On the index of Simon's congruence for piecewise testability

P. Karandikar[a,b,1,2], M. Kufleitner[c,3], Ph. Schnoebelen[a,2]

[a] *Lab. Specification & Verification, CNRS UMR 8643 & ENS Cachan, France*
[b] *Chennai Mathematical Institute, Chennai, India*
[c] *Institut für Formale Methoden der Informatik, University of Stuttgart, Germany*

Abstract

Simon's congruence, denoted $\sim_n$, relates words having the same subwords of length up to n. We show that, over a k-letter alphabet, the number of words modulo $\sim_n$ is in $2^{\Theta(n^{k-1}\log n)}$.

Keywords: Combinatorics of words; Piecewise testable languages; Subwords and subsequences.

1. Introduction

Piecewise testable languages, introduced by Imre Simon in the 1970s, are a family of star-free regular languages that are definable by the presence and absence of given (scattered) subwords [1, 2, 3]. Formally, a language $L \subseteq A^*$ is n-piecewise testable if $x \in L$ and $x \sim_n y$ imply $y \in L$, where $x \sim_n y \overset{\text{def}}{\Leftrightarrow} x$ and y have the same subwords of length at most n (see next section for all definitions missing in this introduction). Piecewise testable languages are important because they are the languages defined by $\mathcal{B}\Sigma_1$ formulae, a simple fragment of first-order logic that is prominent in database queries [4]. They also occur in learning theory [5], computational linguistics [6], etc.

It is easy to see that $\sim_n$ is a congruence with finite index and Sakarovitch and Simon raised the question of how to better characterize or evaluate this number [2, p. 110]. Let us write $C_k(n)$ for the number of $\sim_n$ classes over k letters, i.e., when $|A| = k$. It is clear that $C_k(n) \geq k^n$ since two words $x, y \in A^{\leq n}$ (i.e., of length at most n) are related by $\sim_n$ only if they are equal. In fact, this reasoning gives

$$C_k(n) \geq k^n + k^{n-1} + \cdots + k + 1 = \frac{k^{n+1}-1}{k-1} \qquad (1)$$

(assuming $k \neq 1$). On the other hand, any congruence class in $\sim_n$ is completely characterized by a set of subwords in $A^{\leq n}$, hence

$$C_k(n) \leq 2^{\frac{k^{n+1}-1}{k-1}} . \qquad (2)$$

Estimating the size of $C_k(n)$ has applications in descriptive complexity, for example for estimating the number of n-piecewise testable languages (over a given alphabet), or for bounding the size of canonical automata for n-piecewise testable languages [7, 8, 9].

Unfortunately the above bounds, summarized as $k^n \leq C_k(n) \leq 2^{k^{n+1}}$, leave a large ("exponential") gap and it is not clear towards which side is the actual value leaning.[4] Eq. (1) gives a lower bound that is obviously very naive since it only counts the simplest classes. On the other hand, Eq. (2) too makes wide simplifications since not every subset of $A^{\leq n}$ corresponds to a congruence class. For example, if `aa` and `bb` are subwords of some x then necessarily x also has `ab` or `ba` among its length 2 subwords.

Since the question of estimating $C_k(n)$ was raised in [2] (and to the best of our knowledge) no progress has been made on the question, until Kátai-Urbán et al. proved the following bounds:

Theorem 1.1 (Kátai-Urbán et al. [10]). *For all $k > 1$,*

$$\frac{k^n}{3^{n^2}} \log k \leq \log C_k(n) < 3^n k^n \log k \quad \textit{if } n \textit{ is even,}$$
$$\frac{k^n}{3^{n^2}} < \log C_k(n) < 3^n k^n \quad \textit{if } n \textit{ is odd.}$$

The proof is based on two reductions, one showing $C_{k+\ell}(n+2) \geq C_k^{\ell+2}(n)$ for proving lower bounds, and one showing $C_k(n+2) \leq (k+1)^{2k} C_k^{2k-1}(n)$ for proving upper bounds. For fixed n, Theorem 1.1 allows to estimate the asymptotic value of $\log C_k(n)$ as a function of k: it is in $\Theta(k^n)$ or $\Theta(k^n \log k)$ depending on the parity of n. However, these bounds do not say how, for fixed k, $C_k(n)$ grows as a function of n, which is a more natural question in settings where the alphabet is fixed, and where n comes from, e.g., the number of variables in a $\mathcal{B}\Sigma_1$ formula. In particular, the lower bound is useless for $n \geq k$ since in this case $k^n/3^{n^2} < 1$.

