HAL
archives-ouvertes.fr

On Boolean Combinations forming Piecewise Testable Languages

Tomáš Masopust, Michaël Thomazo

▶ **To cite this version:**

Tomáš Masopust, Michaël Thomazo. On Boolean Combinations forming Piecewise Testable Languages. Theoretical Computer Science, Elsevier, 2017, 682. hal-01637057

HAL Id: hal-01637057
https://hal.inria.fr/hal-01637057

Submitted on 17 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On Boolean Combinations forming Piecewise Testable Languages

Tomáš Masopust[a,1,*], Michaël Thomazo[b,2]

[a]*Institute of Theoretical Computer Science and Center of Advancing Electronics Dresden (cfaed), TU Dresden, Germany and Institute of Mathematics, Czech Academy of Sciences, Czech Republic*
[b]*Inria, France*

Abstract

A regular language is k-piecewise testable (k-PT) if it is a Boolean combination of languages of the form $L_{a_1a_2\dots a_n} = \Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$, where $a_i \in \Sigma$ and $0 \leq n \leq k$. Given a finite automaton $\mathscr{A}$, if the language $L(\mathscr{A})$ is piecewise testable, we want to express it as a Boolean combination of languages of the above form. The idea is as follows. If the language is k-PT, then there exists a congruence $\sim_k$ of finite index such that $L(\mathscr{A})$ is a finite union of $\sim_k$-classes. Every such class is characterized by an intersection of languages of the from L_u, for $|u| \leq k$, and their complements. To represent the $\sim_k$-classes, we make use of the $\sim_k$-canonical DFA. We identify the states of the $\sim_k$-canonical DFA whose union forms the language $L(\mathscr{A})$ and use them to construct the required Boolean combination. We study the computational and descriptional complexity of related problems.

1. Introduction

A regular language L over an alphabet Σ is *piecewise testable* (PT) if it is a finite Boolean combination of languages of the form $L_{a_1a_2\dots a_n} = \Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$, where $a_i \in \Sigma$ and $n \geq 0$. If the language is piecewise testable, then it is a finite Boolean combination of languages of the form L_u, where the length of $u \in \Sigma^*$ is at most k. In this case, the language is called *k-piecewise testable* (k-PT).

In this paper, we study the problem of translating an automaton representing a piecewise testable language into a Boolean combination of languages of the form L_u. The motivation comes from the simplification of XML Schema, since such expressions resemble XPath-like expressions used in the BonXai schema language. The reader is referred to Martens et al. [21] for more details. Since every piecewise testable language is k-PT for some $k \geq 0$, and a k-PT language is also $(k+1)$-PT, we focus on the Boolean combination of languages L_u, where the length of u is bounded by the minimal k for which the language is k-PT. From this point of view, we are interested in translating an automaton to the form of a generalized regular expression (a regular expression allowing the operation of complement). Generalized regular expressions can be non-elementary more succinct than classical regular expressions [6, 29, 9] and not much is known about these transformations [7]. There are many different Boolean combinations describing the same language, and it is not clear which of them is the best representation. The choice significantly depends on applications. We are interested in those Boolean combinations that resemble the disjunctive normal form of logical formulas rather than in the most concise representation.

The basic idea to perform this translation can be outlined as follows. Let L be a language over Σ (represented by its minimal DFA) and let the equivalence relation $\sim_k$ on Σ^* be defined by $u \sim_k v$ if u and v have the same sets of (scattered) subwords up to length k, denoted by $sub_k(u) = sub_k(v)$. Then L is piecewise testable if and only if there exists a nonnegative integer k such that $\sim_k \subseteq \sim_L$, where $\sim_L$ is the Myhill congruence [24], that is, every k-PT language is a finite union of $\sim_k$-classes. As shown, e.g., by Klíma [17], the $\sim_k$-classes can be described by languages of the form $[w]_{\sim_k} = \bigcap_{u \in sub_k(w)} L_u \cap \bigcap_{u \notin sub_k(w), |u| \leq k} \overline{L_u}$, where $\overline{L_u}$ denotes the complement of L_u. The high-level approach is thus:

*Corresponding author.
Email addresses: tomas.masopust@tu-dresden.de (Tomáš Masopust), michael.thomazo@inria.fr (Michaël Thomazo)
[1]Research supported by the German Research Foundation (DFG) in Emmy Noether grant KR 4381/1-1 (DIAMOND).
[2]Research supported by the Alexander von Humboldt Foundation

1. Check whether the regular language L is piecewise testable.
2. If so, compute the minimal $k \geq 0$ for which L is k-piecewise testable.
3. Compute the finite number of representatives of the equivalence classes that form the union of the language L, express them as above and form their union.

We study the computational and descriptional complexity of this approach, provide an overview of related results, and formulate several open problems.

The complexity of the first step has been studied in the literature. Simon [26] proved that PT languages are exactly those regular languages whose syntactic monoid is $\mathscr{J}$-trivial, which gives decidability. Stern [28] showed that the problem is decidable in polynomial time for languages represented by DFAs and Cho and Huynh [5] proved NL-completeness for DFAs. Later, Trahtman [31] showed that the problem is solvable in time quadratic with respect to the number of states of the DFA and linear with respect to the size of the alphabet, and Klíma and Polák [19] gave an algorithm that is quadratic in the size of the input alphabet and linear in the number of states of the DFA. For languages represented by NFAs, the problem is PSPACE-complete [11].

The second step gives rise to the *k-piecewise testability* problem formulated as follows:

INPUT: an automaton (DFA or NFA) $\mathscr{A}$
OUTPUT: YES if and only if $L(\mathscr{A})$ is k-piecewise testable

The problem is trivially decidable for any k because there are only finitely many k-PT languages over the alphabet of $\mathscr{A}$. We investigate and overview the computational complexity of this problem. The upper bound complexity for DFAs has been independently solved in [10, 18, 23]. The co-NP upper bound on the k-piecewise testability problem for DFAs first appeared in [10] without proof, formulated in terms of separability.[3] In this paper, we recall (without proof) the result of [18] showing that the problem is co-NP-complete for DFAs if $k \geq 4$. We then focus on the complexity of the problem for $k < 4$. In particular, for the input given as the minimal DFA, the problem is trivial for $k = 0$, belongs to AC^0 for $k = 1$ (Theorem 6), and is NL-complete for $k = 2, 3$ (Theorems 13 and 18). For NFAs, we show that the problem is PSPACE-complete for any $k \geq 0$ (Theorem 20).

There is an interesting observation by Klíma and Polák [19] that if the depth of a minimal DFA recognizing a PT language is k, then the language is k-PT. (Bounds for finite languages and upward and downward closures have recently been investigated by Karandikar and Schnoebelen [16].) The observation reduces Step 2 to solving a finite number of k-piecewise testability problems, since the upper bound on k is given by the depth of the minimal DFA equivalent to $\mathscr{A}$. The opposite implication does not hold, therefore we investigate the relationship between the depth of an NFA and k-piecewise testability of its language. We show that, for every $k \geq 0$, there exists a k-PT language with an NFA of depth $k-1$ and with the minimal DFA of depth $2^k - 1$ (Theorem 27). Although it is well known that DFAs can be exponentially larger than NFAs, a by-product of our result is that all the exponential number of states of the DFA form a simple path, which is, in our opinion, a result of interest by its own. In addition, the reverse of the NFAs constructed in the proof is deterministic, partially ordered and locally confluent. Therefore, our result also provides a further insight into the complexity of the reverse of piecewise testable languages previously studied in [4, 14].

The last step of the approach requires to compute those $\sim_k$-classes, whose union forms the language L, and to express them as the intersection of languages of the form L_u or its complements. To identify these equivalence classes, we make use of the $\sim_k$-canonical DFA, whose states correspond to $\sim_k$-classes. We construct the $\sim_k$-canonical DFA and compute its accepting states by intersection with the input automaton. The accepting states then represent the $\sim_k$-classes forming the language L. The $\sim_k$-canonical DFA can be effectively constructed. Moreover, although the precise size of the $\sim_k$-canonical DFA is not known, see the estimations in [15], we show that the tight upper bound on its depth is $\binom{k+n}{k} - 1$, where n is the cardinality of the alphabet (Theorem 31).

This paper is an extended version of paper [23] presented at the DLT 2015 conference, containing full proofs and updated with the latest results and open problems. After introducing the necessary notions (Section 2), we introduce the approach on an example (Section 3), before studying the complexity of the k-piecewise testability problem for DFAs (Section 4) and NFAs (Section 5). We finish by investigating the depth of minimal DFAs (Section 6).

[3] The result is a consequence of a proof that is omitted in the conference version.

2. Preliminaries and Definitions

We assume that the reader is familiar with automata theory [20]. 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 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 $\mathscr{A} = (Q, \Sigma, \cdot, I, F)$, 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 $\mathscr{A}$ is the set $L(\mathscr{A}) = \{w \in \Sigma^* \mid I \cdot w \cap F \neq \emptyset\}$. We sometimes 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 $i = 0, 1, \ldots, n-1$. Path π is *accepting* if $q_0 \in I$ and $q_n \in F$. We write $q_0 \xrightarrow{a_1 a_2 \cdots a_n} q_n$ 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 different. The number of states on the longest simple path of $\mathscr{A}$, starting in the initial state, decreased by one (the number of transitions on the path) is called the *depth* of automaton $\mathscr{A}$, denoted by $\mathit{depth}(\mathscr{A})$.

The NFA $\mathscr{A}$ is *deterministic* (DFA) if $|I| = 1$ and $|q \cdot a| = 1$ for every $q \in Q$ and $a \in \Sigma$. The transition function $\cdot$ is then 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.

The reachability relation $\leq$ on the set of states is defined by $p \leq q$ if there is a word $w \in \Sigma^*$ such that $q \in p \cdot w$. The NFA $\mathscr{A}$ is *partially ordered* if the reachability relation $\leq$ is a partial order. For two states p and q of $\mathscr{A}$, 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 $p < q$. Partially ordered automata are sometimes called *acyclic*. In this terminology, a cycle is a nontrivial loop, since self-loops are allowed in partially ordered automata.

Let $\mathscr{A} = (Q, \Sigma, \cdot, i, F)$ be a DFA, and let $\Gamma \subseteq \Sigma$. The DFA $\mathscr{A}$ is Γ-confluent if, for every state $q \in Q$ and every pair of words $u, v \in \Gamma^*$, there exists a word $w \in \Gamma^*$ such that $(qu)w = (qv)w$. The DFA $\mathscr{A}$ is *confluent* if it is Γ-confluent for every subalphabet Γ of Σ. The DFA $\mathscr{A}$ is *locally confluent* if, for every state $q \in Q$ and every pair of letters $a, b \in \Sigma$, there exists a word $w \in \{a, b\}^*$ such that $(qa)w = (qb)w$.

