Every Binary Pattern of Length Greater Than 14 Is Abelian-2-Avoidable

Matthieu Rosenfeld

Lip, ENS de lyon, 46 Allée d'Italie Lyon 69364 France; and CNRS, UCBL, Université de Lyon
matthieu.rosenfeld@ens-lyon.fr

Abstract

We show that every binary pattern of length greater than 14 is abelian-2-avoidable. The best known upper bound on the length of abelian-2-unavoidable binary pattern was 118, and the best known lower bound is 7.

We designed an algorithm to decide, under some reasonable assumptions, if a morphic word avoids a pattern in the abelian sense. This algorithm is then used to show that some binary patterns are abelian-2-avoidable. We finally use this list of abelian-2-avoidable pattern to show our result. We also discuss the avoidability of binary patterns on 3 and 4 letters.

1998 ACM Subject Classification G.2.1 Combinatorial Algorithms

Keywords and phrases Combinatorics on words; Pattern avoidance; Abelian repetitions

Digital Object Identifier 10.4230/LIPIcs.MFCS.2016.81

1 Introduction

The avoidability of patterns in words has been widely studied since the work of Thue on avoidability of repetitions [22, 23]. Thue wanted to know whether, for any word u and long enough word w, there is always a non-erasing morphism h such that $h(u)$ is a factor of w. He answered negatively to the question by constructing an infinite word over three letters that does not contain any image of AA, and an infinite word over two letters that does not contain any image of AAA.

The formal notion of pattern was introduced in [2]. For two words P and w, we say that w avoids the pattern P if there is no non-erasing morphism h such that $h(P)$ is a factor of w, or equivalently if there is no factor $w_1w_2\dots w_{|P|}$ in w such that $\forall i,j,\ P_i = P_j \implies w_i = w_j$. The avoidability of patterns was studied by Zimin [24] and many other authors worked on the classification of avoidable patterns [4, 14, 15, 20, 21]. In particular Roth proved in [20] that binary patterns of length greater than 6 are avoidable over the binary alphabet and in [1] authors showed the existence of a pattern avoidable over 4 letters, but not avoidable over 3 letters. More recently it has been showed that patterns with m different letters of length at least $3(2^{m-1})$ are 2-avoidable [3, 16].

Erdős proposed a commutative version of the results of Thue [8, 9]. An *abelian square* is any non-empty word uv where u and v are permutations of each other. Erdős asked whether there is an infinite abelian-square-free word over an alphabet of size 4 [8, 9]. After some intermediary results (alphabet of size 25 by Evdokimov [10] and size 5 by Pleasant [17]), Keränen answered positively to Erdős's question by giving a 85-uniform morphism (found with the assistance of a computer) whose fixed point is abelian-square-free [11]. Moreover, Dekking showed that it is possible to avoid abelian cubes on a ternary alphabet and abelian-4-powers over a binary alphabet [7].

41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016).
Editors: Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier; Article No. 81; pp. 81:1–81:11
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Following the question of Erdős we say that two words u and v are *abelian equivalent*, denoted $u \approx_a v$, if they are permutations of each other, for example: *listen* $\approx_a$ *silent*. Let $P = P_1 P_2 \dots P_n$ be a pattern, where the P_i are letters. Then we say that a word $w \in \Sigma^*$ *realizes P in the abelian sense* if there are $w_1, \dots, w_n \in \Sigma^+$ such that $w = w_1 w_2 \dots w_n$ and $\forall i, j,\ P_i = P_j \implies w_i \approx_a w_j$. If a word w has no factor that realizes a pattern P in the abelian sense, then w *avoids P* in the abelian sense, w is *abelian-P-free*. In Section 2 we show that one can decide if the fixed point of a morphism avoids a given pattern. This generalizes a result from [5] that tells that under some conditions one can decide if the fixed point of a morphism avoids abelian-k-powers.

We say that a pattern is *abelian-k-avoidable* if there is a word from an alphabet of size k that avoids this pattern. For any pattern $P \in \Delta^*$, the *abelian-avoidability index* of P (denoted by $\lambda_a(P)$) is the smallest integer k such that P is abelian-k-avoidable or ∞ if there is no such k. It is an abelian analog of the usual avoidability index of a pattern P. For example $\lambda_a(ABA) = \lambda_a(ABACABA) = \infty$, $\lambda_a(AA) = 4$ [11], $\lambda_a(AAA) = 3$ and $\lambda_a(AAAA) = 2$ [7]. In [6] authors showed that binary pattern of length greater than 118 are abelian-2-avoidable and asked for a more precise characterization. We can use the algorithm of Section 2 to show that every binary pattern of length greater than 14 are abelian-2-avoidable.

In Section 2 we explain how to decide, under some conditions, whether a morphic word avoids a given pattern in the abelian sense. In Section 3 we show that binary patterns of length greater than 14 are abelian-2-avoidable. We also discuss the avoidability of binary patterns over any finite alphabet and we raise some open questions.

2 Proving the decidability

In this part we explain how to decide, under some conditions, if the fixed point of a morphism avoids some given pattern in the abelian sense.

