# $k$-Abelian Equivalence and Rationality

Julien Cassaigne[1], Juhani Karhumäki[2,⋆], Svetlana Puzynina[3,4,⋆⋆], and Markus A. Whiteland[2,⋆⋆⋆]

[1] Institut de mathématiques de Marseille, France
julien.cassaigne@math.cnrs.fr
[2] Department of Mathematics and Statistics, University of Turku, Turku, Finland
{karhumak,mawhit}@utu.fi
[3] LIP, ENS de Lyon, Université de Lyon, Lyon, France
[4] Sobolev Institute of Mathematics, Russia
svepuz@utu.fi

**Abstract.** Two words $u$ and $v$ are said to be $k$-abelian equivalent if, for each word $x$ of length at most $k$, the number of occurrences of $x$ as a factor of $u$ is the same as for $v$. We study some combinatorial properties of $k$-abelian equivalence classes. Our starting point is a characterization of $k$-abelian equivalence by rewriting, so-called $k$-switching. We show that the set of lexicographically least representatives of equivalence classes is a regular language. From this we infer that the sequence of the numbers of equivalence classes is $\mathbb{N}$-rational. We also show that the set of words defining $k$-abelian singleton classes is regular.

**Keywords:** $k$-abelian equivalence, regular languages, rational sequences

## 1 Introduction

$k$-abelian equivalence has attracted quite a lot of interest recently, see, e.g., [1,2,8,10,12,15]. It is an equivalence relation extending abelian equivalence and allowing an infinitary approximation of the equality of words defined as follows: for an integer $k$, two words $u$ and $v$ are *$k$-abelian equivalent*, denoted by $u \sim_k v$, if, for each word $w$ of length at most $k$, $w$ occurs in $u$ and $v$ equally often.

$k$-abelian equivalence, originally introduced in [7], has been studied, e.g., in the following directions: avoiding *$k$-abelian powers* [6,15], estimating the number of $k$-abelian equivalence classes, that is, *$k$-abelian complexity* [11], analyzing the growth and the fluctuation of the *$k$-abelian complexity of infinite words* [1], analyzing *$k$-abelian palindromicity* [8], and studying *$k$-abelian singletons* [9]. We continue the approach of analyzing the structure of $k$-abelian equivalence classes. We also study some numerical properties of the equivalence classes.

⋆ Supported by the Academy of Finland, grant 257857.
⋆⋆ Supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).
⋆⋆⋆ Supported by the Academy of Finland, grant 257857.

Our starting point is a *k-switching* lemma, proved in [9], which allows a characterization of $k$-abelian equivalence in terms of rewriting. This is quite different from the other existing characterizations, so it is no surprise that it opens new perspectives of $k$-abelian equivalence. This is what we intend to explore here.

A fundamental observation from the characterization of $k$-abelian equivalence using $k$-switching is that certain languages related to $k$-abelian equivalence classes are *regular* (or *rational*). More precisely, the union of all singleton classes forms a regular language, for any parameter $k$, and any size $m$ of the alphabet. Similarly, the set of lexicographically least (or greatest) representatives of $k$-abelian equivalence classes forms a regular language. Summing up all minimal elements of a fixed length we obtain the number of equivalence classes of words of this length. As a consequence, we conclude that the complexity function of $k$-abelian equivalence, that is, the function computing the number of the equivalence classes of all lengths, is a rational function.

Everything above is algorithmic. So, given the parameter $k$ and the size $m$ of the alphabet, we can algorithmically compute a rational generating function giving the numbers of all equivalence classes of words of length $n$. However, the automata involved are – due to the non-determinism and the complementation – so huge that in practice this can be done only for very small values of the parameters. We illustrate these in a few examples.

Inspired by the connection to automata theory, we study $k$-switching in connection with regular languages. We show that regular languages are closed under the *k-switching operation*. On the other hand, we show that regular languages are not closed under the transitive closure of this operation. Using the former result, we conclude that the union of $k$-abelian equivalence classes of size two is regular. On the other hand, it remains open whether this extends, instead of classes of size two, to larger classes. Another open problem is to determine the asymptotic behavior of the complexity function of equivalence classes.

## 2 Preliminaries and Notation

We recall some notation and basic terminology from the literature of combinatorics on words. We refer the reader to [13] for more on the subject.

The set of finite words over an *alphabet* $\Sigma$ is denoted by $\Sigma^*$ and the set of non-empty words is denoted by $\Sigma^+$. The empty word is denoted by $\varepsilon$. A set $L \subseteq \Sigma^*$ is called a *language*. We let $|w|$ denote the length of a word $w \in \Sigma^*$. By convention, we set $|\varepsilon| = 0$. The language of words of length $n$ over the alphabet $\Sigma$ is denoted by $\Sigma^n$.

For a word $w = a_1 a_2 \cdots a_n \in \Sigma^*$ and indices $1 \leq i \leq j \leq n$, we let $w[i, j]$ denote the factor $a_i \cdots a_j$. For $i > j$ we set $w[i, j] = \varepsilon$. Similarly, for $i < j$ we let $w[i, j)$ denote the factor $a_i \cdots a_{j-1}$, and we set $w[i, j) = \varepsilon$ when $i \geq j$. We say that a word $x \in \Sigma^*$ *has position* $i$ *in* $w$ if the word $w[i, |w|]$ has $x$ as a prefix. For $u \in \Sigma^+$ we let $|w|_u$ denote the number of occurrences of $u$ as a factor of $w$.

