is 2-piecewise testable.*

Considering Theorem 18, the lower bound for DFAs is indeed a lower bound for ptNFAs. Thus, we immediately have that the k-piecewise testability problem for ptNFAs is coNP-hard for $k \geq 4$. We now show that it is actually coNP-hard for every $k \geq 0$. The proof is split into two lemmas.

The proof of the following lemma is based on the proof that the non-equivalence problem for regular expressions with operations union and concatenation is NP-complete, even if one of them is of the form Σ^n for some fixed n [17, 35].

▶ **Lemma 23.** *The 0-piecewise testability problem for ptNFAs is coNP-hard (even if the alphabet is binary).*

It seems natural that the $(k + 1)$-piecewise testability problem is not easier then the k-piecewise testability problem. We now formalize this intuition. We also point out that our reduction introduces a new symbol to the alphabet.

▶ **Lemma 24.** *For $k \geq 0$, k-piecewise testability is polynomially reducible to $(k+1)$-piecewise testability.*

Together, since the k-piecewise testability problem for NFAs is in PSPACE [25], we have the following result.

▶ **Theorem 25.** *For $k \geq 0$, the k-piecewise testability problem for ptNFAs is coNP-hard and in PSPACE.*

The case of a fixed alphabet. The previous discussion is for the general case where the alphabet is arbitrary and considered as part of the input. In this subsection, we assume that the alphabet is fixed. In this case, it is shown in the arxiv versions v1–v4 of [18] that the length of the shortest representatives of the $\sim_k$-classes is bounded by the number $\left(\frac{k+2c-1}{c}\right)^c$, where c is the cardinality of the alphabet. This gives us the following result for 0-piecewise testability for ptNFAs.

▶ **Lemma 26.** *For a fixed alphabet Σ with $c = |\Sigma| \geq 2$, the 0-piecewise testability problem for ptNFAs is coNP-complete.*

Proof. The hardness follows from Lemma 23, since it is sufficient to use a binary alphabet.

We now prove the membership. Let $\mathcal{A}$ be a ptNFA over Σ of depth d recognizing a nonempty language (this can be checked in NL). Then the language $L(\mathcal{A})$ is d-piecewise testable by Theorem 8. This means that if $v \sim_d u$, then either both u and v are accepted or both are rejected by $\mathcal{A}$. Now, the language $L(\mathcal{A}) \neq \emptyset$ is not 0-piecewise testable if and only if $L(\mathcal{A})$ is non-universal. Since Σ is fixed, the shortest representative of any of the $\sim_d$-classes is of length less than $\left(\frac{d+2c-1}{c}\right)^c = O(d^c)$, which is polynomial in the depth of $\mathcal{A}$. Thus, if the language $L(\mathcal{A})$ is not universal, then the nondeterministic algorithm can guess a shortest representative of a non-accepted $\sim_d$-class and verify the guess in polynomial time. ◀

We can now generalize this result to k-piecewise testability.

▶ **Theorem 27.** *Let Σ be a fixed alphabet with $c = |\Sigma| \geq 3$, and let $k \geq 0$. Then the problem to decide whether the language of a ptNFA $\mathcal{A}$ over Σ is k-piecewise testable is coNP-complete.*

Table 1 Complexity of k-piecewise testability – an overview.

	Unary alphabet	Fixed alphabet	Arbitrary alphabet $k \leq 3$	Arbitrary alphabet $k \geq 4$
DFA	P	P [19]	NL-complete [25]	coNP-complete [19]
ptNFA	P	coNP-complete	PSPACE & coNP-hard	
NFA	coNP-complete	PSPACE-complete [25]	PSPACE-complete [25]	

Note that this is in contrast with the analogous result for DFAs, cf. Theorem 18, where the problem is in P for DFAs over a fixed alphabet. In addition, the hardness part of the proof of the previous theorem gives us the following corollary, which does not follow from the hardness proof of [19], since the proof there requires a growing alphabet.

▶ **Corollary 28.** *The k-piecewise testability problem for ptNFAs over an alphabet Σ is coNP-hard for $k \geq 0$ even if $|\Sigma| = 3$.*

The case of a unary alphabet. Lemma 26 (resp. Lemma 23) requires at least two letters in the alphabet to prove coNP-hardness. Thus, it remains to consider the case of a unary alphabet. We now show that the problem is simpler in the unary case, unless P=NP. Namely, a similar argument as in the proof of Lemma 26, improved by the fact that the length of the shortest representatives of $\sim_k$-classes is bounded by the depth of the ptNFA, gives the following result.

