On pseudo-repetitions in words*

Florin Manea[1], Robert Mercaş[2], and Cătălin Tiseanu[1]

1 **Faculty of Mathematics and Computer Science, University of Bucharest**
Str. Academiei 14, RO-010014 Bucharest, Romania,
`flmanea@fmi.unibuc.ro, ctiseanu@gmail.com`

2 **Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik**
PSF 4120, D-39016 Magdeburg, Germany
`robertmercas@gmail.com`

Abstract

The notion of repetition of factors in words was studied even from the beginnings of the combinatorics on words area. One of the recent generalizations regarding this concept was introduced by L. Kari et al., and considers a word to be an f-repetition if it is the iterated concatenation of one of its prefixes and the image of this prefix through an anti-/morphic involution f. In this paper, we extend the notion of f-repetitions to arbitrary increasing anti-/morphisms, and investigate a series of algorithmic problems arising in this context. Further, we present a series of results in the fashion of the Fine and Wilf theorem for f-repetitions, when f is an iso(anti)morphism.

1998 ACM Subject Classification Combinatorial algorithms

Keywords and phrases Combinatorics on Words, Repetitions, Algorithmics, Morphisms

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

The notions of repetition and primitivity play an important role in several computer science fields, among them being combinatorics on words [16, 5] and algebraic coding theory. A word is said to be a repetition if it is written as the repeated concatenation of one of its factors; alternatively, in this case, the word is said to be a power of its factor. Fine and Wilf [10] proved in a general context that if one can construct using two different words u and v two different sequences in such a way that one starts with u and the other with v, and they share a common prefix of at least the sum of the lengths of the two words minus their greatest common divisor, then the two sequences are equal and, moreover, u and v are both powers of a factor of length equal to the greatest common divisor of their lengths. In particular, one can restrict this result and say that if the powers of two words u and v share a common prefix of length at least equal to the sum of the lengths of the two words minus their greatest common divisor, then the two words are powers of the same word t.

Up to now several generalizations of this theorem have been investigated, [3, 6, 7, 9].

Having as a strong biological motivation the fact that Watson-Crick complementarity can be formalized as an antimorphic involution, and the fact that both a DNA-single stranded molecule and its complementary basically encode the same information, the authors of [9] introduce the notions of *pseudo-repetition* and *pseudo-primitivity*. In particular, a word is a *pseudo-repetition* if it can be expressed as the iterated concatenation between one of its prefixes and its image through a function f; a word is *pseudo-primitive* if it is not a

* The work of Robert Mercaş is supported by the *Alexander von Humboldt Foundation*.

STACS 2012.
Editors: Billy Editor, Bill Editors; pp. 1–19
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

pseudo-repetition. Until now, the considered functions were quite simple, being restricted to cases of anti-/morphic involutions, following the original motivation.

A natural extension of these concepts is to consider this concept for some more general classes of anti-/morphisms. Thus, we discuss here the concept of f-repetition and f-primitivity, for increasing anti-/morphisms. Considering that the notion of repetition is central in the study of combinatorics of words and the plethora of applications that this concept has in many parts of computer science, the study of pseudo-repetitions seems even more attractive, at least from a theoretical point of view. While the biological motivation seems appropriate only for the case when f is an antimorphic involution, one can imagine a series of real-life scenarios where we are interested in identifying factors of words which can be written as the iterated concatenation of a word and its encoding through some simple function f.

The contents of our paper are as follows. In Section 2 we introduce, and then discuss some algorithmic and complexity-theoretic results relating to f-repetitions. In particular, we approach and solve efficiently the basic problem of testing whether a word is an f-repetition or not, then we improve some algorithmic results from [4] in a more general context, and, finally, we identify the main tractable and intractable cases for the problem of deciding whether there exists a function f for which a given word is an f-repetition. We stress out that the theory of pseudo-repetitions lacked so far a developed algorithmic part, something that is usually quite important in bioinformatics applications; our algorithmic results aim to fill this gap. In Section 3 we extend the notion of pseudo-repetitions to literal bijective anti-/morphisms, and show some combinatorial results related to the Fine and Wilf theorem for this case.

We end this section with an overview of some basic concepts. For a complete view on basic combinatorial definitions we refer to [5, 16, 9].

For a word w over a finite alphabet V, the *length* of it is denoted by $|w|$ and represents the number of letters in the sequence. The *empty word* is the sequence of length zero and is denoted by ε. Moreover, we denote by $\mathrm{alph}(w)$ the alphabet of all letters that occur in w. A word u is a *factor* of a word v, if $v = xuy$, for some x, y. We say that u is a *prefix* of v, if $x = \varepsilon$ and a *suffix* of v if $y = \varepsilon$. We denote by $w[i]$ the symbol at position i in w, and by $w[i..j]$ the factor of w starting at position i and ending at position j, consisting of the concatenation of the symbols $w[i], \ldots, w[j]$, where $1 \leq i \leq j \leq n$. Moreover, we denote by $w = u^{-1}v$, whenever $v = uw$. The powers of a word w are defined recursively by $w^0 = \varepsilon$, for $n \geq 1$, $w^n = ww^{n-1}$, and $w^\omega = ww\cdots$, an infinite concatenation of the word w. If w cannot be expressed as a power of another word, then w is said to be *primitive*. The following lemmata is well known:

▶ **Lemma 1.** *For a word w, if $ww = xwy$ with $x \neq \varepsilon$ and $y \neq \varepsilon$, then x, y and w are powers of the same word t.*

The following result is well known and plays an important role in our investigation:

▶ **Theorem 2** (Fine and Wilf [10]). *Let u and v be two words over an alphabet A and $d = \gcd(|u|, |v|)$. If two words $\alpha \in u\{u, v\}^*$ and $\beta \in v\{u, v\}^*$ have a common prefix of length greater or equal to $|u| + |v| - d$, then u and v are powers of a common word. Moreover, the bound $|u| + |v| - d$ is optimal.*

For some anti-/morphism $f : V^* \to V^*$ we say that f is literal if $f(a) \in V$, for all $a \in V$, and increasing if $f(a) \neq \varepsilon$ for all $a \in V$. Moreover, we say that f is uniform whenever there exists a number k with $f(a) \in V^k$, for all $a \in V$.

