On k-piecewise testability
(preliminary report)

Tomáš Masopust¹, Michaël Thomazo²
TU Dresden, Germany

Abstract

A language is k-piecewise testable if it is a finite boolean combination of languages of the form $\Sigma^*a_1\Sigma^*\cdots\Sigma^*a_n\Sigma^*$, where $a_i \in \Sigma$ and $0 \leq n \leq k$. We investigate the problem, given a minimal DFA recognizing a piecewise testable language, what is the minimal $k$ for which the language is $k$-piecewise testable? It was shown by Klíma and Poláč that such a $k$ is bounded by the depth of the minimal DFA. We first provide the complexity bound to decide whether a given minimal DFA represents a $k$-piecewise testable language for a fixed $k$, which then results in an algorithm that is single exponential with respect to the size of the DFA and double exponential with respect to the resulting $k$. We provide a detailed complexity analysis for $k \leq 2$. Finally, we generalize a result valid for DFAs to NFAs and show that the upper bound given by the depth of the minimal DFA can be exponentially far from the minimal $k$.

1. Introduction

A regular language is piecewise testable if it is a finite boolean combination of languages of the form

$$\Sigma^*a_1\Sigma^*\cdots\Sigma^*a_n\Sigma^*$$

where $a_i \in \Sigma$ and $n \geq 0$. These languages were introduced by Simon in his PhD thesis [25], see also [26]. Simon proved that piecewise testable languages are exactly those regular languages whose syntactic monoid is $J$-trivial. He also provided various characterizations of piecewise testable languages in terms of monoids, automata, etc.

Piecewise testable languages are of interest in mathematics and computer science, for instance in algebra [1,2,19], since their syntactic monoids are of some special form ($J$-trivial), in logic [6,21,22], since they can be characterized by a two-variable fragment of first-order logic (boolean combinations of the lowest level of the quantifier alternation hierarchy of FO), in formal languages and automata theory [5,11,15,14,20], since their automata are of a special simple form and piecewise testable languages form a strict subclass of the class of star-free languages, in natural language [23,8], since they can describe some non-local patterns, in learning theory [9,16], since they are identifiable from positive data in the limit, or in XML databases [6].

We now give a brief overview of 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 piecewise testability of regular languages given by DFAs is decidable in polynomial time [27]. In 1991, Cho and Huyhn [4] proved that to decide whether a minimal DFA recognizes a piecewise testable language is NL-complete. In 2001, Trahtman [29] improved Stern’s algorithm to obtain a quadratic-time algorithm to decide piecewise testability for deterministic finite automata. Another quadratic-time algorithm can be found in [15].

A piecewise testable 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 $0 \leq n \leq k$. The $k$-piecewise testability problem asks whether, given a finite automaton $\mathcal{A}$, the language $L(\mathcal{A})$ is $k$-piecewise testable. Note that if a language is $k$-piecewise testable, it is also

¹Research supported by the DFG in grant KR 4381/1-1
²Research supported by the Alexander von Humboldt Foundation

Email addresses: tomas.masopust@tu-dresden.de (Tomáš Masopust), michael.thomazo@tu-dresden.de (Michaël Thomazo)
(k + 1)-piecewise testable. In [15], it was shown that, given a piecewise testable language, the language is k-piecewise testable for k equal to the depth of the minimal DFA. To the best of our knowledge, however, no efficient algorithm to find the minimal k for which a piecewise testable language is k-piecewise testable nor an algorithm to decide whether a language is k-piecewise testable is known in the literature. Note that by the result of [15], the depth of the minimal DFA is the upper bound on such a k.

In this paper, we investigate the following problem. Given a minimal DFA recognizing a piecewise testable language. What is the minimal k for which the language is k-piecewise testable? Note that we focus only on minimal DFAs, since piecewise testability for NFAs is PSPACE-complete [12]. It is PSPACE-complete even to decide whether an NFA recognizes a 0-piecewise testable language (Theorem 10). We first provide a co-NP bound to decide whether a given minimal DFA represents a k-piecewise testable language for a fixed k (Theorem 5), which then results in an algorithm that is single exponential with respect to the size of the DFA and double exponential with respect to the resulting k. We also provide a detailed complexity analysis for small k. In particular, we show that to decide whether a given minimal DFA recognizes a k-piecewise testable language is trivial for k = 0 (Theorem 7), decidable in logarithmic space for k = 1 (Theorem 8), and NL-complete for k = 2 (Theorem 12). Finally, we generalize a result valid for DFAs to NFAs and use it to investigate the relationship between the depth of an NFA and the k-piecewise testability problem. We show that the upper bound on k given by the depth of the minimal DFA can be exponentially far from being precise, that is, for every k ≥ 1, there exists a k-piecewise testable language with an NFA of depth 2^k − 1 and with the minimal DFA of depth 2^k − 1 (Theorem 21).

2. Preliminaries

We assume that the reader is familiar with automata and formal language theory [17]. 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 ∈ Σ^*, alph(w) ⊆ Σ denotes the set of all letters occurring in w; and |w| denotes the number of occurrences of letter a in w. A language over Σ is a subset of Σ^*.

Automata. A nondeterministic finite automaton (NFA) is a quintuple A = (Q, Σ, ·, I, F), where Q is the finite nonempty set of states, Σ is the input alphabet, I ⊆ Q is the set of initial states, F ⊆ Q is the set of accepting states, and · : Q × Σ → 2^Q is the transition function that can be extended to the domain 2^Q × Σ^*. The language accepted by A is the set L(A) = {w ∈ Σ^* | I · w ∩ F ≠ Ø}. We usually omit · and write simply Iw instead of I · w.

