On the Complexity of k-Piecewise Testability and the Depth of Automata

Tomáš Masopust [⋆] and Michaël Thomazo [⋆⋆]

TU Dresden, Germany
firstname.lastname@tu-dresden.de

Abstract. For a non-negative integer k, a language is k-piecewise testable (k-PT) if it is a finite boolean combination of languages of the form $\Sigma^* a_1 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$ for $a_i \in \Sigma$ and $0 \leq n \leq k$. We study the following problem: Given a DFA recognizing a piecewise testable language, decide whether the language is k-PT. We provide a complexity bound on this problem and a detailed analysis for small k's. The result can be use to find the minimal k for which the language is k-PT. We show that the upper bound on k given by the depth of the minimal DFA can be exponentially bigger than the minimal possible k, and provide a tight upper bound on the depth of the minimal DFA recognizing a k-PT language.

1 Introduction

A regular language is *piecewise testable* (PT) if it is a finite boolean combination of languages of the form $\Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$, where $a_i \in \Sigma$ and $n \geq 0$. It is *k-piecewise testable* (k-PT) if $n \leq k$. These languages were introduced by Simon in his PhD thesis [31]. Simon proved that PT languages are exactly those regular languages whose syntactic monoid is $\mathcal{J}$-trivial. He provided various characterizations of PT languages in terms of monoids, automata, etc.

In this paper, we study the *k-piecewise testability* problem, that is, to decide whether a PT language is k-PT.

Name: k-PiecewiseTestability
Input: an automaton (minimal DFA or NFA) $\mathcal{A}$
Output: Yes if and only if $\mathcal{L}(\mathcal{A})$ is k-piecewise testable

Note that the problem is trivially decidable, since there is only a finite number of k-PT languages over the input alphabet of $\mathcal{A}$.

We investigate the complexity of the problem and the relationship between k and the depth of the input automaton. The motivation to study this relationship comes from the result showing that a PT language is k-PT for any k bigger than or equal to the depth of its minimal DFA [21].

Our motivation is twofold. The first motivation is theoretical and comes from the investigation of various fragments of first-order logic over words, namely the

[⋆] Research supported by the DFG in grant KR 4381/1-1.
[⋆⋆] Research supported by the Alexander von Humboldt Foundation.

Straubing-Thérien and dot-depth hierarchies. For instance, the languages of levels 1/2 and 1 of the dot-depth hierarchy are constructed as boolean combinations of variants of languages of the form $\Sigma^* w_1 \Sigma^* \ldots \Sigma^* w_n \Sigma^*$, where $w_i \in \Sigma^*$, cf. [23, Table 1]. The reader can notice a similarity to PT languages. For these fragments, a problem similar to k-piecewise testability is also relevant.

The second, practical motivation comes from simplifying the XML Schema specification language.

Simplification of XML Schema XML Schema is currently the only schema language that is widely accepted and supported by industry. However, it is rather machine-readable than human-readable. It increases the expressiveness of DTDs, but this increase goes hand in hand with loss of simplicity. Moreover, its logical core does not seem to be well understood by users [24]. Therefore, the BonXai schema language has recently been proposed as an attempt to design a human-readable schema language. It combines the simplicity of DTDs with the expressiveness of XML Schema. Its aim is to simplify the development and analysis of XML Schema Definitions (XSDs). The BonXai schema is a set of rules of the form $L_i \to R_i$, where L_i and R_i are regular expressions. An XML document (unranked tree) belongs to the language of the schema if, for every node of the tree, the labels of its children form a word that belongs to R_i and its ancestors form a word that belongs to L_i, see [24] for more details.

When translating an XSD into an equivalent BonXai schema, the regular expressions L_i are obtained from a finite automaton embedded in the XSD. However, the current techniques of translating automata to regular expressions do not yet generate human-readable results. Therefore, we restrict ourselves to simpler classes of expressions that suffice in practice. Practical and theoretical studies show evidence that expressions of the form $\Sigma^* a_1 \Sigma^* \cdots \Sigma^* a_n$, where $a_i \in \Sigma$, and their variations are suitable for this purpose [15, 25].