We say that a word w is an f-repetition, or, alternatively, an f-power, if w is in $t\{t, f(t)\}^+$, for some prefix t of w. If w is not an f-power, we say that w is f-primitive. As an example, we see that $abcaab$ is primitive from the classical point of view (that is, $\mathbf{1}$-primitive, where $\mathbf{1}$ is the identical morphism) and, moreover, for some f with $f(a) = b$, $f(b) = a$ and $f(c) = c$, the word is also f-primitive. However, when considering the morphism $f(a) = b$, $f(b) = c$ and $f(c) = a$, we note that $abcaab$ is the concatenation of ab, $f(ab) = ca$ and ab, thus, being an f-power. In [9, 13] the authors were able to prove the following two important results:

▶ **Theorem 3** ([9]). *Let u and v be two words over an alphabet V and $f : V^* \to V^*$ a morphic involution (that is f^2 is the identity). If $u\{u, f(u)\}^*$ and $v\{v, f(v)\}^*$ have a common prefix of length greater or equal to $|u| + |v| - \gcd(|u|, |v|)$, then there exists $t \in V^*$ such that $u, v \in t\{t, f(t)\}^*$. Moreover, the bound $|u| + |v| - \gcd(|u|, |v|)$ is optimal.*

▶ **Theorem 4** ([9]). *Let u and v be two words over an alphabet V and $f : V^* \to V^*$ an antimorphic involution. If $u\{u, f(u)\}^*$ and $v\{v, f(v)\}^*$ have a common prefix of length greater or equal to $2|u| + |v| - \gcd(|u|, |v|)$, then there exists $t \in V^*$ such that $u, v \in t\{t, f(t)\}^*$.*

2 Algorithms

2.1 Algorithmic Problems: Overview

We state from the beginning that the model that we use is the unit-cost RAM model [15], and that in all upcoming problems we assume that the size of the alphabet V is constant.

There are three main problems that we discuss in the following. First, we are interested to decide whether a word is an f-repetition, for a given morphism or antimorphism f.

▶ **Problem 1.** *Let $f : V^* \to V^*$ be an increasing anti-/morphism. Given $w \in V^*$, decide whether for some word t we have $w \in t\{t, f(t)\}^+$.*

We are able to solve this problem in the general case in time $\mathcal{O}(n \lg n)$. In the particular case of uniform anti-/morphisms we obtain that Problem 1 is solved in time $\mathcal{O}(n(\lg\lg n))$. This latter case includes the anti-/morphic involutions from [9, 13]. Further, we extend, within the same time complexity, all our solutions to a more general form of Problem 1, testing whether $w \in \{t, f(t)\}\{t, f(t)\}^+$.

Another natural problem is identifying which factors of a word are pseudo-repetitions:

▶ **Problem 2.** *Let $f : V^* \to V^*$ be an increasing anti-/morphism. Given $w \in V^*$ construct data structures that enable us to answer the following queries in constant time:*
"Is $w[i..j] \in \{t, f(t)\}^k$?", denoted rep$(i, j, k)$, for $1 \le i \le j \le |w|$, $1 \le k \le |w|$ and some t.

Clearly, in the general case, one can produce in polynomial time some data structures, that fulfil the requested conditions, using a naive approach. However, in the case when f is a literal anti-/morphism we are able to construct such data structures quite efficiently, in time $\Theta(n^2)$. Using these data structures we give an algorithm that enumerates, in $\Theta(n^2)$ time, all the factors of w that belong to $\{t, f(t)\}^k$ for a fixed k. Note that, there are words w and functions f for which w has $\Theta(n^2)$ factors from $\{t, f(t)\}^k$, thus, in the worst case, any other algorithm enumerating these factors has an asymptotically similar running time as ours. This result improves significantly the algorithmic results reported in [4]. We also get an algorithm that efficiently enumerates, in $\Theta(n^2 \lg n)$ time, all triples (i, j, k), such that there exists t with $w[i..j] \in \{t, f(t)\}^k$. Again, there are words w and functions f for which there exist $\Theta(n^2 \lg n)$ triples (i, j, k) that fulfil the conditions above, thus, the worst case.

Finally, we consider the problem of deciding whether a word is a pseudo-repetition.

▶ **Problem 3.** *Given $w \in V^*$, decide whether there exist an increasing anti-/morphism $f : V^* \to V^*$ and a prefix t of w, such that $w \in t\{t, f(t)\}^+$.*

In general, the problem is trivial, since for a factorization $w = aw'$ with $a \in V$ and nonempty word w', we set f to be any anti-/morphism with $f(a) = w'$. However, there are cases when the problem is more interesting. We show that when we restrict our search to uniform anti-/morphisms, Problem 3 is solved in $\mathcal{O}(n(\lg\lg n))$ time. Furthermore, Problem 3 becomes NP-complete when $|t| \geq 2$ and f is an increasing morphism, but there exist cases when for f an increasing antimorphism it is solved in polynomial time. When we ask that length of t is given as input, both the morphic and antimorphic cases are NP-complete. Finally, the problem is also NP-complete in the antimorphic case, whenever we ask for $|w| \geq 3|t|$.

2.2 Prerequisites: Number Theoretic Properties and Data Structures

Before presenting the proofs of the claims made in the previous section, we give some basic number theoretic properties that are useful in the sequel. Given two natural numbers k and n, if k divides n we say that k is a divisor of n, and we write $k \mid n$. For a number n we denote by $d(n)$ the number of its divisors, and by $\sigma(n)$ the sum of its divisors.

The following lemma is important for our proofs.

▶ **Lemma 5.** *Let n be a natural number. The following statements hold:*

1. $\sum_{1\leq\ell\leq n} d(\ell) \in \Theta(n \lg n)$*; we also have* $\sum_{1\leq\ell\leq n} d(\ell) \geq n \lg n$.
2. $\sigma(n) \in \mathcal{O}(n(\lg\lg n))$.
3. $\sum_{1\leq\ell\leq n}(n-\ell+1)d(\ell) \in \Theta(n^2 \lg n)$.

Proof. Statement 1 was shown by Dirichlet in [2], while the second is known as Gronwall's theorem, [11]. The proof of the third part is skipped due to space reasons (see Appendix). ◀

Now, for a string x of length n, over a fixed alphabet V, we define a suffix-array data structure that contains two arrays Suf and LCP each with n elements from $\{1,\ldots,n\}$.

Basically, Suf is defined such that $x[Suf[i]..n]$ is the i^{th} suffix of x, in the lexicographical order. The array LCP is defined by $LCP[1] = 1$ and $LCP[r]$ is the length of the longest prefix of $x[Suf[r-1]..n]$ and $x[Suf[r]..n]$. These data structures are constructed in time $\mathcal{O}(n)$. For more details, see [14], and the references therein.