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

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 [19]. Furthermore, for two words $v = a_1 a_2 \cdots a_n$ and $w \in L_v$, we say that v is a *subword* of w or that v can be *embedded* into w, denoted by $v \preccurlyeq w$. For $k \geq 0$, let $\mathit{sub}_k(v) = \{u \in \Sigma^* \mid u \preccurlyeq v, |u| \leq k\}$. For two words w_1, w_2, we define $w_1 \sim_k w_2$ if and only if $\mathit{sub}_k(w_1) = \mathit{sub}_k(w_2)$. If $w_1 \sim_k w_2$, we say that w_1 and w_2 are *k-equivalent*. Note that $\sim_k$ is a congruence with finite index.

The *$\sim_k$-canonical DFA* is the DFA $\mathscr{A} = (Q, \Sigma, \cdot, [\varepsilon], F)$, where $Q = \left\{[w] \mid w \in \Sigma^{\leq k}\right\}$, $[w] = \{w' \mid w' \sim_k w\}$, and the transition function $\cdot$ is defined so that, for a state $[w]$ and a letter a, $[w] \cdot a = [wa]$.

Let $\sim_L$ denote the Myhill congruence [24].

Fact 1 (Simon [26])**.** *A regular language L is k-PT if and only if $\sim_k \subseteq \sim_L$. Moreover, L is then a finite union of $\sim_k$ classes.*

Fact 1 says that if L is k-PT, then any two k-equivalent words either both belong to L or neither does. In terms of a minimal DFA, two k-equivalent words end up in the same state.

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

1. *The language L is PT.*
2. *The minimal DFA $\mathscr{A}$ is partially ordered and (locally) confluent [19].*
3. *The minimal DFA $\mathscr{A}$ is partially ordered and satisfies the UMS property [31].*

Figure 1: NFA $\mathscr{A}$ recognizing the language L (left) and its reverse with added sink state 0 (right)

3. Example

Before we investigate the individual steps of the approach, we provide a simple example demonstrating it. Let L be the language recognized by the NFA depicted in Figure 1 (left). Since the automaton is nondeterministic, neither [31] nor [19] applies to decide whether L is piecewise testable. In Theorem 25, we generalize Fact 2 to NFAs, which then gives that L is piecewise testable. Another way how to see this is to notice that the reverse of the NFA for L, depicted in Figure 1 (right), is deterministic. Since, by definition, L is k-PT if and only if $L^R = \{a_n \dots a_2 a_1 \mid a_1 a_2 \dots a_n \in L\}$ is k-PT, the results of [19, 31] can be used to decide whether L is piecewise testable. The upper bound on k given by the depth of the DFA [19] gives that L is 2-PT. It could be that the language is 1-PT. However, the characterization of 1-PT languages in Lemma 5 shows that it is not the case. Thus, the language L is piecewise testable and the minimal k for which it is k-PT is $k = 2$.

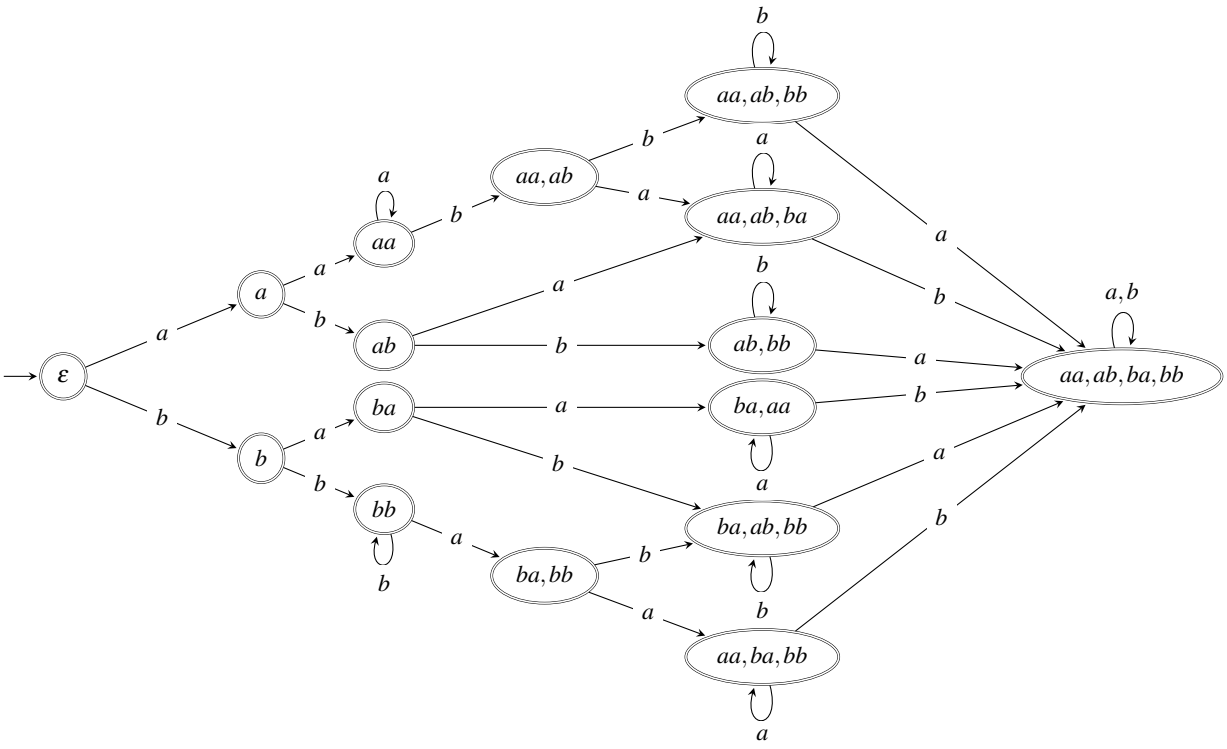

Figure 2: The $\sim_2$-canonical DFA $\mathscr{C}$ over the binary alphabet $\{a,b\}$

Having this information, we can construct the $\sim_2$-canonical DFA $\mathscr{C}$ over the binary alphabet $\{a,b\}$ as depicted in Figure 2. The states of $\mathscr{C}$ correspond to the $\sim_2$-classes and are labeled by their maximal elements with respect to the relation of embedding $\preccurlyeq$. The initial state of $\mathscr{C}$ is the class $[\varepsilon]$ and all its states are accepting. The label of state $[w]$ is the set of maximal elements of the set $sub_2(w)$. For instance, because $sub_2(aab) = \{\varepsilon, a, b, aa, ab\}$, the label of state $[aab]$ is $\{aa, ab\}$. The choice to have all states accepting is made because we now compute the product of the automata $\mathscr{C}$ and $\mathscr{A}$. The result, depicted in Figure 3, says that language L is a union of the following four $\sim_2$

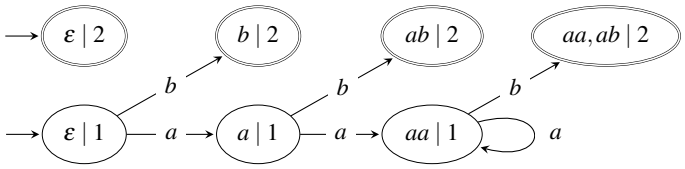

Figure 3: Product of $\mathscr{C}$ and $\mathscr{A}$ restricted to reachable and co-reachable states

classes: $[\varepsilon]$, $[b]$, $[ab]$, $[aab]$. By [17], these classes are characterized by the intersections of the required languages or their complements. Namely, $L = [\varepsilon] \cup [b] \cup [ab] \cup [aab]$, where $[\varepsilon] = \overline{L_a} \cap \overline{L_b} \cap \overline{L_{aa}} \cap \overline{L_{ab}} \cap \overline{L_{ba}} \cap \overline{L_{bb}} = \overline{L_a} \cap \overline{L_b}$, $[b] = L_b \cap \overline{L_a} \cap \overline{L_{aa}} \cap \overline{L_{ab}} \cap \overline{L_{ba}} \cap \overline{L_{bb}} = L_b \cap \overline{L_a} \cap \overline{L_{bb}}$, $[ab] = L_a \cap L_b \cap L_{ab} \cap \overline{L_{aa}} \cap \overline{L_{ba}} \cap \overline{L_{bb}} = L_{ab} \cap \overline{L_{aa}} \cap \overline{L_{ba}} \cap \overline{L_{bb}}$, and $[aab] = L_a \cap L_b \cap L_{aa} \cap L_{ab} \cap \overline{L_{ba}} \cap \overline{L_{bb}} = L_{aa} \cap L_{ab} \cap \overline{L_{ba}} \cap \overline{L_{bb}}$. We used a simple observation formulated below as Lemma 3 to reduce the number of elements in the intersection.

Lemma 3. *If $u \preccurlyeq v$, then $L_v \subseteq L_u$ and $\overline{L_u} \subseteq \overline{L_v}$.* □

The reader can notice that $[ab] \cup [aab] = L_{ab} \cap \overline{L_{ba}} \cap \overline{L_{bb}}$. Thus,

$$L = (\overline{L_a} \cap \overline{L_b}) \cup (L_b \cap \overline{L_a} \cap \overline{L_{bb}}) \cup (L_{ab} \cap \overline{L_{ba}} \cap \overline{L_{bb}}).$$

To justify our choice of a 2-PT language, we point out that if we considered a 3-PT language, then the size of the $\sim_3$-canonical DFA over a binary alphabet would contain 68 states [15] and it would not be possible to present it here in a reasonable form. It is an open question whether it is possible to avoid the use of the $\sim_k$-canonical DFA. One natural way how to reduce the complexity is to rather build the canonical DFA on-the-fly during the computation of its product with the input automaton $\mathscr{A}$ rather than precomputing it and storing it in memory. In this way the algorithm would construct only a relevant part of the canonical DFA, compare the Figures 2 and 3.

4. Complexity of k-Piecewise Testability for DFAs

The *k-piecewise testability problem for DFAs* asks whether, given a minimal DFA $\mathscr{A}$, the language $L(\mathscr{A})$ is k-PT. It has been independently proved to be in co-NP in [10, 18, 23]. We recall the result of [18] that also proves co-NP-completeness if $k \geq 4$.

Theorem 4 (Klíma, Kunc, Polák [18])**.** *For $k \geq 4$, the k-piecewise testability problem for DFAs is co-NP-complete.*

It is shown in [18] that the problem remains co-NP-complete even if the parameter $k \geq 4$ is given as part of the input, and that it is decidable in polynomial time if the alphabet is fixed.

We now study the complexity of the problem for $k \leq 3$.

0-Piecewise Testability. Let $\mathscr{A}$ be a minimal DFA over an alphabet Σ. The language $L(\mathscr{A})$ is 0-PT if and only if it has a single state, that is, it recognizes either Σ^* or $\emptyset$. Thus, it is decidable in $O(1)$ whether $L(\mathscr{A})$ is 0-PT.