We use terminology and notations of Lothaire [13]. For any morphism $h : \Sigma^* \mapsto \Sigma^*$, if there is $a \in \Sigma$ and $w \in \Sigma^*$ such that $h(a) = aw$, then the sequence $(h^n(a))_{n \geq 0}$ converges for the usual topology on $\Sigma^* \cup \Sigma^\omega$ and we denote by $h^\omega(a) = \lim_{n \to 0} h^n(a)$. Note that $h^\omega(a)$ is a fixed point of h. To any pattern P we associate the function $\varphi_P : [1, |P|] \mapsto [1, |P|]$ such that $\varphi_P(i) = \min\{j : P_j = P_i\}$ is the position of the first occurrence of the letter P_i in P.

For any word w, we denote by $[w]_i$ the letter at position i in w (or w_i if it is clear in the context). For any word w, we denote by $|w|$ the length of w and for any letter $a \in \Sigma$, $|w|_a$ is the number of occurrences of a in w. The *Parikh vector* of a word $w \in \Sigma^*$, denoted by $\Psi(w)$, is the vector indexed by Σ such that for every $a \in \Sigma$, $\Psi(w)[a] = |w|_a$. Note that by definition for any two words u and v, $u \approx_a v$ iff $\Psi(u) = \Psi(v)$.

We associate to any morphism $h : \Sigma^* \mapsto \Sigma^*$ a matrix M_h on $\Sigma \times \Sigma$ such that $(M_h)_{a,b} = |h(b)|_a$. Note that from the definition we can deduce for any morphism h and any word u the formula:

$$\Psi(h(u)) = M_h \Psi(u).$$

The induced norm of a matrix $M \in \mathbb{R}_{m \times m}$ is given by $\|M\|_2 = sup_{x \in \mathbb{R}^m} \frac{\|Mx\|_2}{\|x\|_2}$ where $\|v\|_2$ is the Euclidean norm of v.

We generalize the notion of k-templates introduced in [5] in order to show Theorem 1. Let Δ and Σ be two alphabets, and let P be a pattern over Δ, then a *P-template* over Σ is a $2(|P|+1)$-tuple of the form: $[w_0, v_1, w_1, v_2 \dots, v_{|P|}, w_{|P|}]$ where for all i, $w_i \in \Sigma^*$, $v_i \in \mathbb{Z}^{|\Sigma|}$. A word $w \in \Sigma^*$ *realizes* (or *is a realization* of) a P-template $t = [w_0, v_1, \dots, v_{|P|}, w_{|P|}]$ if there are $u_1, \ldots, u_{|P|} \in \Sigma^+$ such that $w = w_0 u_1 w_1 u_2 w_2 \ldots u_{|P|} w_{|P|}$ and $\forall i, j$ $P_i = P_j \implies \Psi(u_i) - \Psi(u_j) = v_i - v_j$.

Each template can be associated to its set of realizations, but many templates are associated to the same set. We say that a P-template is *normalized* if for all $i \in [1, |P|]$, $\varphi_P(i) = i \implies v_i = \overrightarrow{0}$. For any P-template, we can compute a normalized template that is realized by the same set of words. Since one doesn't change the set of realizations by adding the same vector to all vectors corresponding to the same letter, one get the *normalization* of a template by taking for all i $v'_i = v_i - v_{\varphi_P(i)}$ and $w'_i = w_i$. Note that there is a natural bijection between the set of k-templates from [5] and the set of normalized A^k-templates. In the following we only use normalized templates.

We say that a morphism $h : \Sigma^* \to \Sigma^*$ is *convenient* if its associated matrix M_h is invertible, $\|M_h^{-1}\|_2 < 1$ and $\forall a \in \Sigma$, $|h(a)| > 1$. We can now state the main theorem:

▶ **Theorem 1.** *For any alphabets Δ and Σ, pattern $P \in \Delta^*$, P-template t, convenient morphism $h : \Sigma^* \mapsto \Sigma^*$ and any letter $a \in \Sigma$ such that $h(a) = as$ for some s, it is possible to decide if $h^\omega(a)$ avoids t.*

By definition, w avoids P if and only if w avoids the P-template $[\varepsilon, \overrightarrow{0}, \varepsilon, \ldots, \overrightarrow{0}, \varepsilon, \overrightarrow{0}, \varepsilon]$. From that we can deduce the following corollary:

▶ **Corollary 2.** *For any alphabets Δ and Σ, pattern $P \in \Delta^*$, any convenient morphism $h : \Sigma^* \mapsto \Sigma^*$ and any letter $a \in \Sigma$ such that $h(a) = as$ for some s, it is possible to decide if $h^\omega(a)$ is abelian-P-free.*

The rest of this section is devoted to the proof of Theorem 1. The main idea of the proof is that we can compute S, a set of templates, such that $t \in S$ and $h^{n+1}(a)$ avoids any template of S if and only if $h^n(a)$ avoids any template of S. Thus $h^\omega(a)$ avoids t if and only if a avoids any template of S which is easy to check. The set S corresponds to what we call *the set of special ancestors*. In the following Δ will always be the alphabet of patterns and Σ the alphabet of words and templates.

2.1 Parents and ancestors of a template

Let $t = [w_0, v_1, \ldots, v_{|P|}, w_{|P|}]$ and $t' = [w'_0, v'_1, \ldots, v'_{|P|}, w'_{|P|}]$ be two normalized P-templates. We say that *t' is a parent of t by h* if there are $p_0, s_0, \ldots, p_{|P|}, s_{|P|} \in \Sigma^*$ such that:

- $\forall i \in [0, |P|]$, p_i is a prefix of the image of the first letter of w'_i (a prefix of $h([w'_i]_1)$), s_i is a suffix of the image of the last letter of w'_i and $h(w'_i) = p_i w_i s_i$,
- $\forall i, j \in [1, |P|]$, $P_i = P_j \implies v_i - v_j = (\Psi(s_{i-1}) + M_h v'_i + \Psi(p_i)) - (\Psi(s_{j-1}) + M_h v'_j + \Psi(p_j))$.