Moreover, one can process the array LCP in linear time $\mathcal{O}(n)$ in order to answer in constant time queries "What is the length of the longest common prefix of $x[i..n]$ and $x[j..n]$?", denoted $LCPref(i,j)$. The idea is to first compute a structure S that associates to each i the value $S[i] = \ell$ if and only if $i = Suf[\ell]$. Further we compute in linear time a range minimum query data structure for the array LCP (see [12]), and answer in constant time queries "What is the minimum number from $LCP[i],\ldots,LCP[j]$?". Now, $LCPref(i,j)$ is obtained as the minimum from $LCP[i'],\ldots,LCP[j']$, where $i' = \min\{S[i],S[j]\}$ and $j' = \max\{S[i],S[j]\}$.

Finally, for an increasing morphism f, compute for each prefix t of $xf(x)$ the length of $f(t)$, and save it in an array len, i.e., $len[i] = |f(x[1..i])|$, and an array with $|f(x)|$ elements $inv[i] = j$ if $len[j] = n+i$ and $inv[i] = -1$ otherwise. These is done in $\mathcal{O}(n)$ time.

2.3 Solution of Problem 1

Let us first assume that f is an increasing morphism, and that the input word w has length n. We can construct in $\mathcal{O}(n)$ time the word $x = wf(w)$ of length $m = n + |f(w)|$ (which is in $\mathcal{O}(n)$); notice that the constant hidden by the $\mathcal{O}$-denotation and the length of x depend on the morphism f. Now, we construct in $\mathcal{O}(n)$ time a suffix-array data structure for x, such that we are able to answer $LCPref$ queries for x, according to Section 2.2.

Setting $t = w[1..i]$, we can check in constant time whether t or $f(t)$ occur at a position $\ell+1$ in w, since $w[\ell+1..\ell+i] = t$ if and only if $LCPref(\ell+1, 1) \geq i$, and $w[\ell+1..\ell+len[i]] = f(t)$ if and only if $LCPref(\ell+1, n+1) \geq len[i]$.

Assume now that $t = w[1..i]$ is a prefix of w and that $w[1..j] \in t\{t, f(t)\}^*$. Moreover, assume that both t and $f(t)$ occur at position $j+1$ in w, that is, $w[j+1..j+i] = t$ and $w[j+1..j+len[i]] = f(t)$. In this setting, if $w[j+len[i]+1..n]$ begins with t or $f(t)$ and $w[j+1..n] \in t\{t, f(t)\}^*$ then, following Theorem 2, both t and $f(t)$ are powers of a word x, with $|x| \leq |t|$, and, thus, w is also a power of x.

Now, we can describe the strategy one can use to test whether w is an f-repetition.

We first test whether there exists a word x such that $w = x^k$, with $k \geq 2$. If the result is positive we decide that w is a trivial f-repetition, where no factor of the form $f(x)$ appears.

Otherwise, for all prefixes $t = w[1..i]$ of w with $i < n$ we set $s = i+1$ and do the following:

1. If $s = n+1$ we halt and conclude that w is an f-repetition.
2. If t occurs at position s in w, and $f(t)$ does not occur at position s in w, set $s = s+i$. We know that $w[1..s-1]$ is an f-repetition, and we continue checking for $w[s..n]$.
3. If $f(t)$ occurs at position s in w, and t does not occur at position s in w, set $s = s+len[i]$. We know that $w[1..s-1]$ is an f-repetition, and we continue checking for $w[s..n]$.
4. If both t and $f(t)$ occur at position s in w, and t or $f(t)$ occur at position $s+i$ in w, we know that $w[1..s+len[i]-1]$ is an f-repetition, and, according to the remark made before giving this strategy, we know that w is either a repetition of one of its factors or $w[s+i..n]$ cannot be an f-repetition. Thus, since we already know that w is not a repetition, we set $s = s+len[i]$, and we continue checking whether $w[s..n]$ is an f-repetition, as well.
5. If neither t nor $f(t)$ occur at position s in w we conclude that w is not in $t\{t, f(t)\}^+$.

Clearly, the algorithm presented above is sound. In the following, we compute its complexity. The first step takes $\mathcal{O}(n)$ time, as it requires locating the occurrences of w in ww (see [8]). The iteration is executed for each prefix t of w, and during each iteration the algorithm makes at most $\mathcal{O}(\lfloor \frac{n}{\ell} \rfloor)$ steps, for $\ell = |t|$, as s can take at most $\lfloor \frac{n}{\ell} \rfloor$ different values. Hence, the total time spent executing this iterative instruction is $\mathcal{O}(\sum_{1\leq\ell\leq n} \lfloor \frac{n}{\ell} \rfloor)$. But $\sum_{1\leq\ell\leq n} \lfloor \frac{n}{\ell} \rfloor \leq \sum_{1\leq\ell\leq n} \frac{n}{\ell} \in \mathcal{O}(n \lg n)$. In conclusion, the running time of the algorithm testing whether w is an f-repetition is $\mathcal{O}(n \lg n)$, which includes the construction of the suffix-arrays data structures.

An interesting observation is made in the case when the morphism f is uniform. Here we only need to run the iterative instruction for t prefix of w with $|t| \mid n$. Indeed, $|f(t)| = k|t|$ and w is in $\{f, f(t)\}^*$ implies that the length of w is divisible by the length of t. Hence, the total running time of the algorithm is $\mathcal{O}(\sum_{\ell|n} \frac{n}{\ell}) = \mathcal{O}(\sigma(n))$. According to Lemma 5, the total running time of the algorithm, in this case, is $\mathcal{O}(n(\lg \lg n))$, including again the construction of the suffix-arrays data structures.

The case when f is an increasing antimorphism is very similar. We construct the word $x = wf(w)$ and the same data structures as in the former case, and for $t = w[1..i]$, we can check, in constant time, whether t or $f(t)$ occur at a position $\ell+1$ in w, in the following manner. We have $w[\ell+1..\ell+i] = t$ if and only if $LCPref(\ell+1, 1) \geq i$, and $w[\ell+1..\ell+len[i]] = f(t)$ if and only if $LCPref(\ell+1, m-len[i]+1) = len[i]$, where $m = |wf(w)|$. The rest of the reasoning remains unchanged.

Finally, we discuss the more general case of testing whether, for an increasing anti-/morphism f, a given word w is in $\{t, f(t)\}\{t, f(t)\}^*$, where t is a factor of w.

We present the case of f being a morphism. The case when f is an antimorphism is treated similarly, following the ideas above. Once again, we construct the word $x = wf(w)$ and compute all the presented data structures. We assume that w is not in $t\{t, f(t)\}^+$, the previously treated case, for none of its prefixes t, and, consequently, test whether it is in $f(t)\{t, f(t)\}^+$, where t is one of the factors of w. But this can be seen as testing whether w is in $x\{x, y\}^+$, where x is a prefix of w and $f(y) = x$.