Every state of the DFA embedded in the XSD represents a language and we need to compute an over-approximation L_i for each of them that is disjoint with the others. This reduces to the language separation problem: Given two languages K and L and a family of languages $\mathcal{F}$, is there a language S in $\mathcal{F}$ such that S includes K and is disjoint with L? It is independently shown in [7] and [28] that the separation problem for regular languages represented by NFAs and the family of PT languages is decidable in polynomial time. A simple method (in the meaning of description) to compute a PT separator is described in [17], where its running time is investigated. Another technique is described in [28].

Assume that we have computed a PT separator. Since the standard algorithms translating automata to regular expressions do not generate human-readable results and mostly use “only” the basic operations (concatenation, Kleene star and union), we face the problem how to generate human-readable expressions of the considered simple forms. Note that the expressions we are interested in contain the operations of intersection and complement (called generalized regular expressions). These operations make them non-elementary more succinct than classical regular expressions [33]. Unfortunately, not much is known about transformations to generalized regular expressions [10].

For a PT language it means to decompose it into a boolean combination of expressions $\Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$. If we knew that the language is k-PT, this could be derived using a brute-force method and/or the $\sim_k$*-canonical DFA*, the DFA whose states are $\sim_k$ classes, cf. Fact 1. Indeed, the lower the k, the lower the complexity. An upper bound on k is given by the depth of the minimal DFA [21]. However, we show later that the minimal k can be exponentially smaller than the depth of the DFA. Note that the number of states of the $\sim_k$-canonical DFA has recently been investigated in [19] and the literature therein.

Applications of PT Languages Piecewise testable languages are of interest in many disciplines of mathematics and computer science. For instance, in semigroup theory [1, 2, 26], since they possess interesting algebraic properties, namely, the syntactic monoid of a PT language is $\mathcal{J}$-trivial, where $\mathcal{J}$ is one of the Green relations; in logic over words [9, 27, 29] because of their close relation to first-order logic—piecewise testable languages can be characterized by a (two-variable) fragment of first-order logic over words, namely, they form level 1 of the Straubing-Thérien hierarchy as already depicted above; in formal languages and automata theory [8, 21, 28], since their automata are of a special simple form (they are partially ordered and confluent) and PT languages form a strict subclass of the class of star-free languages, that is, languages definable by LTL formulas; in natural language processing, since they can describe some non-local patterns [12, 30]; in learning theory, since they are identifiable from positive data in the limit [13, 22]; in XML databases [7], which is our original motivation described in detail above. The list is not comprehensive and many other interesting results concerning PT languages can be found in the literature. It is also worth mentioning that PT languages and several results have recently been generalized from word languages to tree languages [5].

We now give a brief overview on the complexity of the problem to decide whether a regular language is piecewise testable. As mentioned above, decidability was shown by Simon. In 1985, Stern showed that the problem is decidable in polynomial time for DFAs [32]. In 1991, Cho and Huynh [6] proved NL-completeness of the problem for DFAs. In 2001, Trahtman [34] improved Stern's result to obtain a quadratic algorithm. Another quadratic algorithm can be found in [21]. The problem is PSPACE-complete if the languages are represented as NFAs [18].

Our Contribution The *k-piecewise testability problem* asks whether, given a finite automaton $\mathcal{A}$, the language $L(\mathcal{A})$ is k-PT. It is easy to see that if a language is k-PT, it is also $(k+1)$-PT. Klíma and Polák [21] have shown that if the depth of a minimal DFA recognizing a PT language is k, then the language is k-PT. However, the opposite implication does not hold, that is, the depth of the minimal DFA is only an upper bound on k. To the best of our knowledge, no efficient algorithm to find the minimal k for which a PT language is k-PT nor an algorithm to decide whether a language is k-PT has been published so far.[1]

[1] Very recently, a co-NP upper bound appeared in [16] in terms of separability.

We first give a co-NP upper bound to decide whether a minimal DFA recognizes a k-PT language for a fixed k (Theorem 1), which results in an algorithm to find the minimal k that runs in the time single exponential with respect to the size of the DFA and double exponential with respect to the resulting k. We then provide a detailed complexity analysis for small k's. In particular, the problem is trivial for $k = 0$, decidable in deterministic logarithmic space for $k = 1$ (Theorem 2), and NL-complete for $k = 2, 3$ (Theorems 3 and 4). As a result, we obtain a PSPACE upper bound to decide whether an NFA recognizes a k-PT language for a fixed k (Theorem 5). Recall that it is PSPACE-complete to decide whether an NFA recognizes a PT language, and it is actually PSPACE-complete to decide whether an NFA recognizes a 0-PT language (Proposition 2).