For any normalized template t we denote $\mathrm{Par}_h(t)$ the set of parents of t by h. The *ancestors of t by h* is the set $\mathrm{Ancestors}_h(t) = \cup_{i=0}^{\infty} \mathrm{Par}_h^i(t)$. The relation "is an ancestor" is the transitive and reflexive closure of the relation "is a parent".

Lemmas 3, 4 and 5 tell us that the set of ancestors of a template is computable.

▶ **Lemma 3.** *For any convenient morphism $h : \Sigma^* \mapsto \Sigma^*$ and normalized P-template t, the set $\mathrm{Par}_h(t)$ is finite and computable.*

Proof. Since the template t is normalized we know that:

$$M_h v'_i = \begin{cases} 0 & \text{if } \varphi_P(i) = i \\ v_i - \Psi(s_{i-1}) - \Psi(p_i) + \Psi(s_{\varphi_P(i)-1}) + \Psi(p_{\varphi_P(i)}) & \text{if } \varphi_P(i) \neq i. \end{cases}$$

Since M_h is invertible, there is at most one parent for a given valuation of $(w'_i)_{0\le i\le |P|}$, $(s_i)_{0\le i\le |P|}$ and $(p_i)_{0\le i\le |P|}$. Moreover the possibilities for the s_i, p_i and hence for the w'_i are finite (if h is injective there is at most one possibility for each w'_i). So we can try all the valuations for $(w'_i)_{0\le i\le |P|}$, $(s_i)_{0\le i\le |P|}$ and $(p_i)_{0\le i\le |P|}$ and we get all the parents. ◀

▶ **Lemma 4.** *For any convenient morphism $h : \Sigma^* \mapsto \Sigma^*$ and normalized P-template t there are $(r_1, \dots, r_{|P|}) \in \mathbb{R}^+$ such that if $t' = [w'_0, v'_1, w'_1, v'_2 \dots, v_{|P|}, w_{|P|}]$ is an ancestor of t by h then for all i $\|v'_i\|_2 < r_i$.*

We omit the details of the proof of Lemma 4 which is similar to the proof of Lemma 4 in [5]. Let v_i be the i-th vector of t, then $v'_i = M^{-n}v_i + \sum_{j=0}^{n-1} M^{-j}(\Psi(s_j) + \Psi(p_j) - \Psi(s'_j) - \Psi(p'_j))$ for some s_j, s'_j and p_j, p'_j being respectively suffixes and prefixes of images of letters by h. Moreover $\|M_h^{-1}\|_2 < 1$, so $\|v'_i\|_2$ is bounded.

▶ **Lemma 5.** *For any normalized P-template t the set of ancestors of t by h is finite and computable.*

Proof. Let $t' = [w_0, v_1, w_1, v_2 \dots, v_{|P|}, w_{|P|}]$ be an ancestor of t by h. From Lemma 4, each of the v_i is bounded and since $v_i \in \mathbb{Z}^{|\Sigma|}$ there are finitely many choices for each of the v_i. Moreover, since for all $a \in \Sigma$, $|h(a)| > 1$, the length of the w_i is bounded and there are finitely many different values for the w_i. It implies that there are only finitely many possible ancestors.

In order to compute the set of ancestors, one starts with the singleton $S = \{t\}$ and repeats the operation $S = S \cup \mathrm{Par}_h(S)$ (computable thanks to Lemma 3) until S reaches a fixed point, which will eventually happen since the set of ancestors is finite. ◀

▶ **Lemma 6.** *For any word w and any P-templates t and $t' \in \mathrm{Par}_h(t)$, if w is a realization of t' then $h(w)$ contains a realization of t.*

Proof. Let $t = [w_0, v_1, \dots, v_{|P|}, w_{|P|}]$ and $t' = [w'_0, v'_1, \dots, v'_{|P|}, w'_{|P|}] \in \mathrm{Par}_h(t)$. Then by definition there are $p_0, s_0, \dots, p_{|P|}, s_{|P|} \in \Sigma^*$ such that:

- $\forall i \in [1, |P|]$, $h(w'_i) = p_i w_i s_i$,
- $\forall i, j \in [1, |P|]$, $P_i = P_j \implies v_i - v_j = (\Psi(s_{i-1}) + M_h v'_i + \Psi(p_i)) - (\Psi(s_{j-1}) + M_h v'_j + \Psi(p_j))$.

Assume there is a word w realizing t'. Then there are $u'_1, \dots, u'_{|P|} \in \Sigma^+$ such that $w = w'_0 u'_1 w'_1 u'_2 w'_2 \dots u'_{|P|} w'_{|P|}$ and $\forall i, j$ $P_i = P_j \implies \Psi(u'_i) - \Psi(u'_j) = v'_i - v'_j$. Then $h(w) = p_0 w_0 s_0 h(u'_1) p_1 w_1 \dots s_{|P|-1} h(u'_{|P|}) p_{|P|} w_{|P|} s_{|P|}$. For all i let $u_i = s_{i-1} h(u'_i) p_i$, then $W = w_0 u_1 w_1 u_2 w_2 \dots u_{|P|} w_{|P|}$ is a factor of $h(w)$.