[1] Partially supported by Tata Consultancy Services.
[2] Supported by ANR grant 11-BS02-001-01.
[3] Supported by DFG grant DI 435/5-2.

[4] Comparing the bounds from Eqs. (1) and (2) with actual values does not bring much light here since the magnitude of $C_k(n)$ makes it hard to compute beyond some very small values of k and n, see Table B.1.

Our contribution. In this article, we provide the following bounds:

Theorem 1.2. *For all $k, n > 1$,*

$$\left(\frac{n}{k}\right)^{k-1} \log_2 \left(\frac{n}{k}\right) < \log_2 C_k(n) < k\left(\frac{n+2k-3}{k-1}\right)^{k-1} \log_2 n \log_2 k \,.$$

Thus, for fixed k, $\log C_k(n)$ is in $\Theta(n^{k-1} \log n)$. Compared with Theorem 1.1, our bounds are much tighter for fixed k (and much wider for fixed n).

The proof of Theorem 1.2 relies on two new reductions that allows us to relate $C_k(n)$ with C_{k-1} instead of relating it with $C_k(n-2)$ as in [10]. The article is organized as follows. Section 2 recalls the necessary notations and definitions; the lower bound is proved in Section 3 while the upper bound is proved in Section 4. An appendix lists the exact values of $C_k(n)$ for small n and k that we managed to compute.

2. Basics

We consider words $x, y, w, \ldots$ over a finite k-letter alphabet $A_k = \{\mathtt{a}_1, \ldots, \mathtt{a}_k\}$ sometimes written more simply $A = \{\mathtt{a}, \mathtt{b}, \ldots\}$. The empty word is denoted ϵ, concatenation is denoted multiplicatively. Given a word $x \in A^*$ and a letter $\mathtt{a} \in A$, we write $|x|$ and $|x|_{\mathtt{a}}$ for, respectively, the length of x, and the number of occurrences of $\mathtt{a}$ in x.

We write $x \preccurlyeq y$ to denote that a word x is a *subsequence* of y, also called a (scattered) *subword*. Formally, $x \preccurlyeq y$ iff $x = x_1 \cdots x_\ell$ and there are words $y_0, y_1, \ldots, y_\ell$ such that $y = y_0 x_1 y_1 \cdots x_\ell y_\ell$. It is well-known that $\preccurlyeq$ is a partial ordering and a monoid precongruence.

For any $n \in \mathbb{N}$, we write $x \sim_n y$ when x and y have the same subwords of length $\leq n$. For example $x \stackrel{\text{def}}{=} \mathtt{abacb} \sim_2 y \stackrel{\text{def}}{=} \mathtt{baaacbb}$ since both words have $\{\epsilon, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{aa}, \mathtt{ab}, \mathtt{ac}, \mathtt{ba}, \mathtt{bb}, \mathtt{bc}, \mathtt{cb}\}$ as subwords of length ≤ 2. However $x \not\sim_3 y$ since $x \succcurlyeq \mathtt{aba} \not\preccurlyeq y$. Note that $\sim_0 \supseteq \sim_1 \supseteq \sim_2 \supseteq \cdots$, and that $x \sim_0 y$ holds trivially. It is well-known (and easy to see) that each $\sim_n$ is a congruence since the subwords of some xy are the concatenations of a subword of x and a subword of y. Simon defined a *piecewise testable* language as any $L \subseteq A^*$ that is closed by $\sim_n$ for some n [1]. These are exactly the languages definable by $\mathcal{B}\Sigma_1(<, \mathtt{a}, \mathtt{b}, \ldots)$ formulae [4], i.e., by Boolean combinations of existential first-order formulae with monadic predicates of the form $\mathtt{a}(i)$, stating that the i-th letter of a word is $\mathtt{a}$. For example, $L = A^*\mathtt{a}A^*\mathtt{b}A^* = \{x \in A^* \mid \mathtt{ab} \preccurlyeq x\}$ is definable with the following Σ_1 formula:

$$\exists i : \exists j : i < j \wedge \mathtt{a}(i) \wedge \mathtt{b}(j) \,.$$

The index of $\sim_n$. Since there are only finitely many words of length $\leq n$, the congruence $\sim_n$ partitions A_k^* in finitely many classes, and we write $C_k(n)$ for the number of such classes, i.e., the cardinal of $A_k^* / {\sim_n}$.

The following is easy to see:

$$C_1(n) = n + 1 \,, \qquad C_k(0) = 1 \,, \qquad C_k(1) = 2^k \,. \qquad (3)$$

Indeed, for words over a single letter $\mathtt{a}$, $x \sim_n y$ iff $|x| = |y| < n$ or $|x| \geq n \leq |y|$, hence the first equality. The second equality restates that $\sim_0$ is trivial, as noted above. For the third equality, one notes that $x \sim_1 y$ if, and only if, the same set of letters is occurring in x and y, and that there are 2^k such sets of occurring letters.

3. Lower bound

The first half of Theorem 1.2 is proved by first establishing a combinatorial inequality on the $C_k(n)$'s (Proposition 3.3) and then using it to derive Proposition 3.4.

Consider two words $x, y \in A^*$ and a letter $a \in A$.

Lemma 3.1. *If $x \sim_n y$, then* $\min(|x|_a, n) = \min(|y|_a, n)$.

Proof (Sketch). If $|x|_a = p < n$ then $a^p \preccurlyeq x \not\succcurlyeq a^{p+1}$. From $x \sim_n y$ we deduce $a^p \preccurlyeq y \not\succcurlyeq a^{p+1}$, hence $|y|_a = p$. □

Fix now $k \geq 2$, let $A = A_k = \{\mathtt{a}_1, \ldots, \mathtt{a}_k\}$ and assume $x \sim_n y$. If $|x|_{\mathtt{a}_k} = p < n$, then x is some $x_0 \mathtt{a}_k x_1 \cdots \mathtt{a}_k x_p$ with $x_i \in A_{k-1}^*$ for $i = 0, \ldots, p$. By Lemma 3.1, y too is some $y_0 \mathtt{a}_k y_1 \cdots \mathtt{a}_k y_p$ with $y_i \in A_{k-1}^*$.

Lemma 3.2. *$x_i \sim_{n-p} y_i$ for all $i = 0, \ldots, p$.*

Proof. Suppose $w \preccurlyeq x_i$ and $|w| \leq n - p$. Let $w' \stackrel{\text{def}}{=} \mathtt{a}_k^i w \mathtt{a}_k^{p-i}$. Clearly $w' \preccurlyeq x$ and thus $w' \preccurlyeq y$ since $x \sim_n y$ and $|w'| \leq n$. Now $w' = \mathtt{a}_k^i w \mathtt{a}_k^{p-i} \preccurlyeq y$ entails $w \preccurlyeq y_i$.

With a symmetric reasoning we show that every subword of y_i having length $\leq n - p$ is a subword of x_i and we conclude $x_i \sim_{n-p} y_i$. □

Proposition 3.3. *For $k \geq 2$, $C_k(n) \geq \sum_{p=0}^{n} C_{k-1}^{p+1}(n-p)$.*

Proof. For words $x = x_0 \mathtt{a}_k x_1 \ldots x_{p-1} \mathtt{a}_k x_p$ with exactly $p < n$ occurrences of $\mathtt{a}_k$, we have $C_{k-1}(n-p)$ possible choices of $\sim_{n-p}$ equivalence classes for each x_i ($i = 0, \ldots, p$). By Lemma 3.2 all such choices will result in $\not\sim_n$ words, hence there are exactly $C_{k-1}^{p+1}(n-p)$ classes of words with $p < n$ occurrences of $\mathtt{a}_k$. By Lemma 3.1, these classes are disjoint for different values of p, hence we can add the $C_{k-1}^{p+1}(n-p)$'s. There remain words with $p \geq n$ occurrences of $\mathtt{a}_k$, accounting for at least 1, i.e., $C_{k-1}^{n+1}(0)$, additional class. □