Two words $u, v \in \Sigma^*$ are *k-abelian equivalent*, denoted by $u \sim_k v$, if $|u|_x = |v|_x$ for all $x \in \Sigma^+$ with $|x| \leq k$. The relation $\sim_k$ is clearly an equivalence

relation; we let $[u]_k$ denote the $k$-abelian equivalence class defined by $u$. A word $u$ is called a *$k$-abelian singleton* if $|[u]_k| = 1$.

In [9], $k$-abelian equivalence is characterized in terms of rewriting, namely by *$k$-switching*. For this we define the following. Let $k \geq 1$ and let $u \in \Sigma^*$. Suppose that there exist $x, y \in \Sigma^{k-1}$, not necessarily distinct, and indices $i, j, l$ and $m$, with $i < j \leq l < m$, such that $x$ has positions $i$ and $l$ in $u$ and $y$ has positions $j$ and $m$ in $u$. In other words, we have

$$u = u[1, i) \cdot u[i, j) \cdot u[j, l) \cdot u[l, m) \cdot u[m, |u|],$$

where both $u[i, |u|]$ and $u[l, |u|]$ begin with $x$ and both $u[j, |u|]$ and $u[m, |u|]$ begin with $y$. Furthermore, $u[i, j)$, $u[l, m) \neq \varepsilon$ but we allow $l = j$, in which case $y = x$ and $u[j, l) = \varepsilon$. We define a *$k$-switching* on $u$, denoted by $S_{u,k}(i, j, l, m)$, as

$$S_{u,k}(i, j, l, m) = u[1, i) \cdot u[l, m) \cdot u[j, l) \cdot u[i, j) \cdot u[m, |u|]. \tag{1}$$

A $k$-switching operation is illustrated in Figure 1.

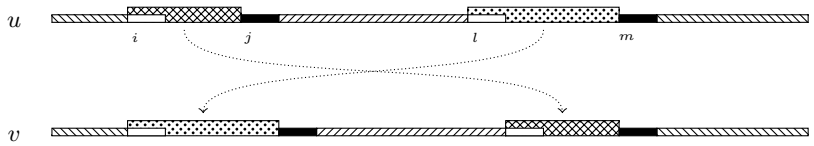


**Fig. 1.** Illustration of a $k$-switching. Here $v = S_{k,u}(i, j, l, m)$; the white rectangles symbolize $x$ and the black rectangles symbolize $y$.

*Example 1.* Let $u = aababababaaabab$ and $k = 4$. Let then $x = aba$, $y = bab$, $i = 2$, $j = 3$, $l = 4$ and $m = 11$. We then have

$$u = a \cdot a \cdot b \cdot ababaaa \cdot bab$$
$$S_{u,4}(i, j, l, m) = a \cdot ababaaa \cdot b \cdot a \cdot bab.$$

Note here that the occurrences of $x$ are overlapping. With $i = 2$, $j = l = 4$, and $m = 10$ we obtain the same word as above:

$$u = a \cdot ab \cdot ababaa \cdot abab$$
$$S_{u,4}(i, j, j, m) = a \cdot ababaa \cdot ab \cdot abab.$$

In this example we have $j = l$, whence $x = y = aba$ and $u[j, l) = \varepsilon$.

Let us define a relation $R_k$ of $\Sigma^*$ by $uR_kv$ if and only if $v$ is obtained from $u$ by a $k$-switching. Now $R_k$ is clearly symmetric, so that the reflexive and transitive closure $R_k^*$ of $R_k$ is an equivalence relation on $\Sigma^*$. In [9], $k$-abelian equivalence is characterized using $R_k^*$:

**Lemma 2.** *For $u, v \in \Sigma^*$, we have $u \sim_k v$ if and only if $uR_k^*v$.*

We need a few basic properties of *regular* (or *rational*) languages, such as equivalent definitions of regular languages with various models of finite automata, e.g., nondeterministic finite automata which can read the empty word ($\varepsilon$-NFA), and some basic closure properties of regular languages. We refer to [3] for this knowledge. In addition to classical language theoretical properties, we use the theory of *languages with multiplicities*. This counts how many times a word occurs in a language. This leads to the theory of $\mathbb{N}$-*rational sets*. Using the terminology of [16], a multiset over $\Sigma^*$ is called $\mathbb{N}$-*rational* if it is obtained from finite multisets by applying finitely many times the rational operations *product*, *union*, and taking *quasi-inverses*, i.e., *iteration* restricted to $\varepsilon$-free languages. Further, a unary $\mathbb{N}$-rational subset is referred to as an $\mathbb{N}$-*rational sequence*. We refer to [16] for more on this topic. The basic result we need is (see [16]):

**Proposition 3.** *Let $\mathcal{A}$ be a nondeterministic finite automaton over the alphabet $\Sigma$. The function $f_{\mathcal{A}} : \Sigma^* \to \mathbb{N}$ defined as*

$$f_{\mathcal{A}}(w) = \#\textit{of accepting paths of } w \textit{ in } \mathcal{A}$$