Since the depth of the minimal DFAs plays a role as an upper bound on k, 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 7). 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. Finally, we investigate the opposite implication and show that the tight upper bound on the depth of the minimal DFA recognizing a k-PT language over an n-letter alphabet is $\binom{k+n}{k} - 1$ (Theorem 8). A relationship with Stirling cyclic numbers is also discussed.

2 Preliminaries and Definitions

We assume that the reader is familiar with automata theory. 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 $w \in \Sigma^*$, $|w|_a$ denotes the number of occurrences of letter a in w. A language over Σ is a subset of Σ^*.

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 $2^Q \times \Sigma^*$. The language *accepted* by $\mathcal{A}$ is the set $L(\mathcal{A}) = \{w \in \Sigma^* \mid I \cdot w \cap F \neq \emptyset\}$. 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 $q_n \in F$. A path is *simple* if all states of the path are pairwise different. The number of states on the longest simple path of $\mathcal{A}$ 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 *deterministic* (DFA) if $|I| = 1$ and $|q \cdot a| = 1$ for every q in Q and a in Σ. Then the transition function $\cdot$ is a map from $Q \times \Sigma$ to Q that can be extended to the domain $Q \times \Sigma^*$. 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. 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 $p < q$. Partially ordered automata are also called *acyclic automata*, see, e.g., [21].

The notion of confluent DFAs was introduced in [21]. Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a DFA and $\Gamma \subseteq \Sigma$ be a subalphabet. The DFA $\mathcal{A}$ is Γ-confluent if, for every state q in Q and every pair of words u, v in Γ^*, there exists a word w in Γ^* such that $(qu)w = (qv)w$. The DFA $\mathcal{A}$ is *confluent* if it is Γ-confluent for every subalphabet Γ. The DFA $\mathcal{A}$ is *locally confluent* if, for every state q in Q and every pair of letters a, b in Σ, there exists a word w in $\{a, b\}^*$ such that $(qa)w = (qb)w$.

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 the 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.*

A regular language is *k-piecewise testable* if it is a finite boolean combination of languages of the form $\Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$, where $0 \leq n \leq k$ and $a_i \in \Sigma$. A regular language is *piecewise testable* if it is k-piecewise testable for some $k \geq 0$. 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 [21]. 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, denoted by $v \preccurlyeq w$. For $k \geq 0$, let $sub_k(v) = \{u \in \Sigma^* \mid u \preccurlyeq v, |u| \leq k\}$. For words w_1, w_2, we define $w_1 \sim_k w_2$ if and only if $sub_k(w_1) = sub_k(w_2)$. 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.

Fact 1 ([31]) *Let L be a regular language, and let $\sim_L$ denote the Myhill congruence. A language L is k-PT if and only if $\sim_k \subseteq \sim_L$. Moreover, L is a finite union of $\sim_k$ classes.*

The theorem says that if L is k-PT, then any two k-equivalent words either both belong to L or neither does. In terms of minimal DFAs, two k-equivalent words lead the automaton to the same state.

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

1. *The language L is PT.*
2. *The minimal DFA $\mathcal{A}$ is partially ordered and (locally) confluent [21].*
3. *The minimal DFA $\mathcal{A}$ is partially ordered and satisfies the UMS property [34].*

3 Complexity of k-Piecewise Testability for DFAs

The *k-piecewise testability problem for DFAs* asks whether, given a minimal DFA $\mathcal{A}$, the language $L(\mathcal{A})$ is k-PT. We show that it belongs to co-NP, which can be used to compute the minimal k for which the language is k-PT in the time single exponential with respect to the size of $\mathcal{A}$ and double exponential with respect to the resulting k. For small k's we then provide precise complexity analyses.