Assume now that $x = w[1..i]$ is a prefix of w and that $w[1..j] = x^\ell$, for some $\ell \geq 1$. Moreover, since w is not in $t\{t, f(t)\}^*$, thus, cannot be written as the repetition of one of its prefixes, we can presume that $w[j+1..j+i] \neq x$. In order to obtain that w is in $x\{x, y\}^+$, for some y with $f(y) = x$, there must exist two numbers k and d such that $ki \leq j$, $d \leq i$, with $w[ki+1..ki+d] = y$. But remark that, when $k \leq \ell - 2$, if $w[ki+1..n]$ is in $y\{y, x\}^+$, then we obtain that $w[ki+1..n]$ and $w[ki+1..j]$, which equals $x^{\ell-k}$, have a common prefix of length at least $2|x| \geq |x| + |y|$. Hence, by Theorem 2, it follows that there exists a word u such that both x and y are powers of u, which leads to the conclusion that w is a power of u, a contradiction. Thus, if we want to have $w[ki+1..n] \in y\{y, x\}^*$, we must have $k \geq t - 1$. Therefore, there are two cases to be analysed: $k = t - 1$ and $k = t$. In both cases, we first determine y, using $LCPref$ queries. First, we verify whether $LCPref(n + len[ki] + 1, 1) \geq x$, that is, whether $w[ki+1..n]$ begins with a word y such that $f(y) = x$. If the answer is positive we check whether $inv[len[ki] + i]$ is defined, and, when this holds, we conclude that $y = w[ki+1..inv[len[ki] + i]]$. If at least one of the above checks does not return a positive answer, then there is no prefix of $w[ki+1..n]$ that has the image through f equal to x. Once we determined y, since $f(y) = x$, we can check, for both possible choices of k, if $w[ki+1..n]$ is in $y\{y, f(y)\}^+$ just as in the iterative instructions of the previous algorithm (see Appendix).

By the already made remarks it is clear that this algorithm works correctly. To compute the complexity, note that checking whether $w \in t\{t, f(t)\}^*$ takes $\mathcal{O}(n \lg n)$ time. Moreover, for each prefix x of the word w we make at most $\lfloor \frac{n}{\ell} \rfloor$ steps, where ℓ is the length of a word y such that $f(y) = x$. But ℓ is at least $c|x|$ for a subunitary constant c that depends on f. Therefore, the total number of steps done by the iterative instruction is $\mathcal{O}(\sum_{1 \leq \ell \leq n} \lfloor \frac{n}{\ell} \rfloor)$. Therefore, the running time of the algorithm is $\mathcal{O}(n \lg n)$. Once again, when f is uniform we can reduce this time complexity, following the same arguments as before, to $\mathcal{O}(n(\lg \lg n))$.

The following proposition summarizes the results obtained in this section:

▶ **Proposition 6.** *Let $f : V^* \to V^*$ be an increasing anti-/morphism. Given $w \in V^*$ one can decide whether $w \in \{t, f(t)\}\{t, f(t)\}^+$ in $\mathcal{O}(n \lg n)$ time. If f is uniform, one can decide whether $w \in \{t, f(t)\}\{t, f(t)\}^+$ in $\mathcal{O}(n(\lg \lg n))$ time.*

2.4 Solution of Problem 2

The first remark that we make is that in the general case, when f is an increasing anti-/morphism one can compute, for a word w of length n, a 3-dimensional array M, such that for all $1 \leq i \leq j \leq n$ and $1 \leq k \leq j - i$ we have $M[i][j][k] = 1$ whenever there exists a factor t of $w[i..j]$ such that $w[i..j] \in \{t, f(t)\}^k$, and $M[i][j][k] = 0$ otherwise. In a naive strategy, this array is easily computed in $\mathcal{O}(n^6)$ time. However, we show that in the particular case of f being a literal anti-/morphism we are able to solve Problem 2 much more efficiently.

The idea that we use is the following. First we construct all data structure from Sections 2.2. Further, we define an $n \times n$ matrix M, such that, for $1 \leq i, d \leq n$, $M[i][d] = (j, i_1, i_2)$ stores the beginning point of the longest word $w[j..i]$ contained in $\{t, f(t)\}^+$, for some word t of length d, as well as the last occurrences of t and $f(t)$ in this word.

By dynamic programming M is computed in $\mathcal{O}(n^2)$ time (see Appendix). In particular, using $LCPref$ queries on $wf(w)$, $M[i][d]$ is obtained in constant time from $M[i-d][d]$.

The matrix M helps us answer rep-queries in constant time. Indeed, we answer yes to the query $rep(i, j, k)$ if and only if $k \mid j - i + 1$ and $M[j][\frac{j-i+1}{k}] \leq i$, and no, otherwise.

Also one can use the just computed matrix to efficiently enumerate, given a word w of length n and a number k, the factors $w[i..j]$ that are in $\{t, f(t)\}^k$, for some word t. We just have to list all the pairs (i, j) for which $rep(i, j, k)$ returns yes. The time needed to do so is $\Theta(n^2)$, as we go through all possible pairs (i, j) and ask rep queries. Note that the time bound does not depend on k. However, any algorithm solving this problem needs $o(n^2)$ operations in the worst case, as, for instance, all the factors $w[i..i + k\ell - 1]$ of the word a^n are in $\{a^\ell\}^k$, and their number is in $o(n^2)$. Our result improve in a more general case the results reported in [4], where the same enumeration problem was solved in time $\mathcal{O}(n^2 \lg n)$.

Finally, we can use rep queries to efficiently enumerate, given a word w of length n, all the triples (i, j, k) such that $w[i..j] \in \{t, f(t)\}^k$, for some t. The algorithm is a bit more complicated in this case. First we compute for all the numbers ℓ, with $1 \leq \ell \leq n$, the list of their divisors. This is done in time $\mathcal{O}(n \lg n)$ using the Sieve of Eratosthenes. Further, for each pair $(i, i + \ell - 1)$ with $\ell \geq 1$, and for all $d \mid \ell$ we verify whether $rep(i, i + \ell - 1, d)$ returns yes. If so, the triple $(i, i + \ell - 1, d)$ is one of the ones we were looking for. Clearly, the algorithm is correct. Its complexity is $\mathcal{O}(n \lg n) + \Theta(\sum_{1 \leq \ell \leq n}(n - \ell + 1)d(\ell))$, and, thus, according to Lemma 5, the overall complexity of this algorithm is $\Theta(n^2 \lg n)$. Moreover, any algorithm solving this problem does $o(n^2 \log n)$ operations in the worst case, since the worst case of this problem is when the word a^n is considered. Clearly, a correct algorithm should output for this word exactly $\sum_{1 \leq \ell \leq n}(n - \ell + 1)d(\ell)$ triples. This means that a correct algorithm makes $o(n^2 \lg n)$ steps for the input a^n, which proves our claim.