*is $\mathbb{N}$-rational. In particular, the function $\ell_{\mathcal{A}} : \mathbb{N} \to \mathbb{N}$,*

$$\ell_{\mathcal{A}}(n) = \#\textit{of accepting paths of length } n \textit{ in } \mathcal{A} \tag{2}$$

*is an $\mathbb{N}$-rational sequence. Consequently, the* generating function *for $\ell_{\mathcal{A}}$ is a rational function.*

## 3 Properties of $k$-Switchings

Our starting point for the study of structural properties of $k$-abelian equivalence classes is the characterization of $k$-abelian equivalence in terms of $k$-switchings. We proceed to describe a $k$-switching operation on languages. We show that this operation preserves regularity. That is, given a regular language $L$, the language obtained by this operation is also regular. This result will be used later on.

We now describe $k$-switchings on languages. For a language $L \subset \Sigma^*$, we define the $k$-switching of $L$, denoted by $R_k(L)$, as the language

$$R_k(L) = \{w \in \Sigma^* \mid wR_kv \text{ for some } v \in L\}.$$

Similarly, we define $R_k^*(L) = \bigcup_{n\in\mathbb{N}} R_k^n(L) = \bigcup_{w\in L}[w]_k$.

Note that, from a regular language $L$, it is straightforward to identify all words that admit a $k$-switching (i.e., the words on the top row of Figure 1). It is not at all clear that, by performing all possible $k$-switchings on all words of $L$ (i.e., taking the union of all words on the bottom row of Figure 1), the obtained language is also regular. We give a direct automata theoretic construction to show this.

**Theorem 4.** *Let $L$ be a regular language. Then $R_k(L)$ is also regular.*

*Proof.* For a language $L$ and fixed words $x, y \in \Sigma^{k-1}$, consider the language

$$\begin{aligned} R_{x,y}(L) = \{w \in \Sigma^* \mid w = & S_{k,u}(i,j,l,m) \text{ for some } i < j \leq l < m, u \in L, \\ & \text{with } u[i, i+k-1) = u[l, l+k-1) = x \text{ and} \\ & u[j, j+k-1) = u[m, m+k-1) = y\}. \end{aligned}$$

We will construct, for a regular language $L$ recognized by a deterministic finite automaton $\mathcal{A} = (Q, \Sigma, \delta, p_{\text{init}}, F)$, an $\varepsilon$-NFA $\hat{\mathcal{A}}$ which recognizes $R_{x,y}(L)$. The claim then follows for $R_k(L)$, as $R_k(L) = \bigcup_{x,y \in \Sigma^{k-1}} R_{x,y}(L)$ is a finite union of regular languages.

In essence, $\hat{\mathcal{A}}$ is a cartesian product of form $\hat{\mathcal{A}} = \mathcal{A}_1 \times \mathcal{A}_x \times \mathcal{A}_y \times \mathcal{A}_x \times \mathcal{A}_y$. The first component automaton $\mathcal{A}_1$ consists of $5|Q|^4$ copies of $\mathcal{A}$, some of which are connected by $\varepsilon$-transitions. The second and fourth components are copies of an automaton $\mathcal{A}_x$ recognizing the language $x\Sigma^*$ and the third and fifth components are copies of an automaton $\mathcal{A}_y$ recognizing the language $y\Sigma^*$. The components $2, 3, 4$, and $5$ are initiated according to the computations performed in $\mathcal{A}_1$. We shall now make this construction more formal.

We first construct $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, \tilde{p}_{\text{init}}, F_1)$ as follows. For each state $p \in Q$, we have $p^{(c,(p_1,p_2),(p_3,p_4))} \in Q_1$ for all $c = 1, \ldots, 5$ and $p_r \in Q$, $r = 1, \ldots, 4$. We also add the initial state $\tilde{p}_{\text{init}}$, from which we have $\varepsilon$-transitions to all the states of form $p_{\text{init}}^{(1,(p_1,p_2),(p_3,p_4))}$, $p_1, p_2, p_3, p_4 \in Q$. Thus the computation of $\mathcal{A}_1$ begins with an $\varepsilon$-transition. We then add the following $\varepsilon$-transitions for all $p_1, p_2, p_3, p_4 \in Q$:

$$\begin{aligned} &p_1^{(1,(p_1,p_2),(p_3,p_4))} \xrightarrow{\varepsilon} p_2^{(2,(p_1,p_2),(p_3,p_4))}, \quad p_3^{(2,(p_1,p_2),(p_3,p_4))} \xrightarrow{\varepsilon} p_4^{(3,(p_1,p_2),(p_3,p_4))}, \\ &p_2^{(3,(p_1,p_2),(p_3,p_4))} \xrightarrow{\varepsilon} p_1^{(4,(p_1,p_2),(p_3,p_4))}, \quad p_4^{(4,(p_1,p_2),(p_3,p_4))} \xrightarrow{\varepsilon} p_3^{(5,(p_1,p_2),(p_3,p_4))}. \end{aligned}$$

Otherwise the computation of $\mathcal{A}_1$ respects the original automaton, that is,

$$\delta_1(p^{(i,(p_1,p_2),(p_3,p_4))}, a) = q^{(i,(p_1,p_2),(p_3,p_4))}$$

if and only if there is a transition $\delta(p, a) = q$ in $\mathcal{A}$. Finally, $F_1$ consists of all states of form $f^{(5,(p_1,p_2),(p_3,p_4))}$, where $f \in F$ and $p_1, p_2, p_3, p_4 \in Q$.