Theorem 1. *The following problem belongs to co-NP:*

Name: k-PiecewiseTestability
Input: *a minimal DFA $\mathcal{A}$*
Output: Yes *if and only if $\mathcal{L}(\mathcal{A})$ is k-PT*

Proof (sketch). One first checks that the automaton $\mathcal{A}$ over Σ recognizes a PT language. If $\mathcal{L}(\mathcal{A})$ is not k-PT, then there exist two k-equivalent words w_1 and w_2. It can be shown that the length of w_1 is at most $k|\Sigma|^k$, w_1 is a subword of w_2, and w_1 and w_2 lead the automaton to two different states. In addition, it can be shown that one can choose w_2 of length at most $\mathit{depth}(\mathcal{A})$ 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 indeed of polynomial length in the size of $\mathcal{A}$ and Σ. □

If we search for the minimal k for which the language is k-PT, we can first check whether it is 0-PT. If not, we check whether it is 1-PT and so on until we find the required k. In this case, the bounds $k|\Sigma|^k$ and $k|\Sigma|^k + \mathit{depth}(\mathcal{A})$ on the length of words w_1 and w_2 that need to be investigated are exponential with respect to k. To investigate all the words up to these lengths then gives an algorithm that is exponential with respect to the size of the minimal DFA and double exponential with respect to the desired k.

Proposition 1. *Let $\mathcal{A}$ be a minimal DFA that is partially ordered and confluent. To find the minimal k for which the language $L(\mathcal{A})$ is k-PT can be done it time exponential with respect to the size of $\mathcal{A}$ and double exponential with respect to the resulting k.*

Theorem 1 gives an upper bound on the complexity to decide whether a language is k-PT for a fixed k. We now show that for $k \leq 3$, the complexity is much simpler.

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

1-*Piecewise Testability* Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal DFA. It can be shown that the language $L(\mathcal{A})$ is 1-PT if and only if (1) for every $p \in Q$ and $a \in \Sigma$, $pa = q$ implies $qa = q$, and (2) for every $p \in Q$ and $a, b \in \Sigma$, $pab = pba$. Since this property can be verified locally in the DFA, we have the following.

Theorem 2. *The problem to decide whether a minimal DFA recognizes a 1-PT language is in LOGSPACE.*

2-*Piecewise Testability* We show that the problem to decide whether a minimal DFA recognizes a 2-PT language is NL-complete. Note that this complexity coincides with the complexity to decide whether the language is PT, that is, whether there exists a k for which the language is k-PT.

Theorem 3. *The problem to decide whether a minimal DFA recognizes a 2-PT language is NL-complete.*

Proof (sketch). To show that the problem is in NL, we need the following structural characterization of 2-PT languages. Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal partially ordered and confluent DFA. The language $L(\mathcal{A})$ is 2-PT 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 NL-hardness is shown by reduction from the monotone graph accessibility problem. □

It was shown in [3] that the syntactic monoids of 1-PT languages are defined by equations $x = x^2$ and $xy = yx$, and those of 2-PT languages by equations $xyzx = xyxzx$ and $(xy)^2 = (yx)^2$. These equations can be used to achieve NL algorithms. However, our characterizations 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 general element of the monoid. It decreases the complexity of the problems. Moreover, the partial order and (local) confluency properties can be checked instead of the equation $(xy)^2 = (yx)^2$.

3-*Piecewise Testability* The equations $(xy)^3 = (yx)^3$, $xzyxvxwy = xzxyxvxwy$ and $ywxvxyzx = ywxvxyxzx$ characterize the variety of 3-PT languages [3]. Non-satisfiability of any of these equations can be check in the DFA in NL by guessing a finite number of states and the right sequences of transitions between them (in parallel, when labeled with the same labels). Thus, we have the following.

Theorem 4. *The problem to decide whether a minimal DFA recognizes a 3-PT language is NL-complete.*

k-*Piecewise Testability* Even though [4] provides a finite sequence of equations to define the k-PT languages over a fixed alphabet for any $k \geq 4$, the equations are more involved and it is not clear whether they can be used to obtain the precise complexity. So far, the k-piecewise testability problem can be shown to be NL-hard (for $k \geq 2$) and in co-NP, and it is open whether it tends rather to NL or to co-NP.[2]

[2] See the acknowledgement for the recent development.