▶ **Theorem 29.** *The k-piecewise testability problem for ptNFAs over a unary alphabet is decidable in polynomial time. The result holds even if k is given as part of the input.*

In contrast to this, the problem is coNP-cotmplete for general NFAs.

▶ **Theorem 30.** *Both piecewise testability and k-piecewise testability problems for NFAs over a unary alphabet are coNP-complete.*

The complexity of k-piecewise testability for considered automata is summarized in Table 1. Note that the precise complexity of k-piecewise testability for ptNFAs is not yet known in the case the alphabet is considered as part of the input even for $k = 0$.

7 Conclusion

In this paper, we have defined a class of nondeterministic finite automata (ptNFAs) that characterize piecewise testable languages. We have shown that their depth (exponentially) improves the known upper bound on k-piecewise testability shown in [20] for DFAs. We have discussed several related questions, mainly in comparison with DFAs and NFAs, including the complexity of k-piecewise testability for ptNFAs. It can be noticed that the results for ptNFAs generalize the results for DFAs in the sense that the results for DFAs are consequences of the results presented here. This, however, does not hold for the complexity results.

The complexity of k-piecewise testability for the case where the alphabet is consider as part of the input is left open. Recall that the results of [18] give a lower bound on the maximal length of the shortest representative of a class. Specifically, let $L_n(k)$ denote the maximal length of the shortest representatives of the $\sim_k$-classes over an n-element alphabet. Then $L_n(n^2) \geq n^{n-1}$. Thus, the representative can be of exponential length with respect to the size of the alphabet.

However, we conjecture that the problem is PSPACE-complete. A partial evidence for this is that it is possible to construct, for an alphabet of cardinality n, an $O(n^2)$-state ptNFA such that the (unique) non-accepted word is of length $\binom{2n}{n} - 1$. We leave this for the future work and provide more details in an extended version.

Acknowledgements. We thank the authors of [14] and [19] for providing the full versions of their papers, and O. Klíma for his comments on the preliminary version of the manuscript.

References

1 A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

2 J. Almeida, J. C. Costa, and M. Zeitoun. Pointlike sets with respect to R and J. *Journal of Pure and Applied Algebra*, 212(3):486–499, 2008.

3 J. Almeida and M. Zeitoun. The pseudovariety J is hyperdecidable. *RAIRO – Theoretical Informatics and Applications*, 31(5):457–482, 1997.

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

5 J. A. Brzozowski and F. E. Fich. Languages of R-trivial monoids. *Journal of Computer and System Sciences*, 20(1):32–49, 1980.

6 S. Cho and D. T. Huynh. Finite-automaton aperiodicity is PSPACE-complete. *Theoretical Computer Science*, 88(1):99–116, 1991.

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

8 W. Czerwiński, W. Martens, and T. Masopust. Efficient separability of regular languages by subsequences and suffixes. In *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 7966 of *LNCS*, pages 150–161, 2013.

9 W. Czerwiński, W. Martens, L. van Rooijen, and M. Zeitoun. A note on decidable separability by piecewise testable languages. In *Fundamentals of Computation Theory (FCT)*, volume 9210 of *LNCS*, pages 173–185, 2015.

10 V. Diekert, P. Gastin, and M. Kufleitner. A survey on small fragments of first-order logic over finite words. *International Journal of Foundations of Computer Science*, 19(3):513–548, 2008.

11 J. Fu, J. Heinz, and H. G. Tanner. An algebraic characterization of strictly piecewise languages. In *Theory and Applications of Models of Computation (TAMC)*, volume 6648 of *LNCS*, pages 252–263, 2011.

12 P. García and J. Ruiz. Learning k-testable and k-piecewise testable languages from positive data. *Grammars*, 7:125–140, 2004.

13 P. García and E. Vidal. Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):920–925, 1990.

14 P. Hofman and W. Martens. Separability by short subsequences and subwords. In *International Conference on Database Theory (ICDT)*, volume 31 of *LIPIcs*, pages 230–246, 2015.

15 Š. Holub, G. Jirásková, and T. Masopust. On upper and lower bounds on the length of alternating towers. In *Mathematical Foundations of Computer Science (MFCS)*, volume 8634 of *LNCS*, pages 315–326, 2014.