We remark the following about $\mathcal{A}_1$. Firstly, once the first $\varepsilon$-transition is taken, the states $p_1, p_2, p_3$, and $p_4$ are fixed for the remainder of the computation. Secondly, the states $p_r$, $r = 1, \ldots, 4$, determine between which states an $\varepsilon$-transition can be performed. Furthermore, the parameter $c$ counts the number of $\varepsilon$-transitions performed. The parameters $c$, $p_1, p_2, p_3$, and $p_4$ together determine at which time and between which states an $\varepsilon$-transition can be performed.

We now describe the behavior of the rest of the component automata of $\hat{\mathcal{A}}$. For $s \in \{2, \ldots, 5\}$, the $s$th component automaton of $\hat{\mathcal{A}}$ is initiated during the $s$th $\varepsilon$-transition performed in $\mathcal{A}_1$ (the first $\varepsilon$-transition being the first computation step of $\mathcal{A}_1$). We also require from $\hat{\mathcal{A}}$ that, after the second and fourth $\varepsilon$-transition performed in $\mathcal{A}_1$, at least one letter is read before performing the next $\varepsilon$-transition. This is not required after the third $\varepsilon$-transition. Note that these requirements can be encoded, e.g., into the parameter $c$ of the states in $A_1$. Finally, $\hat{\mathcal{A}}$ accepts if and only if all its components are in accepting states.

We first show that $R_{x,y}(L) \subseteq L(\hat{\mathcal{A}})$. In order to see this, let $u \in L$ and let $v = S_{k,u}(i,j,l,m) \in R_{x,y}(L)$. Let $q_t$, $t = 1, \ldots, |u|$, denote the state $\delta(p_{\text{init}}, u[1,t))$ (note that some of the states $q_t$ can be the same). We then find an accepting computation of $\mathcal{A}_1$ for $v$ as follows. We first take the $\varepsilon$-transition from $\tilde{p}_{\text{init}}$ to the state $p_{\text{init}}^{(1,(q_i,q_l),(q_j,q_m))}$. After this, the computation is as in Figure 2 by following the dashed lines. The computation of $\mathcal{A}$ on $u$ follows the continuous lines. Note that the other components of $\hat{\mathcal{A}}$ also end up in accepting states, since by the definition of the $k$-switching $S_{k,u}(i,j,l,m)$, $x$ and $y$ have positions in $v$ corresponding to the initiations of the copies of the automata $\mathcal{A}_x$ and $\mathcal{A}_y$. Thus $R_{x,y}(L) \subseteq L(\hat{\mathcal{A}})$.

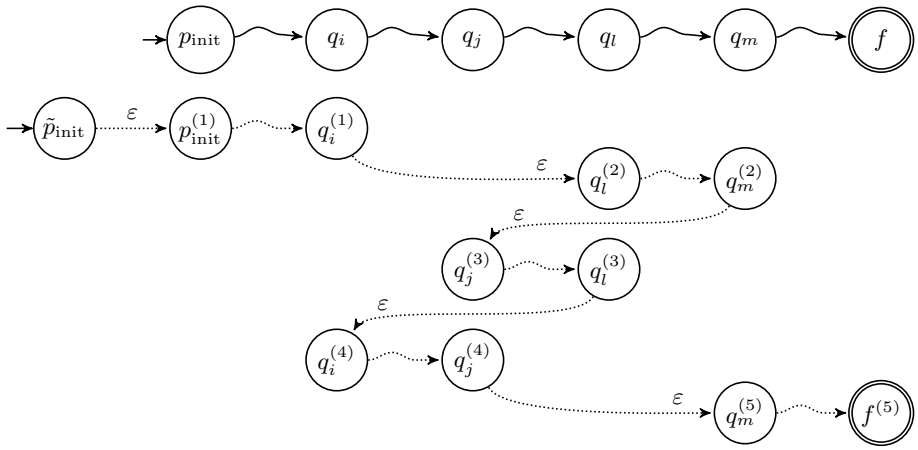


**Fig. 2.** The computation of automaton $\mathcal{A}$ on an accepted word $u$ (in continuous lines) and a computation of $\mathcal{A}_1$ on $S_{k,u}(i,j,l,m)$ (in dotted lines). We have abbreviated the states $q_r^{(c,(q_i,q_l),(q_j,q_m))}$ by $q_r^{(c)}$ (for $c \in \{1,\ldots,5\}$, $r \in \{\text{init}, i, j, l, m\}$).