4 Complexity of k-Piecewise Testability for NFAs

The *k-piecewise testability problem for NFAs* asks whether, given an NFA $\mathcal{A}$, the language $L(\mathcal{A})$ is k-PT. A language is 0-PT if and only if it is either empty or universal. Since the universality problem for NFAs is PSPACE-complete [14], the 0-PT problem for NFAs is PSPACE-complete. Using the same argument as in [18] then gives us the following result.

Proposition 2. *For every integer $k \geq 0$, the problem to decide whether an NFA recognizes a k-PT language is PSPACE-hard.*

Since k is fixed, we can make use of the idea of Theorem 1 to decide whether an NFA recognizes a k-PT language. The length of the word w_2 is now bounded by 2^n, where n is the number of states of the NFA. Guessing the word w_2 on-the-fly then gives that the k-piecewise testability problem for NFAs is in PSPACE.

Theorem 5. *The following problem is PSPACE-complete:*

Name: k-PiecewiseTestabilityNFA
Input: *an NFA $\mathcal{A}$*
Output: Yes *if and only if $\mathcal{L}(\mathcal{A})$ is k-PT*

The problem to find the minimal k for which the language recognized by an NFA is k-PT is PSPACE-hard, since a language is PT if and only if there exists a minimal $k \geq 0$ for which it is PT.

5 Piecewise Testability and the Depth of NFAs

In this section, we generalize a result valid for DFAs to NFAs and investigate the relationship between the depth of an NFA and the minimal k for which its language is k-PT. We show that the upper bound on k given by the depth of the minimal DFA can be exponentially far from such a minimal k. More specifically, we show that for every $k \geq 0$, there exists a k-PT language L recognized by an NFA $\mathcal{A}$ of depth $k-1$ and by the minimal DFA $\mathcal{D}$ of depth $2^k - 1$.

Recall that a regular language is PT if and only if its minimal DFA satisfies some properties that can be tested in a quadratic time, cf. Fact 2. We now show that this characterization generalizes to NFAs. We say that an NFA $\mathcal{A}$ over an alphabet Σ 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.

Theorem 6. *A regular language is PT if and only if there exists a complete NFA that is partially ordered and satisfies the UMS property.*

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.

5.1 Exponential Gap between k and the Depth of DFAs

It was shown in [21] 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 7. *For every $n \geq 2$, 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 (sketch). For $k \geq 0$, let $\mathcal{A}_k = (I_k, \{a_0, a_1, \dots, a_k\}, \cdot, I_k, \{0\})$ be an NFA with $I_k = \{0, 1, \dots, k\}$ and the transition function consisting of the self-loops under a_i in all states $j > i$ and transitions under a_i from the state i to all states $j < i$ as depicted in Fig. 1.

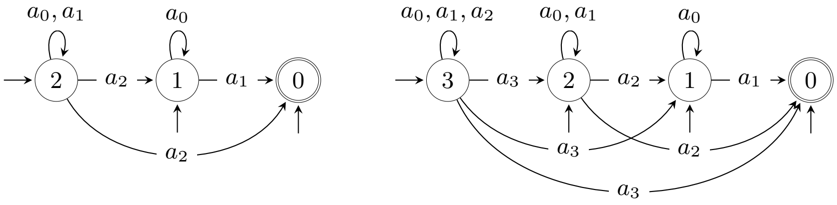

Fig. 1. Automata $\mathcal{A}_2$ and $\mathcal{A}_3$.

Every NFA $\mathcal{A}_k$ has depth k. Using Theorem 6 or noticing that the reversed automata are deterministic, we can show that it accepts a $(k+1)$-PT language. It can be shown that the language is not k-PT and that its minimal DFA has depth $2^{k+1} - 1$. □

Although it is well known that DFAs can be exponentially larger than NFAs, an interesting by-product of this result is that there are NFAs such that all the exponential number of states of their minimal DFAs form a simple path.

It could seem that NFAs are more convenient to provide upper bounds on the k. 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 is provided in the next section.

6 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 5 shows that the language can be $(\ell - i)$-PT for exponentially large i's. Therefore, we study the opposite implication of the relationship between k-piecewise testability and the depth of the minimal DFA in this section. 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 first investigate the following problem.