Moreover for all $i, j \in [1, |P|]$, if $P_i = P_j$ then:

$$\begin{aligned}
\Psi(u_i) - \Psi(u_j) &= \Psi(s_{i-1} h(u'_i) p_i) - \Psi(s_{j-1} h(u'_j) p_j) \\
&= (\Psi(s_{i-1}) + \Psi(h(u'_i)) + \Psi(p_i)) - (\Psi(s_{j-1}) + \Psi(h(u'_j)) + \Psi(p_j)) \\
&= (\Psi(s_{i-1}) + M_h v'_i + \Psi(p_i)) - (\Psi(s_{j-1}) + M_h v'_j + \Psi(p_j)) \\
&= v_i - v_j
\end{aligned}$$

So W realizes t and is a factor of $h(w)$. ◀

It tells us that if one of the ancestors of t is not avoided by $h^n(a)$ for some $n \in \mathbb{N}$, then there is $m > n$ such that t is not avoided by $h^m(a)$.

2.2 Specializations of a template

Let $P \in \Delta^*$ and $L \subseteq \Delta$ then we denote by $P_{|L}$ the pattern which is obtained by deleting from P every letter from $\Delta - L$. For example $ABCBBCCA_{|\{A,C\}} = ACCCA$.

Let $\mathrm{Pos}_{(P,L)} : [1, |P_{|L}|] \mapsto [1, |P|]$ be such that $\mathrm{Pos}_{(P,L)}(i) = \min\{j : |(P_1 \dots P_j)_{|L}| = i\}$, where P_i is the i-th letter of P. $\mathrm{Pos}_{(P,L)}(i)$ is the position of the letter in P that is sent to position i in $P_{|L}$.

Let $t = [w_0, v_1, w_1, \dots, v_{|P|}, w_{|P|}]$ be a P-template and $t_{|L} = [w'_0, v'_1, w'_1, \dots, v'_{|P_{|L}|}, w'_{|P_{|L}|}]$ be a $P_{|L}$-template. We say that $t_{|L}$ is a *L-specialization* of t if there are $(u_i)_{i:P_i \notin L} \in \Sigma^+$ such that:

- $\forall i\ v'_i = v_{\mathrm{Pos}_{(P,L)}(i)}$,
- $\forall i, j,\ P_i = P_j \notin L \implies \Psi(u_i) - \Psi(u_j) = v_i - v_j$,
- $\forall i\ w'_i = w_{i_b} u_{i_b+1} w_{i_b+1} \dots w_{i_e-1} u_{i_e} w_{i_e}$, where $i_b = \mathrm{Pos}_{(P,L)}(i)$ and $i_e = \mathrm{Pos}_{(P,L)}(i+1) - 1$.

▶ **Lemma 7.** *Let $P \in \Delta^*$ be a pattern and $L \subseteq \Delta$. For any P-template t and any L-specialization $t_{|L}$ of t if there is a word w realizing $t_{|L}$ then w realizes t.*

We omit the proof of Lemma 7 which is technical but straightforward.

With this definition a P-template t has infinitely many L-specializations, but in most cases the parents of a given L-specialization are included in the L-specializations of the parents. Thus we need to introduce the set of *small L-specializations* in order to keep a finit subset of them. A L-specialization of a P-template t is a *small L-specialization* if, with the notations from the definition of L-specialization, for any $A \in \Delta - L$ there is $i \in [1, |P|]$ such that $P_i = A$ and $|u_i| \le 2 \cdot \max_{a \in \Sigma} |h(a)|$.

▶ **Lemma 8.** *For any pattern $P \in \Delta^*$, P-template t and $L \subseteq \Delta$ the set of small L-specializations of t is finite and computable.*

Proof. Let $P \in \Delta^*$, $t = [w_0, v_1, w_1, \dots, v_{|P|}, w_{|P|}]$ be a P-template and $L \subseteq \Delta$. Let $t_{|L} = [w'_0, v'_1, w'_1, \dots, v_{|P_{|L}|}, w_{|P_{|L}|}]$ be a small L-specialization of t. Since $t_{|L}$ is a small L-specialization of t, for any letter $A \notin L$ there is an index i_A such that $P_{i_A} = A$ and $|u_{i_A}| \le 2 \cdot max_{a\in\Sigma}|h(a)|$, and there are only finitely many possible values for the u_{i_A}. Then from the definition for all j, $(P_j = A \text{ and } j \ne i_A) \implies \Psi(u_j) = \Psi(u_{i_A}) + v_{P_j} - v_{P_{i_A}}$. So there are only finitely many possible values for each u_j.

Once we have chosen the $(u_i)_{i:P_i \notin L}$ the w'_i and v'_i are fixed. Hence by trying all the possible values for $(u_i)_{i:P_i \notin L}$ we can compute the set of all small L-specializations of t. ◀

We denote by $\mathrm{SmallSpec}_L(t)$ the set of small L-specializations of a P-template t.

2.3 Special ancestors of a template

The set of *special ancestors* of a P-template t by h is the smallest set of templates containing t and any ancestor or small-L-specialization of any of its element. Let us first show that we can compute this set:

▶ **Theorem 9.** *For any alphabets Δ and Σ, pattern $P \in \Delta^*$, normalized P-template t and convenient morphism $h : \Sigma^* \mapsto \Sigma^*$, one can compute the set of special ancestors of t by h.*

Proof. The following algorithm computes this set for any P, t and h.