A path π from a state q₀ to a state qₙ under a word a₁a₂ · · · aₙ, for some n ≥ 0, is a sequence of states and input symbols q₀a₁q₁a₂ · · · qₙ₋₁aₙqₙ such that qᵢ₊₁ ∈ qᵢ · aᵢ₊₁, for all i = 0, 1, ..., n − 1. The path π is accepting if q₀ ∈ I and qₙ ∈ F. We use the notation q₀ →ₚ qₙ to denote that there exists a path from q₀ to qₙ under the word a₁a₂ · · · aₙ.

A path is simple if all states of the path are pairwise different. The number of states on the longest simple path of A decreased by one (or, equivalently, the number of input symbols/transitions on the longest simple path) is called the depth of the automaton A, denoted by depth(A). Note that the depth is bounded by the number of states minus one.

The NFA A is deterministic (DFA) if |I| = 1 and |q · a| = 1 for every q ∈ Q and a ∈ Σ. The transition function · is a map from Q × Σ to Q that can be extended to the domain Q × Σ^*. 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.

Partially ordered automata. Let A = (Q, Σ, ·, I, F) be an NFA. The reachability relation ≤ on the set of states Q is defined by p ≤ q if there exists a word w in Σ^* such that q ∈ p · w. The NFA A is partially ordered if the reachability relation ≤ is a partial order. For two states p and q of A, we write p < q if p ≤ q and p ≠ 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., [15].

Confluence. The notion of (locally) confluent DFAs has been introduced in [15]. Let A = (Q, Σ, ·, I, F) be a DFA and Γ ⊆ Σ be a subalphabet. The DFA 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 A is confluent if it is Γ-confluent for every sub-alphabet Γ. The DFA 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.
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 property (UMS property).

Piecewise testable languages. A regular language is $k$-piecewise testable if it is a finite boolean combination of languages of the form $\Sigma^*a_1\Sigma^*a_2\Sigma^*\cdots a_k\Sigma^*$, where $0 \leq n \leq k$ and $a_i \in \Sigma$. A language is piecewise testable if it is $k$-piecewise testable for some $k \geq 0$.

We adopt the notation $L_{a_1a_2\cdots a_k} = \Sigma^*a_1\Sigma^*a_2\Sigma^*\cdots a_k\Sigma^*$ from [13]. For two words $v = a_1a_2\cdots a_n$, and $w \in L_{a_1a_2\cdots a_n}$, we say that $v$ is a subsequence of $w$ or that $v$ can be embedded into $w$, denoted by $v \preceq w$. For $k \geq 0$, let $\text{sub}_k(v) = \{u \in \Sigma^* | u \preceq v, |u| \leq k\}$. Let $w_1$, $w_2$ be two words, and let $w_1 \sim_k w_2$ if and only if $\text{sub}_k(w_1) = \text{sub}_k(w_2)$. We say that $w_1$ and $w_2$ are $k$-equivalent if $w_1 \sim_k w_2$. Note that $\sim_k$ is a congruence.

Theorem 1 ([13]). Let $L$ be a regular language, and let $\sim_L$ denote the Myhill congruence [13]. A language $L$ is $k$-piecewise testable if and only if $\sim_k \subseteq \sim_L$. \hfill $\Box$

In other words, the theorem says that whenever the language is $k$-piecewise testable, then any two words $w_1$ and $w_2$ such that $w_1 \sim_k w_2$ either both belong to the language or neither does. Equivalently, if the language is given by the minimal DFA, then any two $k$-equivalent words lead the automaton to the same state.

Theorem 2 ([18, 29]). Let $L$ be a language recognized by the minimal DFA $\mathcal{A}$. The following is equivalent.

- The language $L$ is piecewise testable.
- The minimal DFA $\mathcal{A}$ is partially ordered and (locally) confluent.
- The minimal DFA $\mathcal{A}$ is partially ordered and satisfies the UMS property. \hfill $\Box$

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 $k$-piecewise testable. We first show that the problem belongs to co-NP. Then we use this result to obtain an algorithm to compute the minimal $k$ for which the language is $k$-piecewise testable that is single exponential with respect to the size of the automaton and double exponential with respect to the resulting $k$. For small $k$’s we then provide precise complexity analyses.

3.1. $k$-piecewise testability

To show that, for any $k$, the problem to decide whether a language represented by a minimal DFA is $k$-piecewise testable belongs to co-NP, we need the following two lemmas that give upper bounds on the length of words that disprove $k$-piecewise testability.

Let $w_1$ and $w_2$ be two words such that $w_1 \preceq w_2$. Let $\varphi : \{1, 2, \ldots, |w_1|\} \rightarrow \{1, 2, \ldots, |w_2|\}$ be a monotonically increasing mapping induced by one of the possible embeddings of $w_1$ into $w_2$, that is, the letter at the $j$th position in $w_1$ coincides with the letter at the $\varphi(j)$th position in $w_2$. Any such $\varphi$ 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 $\varphi$, we mean an occurrence of $a$ in $w_2$ whose position does not belong to the range of $\varphi$.

Lemma 3. Let $\mathcal{A}$ be a minimal DFA recognizing a piecewise testable language. If there exist two words $w_1$ and $w_2$ that are $k$-equivalent and lead to two different states from the initial state, 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 depth($\mathcal{A}$) more letters than $w_1$. 