We now show the converse. For this, let $v \in L(\hat{\mathcal{A}})$ and consider an accepting path of $\hat{\mathcal{A}}$ on $v$. By construction, the automaton $\mathcal{A}_1$ starts with an $\varepsilon$-transition to a state $p_{\text{init}}^{(1,(p_1,p_2),(p_3,p_4))}$. After this, the computation contains four more $\varepsilon$-transitions, suppose they occur just before reading the $i$th, $j$th, $l$th and $m$th letter, with $i < j \leq l < m$, respectively. (Here we use the requirement for not allowing an $\varepsilon$-transition immediately after the second and fourth $\varepsilon$-transitions.) Furthermore, by the acceptance of the other component automata of $\hat{\mathcal{A}}$, $x$ has positions $i$ and $l$, and $y$ has positions $j$ and $m$ in $v$. We claim that $u = S_{k,v}(i,j,l,m) \in L$. It then follows, by the symmetry of the $k$-switching relation, that $v \in R_{x,y}(L)$. Indeed, turning back to the computation of $\mathcal{A}_1$ on $v$, we obtain the following paths in $\mathcal{A}$:

1. a path from $p_{\text{init}}$ to $p_1$ labeled by $v[1,i)$,

2. a path from $p_2$ to $p_3$ labeled by $v[i, j)$,
3. a path from $p_4$ to $p_2$ labeled by $v[j, l)$,
4. a path from $p_1$ to $p_4$ labeled by $v[l, m)$, and
5. a path from $p_3$ to an accepting state of $\mathcal{A}$ labeled by $v[m, |v|]$.

Thus $u = v[1, i)v[l, m)v[j, l)v[i, j)v[m, |v|] \in L$, as was claimed. □

*Remark 5.* This result may also be proved using MSO logic for words, as suggested by one of the anonymous referees.

The following example shows that the family of regular languages is not closed under the language operation $R_k^*$.

*Example 6.* Fix $k \geq 1$ and let $L = (ab^k)^+$. It is straightforward to verify by, e.g., comparing the number of occurrences of factors of length $k$, that

$$R_k^*(L) = \left\{ ab^{r_1}ab^{r_2} \cdots ab^{r_n} \mid n \geq 1, r_i \geq k-1, \sum_{i=1}^{n} r_i = nk \right\}.$$

Let now $h$ be a morphism defined by $h(a) = ab^{k-1}$ and $h(b) = b$. It is again straightforward to show that $h^{-1}(R_k^*(L)) = \{w \in a\{a, b\}^* \mid |w|_a = |w|_b\}$, which is clearly not regular. It follows that $R_k^*(L)$ is not regular.

## 4 On the Number of $k$-Abelian Equivalence Classes

In this section we focus on the number $\mathcal{P}_{k,m}(n)$ of $k$-abelian equivalence classes of words of length $n$ over $\Sigma$, $|\Sigma| = m$, where $k$ and an $m$ are fixed. We first recall a result from [11]:

**Theorem 7.** *We have, for $k$ and $m$ fixed, $\mathcal{P}_{k,m}(n) = \Theta(n^{m^{k-1}(m-1)})$, where the constants in $\Theta$ depend on $k$ and $m$.*

We are also interested in the number $\mathcal{S}_{k,m}(n)$ of $k$-abelian singletons of length $n$ over $\Sigma$, $|\Sigma| = m$, where $k$ and an $m$ are fixed. We recall a result proved in [9].

**Theorem 8.** *For $k$ and $m$ fixed, we have $\mathcal{S}_{k,m}(n) = \mathcal{O}(n^{N_m(k-1)-1})$, where the constants in $\mathcal{O}$ depend on $k$ and $m$. Here $N_m(l) = \frac{1}{l}\sum_{d|l} \varphi(d)m^{l/d}$ is the number of conjugacy classes (or necklaces) of words in $\Sigma^l$, where $|\Sigma| = m$.*

The main result of this section is the following:

**Theorem 9.** *The sequences $\mathcal{P}_{k,m}(n)$ and $\mathcal{S}_{k,m}(n)$ are $\mathbb{N}$-rational.*

In order to prove this, we define the following languages. Here $\leq$ denotes a lexicographic ordering of $\Sigma^*$.

$$\begin{aligned} L_{\min} &= \{w \in \Sigma^* \mid w \leq u \text{ for all } w \sim_k u\}, \\ L_{\max} &= \{w \in \Sigma^* \mid w \geq u \text{ for all } w \sim_k u\}, \text{ and} \\ L_{\text{sing}} &= \{w \in \Sigma^* \mid |[w]_k| = 1\}. \end{aligned}$$

In other words, $L_{\min}$ (resp., $L_{\max}$) is the language of lexicographically minimal (resp., maximal) representatives of $k$-abelian equivalence classes, while $L_{\text{sing}}$ is the language of $k$-abelian singletons. We also recall a technical lemma from [9], a refinement of Lemma 2.