Input : P, t, h.
Output: The set S of special ancestors of t.

```
S = Ancestors_h(t);
for i = |Δ| − 1 ... 0 do
    for L ⊆ Δ, |L| = i do
        S = S ∪ SmallSpec_L(S);
    end
    S = S ∪ Ancestors_h(S);
end
```

Algorithm 1: How to compute special ancestors.

This algorithm halts because if S is finite at some point then by Lemmas 5 and 8 one can execute $S = S \cup \text{Ancestors}(S)$ and $S = S \cup \text{SmallSpec}_L(S)$ and keep S finite.

For any $D \subseteq L \subseteq \Delta$ and any pattern $P \in \Delta^*$, $(P_{|D})_{|L} = P_{|D}$. So for any L-specialization t_{DL} of a D-specialization t_D of a P-template t, $t_{DL} = t_D$. It implies that at the end for any $L \subseteq \Delta$, every element of S has all of its small L-specializations in S. Since the last operation of the algorithm adds the ancestors, every ancestor of any element of S is in S. ◀

In some reasonable implementation of the algorithm, it is important to use for S a data-structure that allows to check if a template is already in S in logarithmic time. Moreover, we are careful with specialization so that we do not obtain twice the same template by two different paths of specialization (dropping the letter A and then the letter B is the same than dropping B and then A).

2.4 Using special ancestors to decide

Under the conditions of Theorem 1, one can compute the set of special ancestors of a template, thanks to Theorem 9. Now we show that this set allows us to decide if the morphism's fixed point avoids the template.

▶ **Theorem 10.** *For any pattern $P \in \Delta^+$, any normalized P-template t, any convenient morphism h and any word $w \in \Sigma^+$, if there is a factor f of $h(w)$ that realizes t, then there is a factor f' of w that realizes a special ancestor of t.*

In fact we show that f' realizes the parent of an L-specialization of t for some well chosen set L. The only thing we do is to unfold the definitions with this set L.

Proof. Let $t = [w_0, v_1, w_1, \ldots, v_{|P|}, w_{|P|}]$ be a normalized P-template and assume there is a factor f of $h(w)$ that realizes t. Then by definition there are $u_1, \ldots, u_{|P|} \in \Sigma^+$ such that $f = w_0 u_1 w_1 u_2 w_2 \ldots u_{|P|} w_{|P|}$ and

$$\forall i, j\ P_i = P_j \implies \Psi(u_i) - \Psi(u_j) = v_i - v_j. \tag{1}$$

Let us introduce the set L:

$$L = \left\{ A \in \Delta : \forall i, P_i = A \implies |u_i| > 2 \cdot \max_{a \in \Sigma} |h(a)| \right\}. \tag{2}$$

Take the $P_{|L}$-template $t_{|L} = [w'_0, v'_1, w'_1, \ldots, v_{|P_{|L}|}, w_{|P_{|L}|}]$ such that:

- $\forall i,\ v'_i = v_{\text{Pos}_{(P,L)}(i)}$,
- $\forall i,\ w'_i = w_{i_b} u_{i_b+1} w_{i_b+1} \ldots w_{i_e-1} u_{i_e} w_{i_e}$, where $i_b = \text{Pos}_{(P,L)}(i)$ and $i_e = \text{Pos}_{(P,L)}(i{+}1){-}1$.

From the equality (1) and the definition of L, $t_{|L}$ is a small L-specialization of t. Let $(u'_i)_{1\leq i\leq |P_{|L}|}$ be such that for all i, $u'_i = u_{\mathrm{Pos}_{(P,L)}(i)}$. Then $f = w'_0 u'_1 w'_1 \dots u'_{|P_{|L}|} w'_{|P_{|L}|}$. Then from the equality (1) we can deduce:

$$\forall i, j \; [P_{|L}]_i = [P_{|L}]_j \implies \Psi(u'_i) - \Psi(u'_j) = v'_i - v'_j. \tag{3}$$

So f is a realization of the $P_{|L}$-template $t_{|L}$.

Since f is a factor of $h(w)$ there is a factor f' of w such that $h(f') = p_0 f s_{|P_{|L}|}$, where $p_0 \in \text{prefixes}(h(f'_1))$ and $s_{|P_{|L}|} \in \text{suffixes}(h(f'_{|f'|}))$. By construction, for all i, $|u'_i| > 2 \cdot \max_{a\in\Sigma} |h(a)|$ so we know that each of the u'_i contains at least the full image of one letter. So there are $u''_1, \dots, u''_{|P_{|L}|} \in \Sigma^+$, $w''_0, \dots, w''_{|P_{|L}|} \in \Sigma^*$ and $s_0, p_1, s_1, \dots, s_{|P_{|L}|-1}, p_{|P_{|L}|} \in \Sigma^*$ such that $f' = w''_0 u''_1 w''_1 u''_2 \dots u''_{|P_{|L}|} w''_{|P_{|L}|}$ and for all $i \in [0, |P|]$:

- p_i is a prefix of the image of the first letter of w''_i,
- s_i is a suffix of the image of the last letter of w''_i,
- $h(w''_i) = p_i w'_i s_i$,
- $u'_i = s_{i-1} h(u''_i) p_i$.