Proposition 3.4. *For all $k, n > 0$:*

$$\log_2 C_k(n) > \left(\frac{n}{k}\right)^{k-1} \log_2 \left(\frac{n}{k}\right) . \qquad (4)$$

PROOF. Eq. (4) holds trivially when $\log_2(\frac{n}{k}) \leq 0$. Hence there only remains to consider the cases where $n > k$. We reason by induction on k. For $k = 1$, Eq. (3) gives $\log_2 C_1(n) = \log_2(n+1) > \log_2 n = \left(\frac{n}{1}\right)^0 \log_2\left(\frac{n}{1}\right)$. For the inductive case, Proposition 3.3 yields $C_{k+1}(n) \geq C_k^{p+1}(n-p)$ for all $p \in \{0, \ldots, n\}$. For $p = \left\lfloor \frac{n}{k+1} \right\rfloor$ this yields

$$\log_2 C_{k+1}(n) \geq (p+1) \log_2 C_k(n-p)$$

$$> (p+1) \left(\frac{n-p}{k}\right)^{k-1} \log_2 \left(\frac{n-p}{k}\right)$$

by ind. hyp., noting that $n - p > 0$,

$$\geq \frac{n}{k+1} \left(\frac{n}{k+1}\right)^{k-1} \log_2 \left(\frac{n}{k+1}\right)$$

since $\frac{n-p}{k} \geq \frac{n}{k+1} \geq 1$,

$$= \left(\frac{n}{k+1}\right)^{k} \log_2 \left(\frac{n}{k+1}\right)$$

as desired. □

4. Upper bound

The second half of Theorem 1.2 is again by establishing a combinatorial inequality on the $C_k(n)$'s (Proposition 4.3) and then using it to derive Proposition 4.4.

Fix $k > 0$ and consider words in A_k^*. We say that a word x is *rich* if all the k letters of A_k occur in it, and that it is *poor* otherwise. For $\ell > 0$, we further say that x is ℓ-rich if it can be written as a concatenation of ℓ rich factors (by extension "x is 0-rich" means that x is poor). The *richness* of x is the largest $\ell \in \mathbb{N}$ such that x is ℓ-rich. Note that $\forall a \in A_k : |x|_a \geq \ell$ does not imply that x is ℓ-rich. We shall use the following easy result:

Lemma 4.1. *If x_1 and x_2 are respectively ℓ_1-rich and ℓ_2-rich, then $y \sim_n y'$ implies $x_1 y x_2 \sim_{\ell_1+n+\ell_2} x_1 y' x_2$.*

PROOF. A subword u of $x_1 y x_2$ can be decomposed as $u = u_1 v u_2$ where u_1 is the largest prefix of u that is a subword of x and u_2 is the largest suffix of the remaining $u_1^{-1}u$ that is a subword of x_2. Thus $v \preccurlyeq y$ since $u \preccurlyeq x_1 y x_2$. Now, since x_1 is ℓ_1-rich, $|u_1| \geq \ell_1$ (unless u is too short), and similarly $|u_2| \geq \ell_2$ (unless ...). Finally $|v| \leq n$ when $|u| \leq \ell_1 + n + \ell_2$, and then $v \preccurlyeq y'$ since $y \sim_n y'$, entailing $u \preccurlyeq x_1 y' x_2$. A symmetrical reasoning shows that subwords of $x_1 y' x_2$ of length $\leq \ell_1 + n + \ell_2$ are subwords of $x_1 y x_2$ and we are done. □