**Lemma 10.** *Let $u \sim_k v$ with $u \neq v$. Let $p$ be the longest common prefix of $u$ and $v$. Then there exists $z \in \Sigma^*$ such that $zR_k u$ and the longest common prefix of $z$ and $v$ has length at least $|p| + 1$.*

**Lemma 11.** *The languages $L_{\min}$, $L_{\max}$, and $L_{sing}$ are regular languages.*

*Proof.* Let $u$ be the minimal element in $[u]_k$. If there exists a $k$-switching on $u$ which yields a new element, it has to be lexicographically greater than $u$. In particular, $u$ does not contain factors from the language

$$((xb\Sigma^* \cap \Sigma^* y)\,\Sigma^* \cap \Sigma^* x)\,a\Sigma^* \cap \Sigma^* y,$$

where $x, y \in \Sigma^{k-1},\ a, b \in \Sigma,\ a < b$. On the other hand, by the above lemma, any word $u$ avoiding such factors is lexicographically least in $[u]_k$. We thus have

$$L_{\min} = \bigcap_{\substack{x,y\in\Sigma^{k-1}\\ a,b\in\Sigma,\ a<b}} \overline{\Sigma^*\,(((xb\Sigma^* \cap \Sigma^* y)\,\Sigma^* \cap \Sigma^* x)\,a\Sigma^* \cap \Sigma^* y)\,\Sigma^*}, \tag{3}$$

where, for a regular expression $R$, $\overline{R}$ denotes the *complement* language $\Sigma^* \setminus R$.

Similarly, for $L_{\max}$, by reversing $a < b$ to $a > b$ in (3), we obtain the claim.

Finally, $L_{\text{sing}} = L_{\min} \cap L_{\max}$ so that $L_{\text{sing}}$ is regular. Another, perhaps more informative, way to see this is as follows: for $k$-abelian singletons, we are avoiding all possible $k$-switchings that give a different word. By requiring $a \neq b$, as opposed to $a < b$, in (3), we obtain the expression for $L_{\text{sing}}$. □

*Proof (of Theorem 9).* Consider first the language $L_{\min}$ and a DFA $\mathcal{A}$ recognizing it. We transform the automaton to a unary NFA $\mathcal{A}'$ by identifying all input letters. Since $\mathcal{A}$ is deterministic, the transformation is *faithful*, that is, for each word $w$ accepted by $\mathcal{A}$, there exists a unique corresponding accepting path in $\mathcal{A}'$, and vice versa. By the construction of $\mathcal{A}'$, $\ell_{\mathcal{A}'}(n) = \mathcal{P}_{k,m}(n)$ for all $n \in \mathbb{N}$, from which the claim follows for $\mathcal{P}_{k,m}$. The case for $\mathcal{S}_{k,m}$ is similar. □

*Remark 12.* Let $A$ be the adjacency matrix of the unary automaton $\mathcal{A}'$ described above. It is known that, for all large enough $n$,

$$\ell_{\mathcal{A}'}(n) = \sum_{\lambda \in Eig(A)} p_\lambda(n)\lambda^n \tag{4}$$

where the summation is taken over all distinct eigenvalues of $A$, and $p_\lambda$ is a complex polynomial of degree at most $\mu_\lambda - 1$. Here $\mu_\lambda$ is the multiplicity of $\lambda$ as a root of the *minimal polynomial* of $A$ (see for instance [3,17]).

### 4.1 Complexities for Small Values of $k$ and $m$

We now give some examples illustrating the results obtained above for small values of $k$ and $m$. We also compute closed formulas for $\mathcal{P}_{k,m}$ and $\mathcal{S}_{k,m}$ for some small values of $k$ and $m$.

*Example 13.* In figure Figure 3, we have two minimal DFAs, one recognizing the minimal representatives of 2-abelian equivalence classes and the other recognizing 2-abelian singletons over $\Sigma = \{a, b\}$. The sink states are not included in the figures. We also note that all other states are accepting, since the languages are defined by avoiding certain patterns.

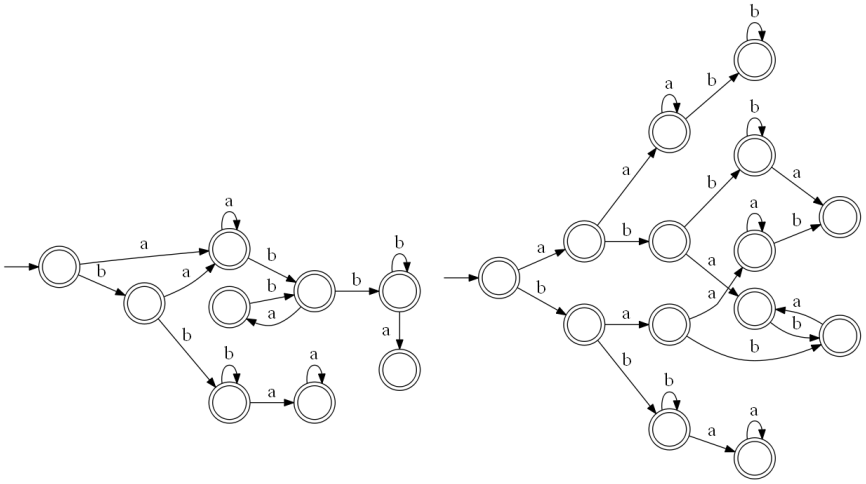

**Fig. 3.** DFAs recognizing the minimal representatives of 2-abelian equivalence classes (left) and 2-abelian singletons (right) over the alphabet $\{a, b\}$.