3
Proof. Let us consider \( w_1 \) and \( w_2 \) as in the statement of the lemma. Let \( \varphi \) 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 \( \varphi \). Let us denote \( w_2 = w_\varphi a w_{\varphi}^a \). If \( i w_\varphi a = i w_\varphi \), then \( i w_\varphi w_{\varphi}^a = i w_2 \). Moreover, since \( a \notin \text{range}(\varphi) \), \( w_1 \) is a subword of \( w_\varphi w_{\varphi}^a \). Thus, \( \text{sub}_k(w_1) \subseteq \text{sub}_k(w_\varphi w_{\varphi}^a) \subseteq \text{sub}_k(w_2) \), which proves that \( w_1 \) and \( w_\varphi w_{\varphi}^a \) 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 \( w_1 \) and that do not trigger a change of state in \( A \), 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. Since in a run of an acyclic automaton there are at most \( \text{depth}(A) \) changes of states, this concludes the proof. \( \square \)

Lemma 4. Let \( A \) be a minimal DFA recognizing a piecewise testable language. If \( L(A) \) is not \( k \)-piecewise testable, there exist two words \( w_1 \) and \( w_2 \) such that:

- \( w_1 \) and \( w_2 \) are \( k \)-equivalent;
- the length of \( w_1 \) is at most \( k|\Sigma|^k \);
- \( w_1 \) is a subword of \( w_2 \);
- \( w_1 \) and \( w_2 \) lead to two different states from the initial state.

Proof. If \( L(A) \) is not \( k \)-piecewise testable, then there exist \( w_1 \) and \( w_2 \) that are \( k \)-equivalent and lead to two different states from the initial state. Let us 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 us denote \( w_i^j \) the prefix of \( w_i \) of length \( k \). Let us assume that there exists \( j \) such that \( \text{sub}_k(w_i^j) = \text{sub}_k(w_i^{j+1}) \). Then the letter at the \( (j+1)^{th} \) 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. Moreover, since \( \text{sub}_k(w_i^j) \subseteq \text{sub}_k(w_i^{j+1}) \), such a \( w_i^* \) contains at most \( \sum_{n=1}^k |\Sigma|^n = \frac{|\Sigma|^{n+1} - 1}{|\Sigma| - 1} \leq k|\Sigma|^k \) letters.

To complete the proof, 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. Let us then consider \( w^* \) that is such that \( w^* \sim_k w_1^* \), and both \( w_1^* \) and \( w_2^* \) are subwords of \( w^* \). Without loss of generality, \( w_1^* \) and \( w^* \) fulfill the required conditions. \( \square \)

We now apply the previous lemmas to prove the following complexity bound.

Theorem 5. The following problem belongs to co-NP:

- **NAME:** \( k \)-PiecewiseTestability
- **INPUT:** a minimal DFA \( A \)
- **OUTPUT:** Yes if and only if \( L(A) \) is \( k \)-piecewise testable.

Proof. One can first check that the automaton \( A \) over \( \Sigma \) recognizes a piecewise testable language. Then, by Lemma 4 if \( L(A) \) is not \( k \)-piecewise testable, 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 Lemma 3 one can choose \( w_2 \) of length at most \( \text{depth}(A) \) 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 \( A \) and \( \Sigma \).

Remark 6. If we search for the minimal \( k \) for which the language is \( k \)-piecewise testable, we can first check whether it is \( 0 \)-piecewise testable. If not, we check whether it is \( 1 \)-piecewise testable and so on until we find the required \( k \). In this case, the bounds \( k|\Sigma|^k \) and \( k|\Sigma|^k + \text{depth}(A) \) 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 \( k \).

Theorem 5 gives an upper bound on the complexity to decide whether a language is \( k \)-piecewise testable for a fixed \( k \). In the following three subsections, we show that for \( k \leq 2 \), the complexity of the problem is much simpler. However, for \( k \geq 3 \), the precise complexity is not known.
3.2. 0-piecewise testability

Given a minimal DFA $\mathcal{A}$ over an alphabet $\Sigma$. The language $L(\mathcal{A})$ is 0-piecewise testable if and only if it has a single state, that is, it recognizes either $\Sigma^*$ or $\emptyset$.

Theorem 7. Given a minimal DFA. It is decidable in $O(1)$ whether its language is 0-piecewise testable.

3.3. 1-piecewise testability

Theorem 1 says that a language $L$ is 1-piecewise testable if and only if for every two words $u, v$ in $\Sigma^*$, $\text{alph}(u) = \text{alph}(v)$ implies that $u \in L$ if and only if $v \in L$. We now provide a characterization of 1-piecewise testable languages that can be verified locally in the DFA.

Theorem 8. Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal DFA. The language $L(\mathcal{A})$ is 1-piecewise testable if and only if both of the following holds:

1. for every $p \in Q$ and $a \in \Sigma$, $pa = q$ implies $qa = q$.
2. for every $p \in Q$ and $a, b \in \Sigma$, $pab = pba$.

Proof. We show successively both directions of the equivalence.

$(\Rightarrow)$ Assume that $L(\mathcal{A})$ is 1-piecewise testable. Since $\mathcal{A}$ is minimal, $p$ is reachable. Thus, there exists $w$ such that $iw = p$. It holds that $\text{alph}(w_1) = \text{alph}(w_2a)$, thus $w_1$ and $w_2a$ lead to the same state, that is, $qa = q$. Similarly, we notice that $\text{alph}(wab) = \text{alph}(whb)$, and thus $pab = pba$.