1-Piecewise Testability. We show that the 1-PT problem belongs to AC^0, which is a strict subset of LOGSPACE. There is an infinite hierarchy of classes Σ_i (Π_i) in AC^0 based on the number of alternating levels of disjunctions and conjunctions. Specifically, Σ_i (Π_i) is the class of problems solvable by uniform families of unlimited fan-in circuits of constant depth and polynomial size with i alternating levels of AND and OR gates (with NOT gates only in the input) and with the output gate being an OR gate (an AND gate) [1]. To prove the result, we make use of the following characterization lemma.

Lemma 5. *Let $\mathscr{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal DFA. Then $L(\mathscr{A})$ is 1-PT if and only if (i) for every $p \in Q$ and $a \in \Sigma$, $pa = q$ implies that $qa = q$, and (ii) for every $p \in Q$ and $a, b \in \Sigma$, $pab = pba$.*

Proof. Assume that $L(\mathscr{A})$ is 1-PT, and let p be a state of $\mathscr{A}$. Since $\mathscr{A}$ is minimal, p is reachable and there exists w such that $iw = p$. It holds that $\text{alph}(wa) = \text{alph}(waa)$, i.e., $wa \sim_1 waa$, thus both wa and waa lead $\mathscr{A}$ to the same state, that is, $paa = pa$. Similarly, $\text{alph}(wab) = \text{alph}(wba)$ implies that $pab = pba$.

On the other hand, we show that for any word w, $iw = ia_1a_2\dots a_n$, where $\text{alph}(w) = \{a_1, a_2, \dots, a_n\}$. For any $a, b \in \Sigma$ and any state q, $qab = qba$ implies that $iw = ia_1^{k_1}a_2^{k_2}\dots a_n^{k_n}$, where k_i is the number of occurrences of a_i in w. By (i), $iw = ia_1a_2\dots a_n$. Thus, if $w_1 \sim_1 w_2$, then $iw_1 = iw_2$. By Fact 1, this shows that $L(\mathscr{A})$ is 1-PT. □

We can now prove the following theorem.

Theorem 6. *To decide whether a minimal DFA recognizes a 1-PT language is in AC^0.*

Proof. To prove the theorem, consider Lemma 5 and notice that the properties can be expressed as a Π_3 formula

$$\bigwedge_{(p,a,q)} [\neg(p,a,q) \vee (q,a,q)] \wedge \bigwedge_{(p,a,r),(r,b,q)} \left[\neg(p,a,r) \vee \neg(r,b,q) \vee \bigvee_{s} ((p,b,s) \wedge (s,a,q))\right],$$

where $(p,a,q) \in Q \times \Sigma \times Q$ is true if and only if there is a transition from state p to state q under a in the minimal DFA. The corresponding family of circuits is of polynomial size with respect to the size of the automaton, namely of size $O(|Q|^2 \cdot |\Sigma|)$, and of constant depth, which proves the theorem. □

Open Problem 7. Is the 1-PT problem Π_3-hard in the AC^0 hierarchy?

As a consequence of Lemma 5, we have that a minimal DFA of a 1-PT language has at most $2^{|\Sigma|}$ states, and this bound is tight as shown in Example 28 below.

Corollary 8. *If a minimal DFA over Σ has more than $2^{|\Sigma|}$ states, then its language is not 1-PT.*

2-Piecewise Testability. We now show that to decide whether a minimal DFA recognizes a 2-PT language is NL-complete. This complexity coincides with the complexity of deciding whether a regular language is PT, that is, whether there exists a k for which the language is k-PT.

We first need the following lemma stating that for any two k-equivalent words ending up in two different states, there exist other two equivalent words ending up in two different states, such that one word is a subword of the other and the words differ only by a single letter. Our proof follows the lines of Simon's original paper [27, Section 2].

Lemma 9. *Let $\mathscr{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal DFA. For every $k \geq 0$, if $w_1 \sim_k w_2$ and $iw_1 \neq iw_2$, then there exist two words w and w' such that $w \sim_k w'$, w' is obtained from w by adding a single letter at some place, and $iw \neq iw'$.*

Proof. Let w_1, w_2 be two words such that $w_1 \sim_k w_2$ and $iw_1 \neq iw_2$. By Theorem 6.2.6 in [25], there is a word w_3 such that w_1 and w_2 are subwords of w_3 and $w_1 \sim_k w_2 \sim_k w_3$. Either w_1 and w_3, or w_2 and w_3, do not end up in the same state of the automaton. Let $v, v' \in \{w_1, w_2, w_3\}$ be such that v is a subword of v' and $iv \neq iv'$. Let $v = u_0, u_1, \dots, u_n = v'$ be a sequence such that u_{j+1} is obtained from u_j by adding a letter at some place. Such a sequence exists since v is a subword of v'. Thus, there is j such that u_j and u_{j+1} end up in two different states and u_{j+1} is obtained from u_j by adding a letter at some place. Setting $w = u_j$ and $w' = u_{j+1}$ completes the proof, since $\mathit{sub}_k(v) \subseteq \mathit{sub}_k(w) \subseteq \mathit{sub}_k(w') \subseteq \mathit{sub}_k(v') = \mathit{sub}_k(v)$. □

We now prove a characterization of 2-PT languages similar to that of Lemma 5.

Lemma 10. *Let $\mathscr{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal partially ordered and confluent DFA. The language $L(\mathscr{A})$ is 2-PT if and only if for every $a \in \Sigma$ and every state p such that $iw = p$ for some word w with $|w|_a \geq 1$, $pua = paua$, for every $u \in \Sigma^*$.*

Proof. ($\Rightarrow$) By contraposition – assume that there exist $u, w \in \Sigma^*$ and a state p such that $iw = p$, w contains a, and $pua \neq paua$. Let $w = w_1aw_2$ with $a \notin \text{alph}(w_1)$. We show that $w_1aw_2ua \sim_2 w_1aw_2aua$, by showing that they have the same set of subwords of length at most 2. The subwords of w_1aw_2ua are subwords of w_1aw_2aua. Conversely, the subwords of w_1aw_2aua that are potentially not subwords of w_1aw_2ua are of two shapes: ca where $c \in \text{alph}(w_1aw_2)$ or

ad where $d \in \mathrm{alph}(ua)$. For any $c \in \mathrm{alph}(w_1aw_2)$, if $ca \preccurlyeq w_1aw_2aua$, then $ca \preccurlyeq w_1aw_2ua$. Similarly for $d \in \mathrm{alph}(ua)$ and $ad \preccurlyeq w_1aw_2aua$. Thus $w_1aw_2ua \sim_2 w_1aw_2aua$. Since $i \cdot wua \neq i \cdot waua$, the minimality of $\mathscr{A}$ gives that there exists a word v such that $wuav \in L(\mathscr{A})$ if and only if $wauav \notin L(\mathscr{A})$. Since $\sim_2$ is a congruence, $wuav \sim_2 wauav$, which violates Fact 1; hence, $L(\mathscr{A})$ is not 2-PT.

($\Leftarrow$) Let w_1 and w_2 be two words such that $w_1 \sim_2 w_2$. We show that $iw_1 = iw_2$. By Lemma 9, it is sufficient to show this direction for two words w and w' such that w' is obtained from w by adding a single letter at some place. Thus, let a be the letter, and let

$$w = a_1 \dots a_k a_{k+1} \dots a_n \text{ and } w' = a_1 \dots a_k a a_{k+1} \dots a_n$$

for $0 \leq k \leq n$. Let $w_{m,j} = a_m a_{m+1} \dots a_j$. We distinguish two cases.

(A) Assume that a does not appear in $w_{1,k}$. Then a must appear in $w_{k+1,n}$. Consider the first occurrence of a in $w_{k+1,n}$. Then $w_{k+1,n} = u_1 a u_2$, where a does not appear in u_1. Let $B = \mathrm{alph}(u_1 a)$. Then $B \subseteq \mathrm{alph}(u_2)$, because if there is no a in $w_{1,k}u_1$, any subword ax, for $x \in B$, that appears in $w' = w_{1,k}au_1au_2$ must also appear in the subword au_2 of $w = w_{1,k}u_1au_2$.

Let $u_2 = x_1 b_1 x_2 b_2 x_3 \dots x_\ell b_\ell x_{\ell+1}$, where $B = \{b_1, b_2, \dots, b_\ell\}$ and b_j does not appear in $x_1 b_1 x_2 \dots x_j$, $j = 1, 2, \dots, \ell$. Let $v = b_1 b_2 \dots b_\ell$. Let $z \in \{i \cdot w_{1,k}u_1a,\ i \cdot w_{1,k}au_1a\}$. We prove (by induction on j) that for every $j = 1, 2, \dots, \ell$, there exists a word y_j such that $z \cdot (b_1 b_2 \dots b_j)^R y_j = z \cdot x_1 b_1 x_2 b_2 x_3 \dots x_j b_j x_{j+1}$. Since b_1 appears in $u_1 a$, we use the assumption from the statement of the lemma to obtain $(z \cdot x_1 b_1) \cdot x_2 = (z \cdot b_1 x_1 b_1) \cdot x_2$, that is, $y_1 = x_1 b_1 x_2$. Assume that it holds for $j < k$. We prove it for $j+1$. Again, b_{j+1} appears in $u_1 a$ implies that

$$\begin{aligned} z \cdot x_1 b_1 x_2 b_2 x_3 \dots x_j b_j x_{j+1} b_{j+1} x_{j+2} &= ((z \cdot x_1 b_1 x_2 b_2 x_3 \dots x_j b_j x_{j+1}) b_{j+1}) x_{j+2} \\ &= ((z \cdot b_j \dots b_2 b_1 y_j) b_{j+1}) x_{j+2} \\ &= z \cdot b_{j+1} \underline{b_j \dots b_2 b_1 y_j b_{j+1}} x_{j+2} \end{aligned}$$

where the second equality is by the induction hypothesis and the third is by the assumption from the statement of the lemma applied to the underlined part. Thus, $y_{j+1} = y_j b_{j+1} x_{j+2}$, which completes the inductive proof. In particular, there exists a word y such that $i \cdot w_{1,k}u_1av^R y = i \cdot w$ and $i \cdot w_{1,k}au_1av^R y = i \cdot w'$.

Let $z_1 = i \cdot w_{1,k}u_1a$ and $z_2 = i \cdot w_{1,k}au_1a$. We prove that $z_1 \cdot v^R = z_2 \cdot v^R$, which then concludes the proof since it implies that $i \cdot w = i \cdot w'$. To prove this, we make use of the following claim.