We summarize the results of this section in the following:

▶ **Proposition 7.** *Take $w \in V^*$ a word of length n and $f : V^* \to V^*$ a literal anti-/morphism.*
1. One can construct in $\mathcal{O}(n^2)$ time data structures answering in constant time the queries: "Is $w[i..j] \in \{t, f(t)\}^k$?", denoted $rep(i, j, k)$, for $1 \leq i \leq j \leq |w|$, $1 \leq k \leq |w|$ and some t.
2. Given $k \leq n$ one can enumerate in time $\mathcal{O}(n^2)$ the factors $w[i..j] \in \{t, f(t)\}^k$, for some t.
3. One can identify in $\mathcal{O}(n^2 \lg n)$ time all triples (i, j, k) with $w[i..j] \in \{t, f(t)\}^k$, for some t.

2.5 Solution of Problem 3

We begin by presenting some of the tractable cases of this problem. Recall that we want to decide, if for a given w there exist a prefix t and an anti-/morphism f such that $w \in t\{t, f(t)\}^+$.

As we have already mentioned, when no restrictions are imposed on the anti-/morphism f or on the form of the factor t, the problem is trivial.

Let us assume in the following that we are interested in finding a literal anti-/morphism f that fulfils the conditions of Problem 3. The problem is solved by checking whether there exist a prefix x and a factor y of w, of equal length, such that $w \in x\{x, y\}^*$. This strategy is implemented in fashion similar to the solutions of Problem 1. Basically we construct a suffix-array structure for w, and then check for each prefix x of w, whose length divides $|w|$, whether w is in $x\{x, y\}^*$, where y is the first factor of w, of length equal to $|x|$, which is not equal to x. Once we obtained such x and y, we define a morphism (respectively, antimorphism) f such that $f(x) = y$ by associating to the i^{th} symbol of x the i^{th} symbol of y, or, respectively, the $(|y| - i)^{th}$ symbol of y. A positive answer is given whenever suitable prefixes and anti-/morphisms are found. The total running time of the algorithm is $\mathcal{O}(\sum_{\ell \mid n}(\ell + \frac{n}{\ell})) = \mathcal{O}(n(\lg \lg n))$, where for a prefix of length ℓ we make $\frac{n}{\ell}$ steps to decide if $w \in x\{x, y\}^+$ and another ℓ steps to construct the morphism. Clearly, an identical strategy can be used to decide whether there exist a factor t of w and an anti-/morphism f such that $w \in \{t, f(t)\}\{t, f(t)\}^+$. The case when f is uniform is solved analogously, with the difference that the word y is now defined as having length $k|x|$, where k is the length of $f(a)$, for $a \in V$.

Assume now that f is an increasing antimorphism. We solve the following problem: *"Given $w \in V^*$, decide whether there exists an increasing antimorphism $f : V^* \to V^*$, and a prefix t of w with $|t| \geq 2$, such that $w \in t\{t, f(t)\}^* f(t)$".*

A key observation is that there exist a prefix t of w and an antimorphism f, such that $w \in t\{t, f(t)\}^* f(t)$, if and only if there exist another prefix $t' = a^k b$ of w with $a \neq b$ letters and $k \geq 1$, and an antimorphism f', such that $w = t' f'(t')$, or w is in $a^k\{a^k, f(a^k)\}^* f(a^k)$ with $k \geq 2$. Indeed, if $w \in t\{t, f(t)\}^*$ and $t \neq a^k$, for $k \geq 2$, then $t = a^k b y$, for some letters a, b and a word y. Here, we take $t' = a^k b$, $f'(a) = f(a)$, and $f'(b) = v$, such that $w = t' v f(a^k)$. To decide whether there exist a prefix $t' = a^k b$ of w with $k \geq 1$, and an antimorphism f', such that $w = t' f'(t')$, we take the single prefix of w that has the form $a^k b$, for some a, b, and check whether there exists a suffix of w that has the form x^k. This takes linear time using a suffix-array data structure. We define $f'(a) = x$ and $f'(b) = v$, such that $w = a^k b v x^k$. To decide whether there exist f with $w \in a^k\{a^k, f(a^k)\}^*$, we choose a suffix of w to be $f(a)$, and check for $k \geq 2$, whether $w \in a^k\{a^k, f(a^k)\}^* f(a^k)$. Since, this naive strategy takes $\mathcal{O}(n^3)$ time, where $|w| = n$, one can solve in polynomial time the problem.

Let us consider now the more general problem:
"Given $w \in V^*$, decide whether there exists an increasing antimorphism $f : V^* \to V^*$, and a prefix t of w with $|t| \geq 2$, such that $w \in t\{t, f(t)\}^* f(t) \cup f(t)\{t, f(t)\}^* t$".

First, run the previous algorithm to check whether $w \in t\{t, f(t)\}^* f(t)$. Otherwise, assume that $w \in f(t)\{t, f(t)\}^* f(t)$ for some proper factor t of w, of length at least 2. Since $w \neq a^k$, for any k, we have $w = a^k x b^\ell$, where x starts with a symbol different from a and ends with one different from b, $1 \leq \ell \leq k$ and $k + |x| > \ell$. So $w = a^k x b^\ell$, where x starts and ends with a symbol different from a. If $|a^{k-\ell} x| \geq 2$, then we take $a^{k-\ell} x = yc$ for some $c \in V \setminus \{b\}$ and $y \in V^*$, and define $t = cb^\ell$ and the antimorphism f that verifies $f(b) = a$ and $f(c) = a^{k-\ell} y$. Clearly $w = f(t)t$. Finally, when $|a^{k-\ell} x| = 1$ we either have $w = a^{\ell+1} b^\ell$, and the problem has no solution, or $w = a^\ell c b^\ell$. For the latter, when $a \neq b$, or $a = b$ and $\ell \leq 2$, the problem has no solution. When $\ell > 2$ the solution is given by $t = aca$, $f(a) = a$ and $f(c) = a^{\ell-2}$, which leads to $w = f(t) t f(t)$. Clearly, the above analysis is done in linear time, and, consequently, the general problem is also solved in polynomial time.

We now present a series of computationally hard cases for Problem 3.