$(\Leftarrow)$ We show that for any word $w$, it holds that $iw = i(a_1a_2 \ldots a_n)$, where $\text{alph}(w) = \{a_1, a_2, \ldots, a_n\}$. This then proves that if $w_1 \sim w_2$, then $iw_1 = iw_2$. Thus, since for any letters $a, b \in \Sigma$ and any state $q$, $qab = qba$, we have that $iw = i(a_1^{k_1}a_2^{k_2} \ldots a_n^{k_n})$, where $k_i$ is the number of appearances of $a_i$ in $w$. By assumption 1 and induction on $k_1 \geq 1$, $i(a_1^{k_1}) = i(a_1)$. By induction on $n$, we thus show that $iw = i(a_1) = a_2 = \ldots = a_n$.

We immediately have the following consequences.

Corollary 9. Given a minimal DFA. To decide whether the DFA recognizes a 1-piecewise testable language is in LOGSPACE.

Corollary 10. If a minimal DFA over $\Sigma$ has more than $2^{|\Sigma|}$ states, then it does not recognize a 1-piecewise testable language.

3.4. 2-piecewise testability

In this section, we show that the problem to decide whether a minimal DFA recognizes a 2-piecewise testable language is NL-complete.

We first need the following general lemma that states that for any two $k$-equivalent words that lead the automaton to two different states, there exist other two equivalent words leading the automaton to two different states, such that one word is a subword of the other and the words differ only by a single letter.

Lemma 11. Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal partially ordered and confluent DFA. For any $k \geq 0$, 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 $w_1$ and $w_2$ be two words such that $w_1 \sim_k w_2$ and $iw_1 = iw_2$. Then, by [24, Theorem 6.2.6], there exists a word $w_3$ such that $w_1$ and $w_2$ are subwords of $w_3$, and $w_1 \sim_k w_2 \sim_k w_3$. Indeed, either $w_1$ and $w_3$, or $w_2$ and $w_3$, do not lead to the same state. Let $v, v' \in \{w_1, w_2, w_3\}$ be such that $v$ is a subword of $v'$ and $iv \neq iv'$. Let $v = u_0, u_1, \ldots, u_n = v'$ be a sequence such that $u_{i+1}$ is obtained from $u_i$ by adding a letter at some place. Such a sequence exists since $v$ is a subword of $v'$. If for every $i$, $v_i$ and $v_{i+1}$ lead to the same state, then $v$ and $v'$ does as well. Thus, there must exist $i$ such that the words $v_i$ and $v_{i+1}$ lead to two different states and $v_i$ is obtained from $v_{i+1}$ by adding a letter at some place. Setting $w = v_i$ and $w' = v_{i+1}$ completes the proof, since $\sub_{k}(w) \subseteq \sub_{k}(w) \subseteq \sub_{k}(w') \subseteq \sub_{k}(w') = \sub_{k}(v')$.

We can now prove the following characterization of 2-piecewise testable languages.
Theorem 12. Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal partially ordered and confluent DFA. The language $L(\mathcal{A})$ is 2-piecewise testable if and only if for every $a \in \Sigma$ and every states $p, q$ such that $pa = q$, $quaw_a = quaw_a$, for every $u_1, u_2 \in \Sigma^*$.

Proof. ($\Rightarrow$) By contrapositive. Assume that there exists $u \in \Sigma^*$ and a state $p$ such that $iw = p$ for some $w \in \Sigma^*$ containing $a$ and such that $pia \neq pawa$. By the assumption, $w = w_1aw_2$, for some $w_1, w_2 \in \Sigma^*$ such that $a \notin \text{alph}(w_1)$, and we want to show that $w_1aw_2au \sim w_1aw_2aa$. However, for any $x \in \text{alph}(w_1aw_2)$, if $xa \neq w_1aw_2au$, then $xa \leq w_1aw_2ua$, since $a \in \text{alph}(w)$. Similarly for $y \in \text{alph}(au)$ and $ay \leq w_1aw_2aa$. Since $i \cdot wua \neq i \cdot waa$, the minimality of $\mathcal{A}$ gives that there exists a word $v$ such that $wuv \in L(\mathcal{A})$ if and only if $wuv \notin L(\mathcal{A})$. Since $\sim$ is a congruence, $wuv \sim wuv$, which violates Theorem[11] hence $L(\mathcal{A})$ is not 2-piecewise testable.

($\Leftarrow$) Let $w_1$ and $w_2$ be two words such that $w_1 \sim w_2$. We want to show that $iw_1 = iw_2$. By Lemma[11] it is sufficient to show this direction of the theorem 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
\[
w = a_1 \ldots a_ka_{k+1} \ldots a_n \quad \text{and} \quad w' = a_1 \ldots a_ka_{k+1} \ldots a_n\]
for $0 \leq k \leq n$. Let $w_{ij} = a_i a_{i+1} \ldots a_j$. We distinguish two cases.

(A) Assume that $a$ does not appear in $w_{k+1,\ell}$. Then a must appear in $w_{k+1,\ell}$. Consider the first occurrence of $a$ in $w_{k+1,\ell}$. Then $w_{k+1} = au_{a_1}$. Let $B = \text{alph}(u_{a_1})$. Then $B \subseteq \text{alph}(u_{a_2})$, because if there is no $a$ in $w_{k,1}$, any subword $ax$, for $x \in B$, that appears in $w_{k,1}au_{a_2}$ must also appear in the subword $au_{a_2}$ of $w = w_{k,1}au_{a_2}$.