Using the idea of the proof of Theorem 9, we first construct deterministic automata for $L_{\min}$ and $L_{\text{sing}}$ for small $k$ and $m$. We then use the automata to compute the function $\ell$ as in Remark 12. We state these conclusions without proofs:

**Proposition 14.**

$$\begin{aligned}
&\textit{For all } n \geq 1,\ \mathcal{P}_{2,2}(n) = n^2 - n + 2,\\
&\textit{for all } n \geq 2,\ \mathcal{P}_{2,3}(n) = \tfrac{1}{18}n^4 - \tfrac{5}{18}n^3 + \tfrac{65}{36}n^2 - \tfrac{23}{6}n - \tfrac{1}{8}(-1)^n + \\
&\qquad\qquad + \tfrac{2}{27}e^{-\frac{\pi i}{3}}(e^{\frac{2\pi i}{3}})^n + \tfrac{2}{27}e^{\frac{\pi i}{3}}(e^{-\frac{2\pi i}{3}})^n + \tfrac{1307}{216},\ \textit{and}\\
&\textit{for all } n \geq 4,\ \mathcal{P}_{3,2}(n) = \tfrac{1}{960}n^6 + \tfrac{7}{320}n^5 + \tfrac{67}{384}n^4 - \tfrac{19}{32}n^3 + \tfrac{1457}{480}n^2 - \\
&\qquad\qquad - (\tfrac{1569}{640} + \tfrac{3}{128}(-1)^n)n + \tfrac{741}{256} + \tfrac{27}{256}(-1)^n.
\end{aligned}$$

**Proposition 15.**

*For all* $n \geq 4$, $\mathcal{S}_{2,2}(n) = 2n + 4$,

*for all* $n \geq 6$, $\mathcal{S}_{2,3}(n) = 3n^2 + 27n - 63$, *and*

*for all* $n \geq 9$, $\mathcal{S}_{3,2}(n) = \frac{1}{2}n^2 + 16n + \frac{2}{3}(e^{\frac{2\pi i}{3}n} + (e^{-\frac{2\pi i}{3}})^n) - \frac{535}{12} - \frac{3}{4}(-1)^n$.

The formulae for $\mathcal{P}_{2,2}$ and $\mathcal{S}_{2,2}$ have previously been proved, using different methods, in [5] and [9], respectively. We note that Eero Harmaala (private communication) has previously computed the values for $\mathcal{P}_{2,3}$ and $\mathcal{P}_{3,2}$ ($n = 2, \ldots, 18$ and $n = 4, \ldots, 21$, respectively). We also note that computing the first few values of $S_{2,3}(n)$ and $S_{3,2}(n)$ is an easy task. The *On-Line Encyclopedia of Integer Sequences* (`http://oeis.org`, accessed June 10, 2016) does not contain any of the above sequences.

The methods used here are far from being practical for computing closed formulae for larger values of $k$ and $m$, as is illustrated by the following example.

*Example 16.* For the binary alphabet, the number of states in the minimal DFA recognizing $L_{\min}$ for $k = 2, 3, 4$ is $10, 49$, and $936$, respectively. This makes computing a closed formula for $\mathcal{P}_{4,2}$ already a computationally challenging problem.

*Remark 17.* The exponential blow-up of the computation time is due to complementation and non-determinism of the automata obtained from the regular expressions (3). Also, by Theorem 7, the automaton obtained from (3) has to grow necessarily exponentially with respect to $k$ when the alphabet is fixed; some of the polynomials $p_\lambda$ in (4) have degree $m^{k-1}(m-1)$.

For the case of $k$-abelian singletons, Theorem 8 does not give a large blow-up immediately, though in [9] it is conjectured that $\mathcal{S}_{k,m}(n) = \Theta(n^{N_m(k-1)-1})$, which would also yield a large blow-up in the number of states.

## 5 Towards a Structure of Fixed Sized Equivalence Classes

The regularity of the languages $L_{\min}$ and $L_{\text{sing}}$ raises questions for the structure of larger equivalence classes. We are thus interested in the $k$-abelian equivalence classes of fixed cardinality. We employ the result of Theorem 4 to obtain a first step in this direction.

**Proposition 18.** *The language* $L_2 = \{w \in \Sigma^* \mid |[w]_k| = 2\}$ *is a regular language.*

*Proof.* Consider the regular language $L = \Sigma^* \setminus (L_{\min} \cup L_{\max})$: we have

$$L = \{w \in \Sigma^* \mid |[w]_k| \geq 3 \text{ and } w \text{ is not minimal or maximal}\},$$

since all classes containing at most two elements are removed. By Lemma 2, $R_k(R_k(L)) \cup R_k(L) \cup L$ then gives exactly the language

$$L' = \{w \in \Sigma^* \mid |[w]_k| \geq 3\},$$

and by Lemma 2, $L'$ is regular. Finally, the complement of $L'$ is the language $\{w \in \Sigma^* \mid |[w]_k| \leq 2\}$. We thus have that $L_2 = \overline{L'} \setminus L_{\text{sing}}$ is a regular language. □

Larger classes were not considered here, but we have no reason to suspect that the corresponding languages would not be regular. In fact, we suspect that modifications of Theorem 4 could yield methods, similar to the ones used in the above, to obtain some structure of larger classes.

## 6 Open Problems and Future Research

The topic of this paper opens up new aspects of $k$-abelian equivalence, and presents a series of questions. Though explicit formulas for the functions $\mathcal{P}_{k,m}$ and $\mathcal{S}_{k,m}$ were obtained, it remains to compute the corresponding generating functions (which, by our results, are rational functions).