Claim 11. *For every $a, b \in \Sigma$ and every state p such that $i \cdot w = p$ and a and b appear in w, $p \cdot ab = p \cdot ba$.*

Proof. By the assumption of the lemma, since a appears in w, $p \cdot ba = p \cdot aba = q_1$. Similarly, since b appears in w, $p \cdot ab = p \cdot bab = q_2$. Then $q_2 \cdot a = (p \cdot ab)a = q_1$ and $q_1 \cdot b = (p \cdot ba)b = q_2$. Since the automaton is partially ordered, $q_1 = q_2$. □

We finish the proof by induction on the length of $v^R = b_\ell \dots b_2 b_1$ by showing that the state $z_i' = z_i \cdot b_\ell \dots b_2 b_1$ has self-loops under B, $i = 1, 2$. Let $z_i \xrightarrow{b_\ell \dots b_2 b_1} z_i' = q_{i,\ell+1} b_\ell q_{i,\ell} b_{\ell-1} q_{i,\ell-1} \dots q_{i,2} b_1 q_{i,1}$ denote the path defined by the word v^R from the state z_i, $i = 1, 2$.

Claim. *Both states z_1' and z_2' have self-loops under all letters of B.*

Proof. Indeed, $q_{i,j} \cdot b_j = q_{i,j+1} \cdot b_j b_j = q_{i,j+1} \cdot b_j = q_{i,j}$, where the second equality is by the assumption from the statement of the lemma, since b_j appears in u_1. Thus, there is a self-loop in $q_{i,j}$ under b_j. Then, $z_i' = q_{i,1} = q_{i,1} b_1 = z_i' b_1$. Now, for every $j = 2, \dots, \ell$, we have $z_i' = q_{i,1} = q_{i,j} \cdot b_{j-1} \dots b_2 b_1 = q_{i,j} \cdot b_j b_{j-1} \dots b_2 b_1 = q_{i,j} \cdot b_{j-1} \dots b_2 b_1 b_j = z_i' b_j$, where the third equality is because there is a self-loop in $q_{i,j}$ under b_j, and the fourth is by several applications of commutativity (Claim 11). □

Thus, since no other states are reachable from z_1' and z_2' under B, and z_1' and z_2' are reachable from $i \cdot w_{1,k}$ by words over B, confluency of the automaton implies that $z_1' = z_2'$, which completes the proof of part (A).

(B) If $a = a_i$ for some $i \leq k$, we consider two cases. First, assume that for every $c \in \Sigma \cup \{\varepsilon\}$, ca is a subword of $w_{1,k}a$ implies that ca is a subword of $w_{1,k}$. Then aa is a subword of $w_{1,k}$. Let $w_{1,k} = w_3 a w_4$, where a does not appear in w_4. Let $q = i \cdot w_3 a$. By the assumption of the lemma, $q = i \cdot w_3 a = i \cdot w_3 aa$, hence there is a self-loop in q under a.

Let $B = \mathrm{alph}(w_4)$. Note that $B \subseteq \mathrm{alph}(w_3)$, since if xa is a subword of $w_{1,k}a$, then it is also in w_3a. By the self-loop under a in q and commutativity (Claim 11), $q \cdot w_4 = q \cdot aw_4 = q \cdot w_4a$. Thus, $i \cdot w_{1,k} = i \cdot w_{1,k}a$.

Second, assume that there exists c in $w_{1,k}$ such that $ca \preccurlyeq w_{1,k}a$ is not a subword of $w_{1,k}$. Then a must appear in $w_{k+1,n}$. Together, there exist $i \leq k < j$ such that $a_i = a_j = a$. By the assumption of the lemma, $i \cdot w_{1,k}aw_{k+1,j} = i \cdot w_{1,k}w_{k+1,j}$, since $w_{k+1,j} = xa$, for some $x \in \Sigma^*$. This implies that $i \cdot w = i \cdot w'$.

This completes the proof of part (B) and, hence, the whole proof. □

The previous result gives a PTIME algorithm to decide whether a minimal DFA recognizes a 2-PT language. To show that the problem is in NL, we need the following lemma providing a characterization of 2-PT languages that can be verified locally in nondeterministic logarithmic space.

Lemma 12. *Let $\mathscr{A} = (Q, \Sigma, \cdot, i, F)$ be a DFA. The following conditions are equivalent:*

1. *For every $a \in \Sigma$ and every state s such that $iw = s$ for some $w \in \Sigma^*$ with $|w|_a \geq 1$, $sua = saua$, for every $u \in \Sigma^*$.*
2. *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 $b \in \Sigma \cup \{\varepsilon\}$.*

Proof. Condition 2 is a special case of Condition 1 for $u = b$. We prove the opposite direction by induction on the length of u. Let $a \in \mathrm{alph}(w)$ such that $iw = s$. If $u = \varepsilon$, we take $b = \varepsilon$; otherwise, $u = u'b$. By the induction hypothesis, we have $su'a = sau'a$. Thus $sua = su'ba = (su')ba = (su')aba = (su'a)ba = (sau'a)ba = (sau')ba = saua$. □

We can now formulate the main result of this paragraph.

Theorem 13. *To decide whether a minimal DFA recognizes a 2-PT language is NL-complete.*

Proof. To check whether a minimal DFA is *not* confluent or does *not* satisfy Condition 2 of Lemma 12 can be done in NL; the reader is referred to [5] for more details. Since NL = co-NL [13, 30], we have an NL algorithm to check 2-piecewise testability of a minimal DFA. NL-hardness follows from Lemma 14 below. □

Lemma 14. *For every $k \geq 2$, the k-PT problem is NL-hard.*

Proof. To prove NL-hardness, we reduce the *monotone graph accessibility problem (2MGAP)*, a special case of the graph reachability problem, known to be NL-complete [5]. An instance of 2MGAP is a graph (G, s, g), where $G = (V, E)$ is a graph with the set of vertices $V = \{1, 2, \ldots, n\}$, the source vertex $s = 1$ and the target vertex $g = n$, the out-degree of each vertex is bounded by 2 and for all edges (u, v), v is greater than u (the vertices are linearly ordered).

We construct the automaton $\mathscr{A} = (V \cup \{i, f_1, f_2, \ldots, f_{k-1}, d\}, \Sigma, \cdot, i, \{f_{k-1}\})$ as follows. For every edge (u, v), we construct a transition $u \cdot a_{uv} = v$ over a fresh letter a_{uv}. Moreover, we add the transitions $i \cdot a = s$, $g \cdot a = f_1$ and $f_j \cdot a = f_{j+1}$, $j = 1, 2, \ldots, k-2$, over a fresh letter a. The automaton is deterministic, but not necessarily minimal, since some of the states may not be reachable from the initial state, or some states may be equivalent. To ensure minimality of the constructed automaton, we add, for each state $v \in V \setminus \{s\}$, new transitions from i to v under fresh letters, and for each state $v \in V \setminus \{g\}$, new transitions from v to f_{k-1} under fresh letters. All undefined transitions go to the sink state d.

Claim. *The automaton $\mathscr{A}$ is deterministic and minimal, and $L(\mathscr{A})$ is finite.*

Proof. By construction, all states are reachable from the initial state i and can reach (except the sink state) the unique accepting state f_{k-1}. In addition, the automaton is deterministic and minimal, since every transition is labeled by a unique label (except for the transitions $ia = s$ and $ga^{k-1} = f_{k-1}$ labeled with the same letter), which makes the states non-equivalent. Finally, $L(\mathscr{A})$ is finite because the monotonicity of the graph (G, s, g) implies that the automaton does not contain a cycle nor a self-loop (but the sink state d). □

The following claim is needed to complete the proof.

Claim 15. *Let w be a word over Σ. If every a from Σ appears at most once in w, that is, $|w|_a \leq 1$, then the language $\{w\}$ is 2-PT.*

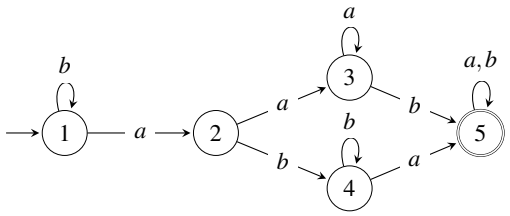

Figure 4: The minimal DFA recognizing L

Proof. Since the language $\{w\}$ is PT, its minimal DFA is partially ordered and confluent. Then the condition of Lemma 10 is trivially satisfied, since, after the second occurrence of the same letter, the minimal DFA accepting $\{w\}$ is in the unique maximal non-accepting state. □

We now finish the proof of Lemma 14 by showing that $L(\mathscr{A})$ is k-PT if and only if g is not reachable from s.

Assume that g is reachable from s. Let w be a sequence of labels of a path from s to g in $\mathscr{A}$. Then awa^{k-1} belongs to $L(\mathscr{A})$ and awa^k does not. However, $awa^{k-1} \sim_k awa^k$, which proves that $L(\mathscr{A})$ is not k-PT.

If g is not reachable from s, then $L(\mathscr{A}) = \{au_1, au_2, \ldots, au_\ell, u_{\ell+1}, \ldots, u_{\ell+t}\} \cup \{w_1a^{k-1}, w_2a^{k-1}, \ldots, w_ma^{k-1}\}$, where u_i and w_i are words over $\Sigma \setminus \{a\}$ that do not contain any letter twice. Then the first part is 2-PT by Claim 15, as well as the second part for $k = 2$. It remains to show that for any $k \geq 3$, the second part of $L(\mathscr{A})$ is k-PT. Assume that $w_ja^{k-1} \sim_k w$, for some $1 \leq j \leq m$ and $w \in \Sigma^*$. Then $w = v_1av_2a \ldots av_k$ for some $v_1, v_2, \ldots, v_k$ such that $|v_1 \ldots v_k|_a = 0$. Since $|w_j|_a = 0$ and, for any letter c of $v_2 \cdots v_{k-1}$ (resp. v_k), the word aca (resp. $a^{k-1}c$) can be embedded into w_ja^{k-1}, that is, into a^{k-1}, we have that $v_2 \cdots v_k = \varepsilon$, i.e., $w = v_1a^{k-1}$. Since $w_ja^{k-1} \sim_k v_1a^{k-1}$, we have that $w_ja \sim_k v_1a$, hence $w_ja = v_1a$, and w_ja^{k-1} and w end up in the same state, which concludes the proof. □

Remark 16 (on 1-PT and 2-PT). It was shown by Blanchet-Sadri [3] that 1-PT languages are characterized as the languages whose syntactic monoids satisfy the equations $x = x^2$ and $xy = yx$, and 2-PT languages are characterized as those whose syntactic monoids satisfy the equations $xyzx = xyxzx$ and $(xy)^2 = (yx)^2$. It can be seen that these equations could be directly used to achieve NL algorithms. Our characterizations, however, improve these results and show that, for 1-PT languages, it is sufficient to verify the equations $x = x^2$ and $xy = yx$ on letters (generators), and that, for 2-PT languages, equation $xyzx = xyxzx$ can be verified on letters (generators) up to the element y, which is a word (a general element of the monoid). Our results thus decrease the complexity of the problems. In addition, the partial order and (local) confluency can be checked instead of the equation $(xy)^2 = (yx)^2$.

The following example demonstrates an application of our characterization lemmas.