Let $u_{a_2} = b_1b_2 \ldots b_{x_j+1}$, where $B = \{b_1, b_2, \ldots, b_j\}$ and $b_j$ does not appear in $b_1b_2 \ldots b_{x_j}$. Let $z \in \{\cdot w_1, w_2a\}$. We prove (by induction on $j$) that for every $j = 1, 2, \ldots, \ell$, there exists a word $y$, such that $z \cdot (b_1b_2 \ldots b_j)^{y_j} = z \cdot x_1b_1b_2 \ldots b_{x_j+1}$. Since $b_j$ appears in $u_{a_1}$, we use the assumption from the statement of the theorem to obtain $(z \cdot x_1b_1) \cdot x_2 = (z \cdot b_1x_1b_1) \cdot x_2$. Assume that it holds for $j < k$. We prove it for $j + 1$. Again, $b_{j+1}$ appears in $u_{a_1}$ implies that
\[
z \cdot x_1b_1b_2 \ldots b_{x_j+1} = ((z \cdot x_1b_1b_2 \ldots b_{x_j+1})x_{j+1})x_{j+2} = (z \cdot b_{j+1}y_{j+1})x_{j+2} = z \cdot b_{j+1}y_{j+1} \cdot x_{j+2}
\]
where the second equality is by the induction hypothesis and the third is by the assumption from the statement of the theorem applied to the underlined part. Thus, in particular, there exists a word $y$ such that $i \cdot w_{1,1}au_{a_2}^{y_j} = i \cdot w$ and $i \cdot w_{1,1}au_{a_2}^{y_j} = i \cdot w'$. Finally, let $z_1 = i \cdot w_1au_{a}a$ and $z_2 = i \cdot w_1au_{a}a$. We prove that $z_1 \cdot v^k = z_2 \cdot v^k$, 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 1 (Commutativity). For every $a, b \in \Sigma$ and every states $p$ 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 theorem, since $a$ appears in $w$, $p \cdot ba = p \cdot aba = q_1$. Similarly, since $b$ appears in $w$, we also have $p \cdot ab = p \cdot bab = q_2$. 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, $q_1 = q_2$. \hfill \(\diamondsuit\)

We can now finish the proof by induction on the length of $v^k = b_1b_2b_1$ by showing that the state $z' = z_1 \cdot b_1 \ldots b_2b_1$ has self-loops under $B$, $i = 1, 2$. Let $z_1 \overset{b_{c_1}, b_{c_2}}{\longrightarrow} z_2 = q_i \cdot b_1q_i \cdot b_{c_1-1}q_i \cdot b_{c_1-1} \ldots q_i \cdot b_1q_i_1$ denote the path defined by the word $\Gamma$ from the state $z_i, i = 1, 2$.

Claim 2. Both states $z'_1$ and $z'_2$ have self-loops under all letters of the alphabet $B$.

Proof. Indeed, $q_{i,j} \cdot b_j = q_{i,j+1} \cdot b_j = q_{i,j+1} \cdot b_j$, where the last equality is by the assumption from the statement of the theorem, since $b_j$ appears in $u_{a_1}$. Thus, we have $z'_i = z_i \cdot b_j = z'_i$. Now, for every $j = 2, \ldots, \ell$, we have $z'_i = q_{i,j} \cdot b_{j-1} \ldots b_2b_1 = q_{i,j} \cdot b_1b_{j-1} \ldots b_2b_1 = q_{i,j} \cdot b_{j-1} \ldots b_2b_1 = z'_j$, where the third equality is because there is a self-loop in $q_{i,j}$ under $b_j$, the fourth is by several applications of commutativity (Claim[11] above), and the fifth is by the assumption from the statement of the theorem. \hfill \(\diamondsuit\)
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 $B$, confluence 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 $w_{1,k}$. Then $ca$ is a subword of $w_{1,k}$. Let $w_{1,k} = w_3a_w$, where $a$ does not appear in $w_3$. Let $q = i \cdot w_3a$, and let $B = \text{alph}(w_3)$. Note that $B \subseteq \text{alph}(w_3)$, since if $xa$ is a subword of $w_{1,k}$, then it is also in $w_3a$. By the assumption of the theorem, $q = i \cdot w_3a = i \cdot w_3aa$, hence we get that there is a self-loop in $q$ under $a$. Now, by the self-loop under a in $q$ and commutativity (Claim[1] above), $q \cdot w_4 = q \cdot aw_4 = q \cdot w_4a$. Thus, $i \cdot w_{1,k} = i \cdot w_{1,k}a$.

Second, assume that there exists $c$ in $w_{1,k}$ such that $ca \subsetneq w_{1,k}$. Then $a$ must appear in $w_{k+1,k}$. Together, there exist $i \leq k < j$ such that $a_i = a_j = a$. By the assumption of the theorem, we obtain that $i \cdot w_{1,k}w_{k+1,j} = i \cdot w_{1,k}w_{k+1,j}$, 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.

This result gives a PTIME algorithm to decide whether a minimal DFA recognizes a 2-piecewise testable language.

**Corollary 13.** Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal DFA. There exists an $O(|\Sigma| \cdot |Q|^3)$-time algorithm to decide whether the automaton $\mathcal{A}$ recognizes a 2-piecewise testable language.

**Proof.** For every $a \in \Sigma$ and for every $(p, pa) \in Q \times Q$ reachable from $i$ under a word containing $a$, we need to check that for every state $(q, q')$ reachable in $(Q, \Sigma, \cdot, p, F)$, it holds that $qa = q'a$.

However, our aim is to show that the problem is NL-complete. To show that the problem is in NL, we need the following lemma, which gives a characterization of 2-piecewise testable languages that can be verified locally in non-deterministic logarithmic space.