The *rich factorization* of $x \in A_k^*$ is the decomposition $x = x_1 a_1 \cdots x_m a_m y$ obtained in the following way: if x is poor, we let $m = 0$ and $y = x$; otherwise x is rich, we let $x_1 a_1$ (with $a_1 \in A_k$) be the shortest prefix of x that is rich, write $x = x_1 a_1 x'$ and let $x_2 a_2 \ldots x_m a_m y$ be the rich factorization of the remaining suffix x'. By construction m is the richness of x. E.g., assuming $k = 3$, the following is a rich factorization with $m = 2$:

$$\overbrace{\texttt{bbaaabbccccaabbbaa}}^{x} = \overbrace{\texttt{bbaaabb}}^{x_1} \cdot \texttt{c} \cdot \overbrace{\texttt{cccaa}}^{x_2} \cdot \texttt{b} \cdot \overbrace{\texttt{bbaa}}^{y}$$

Note that, by definition, $x_1, \ldots, x_m$ and y are poor.

Lemma 4.2. *Consider two words x, x' of richness m and with rich factorizations $x = x_1 a_1 \ldots x_m a_m y$ and $x' = x'_1 a_1 \ldots x'_m a_m y'$. Suppose that $y \sim_n y'$ and that $x_i \sim_{n+1} x'_i$ for all $i = 1, \ldots, m$. Then $x \sim_{n+m} x'$.*

PROOF. By repeatedly using Lemma 4.1, one shows

$$\begin{aligned} x_1 a_1 x_2 a_2 \ldots x_m a_m y &\sim_{n+m} x'_1 a_1 x_2 a_2 \ldots x_m a_m y \\ &\sim_{n+m} x'_1 a_1 x'_2 a_2 \ldots x_m a_m y \\ &\vdots \\ &\sim_{n+m} x'_1 a_1 x'_2 a_2 \ldots x'_m a_m y \\ &\sim_{n+m} x'_1 a_1 x'_2 a_2 \ldots x'_m a_m y' , \end{aligned}$$

using the fact that each factor $x_i a_i$ is rich. □

Proposition 4.3. *For all $n \geq 0$ and $k \geq 2$,*

$$C_k(n) \leq 1 + \sum_{m=0}^{n-1} k^{m+1} C_{k-1}^m(n-m+1) C_{k-1}(n-m) .$$

Furthermore, for $k = 2$,

$$C_2(n) \leq 2 \sum_{m=0}^{2n-1} n^m = 2 \frac{n^{2n}-1}{n-1} . \tag{5}$$

PROOF. Consider two words x, x' and their rich factorization $x = x_1 a_1 \ldots x_m a_m y$ and $x' = x'_1 a'_1 \ldots x'_\ell a'_\ell y'$. By Lemma 4.2 they belong to the same $\sim_n$ class if $\ell = m$, $y \sim_{n-m} y'$, and $a_i = a'_i$ and $x_i \sim_{n-m+1} x'_i$ for all $i = 1, \ldots, m$. Now for every fixed m, there are at most k^m choices for the a_i's, $C_{k-1}^m(n-m+1)$ non-equivalent choices for the x_i's, $kC_{k-1}(n-m)$ choices for y and a letter that is missing in it. We only need to consider m varying up to $n-1$ since all words of richness $\geq n$ are $\sim_n$-equivalent, accounting for one additional possible $\sim_n$ class.

For the second inequality, assume that $k = 2$ and $A_2 = \{\texttt{a}, \texttt{b}\}$. A word $x \in A_2^*$ can be decomposed as a sequence of m non-empty blocks of the same letter, of the form, e.g., $x = \texttt{a}^{\ell_1} \texttt{b}^{\ell_2} \texttt{a}^{\ell_3} \texttt{b}^{\ell_4} \cdots \texttt{a}^{\ell_m}$ (this example assumes that x starts and ends with $\texttt{a}$, hence m is odd). If two words like $x = \texttt{a}^{\ell_1} \texttt{b}^{\ell_2} \texttt{a}^{\ell_3} \texttt{b}^{\ell_4} \cdots \texttt{a}^{\ell_m}$ and $x' = \texttt{a}^{\ell'_1} \texttt{b}^{\ell'_2} \texttt{a}^{\ell'_3} \texttt{b}^{\ell'_4} \cdots \texttt{a}^{\ell'_m}$ have the same first letter $\texttt{a}$, the same alternation depth m, and have $\min(\ell_i, n) = \min(\ell'_i, n)$ for all $i = 1, \ldots, m$, then they are $\sim_n$-equivalent. For a given $m > 0$, there are 2 possibilities for choosing the first letter and n^m non-equivalent choices for the ℓ_i's. Finally, all words with alternation depths $m \geq 2n$ are $\sim_n$-equivalent, hence we can restrict our attention to $1 \leq m \leq 2n-1$. The extra summand $2\,n^0$ in Eq. (5) accounts for the single class with $m \geq 2n$ and the single class with $m = 0$. □