Let us first show that the following problem is NP-complete:
"Given $w \in V^*$ with $|w| \geq 4$, decide whether there exist an increasing morphism $f : V^* \to V^*$, and a prefix t of w with $|t| \geq 2$, such that $w \in t\{t, f(t)\}^+$".

Clearly, this problem is in NP. Now we show that it is NP-complete by giving a polynomial-time reduction from the pattern-description problem [1]. This problem is described as follows:

"Given two words x and y decide whether there exists an increasing morphism f, such that $f(x) = y$"; this problem was shown to be NP-complete in [1].

Assume now that we have an input instance of the pattern-description problem, namely two words x and y, over an alphabet V, and want to decide whether there exists an increasing morphism f such that $f(x) = y$. We construct the word $w = a^n x b^n y$, where $n = 2\max\{|x|, |y|\}$ and $a, b \notin V$. We show there exists an increasing morphism f such that $f(x) = y$ if and only if there exist an increasing morphism $f' : (V \cup \{a, b\})^* \to (V \cup \{a, b\})^*$, and a prefix t of w with $|t| \geq 2$, such that $w \in t\{t, f'(t)\}^+$. The left to right implication is immediate. For the other implication, assuming first that $t = a^k$, it is clear that, since b appears in $f'(a)$ and $k \geq 2$, we immediately obtain a contradiction. Therefore, $t = a^n x'$. We obtain that $f'(t) = (f'(a))^n f'(x')$. But the only two factors of the form x^n of w are a^n and b^n, so $(f'(a))^n = b^n$ and $f'(a) = b$. Now, we immediately obtain that $x' = x$ and $f'(x) = y$. This concludes the proof of this implication, and the equivalence that we have just shown exhibits a polynomial time reduction from the pattern-description problem to our problem. Therefore, our problem is NP-complete.

Similarly, one can show the NP-completeness of a more general problem:
"Given $w \in V^*$ with $|w| \geq 4$, decide whether there exist an increasing morphism $f : V^* \to V^*$, and a prefix t of w with $|t| \geq 2$, such that $w \in \{t, f(t)\}\{t, f(t)\}^+$".

We have proved that the two previously discussed hard problems are tractable whenever f is an antimorphism. Let us show some hard results for this case, as well. We begin with:
"Given $w \in V^*$ with $|w| \geq 4$, and a number $\ell \geq 2$, decide whether there exists an increasing antimorphism $f : V^* \to V^*$, such that $w \in t\{t, f(t)\}^+$ whenever $|t| = \ell$".

The NP-completeness is proven again by a polynomial time reduction from the pattern-description problem. Assume the words $x, y \in V^*$ are an input instance of the pattern-description problem. For a word $w = xaby^R$, where $a, b \notin V$, it is not hard to see that there exists an increasing morphism f such that $f(x) = y$ if and only if there exists an increasing antimorphism $f' : (V \cup \{a, b\})^* \to (V \cup \{a, b\})^*$ with $w \in t\{t, f'(t)\}^+$, for $|t| = |x| + 1$.

Following the same lines one obtains easily the NP-completeness for the problem:
"Given $w \in V^*$ with $|w| \geq 4$, and a number $\ell \geq 2$, decide whether there exists an increasing antimorphism $f : V^* \to V^*$, such that $w \in \{t, f(t)\}\{t, f(t)\}^+$ whenever $|t| = \ell$".

Finally let us approach the following problem:
"Given $w \in V^*$ decide whether there exist an increasing antimorphism $f : V^* \to V^*$, and a prefix t of w with $|t| \geq 2$, such that $w \in t\{t, f(t)\}\{t, f(t)\}^+$".

We sketch the hardness of this proof by polynomial time reduction from the pattern-description problem (for the complete proof see Appendix). Assume that we have $x, y \in V$, an input instance of the pattern-description problem, and take $w = ea^nxb^nc^ny^Rd^nfc^ny^Rd^nf$, where $n = 2\max\{|x|, |y|\}$ and $a, b, c, d, e, f \notin V$ are pairwise distinct. We show there exists an increasing morphism f such that $f(x) = y$ if and only if there exists an increasing antimorphism f' and a prefix t of w with $|t| \geq 2$, such that $w \in t\{t, f'(t)\}\{t, f'(t)\}^+$. Since the left to right implication is easy, and for $t = e$ or $t = ea^k$ with $k \leq n$, we easily obtain contradictions we take $t = ea^nx'$. We have $f'(t) = f'(x')(f'(a))^nf'(e)$, and, clearly, $f'(t)$ is a suffix of w, and easily follows that $f'(e) = f$ and $f'(a) = d$. Since $w \in t\{t, f'(t)\}\{t, f'(t)\}^+$, we obtain that $w = tf'(t)f'(t)$ and $f'(t) = c^ny^Rd^nf$. Thus, $t = ea^nxb^n$ and $f'(b) = c$, and we get that $f'(x) = y^R$. This shows the existence of a morphism f that makes $f(x) = y$ and concludes the proof. This equivalence exhibits a polynomial time reduction from the pattern-description problem to our problem, providing the NP-completeness.

Similarly, one can show that also NP-complete is the more general problem:
"Given $w \in V^*$ with $w \geq 6$, decide whether there exist an increasing antimorphism $f : V^* \to V^*$, and a prefix t of w with $|t| \geq 2$, such that $w \in \{t, f(t)\}^2\{t, f(t)\}^+$".

Also, one can show, by similar ideas that deciding for a given $w \in V^*$ whether there exist an increasing antimorphism $f : V^* \to V^*$ and a prefix t of w with $|t| \geq 2$, such that w in $t\{t, f(t)\}^+t$ or $f(t)\{t, f(t)\}^+f(t)$ is generally computationally hard.

We summarize the results of this section in the following proposition:

▶ **Proposition 8.** *For a word $w \in V^*$,*
1. One can decide in $\mathcal{O}(n(\lg\lg n))$ time whether there exist a uniform anti-/morphism $f : V^* \to V^*$ and a prefix t of w, such that $w \in \{t, f(t)\}\{t, f(t)\}^+$.
2. One can decide in polynomial time whether there exists an increasing antimorphism $f : V^* \to V^*$, and a prefix t of w with $|t| \geq 2$, such that $w \in f(t)\{t, f(t)\}^*t \cup t\{t, f(t)\}^*f(t)$.
3. The problem of deciding whether there exist an increasing morphism $f : V^* \to V^*$ and a prefix t of w with $|t| \geq 2$, such that $w \in \{t, f(t)\}\{t, f(t)\}^+$, is NP-complete.
4. Given a number $\ell \geq 2$, the problem of deciding whether there exists an increasing

antimorphismf $: V^* \to V^*$*, such that* $w \in \{t, f(t)\}\{t, f(t)\}^+$*, for* $|t| = \ell$*, is NP-complete.*
5. The problem of deciding whether there exists an increasing antimorphism $f : V^* \to V^*$ *and a prefix* t *of* w *with* $|t| \geq 2$*, such that* $w \in \{t, f(t)\}^2\{t, f(t)\}^+$*, is NP-complete.*

3 Extension of the Fine and Wilf Theorem

The results presented in this section generalize the original Fine and Wilf Theorem [10], as well as the case of involutions, presented by Kari et al. [9, 13]. Due to the limited space all proofs of this Section are placed in the Appendix.