**Lemma 14.** Let $\mathcal{A} = (Q, \Sigma, \cdot, i, F)$ be a minimal DFA (that is partially ordered and confluent). Then the following is equivalent:

1. For every $a \in \Sigma$ and every states $p, q$ such that $pa = q, qa_1u_2 = qa_1u_2a$, for every $a_1, u_2 \in \Sigma^*$;
2. For every $a \in \Sigma$ and every state $s$ such that $is = s$ for some $w \in \Sigma^*$ with $|w|_b \geq 1$, $sba = saba$ for every $b \in \Sigma \cup \{\varepsilon\}$.

**Proof.** ($1 \Rightarrow 2$) Let $w = uav_1v_2$, and let $q = iua$ and $s = qv_1$. Then, for $u_1 = v_1$ and $u_2 = b \in \Sigma \cup \{\varepsilon\}$, we obtain $sba = qv_1ba = qv_1aba = saba$.

($2 \Rightarrow 1$) We prove this direction by induction on the length of $u_2$. Let $a \in \Sigma$ and $pa = q$. If $u_2 = \varepsilon$, let $b = \varepsilon$. Then $qa_1u = (qa_1)u = (qa_1)uu = qa_1u_2$. Thus, let $u_2 = ab$, for some $b \in \Sigma$. By the induction hypothesis, we have that $qa_1ua = qa_1ua$. Then $qa_1uaba = (qa_1u)ba = (qa_1u)aba = (qa_1u)ba = (qa_1u)aba = (qa_1u)aba = qa_1uaba$.

Now we prove that 2-piecewise testability is NL-complete.

**Theorem 15.** To decide whether a minimal DFA recognizes a 2-piecewise testable languages is NL-complete.

**Proof.** The check of whether a minimal DFA is not confluent or does not satisfy condition 2 of Lemma[14] can be done in NL; the reader is referred to [4] for a proof how to check confluency in NL. Since NL=co-NL [13, 28], we have an NL algorithm to check 2-piecewise testability of a minimal DFA.

To prove NL-hardness, we reduce an NL-complete problem monotone graph accessibility (2MGAP) [4], which is a special case of the graph reachability problem, to the 2-piecewise testability problem. 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 $\mathcal{A} = (V \cup \{i, f, d\}, \Sigma, i, \{f\})$ as follows. For every edge $(u, v)$, we construct a transition $u \cdot d_{uv} = v$ over a fresh letter $d_{uv}$. Moreover, we add two transitions $i \cdot a = s$ and $g \cdot a = f$ 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 states, 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 $f$ under fresh letters. All undefined transitions go to the sink state $d$. 

7
Claim 3. The automaton $\mathcal{A}$ is deterministic and minimal, and $L(\mathcal{A})$ is finite.

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

The following claim is needed to complete the proof.

Claim 4. Let $w$ be a word over $\Sigma$. If every $a$ from $\Sigma$ appears at most once in $w$, that is, $|w_a| \leq 1$, then the language $\{w\}$ is 2-piecewise testable.

Proof. Note that the condition of Theorem 12 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 show that the language $L(\mathcal{A})$ is 2-piecewise testable if and only if $g$ is not reachable from $s$. By contraposition, we first assume that $g$ is reachable from $s$. Let $w$ be a sequence of labels of such a path from $s$ to $g$ in $\mathcal{A}$. Then the word $awa$ belongs to $L(\mathcal{A})$ and $awaa$ does not. However, $awa \sim_a awaa$, which proves that the language $L(\mathcal{A})$ is not 2-piecewise testable. On the other hand, if $g$ is not reachable from $s$, then any word of the finite language $L(\mathcal{A})$ does not contain the same letter twice. Thus, by the previous claim, the language $L(\mathcal{A})$ is a finite union of 2-piecewise testable languages, hence it is 2-piecewise testable.

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 $k$-piecewise testable.

Theorem 16. The 0-piecewise testability problem for NFAs is PSPACE-complete.

Proof. A language is 0-piecewise testable if and only if it is either empty or universal. The result now follows from the fact that universality is PSPACE-complete [10].

Since $k$ is fixed, we can make use of the idea of Theorem 5 to show that to decide whether an NFA accepts a $k$-piecewise testable language is in PSPACE.

Corollary 17. The $k$-piecewise testability problem for NFAs is in PSPACE.

Proof. Let $\mathcal{A}$ be an NFA over the alphabet $\Sigma$. Let $\mathcal{A}'$ denote the minimal DFA obtained from $\mathcal{A}$ by the standard subset construction and minimization. By Theorem 5 and since it is well known that NPSPACE=PSPACE=co-PSPACE, we can guess and store a word $w_1$ of length at most $k|\Sigma|^k$ and to 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 $\mathcal{A}'$ to two different states. Note that because of the space limits the minimal DFA $\mathcal{A}'$ cannot be stored in the memory, but must be simulated on-the-fly during the word $w_2$ is being guessed. The state of $\mathcal{A}'$ defined by the word $w_2$ can then be compared with the state of $\mathcal{A}'$ defined by the word $w_1$, which is either computed at the end or stored from the beginning.

Considering the problem to find the minimal $k$ for which the language recognized by a given NFA is $k$-piecewise testable, we can show that it is PSPACE-hard.
Theorem 18. The problem to find the minimal \( k \) for which a given NFA recognizes a \( k \)-piecewise testable language is PSPACE-hard.

Proof. It is known that if a language given by an NFA is \( k \)-piecewise testable, then \( k \) is limited by \( 2^n \), where \( n \) is the number of states of the given NFA [13]. Thus, we can reduce the piecewise testability problem for NFAs to the problem under consideration so that we use the binary search to find the minimal \( k \) between 0 and \( 2^n \) if it exists. Since the piecewise testability problem for NFAs is PSPACE-complete [13], the proof is complete.