Proposition 4.4. *For all $k, n > 1$:*

$$C_k(n) < 2^{k\left(\frac{n+2k-3}{k-1}\right)^{k-1}\log_2 n \log_2 k}.$$

PROOF. By induction on k. For $k = 2$, Eq. (5) yields:

$$C_2(n) \leq 2\frac{n^{2n}-1}{n-1} < n\frac{n^{2n+1}}{1}$$

since $n \geq 2$,

$$= n^{2n+2} = 2^{2(n+1)\log_2 n}$$
$$= 2^{k\left(\frac{n+2k-3}{k-1}\right)^{k-1}\log_2 n \log_2 k}.$$

For the inductive case, Proposition 4.3 yields:

$$C_{k+1}(n) \leq 1 + \sum_{m=0}^{n-1}(k+1)^{m+1}C_k^m(n-m+1)C_k(n-m)$$
$$= 1 + (k+1)C_k(n) + \sum_{m=1}^{n-1}(k+1)^{m+1}C_k^m(n-m+1)C_k(n-m)$$
$$< (k+1)^n C_k(n) + \sum_{m=1}^{n-1}(k+1)^n C_k^{m+1}(n-m+1)$$

since $C_k(q) \leq C_k(q+1)$,

$$< (k+1)^n 2^{k\left(\frac{n+2k-3}{k-1}\right)^{k-1}\log_2 n \log_2 k} + \sum_{m=1}^{n-1}(k+1)^n 2^{k(m+1)\left(\frac{n-m+2k-2}{k-1}\right)^{k-1}\log_2 n \log_2 k}$$

by ind. hyp.,

$$< (k+1)^n \sum_{m=0}^{n-1} 2^{k(m+1)\left(\frac{n-m+2k-2}{k-1}\right)^{k-1}\log_2 n \log_2 k}.$$

Since $(m+1)\left(\frac{n-m+2k-2}{k-1}\right)^{k-1} \leq \left(\frac{n+2k-1}{k}\right)^k$ for all $m \in \{0, \ldots, n-1\}$ —see Appendix A—, we may proceed with:

$$C_{k+1}(n) < (k+1)^n \sum_{m=0}^{n-1} 2^{k\left(\frac{n+2k-1}{k}\right)^k \log_2 n \log_2 k}$$
$$= n(k+1)^n 2^{k\left(\frac{n+2k-1}{k}\right)^k \log_2 n \log_2 k}$$
$$= 2^{\log_2 n + n\log_2(k+1) + k\left(\frac{n+2k-1}{k}\right)^k \log_2 n \log_2 k}$$
$$< 2^{\left(\log_2 n + n + k\left(\frac{n+2k-1}{k}\right)^k \log_2 n\right)\log_2(k+1)}$$
$$< 2^{(k+1)\left(\frac{n+2k-1}{k}\right)^k \log_2 n \log_2(k+1)}$$

since $\log_2 n + n < \left(\frac{n+2k-1}{k}\right)^k \log_2 n$ (see below). This is the desired bound.

To see that $\log_2 n + n < \left(\frac{n+2k-1}{k}\right)^k \log_2 n$, we use

$$\left(\frac{n+2k-1}{k}\right)^k > \left(\frac{n}{k}+1\right)^k = \sum_{j=0}^{k}\binom{k}{j}\cdot\left(\frac{n}{k}\right)^j$$
$$= 1 + k\cdot\left(\frac{n}{k}\right) + \cdots \geq n+1\,.$$

This completes the proof. □

By combining the two bounds in Propositions 3.4 and 4.4 we obtain Theorem 1.2, implying that $\log C_k(n)$ is in $\Theta(n^{k-1}\log n)$ for fixed alphabet size k.