For all $i \in [1, |P_{|L}|]$, let $v''_i = \Psi(u''_i) - \Psi(u''_{\varphi_{P_{|L}}(i)})$ and let t'' be the $P_{|L}$-template $t'' = [w''_0, v''_1, w''_1, v''_2, \dots, v''_{|P_{|L}|}, w''_{|P_{|L}|}]$. Then t'' is the normalization of the $P_{|L}$-template $[w''_0, \Psi(u''_1), w''_1, \Psi(u''_2), \dots, \Psi(u''_{|P_{|L}|}), w''_{|P_{|L}|}]$ which is realized by f', thus f' is a realization of t''.

From $u'_i = s_{i-1} h(u''_i) p_i$ we get:

$$\Psi(u'_i) = \Psi(s_{i-1}) + M_h \Psi(u''_i) + \Psi(p_i). \tag{4}$$

Let $i, j \in [1, |P_{|L}|]$ such that $[P_{|L}]_i = [P_{|L}]_j$ then $\varphi(i) = \varphi(j)$ and hence

$$\Psi(u''_{\varphi(i)}) = \Psi(u''_{\varphi(j)}). \tag{5}$$

Now if we put all of that together we get:

$$\begin{aligned} v'_i - v'_j =& \Psi(u'_i) - \Psi(u'_j) \text{ (from (3))} \\ =& (\Psi(s_{i-1}) + M_h\Psi(u''_i) + \Psi(p_i)) - (\Psi(s_{j-1}) + M_h\Psi(u''_j) + \Psi(p_j)) \text{ (from (4))} \\ =& (\Psi(s_{i-1}) + M_h v''_i + \Psi(p_i)) - (\Psi(s_{j-1}) + M_h v''_i + \Psi(p_j)) \text{ (from (5))} \end{aligned}$$

Thus t'' is a parent of $t_{|L}$. So t'' is a parent of a specialization of t and is realized by a factor f' of w. ◀

Theorem 10 together with the fact that, by definition, a special ancestor of a special ancestor of t is itself a special ancestor of t gives:

▶ **Theorem 11.** *For any pattern $P \in \Delta^*$, any normalized P-template t, any convenient morphism h and any letter $a \in \Sigma$, if there is a positive integer n and a factor of $h^n(a)$ that realizes t, then a realizes a special ancestor of t.*

We also need the converse, that is:

▶ **Theorem 12.** *For any pattern $P \in \Delta^*$, any normalized P-template t, any convenient morphism h and any letter $a \in \Sigma$, if a realizes a special ancestor t' of t, then there is a positive integer n and a factor of $h^n(a)$ that realizes t'.*

Proof. We first take the sequence of parent and L-specialization that reaches t' from t. Then we use Lemmas 6 and 7 to reverse operations on a and we reach the factor of $h^n(a)$ that realizes t. ◀

From Theorems 11 and 12 we deduce the following one:

▶ **Theorem 13.** *For any pattern $P \in \Delta^*$, any normalized P-template t, any convenient morphism h and any letter $a \in \Sigma$, $h^\omega(a)$ avoids t if and only if a does not realize any special ancestor of t.*

Since we can compute the set of special ancestors and compare it to the letter a, we can decide if $h^\omega(a)$ avoids t. We implemented this algorithm in c++ and thus we can check if a pattern is avoided by the fixed point of a morphism.

3 Abelian avoidability of patterns

Patterns are words so we can say that a pattern avoids another pattern in the abelian sense. Moreover, for two patterns P, P' and word w, if P' is not abelian-P-free and the word w is abelian-P-free, then w is abelian-P'-free. It means that if P' is not abelian-P-free, then $\lambda_a(P') \leq \lambda_a(P)$. For instance, since $\lambda_a(AA) = 4$ for any $P \in \{A, B\}^*$, if $P \notin \{A, B, AB, BA, ABA, BAB\}$, then $\lambda_a(P) \leq 4$ because all the other binary patterns are not abelian-AA-free. So every binary pattern is either abelian-4-avoidable or abelian unavoidable. It is interesting to know which of them have avoidability index 2 or 3.

▶ **Theorem 14.** *Binary patterns of length greater than 8 are abelian-3-avoidable. More precisely every pattern that does not appear up to symmetry on the following list is abelian-3-avoidable:*
A, AA, AB, AAB, ABA, $AABA$, $AABB$, $ABAB$, $ABBA$, $AABAA$, $AABAB$, $AABBA$, $ABAAB$, $ABABA$, $AABAAB$, $AABABA$, $AABABB$, $AABBAA$, $ABAABA$, $AABAABA$, $AABABAA$, $ABBABBA$, $AABAABAA$, $ABAABAAB$.

Proof. It is well known that AAA is abelian-3-avoidable [7] and it is already enough to show the upper bound. Moreover, we can use the algorithm from Theorem 1 to show that any fixed point of $a \mapsto aabaac$, $b \mapsto cbbbab$, $c \mapsto cbccac$ is abelian-$AABBAB$-free. So we only need to find exhaustively all the words that avoid AAA, $AABBAB$ and $ABAABB$. This gives the list of Theorem 14. ◀

Conversely, if there is a word that avoids $AABAA$, there is also a recurrent word w that avoids $AABAA$ and then w avoids AA, thus $\lambda_a(AABAA) = 4$. So the patterns $A, AA, AB, AAB, ABA, AABA, AABAA$ are not abelian-3-avoidable. But, for the rest of the list, we do not know which of them are abelian-3-avoidable

▶ **Problem 1.** *Which of the following patterns are abelian-3-avoidable?*
$ABAB$, $ABBA$, $AABAB$, $AABBA$, $ABAAB$, $ABABA$,$AABAAB$, $AABABA$, $AABABB$, $AABBAA$, $ABAABA$, $AABAABA$, $AABABAA$, $ABBABBA$, $AABAABAA$, $ABAABAAB$.