Example 17. Consider the language L recognized by the minimal DFA depicted in Figure 4. By [19, 31] or Theorem 25 below, L is piecewise testable. Since the depth of the DFA is 3, L is 3-PT [19]. Using Lemmas 5 and 12, the reader can verify that the language is 2-PT but not 1-PT. Furthermore, the technique studied in this paper and demonstrated in Section 3 results in $L = [aab] \cup [aabb] \cup [aaba] \cup [abba]$, where (using Lemma 3) $[aab] = L_{aa} \cap L_{ab} \cap \overline{L_{ba}} \cap \overline{L_{bb}}$, $[aabb] = L_{aa} \cap L_{ab} \cap L_{bb} \cap \overline{L_{ba}}$, $[aaba] = L_{aa} \cap L_{ab} \cap L_{ba} \cap \overline{L_{bb}}$, $[abba] = L_{aa} \cap L_{ab} \cap L_{ba} \cap L_{bb}$. By the standard De Morgan's laws, $L = L_{aa} \cap L_{ab} \cap \big((\overline{L_{ba}} \cap \overline{L_{bb}}) \cup (\overline{L_{ba}} \cap L_{bb}) \cup (L_{ba} \cap \overline{L_{bb}}) \cup (L_{ba} \cap L_{bb})\big) = L_{aa} \cap L_{ba} \cap \{a,b\}^* = L_{aa} \cap L_{ab}$.

3-Piecewise Testability. In this paragraph, we make use of the known equations $(xy)^3 = (yx)^3$, $xzyxvxwy = xzxyxvxwy$ and $ywxvxyzx = ywxvxyxzx$ characterizing the variety of 3-PT languages [3] to show NL-completeness of the 3-piecewise testability problem. The hardness is shown in Lemma 14. For the membership, we make use of the closure of NL under complement. To show that one of these equations is not satisfied, we guess a fix number of states (at most 18) and step by step (in parallel) the transitions. For instance, to check that $xy = yx$ is not satisfied, we guess states q, p_1, p_2, r_1, r_2 such that (i) $r_1 \neq r_2$ and (ii) $q \xrightarrow{x} p_1 \xrightarrow{y} r_1$ and $q \xrightarrow{y} p_2 \xrightarrow{x} r_2$. This requires several reachability checks, where we also ensure that the guessed paths from q to p_1 and from p_2 to r_2 are under the same label, x, and similarly for the paths from p_1 to r_1 and q to p_2 under y. It can be done by guessing the transitions for the four-tuple of labels in parallel. Namely, in the first step, the algorithm guesses a tuple of transitions $(q \xrightarrow{a} p'_1, q \xrightarrow{b} p'_2, p_1 \xrightarrow{b} r'_1, p_2 \xrightarrow{a} r'_2)$, which ensures that the related path labels begin with the same letter. It then continues until the paths satisfying (ii) are found. This method can easily be extended to any such an equation, thus we have the following.

Theorem 18. *To decide whether a minimal DFA recognizes a 3-PT language is NL-complete.*

Open Problem 19. Is there a better characterization for 3-PT languages similar to that of 1-PT and 2-PT languages?

5. Complexity of k-Piecewise Testability for NFAs

The *k-piecewise testability problem for NFAs* asks whether, given an NFA $\mathscr{A}$, the language $L(\mathscr{A})$ is k-PT.

Theorem 20. *The k-piecewise testability problem for NFAs is PSPACE-complete.*

Proof. Hunt III and Rosenkrantz [12] have shown that a property P of languages over $\{0,1\}$ such that (i) $P(\{0,1\}^*)$ is true and (ii) there exists a regular language that is not expressible as a quotient $x\backslash L$, for some L for which $P(L)$ is true, is as hard as to decide "$= \{0,1\}^*$". Since k-piecewise testability is such a property (the class of k-PT languages is closed under quotient) and universality is PSPACE-hard for NFAs, the result implies that k-piecewise testability for NFAs is PSPACE-hard.

We now prove membership. To do this, we show a co-NP upper bound for DFAs and use it to prove the rest of the theorem. Let w_1, w_2 be two words such that $w_1 \preccurlyeq w_2$. Let $\varphi : \{1,2,\ldots,|w_1|\} \to \{1,2,\ldots,|w_2|\}$ be a monotonically-increasing mapping induced by an embedding of w_1 into w_2, that is, the letter at the j^{th} position in w_1 coincides with the letter at the $\varphi(j)^{\text{th}}$ position in w_2. Any such φ is called a *witness (of the embedding) of* w_1 *in* w_2. If we speak about *a letter* a *of* w_2 *that does not belong to the range of* φ, we mean an occurrence of a in w_2 whose position does not belong to the range of φ.

Let $\mathscr{B}$ be an NFA over Σ, and let $\mathscr{A}$ be the minimal DFA obtained from $\mathscr{B}$ by the standard subset construction and minimization.

Claim 21. *If there are two words w_1, w_2 that are k-equivalent and lead to two different states from the initial state of $\mathscr{A}$, such that w_1 is a subword of w_2, then there exists a w_2' that is k-equivalent to w_1 leading to the same state as w_2 such that w_2' contains at most $\mathsf{depth}(\mathscr{A})$ more letters than w_1.*

Proof. Consider w_1, w_2 from the statement. Let φ be a witness of w_1 in w_2. Let a be a letter of w_2 that does not belong to the range of φ. We denote $w_2 = w_a a w_a^c$. If $i w_a a = i w_a$, then $i w_a w_a^c = i w_2$. Since $a \notin \mathsf{range}(\varphi)$, w_1 is a subword of $w_a w_a^c$. Thus, $\mathsf{sub}_k(w_1) \subseteq \mathsf{sub}_k(w_a w_a^c) \subseteq \mathsf{sub}_k(w_2)$, which proves that w_1 and $w_a w_a^c$ are k-equivalent. By induction on the number of letters in w_2 that do not belong to the range of the given witness of w_1 in w_2 and that do not trigger a change of state in $\mathscr{A}$, one can show that there exists a word equivalent to w_1 and leading to the same state as w_2 that does not contain any such letter. Note that if $\mathscr{A}$ were not acyclic, $L(\mathscr{B})$ would not be piecewise testable. This can be checked in PSPACE. Since in a run of an acyclic automaton there are at most $\mathsf{depth}(\mathscr{A})$ changes of states, this concludes the proof. □

Claim 22. *If $L(\mathscr{A})$ is not k-PT, there are two words w_1, w_2 such that (i) w_1 and w_2 are k-equivalent, (ii) the length of w_1 is at most $k|\Sigma|^k$, (iii) w_1 is a subword of w_2, and (iv) w_1 and w_2 lead to two different states from the initial state.*

Proof. If $L(\mathscr{A})$ is not k-PT, then there are w_1 and w_2 that are k-equivalent and lead to two different states from the initial state. We show that for $i \in \{1,2\}$, there exists w_i' such that $w_i \sim_k w_i'$ and the length of w_i' is at most $k|\Sigma|^k$. Let w_i^j denote the prefix of w_i of length j, for any j smaller than the length of w_i. Assume that there exists j such that $\mathsf{sub}_k(w_i^j) = \mathsf{sub}_k(w_i^{j+1})$. Then the letter at the $(j+1)^{\text{th}}$ position of w_i can be removed while keeping the same set of subwords of length k. Thus there exists w_i' equivalent to w_i such that any two different prefixes of w_i' are not k-equivalent. Since $\mathsf{sub}_k(w_i'^j) \subsetneq \mathsf{sub}_k(w_i'^{j+1})$, such a w_i' contains at most $\sum_{n=1}^{k} |\Sigma|^n \leq k|\Sigma|^k$ letters.

To complete the proof, there are two cases. Either w_1' and w_2' lead to the same state: then, without loss of generality, w_1' and w_1 lead to two different states, which proves the claim. Or w_1' and w_2' lead to two different states: then consider w' such that $w' \sim_k w_1'$, and both w_1' and w_2' are subwords of w', which exists by [25, Theorem 6.2.6]. Without loss of generality, w_1' and w' fulfill the required conditions. □

Claim 23. *The k-piecewise testability problem for DFAs belongs to co-NP.*

Proof. One can first check whether $\mathscr{A}$ recognizes a PT language. By Claim 22, if $L(\mathscr{A})$ is not k-PT, there exist two k-equivalent words w_1 and w_2, with the length of w_1 being at most $k|\Sigma|^k$, w_1 being a subword of w_2, and w_1 and w_2 leading the automaton to two different states. By Claim 21, one can choose w_2 of length at most $\mathit{depth}(\mathscr{A})$ bigger than the length of w_1. A polynomial certificate for non-k-piecewise testability can thus be given by providing such w_1 and w_2, which are of polynomial length in the size of $\mathscr{A}$ and Σ. □

We now continue to prove the theorem. By Claim 23 and the fact that NPSPACE=PSPACE=co-PSPACE, we can guess and store a word w_1 of length at most $k|\Sigma|^k$ and enumerate and store all words of length at most k. There are $\sum_{i=1}^{k}|\Sigma|^i$ such words, which is polynomial, since k is a constant. First, we mark all of these words that appear as subwords of w_1. Then we guess (letter by letter) a word w_2 such that w_1 is a subword of w_2 (which can be checked by keeping a pointer to w_1) and such that the length of w_2 is at most $|w_1|+2^n = O(2^n)$, where n is the number of states of the NFA. With each guess of the next letter of w_2, we correspondingly move all the pointers to all the stored subwords to keep track of all subwords of w_2. We accept if w_1 and w_2 have the same subwords, w_1 is a subword of w_2, and w_1 and w_2 lead the minimal DFA $\mathscr{A}$ to two different states. Because of the space limits the minimal DFA $\mathscr{A}$ cannot be stored in memory, but must be simulated on-the-fly while the word w_2 is being guessed. The state of $\mathscr{A}$ defined by w_2 can then be compared with the state defined by w_1. □

Open Problem 24. What is the complexity of k-piecewise testability for NFAs if k is given as input?

6. Piecewise Testability and the Depth of NFAs

We now generalize the structural automata characterization of Fact 2 to NFAs. Then we investigate the relationship between the depth of an NFA and the minimal k for which its language is k-PT and show that the upper bound on k given by the depth of the minimal DFA can be exponentially far from minimality.

6.1. The UMS property and NFAs

We say that an NFA $\mathscr{A}$ over an alphabet Σ is *complete* if for every state q of $\mathscr{A}$ and every letter $a \in \Sigma$, the set $q \cdot a$ is nonempty, that is, in every state, a transition under every letter is defined.

Theorem 25. *A regular language is piecewise testable if and only if there exists a complete NFA that is partially ordered and satisfies the UMS property.*

Proof. If a regular language is PT, then its minimal DFA is partially ordered and satisfies the UMS property by [31].