We start with the simple remark that for a two letter alphabet $\{a, b\}$, the case of bijective literal morphisms, is quite trivial, since we either have the identity, or that $f(a) = b$ and $f(b) = a$, the results being given by Theorems 2 and 3. For the literal antimorphisms, the results follow from Theorem 4.

Also note that, while in the morphism case the identity refers to the original Fine and Wilf result, in the case when $f(a) = b$ and $f(b) = a$, both for morphisms and antimorphisms, all words starting with the same letter a, are from $a\{a, f(a)\}^*$. Thus, for the rest of this section we consider alphabets of three or more letters.

The following small observation helps us with all proofs throughout the Section.

▶ **Lemma 9.** *Let* w *be a word over* V *and* $f : V^* \to V^*$ *a bijective literal anti-/morphism. If* $u = f(u)$*, then, for any letter* $a \in alph(u)$*, we have* $f^2(a) = a$*.*

Note that if f is a bijective function from V to V, then one can see f as a permutation of V. Therefore, there exists a minimum $m > 0$ such that f^m is the identity of V. Generally, this value is denoted by $ord(f)$, called the order of f, and is less than $g(|V|)$, where g is the Landau function. Using techniques similar to [17], one can prove the following result:

▶ **Theorem 10.** *Let* u *and* v *be words over* V *and* $f : V^* \to V^*$ *an isomorphism with* $ord(f) = k+1$*. If* $\alpha \in u\{u, f(u), \ldots, f^k(u), v, f(v), \ldots, f^k(v)\}^*$ *has a common prefix of length greater or equal to* $|u| + |v| - \gcd(|u|, |v|)$ *with* $\beta \in v\{u, f(u), \ldots, f^k(u), v, f(v), \ldots, f^k(v)\}^*$*, then there exists* $t \in V^*$*, such that* $u, v \in t\{t, f(t), \ldots, f^k(t)\}^*$*.*

As consequence, we generalize both Fine and Wilf and Kari et al. periodicity results.

▶ **Corollary 11.** *Let* u *and* v *be two words over* V *and* $f : V^* \to V^*$ *an isomorphism with* $ord(f) = k + 1$*. If* $u\{u, f(u), \ldots, f^k(u)\}^*$ *and* $v\{v, f(v), \ldots, f^k(v)\}^*$ *have a common prefix of length greater or equal to* $|u| + |v| - \gcd(|u|, |v|)$*, then there exists* $t \in V^*$*, such that* $u, v \in t\{t, f(t), \ldots, f^k(t)\}^*$*. Moreover, the bound is optimal.*

Next we show that in the case of arbitrary bijective literal morphisms the result of Theorem 10 is optimal also regarding the number of different iterations of the function f that are used in expressing both u and v.

▶ **Proposition 12.** *Let* $f : V^* \to V^*$ *be an arbitrary isomorphism with* $ord(f) = k+1$*. There exist two words* u *and* v *with* $|u| = |v| + \gcd(|u|, |v|)$ *and* $vf(v)$ *a prefix of* u^2*, such that* u *is not part of* $t\{f^{i_1}(t), \ldots, f^{i_\ell}(t)\}^*$ *for any prefix* t *of* u *and any set* $\{i_1, \ldots, i_\ell\}$ *strictly included in* $\{1, \ldots, k\}$*.*

Of course one of the first natural questions that comes up is what is then a good bound for this case. The following results and examples give us the optimal bounds.

▶ **Example 13.** Let us consider the words $u = bdacae$ and $v = bdac$, and the isomorphism f with $f(a) = b$, $f(b) = a$, $f(c) = d$, $f(d) = e$ and $f(e) = c$. The words u^2 and $vf(v)^2$ share a prefix of length $|u| + |v| - 1$ and no word t exists, such that $u, v \in t\{t, f(t)\}^*$.

▶ **Proposition 14.** *Let u and v be two words over V, such that $|u| > |v| = 2gcd(|u|, |v|)$, and $f : V^* \to V^*$ an isomorphism. If $\alpha \in u\{u, f(u)\}^*$ and $\beta \in v\{v, f(v)\}^*$ have a common prefix of length greater or equal to $|u| + |v|$, then there exists $t \in V^*$, such that $u, v \in t\{t, f(t)\}^*$. Moreover, the bound is optimal.*

However, when we have the length of the shortest word strictly greater than two times their greatest common divisor the result is a bit more complicated. The following results are proved by case analysis with the help of Lemma 1, and Theorems 2, 3 and 4, the examples providing the optimality argument.

▶ **Example 15.** Let us consider the words $u = abcc$ and $v = abc$, and the isomorphism f with $f(a) = c$, $f(b) = a$ and $f(c) = b$. The words u^2 and $vf(v)^2$ share a common prefix of length $2|u| - 1$ and no word t exists, such that $u, v \in t\{t, f(t)\}^*$

▶ **Proposition 16.** *Let u and v be two words over V, such that $|u| > |v| > 2gcd(|u|, |v|)$, and $f : V^* \to V^*$ an isomorphism. If $\alpha \in uu\{u, f(u)\}^*$ and $\beta \in v\{v, f(v)\}^*$ have a common prefix of length greater or equal to $2|u|$, then there exists $t \in V^*$, such that $u, v \in t\{t, f(t)\}^*$. Moreover, the bound is optimal.*

▶ **Example 17.** Let us consider the words $u = abcabdabeabc$ and $v = abcabdabe$ and the isomorphism f with $f(a) = a$, $f(b) = b$, $f(c) = d$, $f(d) = e$ and $f(e) = c$. The words $uf(u)ab$ and v^3 share a common prefix of length $2|u| + gcd(|u|, |v|) - 1$ and no word t exists, such that $u, v \in t\{t, f(t)\}^*$

▶ **Proposition 18.** *Let u and v be two words over V, such that $|u| > |v| > 2gcd(|u|, |v|)$, and $f : V^* \to V^*$ an isomorphism. If $\alpha \in uf(u)\{u, f(u)\}^*$ and $\beta \in v\{v, f(v)\}^*$ have a common prefix of length greater or equal to $2|u| + \gcd(|u|, |v|)$, then there exists $t \in V^*$, such that $u, v \in t\{t, f(t)\}^*$. Moreover, the bound is optimal.*

We now turn our attention to the case of the literal antimorphisms. A first disappointment is that a result similar to that of Theorem 10 does not hold in this case, even when we allow the size of the common prefix to be arbitrarily large.