There is a direct link with the following question:

▶ **Problem 2** (Mäkelä (see [12])). *Can you avoid abelian squares of the form uv where $|u| \geq 2$ over three letters ? - Computer experiments show that you can avoid these patterns at least in words of length 450.*

If the answer to the question from Mäkelä is positive then all the patterns from Problem 1 are abelian-3-avoidable. In [18] we showed that abelian squares of the form uv where $|u| \geq 6$ are avoidable over three letters.

Abelian-2-avoidability. For the binary case it was showed in [6] that:

▶ **Theorem 15** (J. D. Currie, T. I. Visentin)**.** *Binary patterns of length greater than 118 are abelian-2-avoidable.*

They also asked:

▶ **Problem 3** (J. D. Currie, T. I. Visentin)**.** *Characterize which binary patterns are abelian-2-avoidable.*

Using the algorithm from Theorem 1 we can improve this result and lower the 118 to 14. First we use the algorithm to check that:

▶ **Lemma 16.** *The fixed point of the morphisms on the left avoid in the abelian sense the corresponding patterns in the right:*

morphisms	*avoided patterns*
$a \mapsto aabaa$ $b \mapsto bbabb$	$AABBBAAAB$, $ABAAABBBA$, $AAABABABBB$, $AAABABBABB$,$AAABABBBAB$, $AABBBABAAB$, $AABBBABABA$, $ABAABABBBA$, $ABAABBBABA$, $ABABAABBBA$, $ABBBABAAAB$, $AABAABBBAB$, $AABBBAABAB$, $AABBBAABAAB$, $AAABABBAAAB$, $AABBBABBBAA$, $ABABABBBABA$, $ABABBABBABA$, $AAABAAABBAB$, $AAABBABAAAB$, $AAABAABAABAB$, $AAABABAAABAB$, $AABAAABABAAB$, $AAABAAABABBA$, $AAABAABABAAB$, $AAABABAABAAB$, $ABBABAAABAAB$, $ABABBBABBBABA$.
$a \mapsto aaaab$ $b \mapsto abbab$	$ABAABBBAAB$, $AAABBABABB$, $AAABBABBAB$, $AABAABBABB$, $AABABABBBA$, $AABABBABBA$, $AABABBBAAB$, $AABABBBABA$, $AABBAABBBA$, $AABBABABBA$, $AABBABBAAB$, $AABBABBABA$, $AABBBAABBA$, $ABAABBABBA$, $AABBABABBBA$, $AABABBBABBBA$,
$a \mapsto abb$ $b \mapsto aaab$	$AAAA$, $AAABAABBB$, $AAABBBABB$, $AABBABBBA$, $AABBBABBA$, $AAABBAAABB$, $AABABAAABB$, $ABBBAABBBA$, $AAABAABBAB$, $AAABAABAABB$, $AAABBAABAAB$, $AABAABAABBA$, $AABAABBAAAB$, $AABABABAAAB$, $AAABBAAABAB$, $AABAAABABAB$, $AABAAABBAAB$, $AAABAABAAABAB$,
$a \mapsto aaab$ $b \mapsto bbba$	$AAABABBBAA$, $AAABBAABBB$, $AAABBABBAA$, $ABABAAABBB$, $ABABBBAABBA$, $AABABBAAABA$, $AABBABAAABA$,
$a \mapsto abaa$ $b \mapsto babb$	$AABBABBABBA$, $AABABBABBBA$, $AABBBABBBABA$, $ABABBABBABBA$, $ABABBABBBABA$, $ABABBBABABBA$, $ABBABABBABBA$,
$a \mapsto aaaba$ $b \mapsto babbb$	$ABAABBBAAA$, $AABABBBAAA$,
$a \mapsto aababbaaaba$ $b \mapsto babbbaababb$	$AABAAABAAABAB$, $ABBBABBBABBBA$, $AAABAAABAAABAAA$,

It implies that, if a pattern contains any of the pattern from Lemma 16, then it can be avoided by a binary word. One can easily check that any binary pattern of length greater than 14 contains at least one of the patterns from the Lemma 16. It implies:

▶ **Theorem 17.** *Binary patterns of length greater than 14 are abelian-2-avoidable.*

In fact, up to symmetry, there are only 284 patterns that avoid all patterns of Lemma 16.

▶ **Theorem 18.** *The patterns from the following list are abelian-2-unavoidable:* A, AA, AB, AAA, AAB, ABA, $AAAB$, $AABA$, $AABB$, $ABAB$, $ABBA$, $AAABA$, $AAABB$, $AABAA$, $AABAB$, $AABBA$, $ABAAB$, $ABABA$, $ABBBA$, $AAABAA$, $AAABAB$, $AABAAB$, $ABAAAB$, $AAABAAA$.

Proof. Let assume that $AAABAAA$ is abelian-2-avoidable, then we can find a recurrent word that avoids $AAABAAA$ in the abelian sense and this words necessarily avoids AAA which is not possible. Thus $AAABAAA$ is abelian-2-unavoidable.

For all the other patterns one can do an exhaustive search and check that they are abelian-2-unavoidable. ◀

For the 260 other patterns we don't know which are abelian-2-avoidable and which are not. For most of them there is probably no fixed point of a binary morphism avoiding them, but they could be avoided by the image of a fixed point by a second morphism.