To prove the other direction, let $\mathscr{A} = (Q, \Sigma, \cdot, I, F)$ be a complete partially ordered NFA that satisfies the UMS property. Let $\mathscr{D}$ be the minimal DFA computed from $\mathscr{A}$ by the standard subset construction and minimization. We represent every state of $\mathscr{D}$ by a nonempty set of states of $\mathscr{A}$.

Claim 26. *The minimal DFA $\mathscr{D}$ is partially ordered.*

Proof. Let $X = \{p_1, p_2, \ldots, p_n\}$ with $p_i < p_j$ for $i < j$ be a state of $\mathscr{D}$, and let $w \in \Sigma^*$ be such that $X \cdot w = X$. By induction on $k = 1, 2, \ldots, n$, we show that $p_i w = \{p_i\}$. Assume that $p_i w = \{p_i\}$ for all $i < k$. We prove it for k. Since $X = Xw = \cup_{i=1}^{n} p_i w$, $p_k \leq p_k w$ and $p_i w = \{p_i\}$ for $i < k$, we have that $p_k \in p_k w$. Thus, $\mathrm{alph}(w) \subseteq \Sigma(p_k)$ and the UMS property of $\mathscr{A}$ implies that $p_k w = \{p_k\}$. Therefore, $p_i a = \{p_i\}$ for every $a \in \mathrm{alph}(w)$ and $i = 1, 2, \ldots, n$. If, for any state Y of $\mathscr{D}$ and any words w_1 and w_2, $Xw_1 = Y$ and $Yw_2 = X$, the previous argument gives that $X = Y$, hence $\mathscr{D}$ is partially ordered. □

Claim. *The minimal DFA $\mathscr{D}$ satisfies the UMS property.*

Proof. As $\mathscr{D}$ is deterministic, for every state X of $\mathscr{D}$, X is a maximal state of $G(\mathscr{D}, \Sigma(X))$. Assume, for the sake of contradiction, that there exist two different states X and Y in the same component of $\mathscr{D}$ that are maximal with respect to alphabet $\Sigma(X)$. That is, there exist a state Z in $\mathscr{D}$ and two words u and v over $\Sigma(X)$ such that $X = Zu$ and $Y = Zv$. If $X \setminus Y \neq \emptyset$, let $x \in X \setminus Y$ and $z \in Z$ be such that $x \in zu$. Since x does not belong to Y, we have that $x \notin zv$. Note that $zv \neq \emptyset$, since $\mathscr{A}$ is complete. Let $y \in zv$ be fixed, but arbitrarily. (If $X \setminus Y = \emptyset$, then there is $y \in Y \setminus X$. In this case, let $z \in Z$ be such that $y \in zv$. Then $y \notin zu$, $zu \neq \emptyset$, and we fix an arbitrary $x \in zu$.) In any case, $x \neq y$. Since $x \in X$,

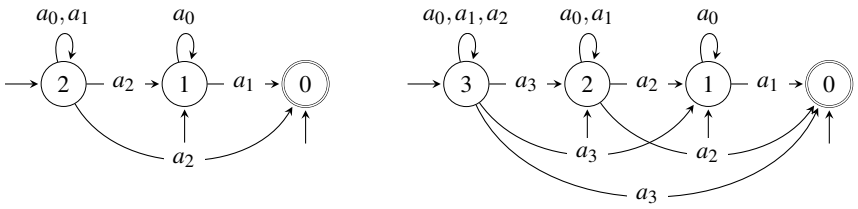

Figure 5: Automata $\mathscr{A}_2$ and $\mathscr{A}_3$.

$y \in Y$, and X and Y are maximal with respect to $\Sigma(X)$, a similar argument as in Claim 26 shows that $xa = \{x\}$ and $ya = \{y\}$ for any $a \in \Sigma(X)$. Thus, we have that $\Sigma(X) \subseteq \Sigma(x) \cap \Sigma(y)$. By the UMS property of $\mathscr{A}$, x must be reachable from y by $\Sigma(x)$, hence $y \leq x$, and y must be reachable from x under $\Sigma(y)$, hence $x \leq y$. Therefore, $y = x$, which is a contradiction. □

Thus, the minimal DFA $\mathscr{D}$ is partially ordered and satisfies the UMS property. Fact 2 now completes the proof. □

As it is PSPACE-complete to decide whether an NFA defines a PT language, it is PSPACE-complete to decide whether, given an NFA, there is an equivalent complete NFA that is partially ordered and satisfies the UMS property. More details on these automata can be found in [22].

6.2. Exponential Gap between k-PT and the Depth of Minimal DFAs

It was shown in [19] that the depth of minimal DFAs does not correspond to the minimal k for which the language is k-PT. Namely, an example of $(4\ell - 1)$-PT languages with the minimal DFA of depth $4\ell^2$, for $\ell > 1$, has been presented. We now show that there is an exponential gap between the minimal k for which the language is k-PT and the depth of a minimal DFA.

Theorem 27. *For every $n \geq 1$, there exists an n-PT language that is not $(n-1)$-PT, it is recognized by an NFA of depth $n-1$, and the minimal DFA recognizing it has depth $2^n - 1$.*

Proof. For every $k \geq 0$, we define the NFA $\mathscr{A}_k = (\{0, 1, \ldots, k\}, \{a_0, a_1, \ldots, a_k\}, \cdot, I_k, \{0\})$ with $I_k = \{0, 1, \ldots, k\}$ and the transition function $\cdot$ consisting of self-loops under a_i in all states $j > i$ and transitions under a_i from state i to all states $j < i$. Formally, $i \cdot a_j = i$ if $k \geq j > i \geq 0$ and $i \cdot a_i = \{0, 1, \ldots, i-1\}$ if $k \geq i \geq 1$. Automata $\mathscr{A}_2$ and $\mathscr{A}_3$ are shown in Figure 5. Note that $\mathscr{A}_k$ is an extension of $\mathscr{A}_{k-1}$, in particular, $L(\mathscr{A}_{k-1}) \subseteq L(\mathscr{A}_k)$.

We define the word w_k inductively by $w_0 = a_0$ and $w_\ell = w_{\ell-1} a_\ell w_{\ell-1}$, for $0 < \ell \leq k$. Note that $|w_\ell| = 2^{\ell+1} - 1$. In [11], we have shown that every prefix of w_k of odd length ends with a_0 and, therefore, does not belong to $L(\mathscr{A}_k)$, while every prefix of even length belongs to $L(\mathscr{A}_k)$. For convenience, we briefly recall the proof here. The empty word belongs to $L(\mathscr{A}_0) \subseteq L(\mathscr{A}_k)$. Let v be a prefix of w_k of even length. If $|v| < 2^k - 1$, then v is a prefix of w_{k-1} and, by the induction hypothesis, $v \in L(\mathscr{A}_{k-1}) \subseteq L(\mathscr{A}_k)$. If $|v| > 2^k - 1$, then $v = w_{k-1} a_k v'$. The definition of $\mathscr{A}_k$ and the induction hypothesis then yield that there is a path $k \xrightarrow{w_{k-1}} k \xrightarrow{a_k} (k-1) \xrightarrow{v'} 0$. Thus, v belongs to $L(\mathscr{A}_k)$.

Let $det(\mathscr{A}_k)$ denote the minimal DFA recognizing the language $L(\mathscr{A}_k)$ obtained from $\mathscr{A}_k$ by the standard subset construction and minimization.

Claim. *For every $k \geq 0$, the depth of $det(\mathscr{A}_k)$ is $2^{k+1} - 1$.*

Proof. By induction on k. For $k = 0$, $det(\mathscr{A}_0) = (\{\{0\}, \emptyset\}, \{a_0\}, \cdot, \{0\}, \{0\})$ has two states, accepts the single word ε, and a_0 goes from the initial state $I_0 = \{0\}$ to the sink state $\emptyset$. Thus, it has depth 1 as required. Consider the word $w_k = w_{k-1} a_k w_{k-1}$ for $k > 0$. By the induction hypothesis, there exists a simple path of length $2^k - 1$ in $det(\mathscr{A}_{k-1})$ defined by the word w_{k-1} starting from the initial state $I_k = \{0, 1, \ldots, k-1\}$ and ending in state $\emptyset$. Let $Q_0, Q_1, \ldots, Q_{2^k-1}$ denote the states of that simple path in the order they appear on the path, that is, $Q_0 = I_k$, $Q_{2^k-1} = \emptyset$, and $Q_i \subseteq Q_0$

for $i = 1, 2, \ldots, 2^k - 1$. The states are pairwise non-equivalent by the induction hypothesis. Let $w_{k-1,i}$ denote the i-th letter of the word w_{k-1}. Then the path

$$\underbrace{(Q_0 \cup \{k\}) \xrightarrow{w_{k-1,1}} (Q_1 \cup \{k\}) \xrightarrow{w_{k-1,2}} (Q_2 \cup \{k\}) \xrightarrow{\cdots} (Q_{2^k-1} \cup \{k\})}_{w_{k-1}} \xrightarrow{a_k} \underbrace{Q_0 \xrightarrow{w_{k-1,1}} Q_1 \xrightarrow{w_{k-1,2}} Q_2 \xrightarrow{\cdots} Q_{2^k-1}}_{w_{k-1}}$$

consists of 2^{k+1} different states. We show that these states are pairwise non-equivalent. Since the letter a_k is accepted from every state $Q_j \cup \{k\}$, but from no state Q_i, for $0 \leq i, j \leq 2^k - 1$, state $Q_j \cup \{k\}$ is distinguishable from state Q_i. Moreover, $Q \cup \{k\}$ and $Q' \cup \{k\}$ are distinguished by the same word as the states Q and Q', which are distinguishable by the induction hypothesis. Thus, we have a simple path of length $2^{k+1} - 1$ as required. □

We now show that $\mathscr{A}_k$ defines a $(k+1)$-PT language that is not k-PT.