5. Piecewise testability and the depth of NFAs

In this section, we generalize a result valid for DFAs to NFAs and use it to investigate the relationship between the depth of an NFA and the minimal \( k \) for which the language of the NFA is \( k \)-piecewise testable. 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 \( k \)-piecewise testable language \( L \) recognized by an NFA \( \mathcal{A} \) of depth \( k - 1 \) (of depth \( k \) if we consider “complete” NFAs defined below) and by the minimal DFA \( \mathcal{D} \) of depth \( 2^k - 1 \).

Recall that a regular language is piecewise testable if and only if its minimal DFA satisfies some properties that can be tested in a quadratic time, cf. Theorem 2. We now show that this characterization generalizes to NFAs. We say that an NFA \( \mathcal{A} \) over an alphabet \( \Sigma \) is complete if for every state \( q \) of \( \mathcal{A} \) 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 19. 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. \((\Rightarrow)\) If a regular language is piecewise testable, then its minimal DFA is partially ordered and satisfies the UMS property by [29].

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

Claim 5. The minimal DFA \( \mathcal{D} \) 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 \( \mathcal{D} \), 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 for all \( i < k \), it holds that \( p_i w = p_i \). We prove it for \( k \). Since \( X = \{p_1, p_2, \ldots, p_n\} = Xw = \bigcup_{p_i \in X} p_i w, p_k \leq p_i w \) and \( p_i w = p_i \) for \( i < k \), we have that \( p_k \in p_i w \). Thus, \( \text{alph}(w) \subseteq \Sigma(p_k) \) and the UMS property of \( \mathcal{A} \) implies that \( p_i w = p_k \). Therefore, for every \( a \in \text{alph}(w) \) and \( i = 1, 2, \ldots, n, p_i a = p_i \). If, for any state \( Y \) of \( \mathcal{D} \), \( Xw_1 = Y \) and \( Yw_2 = X \), the previous argument gives that \( X = Y \), hence \( \mathcal{D} \) is partially ordered.

Claim 6. The minimal DFA \( \mathcal{D} \) satisfies the UMS property.

Proof. Assume, for the sake of contradiction, that there exist two different states \( X \) and \( Y \) in the same component of \( \mathcal{D} \) that are maximal with respect to the alphabet \( \Sigma(X) \). That is, there exist a state \( Z \) in \( \mathcal{D} \) and two words \( u \) and \( v \) over \( \Sigma(X) \) such that \( X = Zu \) and \( Y = Zv \). The previous argument gives that for every \( a \in \Sigma(X) \) and \( p \) in \( X \cup Y \), \( pa = p \). Without loss of generality, we may assume that there exists a state \( x \) in \( X \setminus Y \). Let \( z \) in \( Z \) be such that \( x = z u \). Since \( x \) does not belong to \( Y, z v \neq x \). Note that \( z v \) is defined, since \( \mathcal{A} \) is complete. By the proof of Claim 3, \( \Sigma(X) \subseteq \Sigma(zv) \) and \( \Sigma(X) \subseteq \Sigma(x) \). If \( x \) is not reachable from \( z v \) by \( \Sigma(zv) \setminus \Sigma(X) \), neither \( z v \) is reachable from \( x \) by \( \Sigma(x) \setminus \Sigma(X) \), we have a contradiction with the UMS property of \( \mathcal{A} \). Assume that \( z v \) reaches \( x \) under \( \Sigma(zv) \setminus \Sigma(X) \). Then \( z v \leq x \). If \( x \) does not reach \( z v \) under \( \Sigma(zv) \), then \( z v \) and a maximal state of \( x \cdot \Sigma(zv) \) are two different maximal states in \( \mathcal{A} \), a contradiction. If \( x \) reaches \( z v \) under \( \Sigma(zv) \setminus \Sigma(X) \), then \( x \leq z v \), which implies, since the NFA is partially ordered, that \( z v = x \), which is again a contradiction. Since the case where \( x \) reaches \( z v \) under \( \Sigma(x) \setminus \Sigma(X) \) is analogous, the proof is complete.

Thus, we have shown that the minimal DFA \( \mathcal{D} \) is partially ordered and satisfies the UMS property. Theorem 2 now completes the proof.
Since it is PSPACE-complete to decide whether an NFA defines a piecewise testable language, we immediately have the following.

**Corollary 20.** Given an NFA, it is PSPACE-complete to decide whether there exists an equivalent complete NFA that is partially ordered and satisfies the UMS property.

### 5.1. Exponential gap between the minimal $k$ and the depth of DFAs

It was shown in [15] that the depth of minimal DFAs does not correspond to the minimal $k$ for which the language is $k$-piecewise testable. Namely, an example of $(4\ell - 1)$-piecewise testable languages with the minimal DFA of depth $4\ell^2$, for $\ell > 1$, has been proposed. We now show that there is an exponential gap between the minimal $k$ for which the language is $k$-piecewise testable and the depth of a minimal DFA.