Problem 1. 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, $sub_k(w_1) \neq sub_k(w_2)$?

The answer to this question is formulated in the following proposition.

Proposition 3. *Let Σ be an alphabet of cardinality n. The length of a longest word, w, satisfying the requirements of Problem 1 is given by the recursive formula $|w| = P_{k,n} = P_{k-1,n} + P_{k,n-1} + 1$, where $P_{1,m} = m = P_{m,1}$, for $m \geq 1$.*

It follows by induction that for any positive integers k and n

$$P_{k,n} = \binom{k+n}{k} - 1\,.$$

We now use this result to show that the depth of the minimal DFA recognizing a k-PT language over an n-letter alphabet is $P_{k,n}$ in the worst case.

Theorem 8. *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 $P_{k,n}$. Moreover, the bound is tight for any k and n.*

A few of these numbers are listed in Table 1. We now present several consequences of these results.

1. Note that it follows from the formula that $P_{k,n} = P_{n,k}$. This gives and interesting observation that increasing the length of the considered subwords has exactly the same effect as increasing the size of the alphabet.
2. Equivalently stated, Problem 1 asks what is the depth of the $\sim_k$-canonical DFA, whose states are $\sim_k$ classes. The number of equivalence classes of $\sim_k$, i.e., the number of states, has recently been investigated in [19].

	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. The table of a few first numbers $P_{k,n}$

3. It provides a precise bound on the length of w_1 of Theorem 1. However, it does not improve the statement of the theorem.

To provide a relationship of $P_{k,n}$ with Stirling cyclic numbers, it can be shown that, for any positive integers k and n,

$$P_{k,n} = \frac{1}{k!} \sum_{i=1}^{k} \begin{bmatrix} k+1 \\ i+1 \end{bmatrix} n^i$$

where $\begin{bmatrix} k \\ n \end{bmatrix}$ denotes the Stirling cyclic numbers.

Finally, note that one could also see a noticeable relation between the columns (resp. rows) of Table 1 and the generalized Catalan numbers of [11]. We leave the details of this correspondence for a future investigation.

Acknowledgements. We would like to thank an anonymous reviewer for informing us about the unpublished manuscript [20] and its authors for providing it. It turns out that we have independently obtained two results—the bound of Theorem 8 and the co-NP bound on the k-PT problem for DFAs. Furthermore, it is shown in [20] that the k-PT problem is co-NP-complete for $k \geq 4$. It also provides a smaller bound on the length of the witnesses, the consequence of which is a single exponential algorithm to find the minimal k. On the other hand, for $k \leq 3$, that paper only says that the k-PT problem belongs to P.

The authors are grateful to Sebastian Rudolph for a fruitful discussion.

References

1. Almeida, J., Costa, J.C., Zeitoun, M.: Pointlike sets with respect to R and J. J. Pure Appl. Algebra 212(3), 486–499 (2008)
2. Almeida, J., Zeitoun, M.: The pseudovariety J is hyperdecidable. Theor. Inform. Appl. 31(5), 457–482 (1997)
3. Blanchet-Sadri, F.: Games, equations and the dot-depth hierarchy. Comput. Math. Appl. 18(9), 809–822 (1989)
4. Blanchet-Sadri, F.: Equations and monoid varieties of dot-depth one and two. Theoret. Comput. Sci. 123(2), 239–258 (1994)
5. Bojanczyk, M., Segoufin, L., Straubing, H.: Piecewise testable tree languages. LMCS 8(3) (2012)
6. Cho, S., Huynh, D.T.: Finite-automaton aperiodicity is PSPACE-complete. Theor. Comput. Sci. 88(1), 99–116 (1991)
7. Czerwiński, W., Martens, W., Masopust, T.: Efficient separability of regular languages by subsequences and suffixes. In: ICALP. LNCS, vol. 7966, pp. 150–161. Springer (2013)
8. Czerwiński, W., Martens, W.: A note on decidable separability by piecewise testable languages. CoRR abs/1410.1042 (2014)
9. Diekert, V., Gastin, P., Kufleitner, M.: A survey on small fragments of first-order logic over finite words. Internat. J. Found. Comput. Sci. 19(3), 513–548 (2008)
10. Ellul, K., Krawetz, B., Shallit, J., Wang, M.: Regular expressions: New results and open problems. J. Autom. Lang. Comb. 10(4), 407–437 (2005)