5. Conclusion

We proved that, over a fixed k-letter alphabet, $C_k(n)$ is in $2^{\Theta(n^{k-1}\log n)}$. This shows that $C_k(n)$ is not doubly exponential in n as Eq. (2) and Theorem 1.1 would allow. It also is not simply exponential, bounded by a term of the form $2^{f(k)\cdot n^c}$ where the exponent c does not depend on k.

We are still far from having a precise understanding of how $C_k(n)$ behaves and there are obvious directions for improving Theorem 1.2. For example, its bounds are not monotonic in k (while the bounds in Theorem 1.1 are not monotonic in n) and it only partially uses the combinatorial inequalities given by Propositions 3.3 and 4.3.

Acknowledgments. We thank J. Berstel, J.-É. Pin and M. Zeitoun for their comments and suggestions.

References

[1] I. Simon, Piecewise testable events, in: Proc. 2nd GI Conf. on Automata Theory and Formal Languages, volume 33 of *Lecture Notes in Computer Science*, Springer, 1975, pp. 214–222. doi:10.1007/3-540-07407-4_23.

[2] J. Sakarovitch, I. Simon, Subwords, in: M. Lothaire (Ed.), Combinatorics on words, volume 17 of *Encyclopedia of Mathematics and Its Applications*, Cambridge Univ. Press, 1983, pp. 105–142.

[3] J.-E. Pin, Varieties of Formal Languages, Plenum, New-York, 1986.

[4] V. Diekert, P. Gastin, M. Kufleitner, A survey on small fragments of first-order logic over finite words, Int. J. Foundations of Computer Science 19 (2008) 513–548.

[5] L. Kontorovich, C. Cortes, M. Mohri, Kernel methods for learning languages, Theoretical Computer Science 405 (2008) 223–236.

[6] J. Rogers, J. Heinz, G. Bailey, M. Edlefsen, M. Visscher, D. Wellcome, S. Wibel, On languages piecewise testable in the strict sense, in: Proc. 10th and 11th Biennal Conf. Mathematics of Language (MOL 10), volume 6149 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 255–265. doi:10.1007/978-3-642-14322-9_19.

[7] W. Czerwiński, W. Martens, T. Masopust, Efficient separability of regular languages by subsequences and suffixes, in: Proc. 40th Int. Coll. Automata, Languages, and Programming (ICALP 2013), volume 7966 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 150–161. doi:10.1007/978-3-642-39212-2_16.

[8] O. Klíma, L. Polák, Alternative automata characterization of piecewise testable languages, in: Proc. 17th Int. Conf. Developments in Language Theory (DLT 2013), volume 7907 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 289–300. doi:10.1007/978-3-642-38771-5_26.

[9] Th. Place, L. van Rooijen, M. Zeitoun, Separating regular languages by piecewise testable and unambiguous languages, in: Proc. 38th Int. Symp. Math. Found. Comp. Sci. (MFCS 2013), volume 8087 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 729–740. doi:10.1007/978-3-642-40313-2_64.

[10] K. Kátai-Urbán, P. P. Pach, G. Pluhár, A. Pongrácz, C. Szabó, On the word problem for syntactic monoids of piecewise testable languages, Semigroup Forum 84 (2012) 323–332.

Appendix A. Additional proofs

We prove that $(m+1)\left(\frac{n-m+2k-2}{k-1}\right)^{k-1} \leq \left(\frac{n+2k-1}{k}\right)^k$ for all $m = 0, \ldots, n-1$, an inequality that was used to establish Proposition 4.4.