16 Š. Holub, T. Masopust, and M. Thomazo. Alternating towers and piecewise testable separators. *CoRR*, 2014. Submitted. URL: `http://arxiv.org/abs/1409.3943`.

17 H. B. Hunt III. *On the Time and Tape Complexity of Languages.* PhD thesis, Department of Computer Science, Cornell University, Ithaca, NY, 1973.

18 P. Karandikar, M. Kufleitner, and Ph. Schnoebelen. On the index of Simon's congruence for piecewise testability. *Information Processing Letters*, 115(4):515–519, 2015.

19 O. Klíma, M. Kunc, and L. Polák. Deciding k-piecewise testability. Submitted.

20 O. Klíma and L. Polák. Alternative automata characterization of piecewise testable languages. In *Developments in Language Theory (DLT)*, volume 7907 of *LNCS*, pages 289–300, 2013.

21 L. Kontorovich, C. Cortes, and M. Mohri. Kernel methods for learning languages. *Theoretical Computer Science*, 405(3):223–236, 2008.

22 M. Krötzsch, T. Masopust, and M. Thomazo. On the complexity of university for partially ordered NFAs. In *Mathematical Foundations of Computer Science (MFCS)*, volume 58 of *LIPIcs*, pages 62:1–62:14, 2016.

23 M. Kufleitner and A. Lauser. Around dot-depth one. *International Journal of Foundations of Computer Science*, 23(6):1323–1340, 2012.

24 W. Martens, F. Neven, M. Niewerth, and T. Schwentick. Bonxai: Combining the simplicity of DTD with the expressiveness of XML schema. In *Principles of Database Systems (PODS)*, pages 145–156. ACM, 2015.

25 T. Masopust and M. Thomazo. On the complexity of k-piecewise testability and the depth of automata. In *Developments in Language Theory (DLT)*, volume 9168 of *LNCS*, pages 364–376, 2015.

26 J. Myhill. Finite automata and representation of events. Technical report, Wright Air Development Center, 1957.

27 D. Perrin and J.-E. Pin. First-order logic and star-free sets. *Journal of Computer and System Sciences*, 32(3):393–406, 1986.

28 D. Perrin and J.-E. Pin. *Infinite words: Automata, semigroups, logic and games*, volume 141 of *Pure and Applied Mathematics.* Elsevier, 2004.

29 T. Place, L. van Rooijen, and M. Zeitoun. Separating regular languages by piecewise testable and unambiguous languages. In *Mathematical Foundations of Computer Science (MFCS)*, volume 8087 of *LNCS*, pages 729–740, 2013.

30 J. Rogers, J. Heinz, G. Bailey, M. Edlefsen, M. Visscher, D. Wellcome, and S. Wibel. On languages piecewise testable in the strict sense. In *Mathematics of Language (MOL)*, volume 6149 of *LNAI*, pages 255–265, 2010.

31 J. Rogers, J. Heinz, M. Fero, J. Hurst, D. Lambert, and S. Wibel. Cognitive and sub-regular complexity. In *Formal Grammar (FG)*, volume 8036 of *LNCS*, pages 90–108, 2013.

32 T. Schwentick, D. Thérien, and H. Vollmer. Partially-ordered two-way automata: A new characterization of DA. In *Developments in Language Theory (DLT)*, volume 2295 of *LNCS*, pages 239–250, 2001.

33 I. Simon. *Hierarchies of Events with Dot-Depth One.* PhD thesis, Department of Applied Analysis and Computer Science, University of Waterloo, Canada, 1972.

34 J. Stern. Complexity of some problems from the theory of automata. *Information and Control*, 66(3):163–176, 1985.

35 L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time: Preliminary report. In *Symposium on the Theory of Computing (STOC)*, pages 1–9. ACM, 1973.

36 W. Thomas. Classifying regular events in symbolic logic. *Journal of Computer and System Sciences*, 25(3):360–376, 1982.

37 A. N. Trahtman. Piecewise and local threshold testability of DFA. In *Fundamentals of Computation Theory (FCT)*, volume 2138 of *LNCS*, pages 347–358, 2001.

38 L. van Rooijen. *A combinatorial approach to the separation problem for regular languages.* PhD thesis, LaBRI, University of Bordeaux, France, 2014.