Claim. *For every $k \geq 0$, $L(\mathscr{A}_k)$ is $(k+1)$-PT.*

Proof. By induction on k. For $k = 0$, $L(\mathscr{A}_0) = \{\varepsilon\} = \cap_{a \in \Sigma} \overline{L_a}$ is 1-PT. Consider the automaton $\mathscr{A}_k$ and let u and v be two words such that $u \sim_{k+1} v$. Assume that $u \in L(\mathscr{A}_k)$. We show that $v \in L(\mathscr{A}_k)$ as well. If u does not contain letter a_k, then $u \in L(\mathscr{A}_{k-1})$ and, since $u \sim_{k+1} v$ implies that $u \sim_k v$, the induction hypothesis gives that $v \in L(\mathscr{A}_{k-1}) \subseteq L(\mathscr{A}_k)$. If u contains a_k, the definition of $\mathscr{A}_k$ gives that u is of the form $u = u_1 a_k u_2$, where $u_1 u_2$ does not contain a_k. Since $u \sim_{k+1} v$, v is also of the form $v = v_1 a_k v_2$, where $v_1 v_2$ does not contain a_k. However, $u_2 \sim_k v_2$, since $w \in sub_k(u_2)$ if and only if $a_k w \in sub_{k+1}(u_1 a_k u_2) = sub_{k+1}(v_1 a_k v_2)$, which is if and only if $w \in sub_k(v_2)$. Since, by the induction hypothesis, $u_2 \in L(\mathscr{A}_{k-1})$ implies that $v_2 \in L(\mathscr{A}_{k-1})$, we obtain that $v \in L(\mathscr{A}_k)$. □

Claim. *For every $k \geq 0$, $L(\mathscr{A}_k)$ is not k-PT.*

Proof. Let $w_k = w_{k-1} a_k w_{k-1}$ be the word defined above. Let w'_k denote its prefix without the last letter (which is a_0), that is, $w_k = w'_k a_0$. We show, by induction on k, that $w_k \sim_k w'_k$. This then implies that $L(\mathscr{A}_k)$ is not k-PT, because w'_k belongs to $L(\mathscr{A}_k)$ and w_k does not. For $k = 0$, $w_0 = a_0 \sim_0 \varepsilon = w'_0$. Thus, assume that $w_k \sim_k w'_k$ for some $k \geq 0$, and consider $w \in sub_{k+1}(w_k a_{k+1} w_k)$. Then the word w can be decomposed to $w = w'w''$, where w' is the maximal prefix of w that can be embedded into the word $w_k a_{k+1}$. Note that w'' is a suffix of w that can be embedded into w_k. Since $|w'| > 0$, we have that $|w''| \leq k$. By the induction hypothesis, $w'' \in sub_k(w_k) = sub_k(w'_k)$. Thus, $w = w'w'' \in sub_{k+1}(w_k a_{k+1} w'_k)$, which proves that $w_{k+1} \sim_{k+1} w'_{k+1}$. □

To finish the proof of Theorem 27, note that every NFA $\mathscr{A}_k$ has depth k, accepts a $(k+1)$-PT language that is not k-PT and its minimal DFA has depth $2^{k+1} - 1$. This completes the proof. □

Although it is well known that DFAs can be exponentially larger than NFAs, an interesting by-product of the previous proof is that there are NFAs such that all the exponential number of states of their minimal DFAs form a simple path. Notice that the reverse of the NFA constructed above is a DFA, hence the result also contributes to the state complexity of the reverse of piecewise testable languages, cf. [4, 14].

It could seem that NFAs are more convenient to provide upper bounds on k-PT. However, the following simple example demonstrates that even for 1-PT languages, the depth of an NFA depends on the size of the input alphabet. Specifically, for any alphabet Σ, the language $L = \bigcap_{a \in \Sigma} L_a$ of all words containing all letters of Σ is a 1-PT language such that any NFA recognizing it requires at least $2^{|\Sigma|}$ states and has depth $|\Sigma|$. A deeper investigation in this direction follows in the next section.

Example 28. Let $L = \bigcap_{a \in \Sigma} L_a$ be a language of all words that contain all letters of the alphabet. Then $2^{|\Sigma|}$ states are sufficient for an NFA to recognize L. Indeed, the automaton $\mathscr{A} = (2^\Sigma, \Sigma, \cdot, \{\emptyset\}, \{\Sigma\})$ with the transition function defined by $X \cdot a = X \cup \{a\}$, for $X \subseteq \Sigma$ and $a \in \Sigma$, recognizes L. The depth of $\mathscr{A}$ is $|\Sigma|$, since every non-self-loop transition goes to a strict superset of the current state.

To prove that every NFA requires at least $2^{|\Sigma|}$ states, we use the fooling set lower-bound technique [2]. A set of pairs of words $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ is a fooling set for L if, for all i, the words $x_i y_i$ belong to L and, for $i \neq j$, at least one of the words $x_i y_j$ and $x_j y_i$ does not belong to L. To construct such a fooling set, for any $X \subseteq \Sigma$, we fix a word w_X such that $\text{alph}(w_X) = X$. Let $S = \{(w_X, w_{\Sigma \setminus X}) \mid X \subseteq \Sigma\}$. Then $\text{alph}(w_X w_{\Sigma \setminus X}) = \Sigma$ and $w_X w_{\Sigma \setminus X}$ belongs to L.

On the other hand, for $X \neq Y$, either $X \cup (\Sigma \setminus Y)$ or $Y \cup (\Sigma \setminus X)$ is different from Σ, which implies that S is a fooling set of size $2^{|\Sigma|}$. The main result of [2] now implies the claim. It remains to prove that the depth is at least $|\Sigma|$. However, the shortest words of L are of length $|\Sigma|$, which completes the proof.

7. Tight Bounds on the Depth of Minimal DFAs

If a PT language is recognized by a minimal DFA of depth ℓ, then it is ℓ-PT. However, the opposite implication does not hold and the analysis of Section 6 shows that the language can be k-PT with ℓ being exponentially bigger than k. Therefore, we study the opposite implication of the relationship between k-piecewise testability and the depth of the minimal DFA. Specifically, given a k-PT language over an n-letter alphabet, we show that the depth of the minimal DFA recognizing it is at most $\binom{k+n}{k} - 1$. To this end, we investigate the following problem.

Problem 29. Let Σ be an alphabet of cardinality $n \geq 1$, and let $k \geq 1$. What is the length of a longest word, w, such that $sub_k(w) = \Sigma^{\leq k} = \{v \in \Sigma^* \mid |v| \leq k\}$ and, for any two distinct prefixes w_1 and w_2 of w, $sub_k(w_1) \neq sub_k(w_2)$?

Equivalently stated, Problem 29 asks what is the depth of the $\sim_k$*-canonical DFA*. The answer is formulated below.

Lemma 30. *Let Σ be an alphabet of cardinality n. Then the length of a longest word satisfying the requirements of Problem 29 is given by the recursive formula $P_{k,n} = P_{k-1,n} + P_{k,n-1} + 1$, where $P_{1,m} = m = P_{m,1}$, for $m \geq 1$. It can be shown by induction that $P_{k,n} = \binom{k+n}{k} - 1$.*

Proof. We show the lemma in two claims. The first claim shows that w is not longer than $P_{k,n}$.

Claim. *Let w' be a word over Σ satisfying the requirements of Problem 29. Then $|w'| \leq P_{k,n}$.*

Proof. It is clear that $P_{1,m} = m = P_{m,1}$. Let $\Sigma = \{a_1, a_2, \ldots, a_n\}$ with the order $a_i < a_j$ if $i < j$ induced by the occurrence of letters in w'. For instance, *abadca* induces the order $a < b < d < c$. Let z denote the first occurrence of a_n in w'. Then $w' = w_1 z w_2$, where $w_1 \in \{a_1, a_2, \ldots, a_{n-1}\}^*$ satisfies the second requirement of Problem 29, hence $|w_1| \leq P_{k,n-1}$. On the other hand, since $\mathrm{alph}(w_1 z) = \Sigma$, any prefix of w_2 extends the set of subwords with a subword of length at least 2. Thus, w_2 cannot be longer than the longest word over Σ containing all subwords up to length $k-1$, that is, $|w_2| \leq P_{k-1,n}$. □

The second claim shows that there exists a word of length $P_{k,n}$.

Claim. *There exists a word w of length $P_{k,n}$ satisfying the requirements of Problem 29.*

Proof. Let Σ_n denote the alphabet $\{a_1, a_2, \ldots, a_n\}$ with the order $a_i < a_j$ if $i < j$. For $n = 1$ and $k \geq 1$, the word $W_{k,1} = a^k$ is of length $P_{k,1}$ and satisfies the requirements, as well as the word $W_{1,n} = a_1 a_2 \ldots a_n$ of length $P_{1,n}$ for $k = 1$ and $n \geq 1$. Assume that we have constructed the words $W_{i,j}$ of length $P_{i,j}$ for all $i < k$ and $j < n$, $W_{i,n}$ of length $P_{i,n}$ for all $i < k$, and $W_{k,j}$ of length $P_{k,j}$ for all $j < n$. We construct the word $W_{k,n}$ of length $P_{k,n}$ over Σ_n as follows:

$$W_{k,n} = W_{k,n-1}\, a_n\, W_{k-1,n}\,.$$

It remains to show that $W_{k,n}$ satisfies the requirements of Problem 29. We first show that any word, w, of length less than or equal to k is a subword of $W_{k,n}$. If w does not contain a_n, then it is a subword of $W_{k,n-1}$ by induction assumption. Otherwise, let $w = w_1 a_n w_2$ with w_1 not containing a_n. By induction, w_1 is a subword of $W_{k,n-1}$, while w_2 is of length strictly less than k, hence a subword of $W_{k-1,n}$. Thus $w = w_1 a_n w_2$ is a subword of $W_{k,n-1} a_n W_{k-1,n}$, which shows that the first condition of Problem 29 holds.

As for the second condition, let w_1 and w_2 be two different prefixes of $W_{k,n}$. Without loss of generality, we may assume that w_1 is a prefix of w_2. If they are both prefixes of $W_{k,n-1}$, the second requirement of Problem 29 follows by induction. If w_1 is a prefix of $W_{k,n-1}$ and w_2 contains a_n, then the second requirement of Problem 29 is satisfied, because w_1 does not contain a_n. Thus, assume that both w_1 and w_2 contain a_n, that is, they both contain $W_{k,n-1} a_n$ as a prefix. Let $w_1 = W_{k,n-1} a_n w_1'$ and $w_2 = W_{k,n-1} a_n w_1' w_2'$. Since, by induction, $sub_{k-1}(w_1') \subsetneq sub_{k-1}(w_1' w_2')$, there exists $v \in sub_{k-1}(w_1' w_2') \setminus sub_{k-1}(w_1')$. Then $a_n v$ belongs to $sub_k(w_2)$, but not to $sub_k(w_1)$, which completes the proof. □

k \ n	n=1	n=2	n=3	n=4	n=5	n=6
k=1	1	2	3	4	5	6
k=2	2	5	9	14	20	27
k=3	3	9	19	34	55	83
k=4	4	14	34	69	125	209
k=5	5	20	55	125	251	461
k=6	6	27	83	209	461	923

Table 1: A few first numbers $P_{k,n}$

This completes the proof of Lemma 30. □

We now establish a bound on the depth of the minimal DFA recognizing a k-PT language over an n-letter alphabet. This result has independently been obtained by Klíma, Kunc and Polák [18].

Theorem 31. *For any natural numbers k and n, the depth of the minimal DFA recognizing a k-PT language over an n-letter alphabet is at most $\binom{k+n}{k}-1$. The bound is tight for any k and n.*

Proof. Let $L_{k,n}$ be a k-PT language over an n-letter alphabet. Since $L_{k,n}$ is a finite union of $\sim_k$ classes [26], there exists F such that the $\sim_k$-canonical DFA $\mathscr{A} = (Q,\Sigma,\cdot,[\varepsilon],F)$ recognizes $L_{k,n}$. The depth of $\mathscr{A}$ is $P_{k,n}$. Let $min(\mathscr{A})$ denote the minimal DFA obtained from $\mathscr{A}$ by the standard minimization procedure. Since the minimization does not increase the depth, the depth of $min(\mathscr{A})$ is at most $P_{k,n} = \binom{k+n}{k}-1$.