For $k > 0$ and $x, y \in \mathbb{R}$, let

$$F_k(x) \stackrel{\text{def}}{=} \left(\frac{x+2k-1}{k}\right)^k ,$$

$$G_{k,x}(y) \stackrel{\text{def}}{=} (y+1)F_k(x-y+1) = \frac{(y+1)(x-y+2k)^k}{k^k} .$$

Let us check that $G_{k,x}\left(\frac{k+x}{k+1}\right) = F_{k+1}(x)$ for any $k > 0$ and $x \geq 0$:

$$\begin{aligned} G_{k,x}\left(\frac{k+x}{k+1}\right) &= \left(\frac{k+x}{k+1}+1\right)\frac{1}{k^k}\left(x - \frac{k+x}{k+1}+2k\right)^k \\ &= \frac{x+2k+1}{k+1}\,\frac{1}{k^k}\left(\frac{kx+2k^2+k}{k+1}\right)^k \\ &= \frac{x+2k+1}{k+1}\,\frac{1}{k^k}\left(\frac{k}{k+1}\right)^k (x+2k+1)^k \\ &= \left(\frac{x+2k+1}{k+1}\right)^{k+1} = F_{k+1}(x) . \qquad (\dagger) \end{aligned}$$

We now claim that $G_{k,x}(y) \leq F_{k+1}(x)$ for all $y \in [0, x]$. For $n, k \geq 2$, the claim entails $G_{k-1,n}(m) \leq F_k(m)$, i.e. $(m+1)\left(\frac{n-m+2k-2}{k-1}\right)^{k-1} \leq \left(\frac{n+2k-1}{k}\right)^k$, for $m = 0, \ldots, n-1$ as announced.

Proof (of the claim). Let $y_{\max} \stackrel{\text{def}}{=} \frac{k+x}{k+1}$. We prove that $G_{k,x}(y) \leq G_{k,x}(y_{\max})$ and conclude using Eq. (†): $G_{k,x}$ is well-defined and differentiable over $\mathbb{R}$, its derivative is

$$\begin{aligned} G'_{k,x}(y) &= \frac{(x-y+2k)^k - (y+1)k(x-y+2k)^{k-1}}{k^k} \\ &= \frac{(x-y+2k)^{k-1}}{k^k}\big((x-y+2k)-(y+1)k\big) \\ &= \frac{(x-y+2k)^{k-1}}{k^k}\big(x+k-y(k+1)\big) . \end{aligned}$$

Thus $G'_{k,x}(y)$ is 0 for $y = y_{\max}$, is strictly positive for $0 \leq y < y_{\max}$, and strictly negative for $y_{\max} < y \leq x$. Hence, over $[0, x]$, $G_{k,x}$ reaches its maximum at $y_{\max}$. □

Appendix B. First values for $C_k(n)$

We computed the first values of $C_k(n)$ by a brute-force method that listed all minimal representatives of $\sim_n$ equivalence classes over a k-letter alphabet. Here x is *minimal* if $x \sim_n y$ implies ($|x| < |y|$ or ($|x| = |y|$ and $x \leq_{\text{lex}} y$)). Every equivalence class has a unique minimal representative. Note that if a concatenation xx' is minimal then both x and x' are. Therefore, when listing the minimal representatives in order of increasing length, it is possible to stop when, for some length ℓ, one finds no minimal representatives. In that case we know that there cannot exist minimal representatives of length $> \ell$.

The cells left blank in the table were not computed for lack of memory.

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	k
$n=0$	1	1	1	1	1	1	1	1	1
$n=1$	2	4	8	16	32	64	128	256	2^k
$n=2$	3	16	152	2 326	52 132	1 602 420	64 529 264	$\geq 173 \cdot 10^7$	
$n=3$	4	68	5 312	1 395 588	1 031 153 002	$\geq 23 \cdot 10^7$			
$n=4$	5	312	334 202	$\geq 73 \cdot 10^7$					
$n=5$	6	1 560	38 450 477						
$n=6$	7	8 528	$\geq 39 \cdot 10^7$						
$n=7$	8	50 864							
$n=8$	9	329 248							
$n=9$	10	2 298 592							
$n=10$	11	17 203 264							
$n=11$	12	137 289 920							
n	$n+1$								

Table B.1: Computed values for $C_k(n)$