To conclude, we suggest the following open problems.

- What are the generating functions for $\mathcal{P}_{k,m}$ and $\mathcal{S}_{k,m}$?
- When is $\mathcal{P}_{k,m}(n) \sim Cn^{m^{k-1}(m-1)}$ for some constant $C$? This is the case for small values of $k$ and $m$.
- Is the language of words $w$ having $|[w]_k| = l$, where $l$ is a fixed constant, a regular language? For $l = 2$, this is settled in the positive by Proposition 18.

## Acknowledgments

The automata used to calculate the functions in Proposition 14 and Proposition 15 were constructed using the java package `dk.brics.automaton` [14]. The automata in Figure 3 were created using the software Graphviz [4]. We would like to thank the anonymous referees for valuable comments which helped to improve the presentation.

## References

1. Cassaigne, J., Karhumäki, J., Saarela, A.: On Growth and Fluctuation of $k$-Abelian Complexity. In: Computer Science - Theory and Applications - 10th International Computer Science Symposium in Russia, CSR 2015, Listvyanka, Russia, July 13-17, 2015, Proceedings. pp. 109–122 (2015), http://dx.doi.org/10.1007/978-3-319-20297-6_8
2. Ehlers, T., Manea, F., Mercas, R., Nowotka, D.: $k$-Abelian pattern matching. Journal of Discrete Algorithms 34, 37–48 (2015), http://dx.doi.org/10.1016/j.jda.2015.05.004
3. Eilenberg, S.: Automata, Languages, and Machines, vol. A. Academic Press, Inc., New York, New York, USA (1974)
4. Gansner, E.R., North, S.C.: An open graph visualization system and its applications to software engineering. SOFTWARE - PRACTICE AND EXPERIENCE 30(11), 1203–1233 (2000), http://www.graphviz.org

5. Huova, M., Karhumäki, J., Saarela, A., Saari, K.: Local Squares, Periodicity and Finite Automata. In: Rainbow of Computer Science - Dedicated to Hermann Maurer on the Occasion of His 70th Birthday. pp. 90–101 (2011), http://dx.doi.org/10.1007/978-3-642-19391-0_7
6. Huova, M., Saarela, A.: Strongly $k$-Abelian Repetitions. In: Combinatorics on Words - 9th International Conference, WORDS 2013, Turku, Finland, September 16-20. Proceedings. pp. 161–168 (2013), http://dx.doi.org/10.1007/978-3-642-40579-2_18
7. Karhumäki, J.: Generalized Parikh Mappings and Homomorphisms. Information and Control 47(3), 155–165 (1980), http://dx.doi.org/10.1016/S0019-9958(80)90493-3
8. Karhumäki, J., Puzynina, S.: On $k$-Abelian Palindromic Rich and Poor Words. In: Developments in Language Theory - 18th International Conference, DLT 2014, Ekaterinburg, Russia, August 26-29, 2014. Proceedings. pp. 191–202 (2014), http://dx.doi.org/10.1007/978-3-319-09698-8_17
9. Karhumäki, J., Puzynina, S., Rao, M., Whiteland, M.A.: On Cardinalities of $k$-Abelian Equivalence Classes. Theoretical Computer Science (in press)
10. Karhumäki, J., Puzynina, S., Saarela, A.: Fine and Wilf's Theorem for $k$-Abelian Periods. International Journal of Foundations of Computer Science 24(7), 1135–1152 (2013), http://dx.doi.org/10.1142/S0129054113400352
11. Karhumäki, J., Saarela, A., Zamboni, L.Q.: On a generalization of Abelian equivalence and complexity of infinite words. Journal of Combinatorial Theory, Series A 120(8), 2189–2206 (2013), http://dx.doi.org/10.1016/j.jcta.2013.08.008
12. Karhumäki, J., Saarela, A., Zamboni, L.Q.: Variations of the Morse-Hedlund Theorem for $k$-Abelian Equivalence. In: Developments in Language Theory - 18th International Conference, DLT 2014, Ekaterinburg, Russia, August 26-29, 2014. Proceedings. pp. 203–214 (2014), http://dx.doi.org/10.1007/978-3-319-09698-8_18
13. Lothaire, M. (ed.): Combinatorics on Words. Cambridge University Press, second edn. (1997), http://dx.doi.org/10.1017/CBO9780511566097, Cambridge Books Online
14. Møller, A.: dk.brics.automaton – finite-state automata and regular expressions for Java (2010), http://www.brics.dk/automaton/
15. Rao, M., Rosenfeld, M.: Avoidability of long $k$-abelian repetitions. Mathematics of Computation (Published electronically: February 18, 2016), http://dx.doi.org/10.1090/mcom/3085
16. Salomaa, A., Soittola, M.: Automata-Theoretic Aspects of Formal Power Series. Texts and Monographs in Computer Science, Springer (1978), http://dx.doi.org/10.1007/978-1-4612-6264-0
17. Weintraub, S.H.: Jordan Canonical Form: Theory and Practice. Synthesis Lectures on Mathematics & Statistics, Morgan & Claypool Publishers (2009), http://dx.doi.org/10.2200/S00218ED1V01Y200908MAS006