We are left with some interesting questions:

▶ **Problem 4.** *What is the length of the longest abelian-2-unavoidable binary pattern?*

We know that the answer is between 7 and 14.

▶ **Problem 5.** *What is the exact list of the abelian-2-unavoidable binary patterns?*

It is probably related somehow to the question 6 which seems really hard.

▶ **Problem 6** (Mäkelä (see [12])). *Can you avoid abelian-cubes of the form uvw where $|u| \geq 2$, over two letters ? - You can do this at least for words of length 250.*

It was showed in [19] that the answer to Problem 6 is negative. But we can replace the 2 in the question by any integer. In particular a proof that abelian cubes of the form uvw where $|u| \geq 3$ are avoidable over two letters would imply that many of the 284 patterns are also abelian-2-avoidable.

Finally we have some more general questions:

▶ **Problem 7.** *For any finite alphabet Δ is it true that:*

- $\exists n \in \mathbb{N}$ *such that any pattern over Δ of length greater than n is abelian-avoidable?*
- $\exists n \in \mathbb{N}$ *such that any pattern over Δ of length greater than n is abelian-$|\Delta|$-avoidable?*
- $\exists n \in \mathbb{N}$ *such that any pattern over Δ of length greater than n is abelian-2-avoidable?*

Acknowledgements. The author would like to thank Pascal Ochem and Michaël Rao for valuable discussions.

References

1. K. A. Baker, G. F. McNulty, and W. Taylor. Growth problems for avoidable words. *Theoretical Computer Science*, 69(3):319–345, 1989. doi:10.1016/0304-3975(89)90071-6.
2. D. R. Bean, A. Ehrenfeucht, and G. F. McNulty. Avoidable patterns in strings of symbols. *Pacific J. Math.*, 85(2):261–294, 1979.
3. F. Blanchet-Sadri and B. Woodhouse. Strict bounds for pattern avoidance. *Theoretical Computer Science*, 506:17–27, 2013.
4. J. Cassaigne. Unavoidable binary patterns. *Acta Informatica*, 30(4):385–395. doi:10.1007/BF01209712.
5. J. D. Currie and N. Rampersad. Fixed points avoiding abelian k-powers. *Journal of Combinatorial Theory, Series A*, 119(5):942–948, July 2012. doi:10.1016/j.jcta.2012.01.006.
6. J. D. Currie and T. I. Visentin. Long binary patterns are abelian 2-avoidable. *Theoretical Computer Science*, 409(3):432–437, 2008. doi:10.1016/j.tcs.2008.08.039.

7 F. M. Dekking. Strongly non-repetitive sequences and progression-free sets. *Journal of Combinatorial Theory, Series A*, 27(2):181–185, 1979. `doi:10.1016/0097-3165(79)90044-X`.

8 P. Erdős. Some unsolved problems. *The Michigan Mathematical Journal*, 4(3):291–300, 1957. `doi:10.1307/mmj/1028997963`.

9 P. Erdős. Some unsolved problems. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 6:221–254, 1961.

10 A. A. Evdokimov. Strongly asymmetric sequences generated by a finite number of symbols. *Dokl. Akad. Nauk SSSR*, 179:1268–1271, 1968.

11 V. Keränen. Abelian squares are avoidable on 4 letters. In *ICALP*, pages 41–52, 1992.

12 V. Keränen. New abelian square-free DT0L-languages over 4 letters. *Manuscript*, 2003.

13 M. Lothaire. *Combinatorics on Words*. Cambridge University Press, 1997.

14 R. Mercas, P. Ochem, A. V. Samsonov, and A. M. Shur. Binary patterns in binary cube-free words: Avoidability and growth. *RAIRO – Theor. Inf. and Applic*, 48(4):369–389, 2014.

15 P. Ochem. Doubled patterns are 3-avoidable. *Electron. J. Combinatorics.*, 23(1), 2016.

16 P. Ochem and A. Pinlou. Application of entropy compression in pattern avoidance. *Electron. J. Combinatorics.*, 21(2), 2014.

17 P. A. B. Pleasants. Non-repetitive sequences. *Mathematical Proceedings of the Cambridge Philosophical Society*, 68:267–274, 9 1970. `doi:10.1017/S0305004100046077`.

18 M. Rao and M. Rosenfeld. On Mäkelä's Conjectures: deciding if a morphic word avoids long abelian-powers. *ArXiv e-prints*, November 2015. `arXiv:1511.05875`.

19 M. Rao and M. Rosenfeld. Avoidability of long k-abelian repetitions. *Mathematics of Computation*, in press 2015.

20 P. Roth. Every binary pattern of length six is avoidable on the two-letter alphabet. *Acta Informatica*, 29(1):95–107. `doi:10.1007/BF01178567`.

21 U. Schmidt. Avoidable patterns on two letters. *Theoretical Computer Science*, 63(1):1–17, 1989. `doi:10.1016/0304-3975(89)90064-9`.

22 A. Thue. Über unendliche Zeichenreihen. *Norske Vid. Selsk. Skr. I. Mat. Nat. Kl. Christiania,*, 1906.

23 A. Thue. Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen. *Norske Vid. Selsk. Skr. I. Mat. Nat. Kl. Christiania,*, 10:1–67, 1912.

24 A. I. Zimin. Blocking sets of terms. *Sbornik: Mathematics*, 47(2):353–364, 1984.