To show that the bound is tight, let $w = x_1x_2\cdots x_\ell$ be a fixed word of length $\ell = P_{k,n}$, where $x_i \in \Sigma$ for $i = 1,\ldots,\ell$. Such a word exists by Lemma 30. Consider the $\sim_k$-canonical DFA $\mathscr{A}' = (Q,\Sigma,\cdot,[\varepsilon],F)$, where $F = \{[w'] \mid w'$ is a prefix of w of even length$\}$. Then w defines a path $\pi_w = [\varepsilon] \xrightarrow{x_1} [x_1] \xrightarrow{x_2} [x_1x_2]\ldots \xrightarrow{x_\ell} [w]$ in $\mathscr{A}'$ of length $P_{k,n}$, where accepting and non-accepting states alternate. Again, let $min(\mathscr{A}')$ denote the minimal DFA obtained from $\mathscr{A}'$. If there were two equivalent states in π_w, then they must be of the same acceptance status. However, between any two states with the same acceptance status, there exists a state with the opposite acceptance status. Therefore, joining the two states creates a cycle in $min(\mathscr{A}')$, which is a contradiction with Fact 2, since the DFA $\mathscr{A}'$ recognizes a PT language. □

A few of these numbers are listed in Table 1. The reader can notice a remarkable relation between the columns (rows) of Table 1 and the generalized Catalan numbers of Frey and Sellers [8]. The interpretation of this correspondence is not yet clear and is left for a future investigation. Another relation between the depth of $\sim_k$-canonical DFAs and Stirling cyclic numbers is depicted below.

Proposition 32. *For positive integers k and n, $P_{k,n} = \frac{1}{k!}\sum_{i=1}^{k}\left[{k+1 \atop i+1}\right]n^i$, where $\left[{k \atop n}\right]$ denotes the Stirling cyclic numbers.*

Proof. We first recall the following well-known properties of Stirling cyclic numbers:

$$\left[{k+1 \atop 1}\right] = k! \qquad \text{and} \qquad \sum_{i=0}^{k}\left[{k \atop i}\right]x^i = x(x+1)\cdots(x+k-1) = \frac{(x+k-1)!}{(x-1)!} \tag{1}$$

Now, $\frac{1}{k!}\sum_{i=1}^{k}\left[{k+1 \atop i+1}\right]n^i = \frac{1}{nk!}\sum_{i=1}^{k}\left[{k+1 \atop i+1}\right]n^{i+1} = \frac{1}{nk!}\sum_{i=2}^{k+1}\left[{k+1 \atop i}\right]n^i = \frac{1}{nk!}\left(\sum_{i=0}^{k+1}\left[{k+1 \atop i}\right]n^i - \left[{k+1 \atop 1}\right]n\right) = \frac{1}{nk!}\left(\frac{(k+n)!}{(n-1)!} - k!n\right) = \frac{(k+n)!}{n!k!} - 1 = P_{k,n}$, where the equations are by multiplication by n/n, changing indexes, adding the cases $i = 0,1$ into the sum, using Equation 1, and by simplification. □

Open Problem 33. What is the interpretation of the relation between the depth of $\sim_k$-canonical DFAs and Stirling/Catalan numbers?

A closely related problem to the question on the depth of the $\sim_k$-canonical DFA over an n element alphabet is the question on the size (number of states) of the $\sim_k$-canonical DFA. Although this problem has been investigated in the literature, see Karandikar, Kufleitner and Schnoebelen [15], the precise answer is still open.

Open Problem 34. Is there a formula on the size of the $\sim_k$-canonical DFA over an n element alphabet based on k and n?

8. Conclusion

We investigated the descriptional and computational complexity of an approach of translating an automaton recognizing a piecewise testable language into a Boolean combination of languages of the form $\Sigma^* a_1 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$, where $a_i \in \Sigma$. We focused on the Boolean combination of languages of the form $\Sigma^* a_1 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$ resembling the disjunctive normal form of logical formulas, where n is at most the minimal k for which the language is k-PT. We also discussed the latest results and formulated open problems.

This work can be seen as translating an automaton into a form of a generalized regular expression allowing the operation of complement. It is known that generalized regular expressions can be non-elementary more succinct than classical regular expressions, however it is not yet known whether this non-elementary succinctness can be witnessed by a piecewise testable language. To the best of our knowledge, not much is known about transformations to generalized regular expressions. This paper contributes to this topic that still needs to be investigated.

Acknowledgements. We thank the authors of [10] and [18] for the full versions of their papers, and Sebastian Rudolph and Markus Krötzsch for fruitful discussions.

References

[1] Barrington, D. A. M., Lu, C., Miltersen, P. B., Skyum, S., 1998. Searching constant width mazes captures the AC^0 hierarchy. In: Morvan, M., Meinel, C., Krob, D. (Eds.), Symposium on Theoretical Aspects of Computer Science (STACS). Vol. 1373 of LNCS. Springer, pp. 73–83.
[2] Birget, J.-C., 1992. Intersection and union of regular languages and state complexity. Information Processing Letters 43, 185–190.
[3] Blanchet-Sadri, F., 1989. Games, equations and the dot-depth hierarchy. Computers & Mathematics with Applications 18 (9), 809–822.
[4] Brzozowski, J. A., Li, B., 2014. Syntactic complexity of $\mathscr{R}$- and $\mathscr{J}$-trivial regular languages. International Journal of Foundations of Computer Science 25 (7), 807–822.
[5] Cho, S., Huynh, D. T., 1991. Finite-automaton aperiodicity is PSPACE-complete. Theoretical Computer Science 88 (1), 99–116.
[6] Dang, Z. R., 1973. On the complexity of a finite automaton corresponding to a generalized regular expression. Doklady Akademii Nauk SSSR 213, 26–29.
[7] Ellul, K., Krawetz, B., Shallit, J., Wang, M., 2005. Regular expressions: New results and open problems. Journal of Automata, Languages and Combinatorics 10 (4), 407–437.
[8] Frey, D. D., Sellers, J. A., 2001. Generalizing Bailey's generalization of the Catalan numbers. The Fibonacci Quarterly 39 (2), 142–148.
[9] Gelade, W., Neven, F., 2012. Succinctness of the complement and intersection of regular expressions. ACM Transactions on Computational Logic 13 (1), 4.
[10] Hofman, P., Martens, W., 2015. Separability by short subsequences and subwords. In: Arenas, M., Ugarte, M. (Eds.), International Conference on Database Theory (ICDT). Vol. 31 of LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 230–246.
[11] Holub, Š., Masopust, T., Thomazo, M., 2014. Alternating towers and piecewise testable separators. CoRR abs/1409.3943. URL `http://arxiv.org/abs/1409.3943`
[12] Hunt III, H. B., Rosenkrantz, D. J., 1978. Computational parallels between the regular and context-free languages. SIAM Journal on Computing 7 (1), 99–114.
[13] Immerman, N., 1988. Nondeterministic space is closed under complementation. SIAM Journal on Computing 17 (5), 935–938.
[14] Jirásková, G., Masopust, T., 2013. On the state complexity of the reverse of $\mathscr{R}$- and $\mathscr{J}$-trivial regular languages. In: Jürgensen, H., Reis, R. (Eds.), Descriptional Complexity of Formal Systems (DCFS). Vol. 8031 of LNCS. Springer, pp. 136–147.
[15] Karandikar, P., Kufleitner, M., Schnoebelen, P., 2015. On the index of Simon's congruence for piecewise testability. Information Processing Letters 115 (4), 515–519.
[16] Karandikar, P., Schnoebelen, P., 2016. The height of piecewise-testable languages with applications in logical complexity. In: Talbot, J., Regnier, L. (Eds.), EACSL Annual Conference on Computer Science Logic (CSL). Vol. 62 of LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 37:1–37:22.
[17] Klíma, O., 2011. Piecewise testable languages via combinatorics on words. Discrete Mathematics 311 (20), 2124–2127.
[18] Klíma, O., Kunc, M., Polák, L., 2014. Deciding k-piecewise testability, submitted.
[19] Klíma, O., Polák, L., 2013. Alternative automata characterization of piecewise testable languages. In: Béal, M., Carton, O. (Eds.), Developments in Language Theory (DLT). Vol. 7907 of LNCS. Springer, pp. 289–300.
[20] Lawson, M. V., 2003. Finite Automata. Chapman and Hall/CRC.

[21] Martens, W., Neven, F., Niewerth, M., Schwentick, T., 2015. Bonxai: Combining the simplicity of DTD with the expressiveness of XML schema. In: Milo, T., Calvanese, D. (Eds.), Principles of Database Systems (PODS). ACM, pp. 145–156.
[22] Masopust, T., 2016. Piecewise testable languages and nondeterministic automata. In: Faliszewski, P., Muscholl, A., Niedermeier, R. (Eds.), Mathematical Foundations of Computer Science (MFCS). Vol. 58 of LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 67:1–67:14.
[23] Masopust, T., Thomazo, M., 2015. On the complexity of k-piecewise testability and the depth of automata. In: Potapov, I. (Ed.), Developments in Language Theory (DLT). Vol. 9168 of LNCS. Springer, pp. 364–376.
[24] Myhill, J., 1957. Finite automata and representation of events. Tech. rep., Wright Air Development Center.
[25] Sakarovitch, J., Simon, I., 1997. Subwords. In: Lothaire, M. (Ed.), Combinatorics on words. Cambridge University Press, pp. 105–142.
[26] Simon, I., 1972. Hierarchies of events with dot-depth one. Ph.D. thesis, University of Waterloo, Canada.
[27] Simon, I., 1975. Piecewise testable events. In: Barkhage, H. (Ed.), GI Conference on Automata Theory and Formal Languages. Springer, pp. 214–222.
[28] Stern, J., 1985. Complexity of some problems from the theory of automata. Information and Computation 66 (3), 163–176.
[29] Stockmeyer, L. J., Meyer, A. R., 1973. Word problems requiring exponential time: Preliminary report. In: Aho, A. V., Borodin, A., Constable, R. L., Floyd, R. W., Harrison, M. A., Karp, R. M., Strong, H. R. (Eds.), Symposium on the Theory of Computing (STOC). ACM, pp. 1–9.
[30] Szelepcsényi, R., 1988. The method of forced enumeration for nondeterministic automata. Acta Informatica 26 (3), 279–284.
[31] Trahtman, A. N., 2001. Piecewise and local threshold testability of DFA. In: Freivalds, R. (Ed.), Fundamentals of Computation Theory (FCT). Vol. 2138 of LNCS. Springer, pp. 347–358.