11. Frey, D.D., Sellers, J.A.: Generalizing Bailey's generalization of the Catalan numbers. Fibonacci Quarterly 39(2), 142–148 (2001)
12. Fu, J., Heinz, J., Tanner, H.: An algebraic characterization of strictly piecewise languages. In: TAMC, LNCS, vol. 6648, pp. 252–263. Springer (2011)
13. García, P., Ruiz, J.: Learning k-testable and k-piecewise testable languages from positive data. Grammars 7, 125–140 (2004)
14. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
15. Gelade, W., Neven, F.: Succinctness of pattern-based schema languages for XML. J. Comput. Syst. Sci. 77(3), 505–519 (2011)
16. Hofman, P., Martens, W.: Separability by short subsequences and subwords. In: ICDT. LIPIcs, vol. 31, pp. 230–246 (2015)
17. Holub, Š., Jirásková, G., Masopust, T.: On upper and lower bounds on the length of alternating towers. In: MFCS. LNCS, vol. 8634, pp. 315–326. Springer (2014)
18. Holub, Š., Masopust, T., Thomazo, M.: Alternating towers and piecewise testable separators. CoRR abs/1409.3943 (2014), http://arxiv.org/abs/1409.3943
19. Karandikar, P., Kufleitner, M., Schnoebelen, P.: On the index of Simon's congruence for piecewise testability. Inform. Process. Lett. 115(4), 515–519 (2015)
20. Klíma, O., Kunc, M., Polák, L.: Deciding k-piecewise testability, manuscript
21. Klíma, O., Polák, L.: Alternative automata characterization of piecewise testable languages. In: DLT. LNCS, vol. 7907, pp. 289–300. Springer (2013)
22. Kontorovich, L.A., Cortes, C., Mohri, M.: Kernel methods for learning languages. Theor. Comput. Sci. 405(3), 223–236 (2008)
23. Kufleitner, M., Lauser, A.: Around dot-depth one. Internat. J. Found. Comput. Sci. 23(6), 1323–1340 (2012)
24. Martens, W., Neven, F., Niewerth, M., Schwentick, T.: Developing and analyzing XSDs through BonXai. PVLDB 5(12), 1994–1997 (2012)
25. Martens, W., Neven, F., Schwentick, T., Bex, G.: Expressiveness and complexity of XML Schema. ACM T. Database Syst. 31(3), 770–813 (2006)
26. Perrin, D., Pin, J.E.: Infinite words: Automata, semigroups, logic and games. Pure and Applied Mathematics, vol. 141, pp. 133–185. Elsevier (2004)
27. Place, T., Zeitoun, M.: Going higher in the first-order quantifier alternation hierarchy on words. In: ICALP. LNCS, vol. 8573, pp. 342–353. Springer (2014)
28. Place, T., van Rooijen, L., Zeitoun, M.: Separating regular languages by piecewise testable and unambiguous languages. In: MFCS. LNCS, vol. 8087, pp. 729–740. Springer (2013)
29. Place, T., Zeitoun, M.: Separating regular languages with first-order logic. In: CSL/LICS. pp. 75:1–75:10. ACM (2014)
30. Rogers, J., Heinz, J., Bailey, G., Edlefsen, M., Visscher, M., Wellcome, D., Wibel, S.: On languages piecewise testable in the strict sense. In: MOL. LNAI, vol. 6149, pp. 255–265. Springer (2010)
31. Simon, I.: Hierarchies of Events with Dot-Depth One. Ph.D. thesis, Department of Applied Analysis and Computer Science, University of Waterloo, Canada (1972)
32. Stern, J.: Complexity of some problems from the theory of automata. Inform. Control 66(3), 163–176 (1985)
33. Stockmeyer, L.J., Meyer, A.R.: Word problems requiring exponential time: Preliminary report. In: STOC. pp. 1–9. ACM (1973)
34. Trahtman, A.N.: Piecewise and local threshold testability of DFA. In: FCT. LNCS, vol. 2138, pp. 347–358. Springer (2001)