**Theorem 21.** For every $n \geq 2$, there exists an $n$-piecewise testable language that is not $(n - 1)$-piecewise testable, 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

$$\mathcal{A}_k = ([0, 1, \ldots, k], \{a_0, a_1, \ldots, a_k\}, I, I_k, \{0\})$$

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

![Figure 1: Automata $\mathcal{A}_2$ and $\mathcal{A}_3$.](image)

We define the word $w_k$ inductively by $w_0 = a_0$ and $w_{\ell} = w_{\ell-1} a_{i} w_{\ell-1}$, for $0 < \ell \leq k$. Note that $|w_{\ell}| = 2^{\ell+1} - 1$. In [12], we have shown that every prefix of $w_k$ of odd length ends with $a_0$ and, thus, does not belong to $L(\mathcal{A}_k)$, while every prefix of even length belongs to $L(\mathcal{A}_k)$. For convenience, we briefly recall the proof here. The empty word belongs to $L(\mathcal{A}_k) \subseteq L(\mathcal{A}_k)$. 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, $v \in L(\mathcal{A}_{k-1}) \subseteq L(\mathcal{A}_k)$. If $|v| > 2^k - 1$, then $v = w_{k-1} a_{i} v'$. The definition of $\mathcal{A}_k$ and the induction hypothesis then yield that there is a path $k \xrightarrow{w_{k-1}} k \xrightarrow{a_{i} v'} (k-1) \xrightarrow{v'} 0$. Thus, $v$ belongs to $L(\mathcal{A}_k)$.

We now discuss the depth of the minimal DFA recognizing the language $L(\mathcal{A}_k)$.

**Claim 7.** For every $k \geq 0$, the depth of the minimal DFA recognizing the language $L(\mathcal{A}_k)$ is $2^{k+1} - 1$.

**Proof.** We prove the claim by induction on $k$. For $k = 0$, the minimal DFA $\det(\mathcal{A}_0) = \{([0], \emptyset), \{a_0\}, \{0, 0\}\}$ obtained from $\mathcal{A}_0$ by the standard subset construction and minimization has two states, accepts the single word $\varepsilon$, 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 $w_k = w_{k-1} a_{i} w_{k-1}$ for $k > 0$. By the induction hypothesis, there exists a simple path of length $2^k - 1$ in $\det(\mathcal{A}_{k-1})$ defined by the word $w_{k-1}$ starting from the initial state $I_k = \{0, 1, \ldots, k - 1\}$ and ending in the state $\emptyset$. Let $Q_0, Q_1, \ldots, Q_{2^{k-1}-1}$ denote the states of that simple path in the order they appear on the path, that is, $Q_0 = I_k$, $Q_{2^{k-1}-1} = \emptyset$, and $Q_i \subseteq Q_0$ for $i = 1, 2, \ldots, 2^{k-1}$. Note that the states are pairwise non-equivalent by the induction hypothesis. Let $w_{k-1, i}$ denote the $i$-th letter of the word $w_{k-1}$.

The path

$$\begin{align*}
(Q_0 \cup \{k\}) & \xrightarrow{w_{k-1,1}} (Q_1 \cup \{k\}) \xrightarrow{w_{k-1,2}} (Q_2 \cup \{k\}) \xrightarrow{w_{k-1}} (Q_{2^{k-1}} \cup \{k\}) \xrightarrow{a_k} Q_0 \xrightarrow{w_{k-1,1}} Q_1 \xrightarrow{w_{k-1,2}} Q_2 \xrightarrow{\ldots} Q_{2^{k-1}}
\end{align*}$$

for $k \geq 1$.
Claim 9. For every $k \geq 0$, the language $L(A_k)$ is not $k$-piecewise testable.

Proof. Let $w_k = w_{k-1}a_1w_{k-1}$ be the word defined above. Let $w'_k$ denote the prefix of $w_k$ without the last letter (which is $a_0$), that is, $w_k = w'_k a_0$. We now show, by induction on $k$, that $w_k \not\sim_0 w'_{k+1}$. This then implies that the language $L(A_k)$ is not $k$-piecewise testable, because $w'_k$ belongs to $L(A_k)$ while $w_k$ does not belong to $L(A_k)$. Indeed, for $k = 0$, we have $w_0 = a_0 \not\sim_0 \varepsilon = w'_0$. Thus, assume that $w_k \not\sim_k w'_k$ for some $k \geq 0$, and consider a word $w$ such that $w \in \text{sub}_{k+1}(w_1w_{k+1}w_k)$.

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 $w_1w_{k+1}$. 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, $w'' \in \text{sub}_k(w_k) = \text{sub}_k(w'_k)$. Thus, $w = w'w'' \in \text{sub}_{k+1}(w_kw_{k+1}w'_k)$, which proves that $w_{k+1} \not\sim_k w'_{k+1}$.

To finish the proof of Theorem 2.1, note that every NFA $A_k$ has depth $k$, accepts a $(k + 1)$-piecewise testable language and its minimal DFA has depth $2^{k+1} - 1$. This completes the proof.

It could seem that NFAs are more convenient to provide upper bounds on the $k$. However, the following simple example demonstrates that even for $k$-piecewise testable languages, the depth of an NFA depends on the size of the input alphabet. Specifically, for any alphabet $\Sigma$, there exists a $1$-piecewise testable language over $\Sigma$ such that any NFA recognizing it requires at least $2^{|\Sigma|}$ states and depth $|\Sigma|$.

Example 22. Let $L = \bigcap_{w \in \Sigma} L_w$ 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 $A = (2^\Sigma, \Sigma, \{\emptyset\}, 2^\Sigma)$ 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 $A$ 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 a fooling set lower-bound technique [3]. 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 $x_i$ belong to $L$ and, for $i \neq j$, at least one of the words $x_iy_j$ and $x_jy_i$ 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 $\text{alph}(w_Xw_{\Sigma \setminus X}) = \Sigma$ and $w_Xw_{\Sigma \setminus X}$ 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 $\Sigma$, which implies that $S$ is a fooling set of size $2^{|\Sigma|}$. The main result of [3] 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.

Note that if we consider union instead of intersection, the resulting minimal DFA has only 2 states and depth 1.

Acknowledgement

The research was supported by the DFG in grant KR 4381/1-1 and by the Alexander von Humboldt Foundation.