▶ **Example 19.** Consider the words $u = abc$ and $v = ab$, and the literal antimorphism f with $f(a) = e$, $f(b) = d$, $f(c) = c$, $f(d) = b$ and $f(e) = a$. It is easy to see that f is even more an involution. The infinite word $w = abc(de)^\omega$ can be written as $w = uf(v)^\omega = vf(u)f(v)^\omega$. Furthermore, all three words u, v and w are aperiodic.

So which are the bounds in the antimorphism case? When $|v| = 2\gcd(|u|, |v|)$ the following result is not difficult to prove with the help of Lemma 9 and Theorem 4:

▶ **Proposition 20.** *Let u and v be two words over V, such that $|u| > |v| = 2gcd(|u|, |v|)$, and $f : V^* \to V^*$ a bijective literal antimorphism with $ord(f) = k + 1$. If $\alpha \in u\{u, f(u)\}^*$ and $\beta \in v\{v, f(v)\}^*$ have a common prefix of length greater or equal to $2|u| + \gcd(|u|, |v|)$, then there exists $t \in V^*$, such that $u, v \in t\{t, f(t), f^k(t)\}^*$. Moreover, the bound is optimal.*

The next example shows how the iterations of the function f describe u and v:

▶ **Example 21.** Let us consider the words $u = acbabcacb$ and $v = acbabc$, and the literal antimorphism f with $f(a) = b$, $f(b) = c$ and $f(c) = a$. Let us now look at the words $uf(u)$ and $vf(v)^2$. One can see that the two new words are equal, but no word t exists such that both u and v are expressed as f-powers of it. Clearly, an infinite iteration of this word still has two different factorizations as f-powers, starting with u and v, respectively. Therefore, in the case when $|v| = 2gcd(|u|, |v|)$, the result of the previous proposition is optimal as far as the number of different iterations of f needed to describe u and v is concerned.

The following result represents a variation of Lemma 1. The proof is done identifying factors that give equalities as in Lemma 9 and conclude that the antimorphism is an involution.

▶ **Lemma 22.** *For a word w and a literal antimorphism f defined on the alphabet of w, if w or $f(w)$ are proper factors of $\{w, f(w)\}^2$, such that not all three factors are equal, it is the case that f is an involution.*

The case of $|v| \geq 3\,\mathrm{gcd}(|u|, |v|)$ is proved with the help of Lemmas 9 and 22, by looking at the alignment of the prefix v, or, respectively, suffix $f(v)$, of the second factor of α with the corresponding factor from β.

▶ **Proposition 23.** *Let u and v be two words over V, such that $|u| > |v| > 2gcd(|u|, |v|)$, and $f : V^* \to V^*$ a bijective literal antimorphism. If $\alpha \in u\{u, f(u)\}^*$ and $\beta \in v\{v, f(v)\}^*$ have a common prefix of length greater or equal to $2|u| + |v| - \mathrm{gcd}(|u|, |v|)$, then there exists $t \in V^*$, such that $u, v \in t\{t, f(t)\}^*$. Moreover, the bound is optimal.*

References

1 Dana Angluin. Finding patterns common to a set of strings. *J. Comput. Syst. Sci.*, 21(1):46–62, 1980.

2 Tom M. Apostol. *Introduction to analytic number theory.* Springer, 1976.

3 Jean Berstel and Luc Boasson. Partial words and a theorem of Fine and Wilf. *Theoretical Computer Science*, 218:135–141, 1999.

4 Ehsan Chiniforooshan, Lila Kari, and Zhi Xu. Pseudopower avoidance. *Fundamenta Informaticae*, to appear, 2011.

5 Christian Choffrut and Juhani Karhumäki. Combinatorics of words. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, volume 1, pages 329–438. Springer-Verlag, 1997.

6 Sorin Constantinescu and Lucian Ilie. Generalised Fine and Wilf's theorem for arbitrary number of periods. *Theoretical Computer Science*, 339(1):49–60, 2005.

7 Sorin Constantinescu and Lucian Ilie. Fine and Wilf's theorem for abelian periods. *Bulletin of EATCS*, 89:167–170, 2006.

8 Maxime Crochemore and Wojciech Rytter. *Jewels of Stringology: Text algorithms.* World Scientific, Singapore, 2002.

9 Elena Czeizler, Lila Kari, and Shinnosuke Seki. On a special class of primitive words. *Theoretical Computer Science*, 411:617–630, 2010.

10 Nathan J. Fine and Herb S. Wilf. Uniqueness theorem for periodic functions. *Proceedings of the American Mathematical Society*, 16:109–114, 1965.

11 Thomas H. Grönwall. Some asymptotic expressions in the theory of numbers. *Trans. Amer. Math. Soc.*, 14:113–122, 1913.

12 Dan Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology.* Cambridge University Press, New York, NY, USA, 1997.

13 Lila Kari and Shinnosuke Seki. An improved bound for an extension of Fine and Wilf's theorem, and its optimality. *Fundamenta Informaticae*, 191:215–236, 2010.

14 Juha Kärkkäinen, Peter Sanders, and Stefan Burkhardt. Linear work suffix array construction. *J. ACM*, 53:918–936, 2006.

15 Donald E. Knuth. *The Art of Computer Programming*, volume 1: Fundamental Algorithms. Addison-Wesley, 1968.

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

17 Jeffrey Shallit. http://www.cs.uwaterloo.ca/ shallit/talks/wilf3.pdf.

A Appendix Section

We say that a function $f : V^* \to V^*$ is a morphism if $f(xy) = f(x)f(y)$, for any words x and y, over V. Further, f is an antimorphism if $f(xy) = f(y)f(x)$, for any words x and y over V. Note that, when we want to define a morphism or an antimorphism it is enough to give the definitions of $f(a)$, for all $a \in V$. We write anti-/morphism whenever we want to say "antimorphism or morphism".

Proof of Lemma 5: The first two results are well known. The first one was shown by Dirichlet (see [2]), while the second is known as Gronwall's theorem (see [11]).

For the third statement let $T(n) = \sum_{1\le\ell\le n}(n-\ell+1)d(\ell)$.

We have $T(n) = T(n-1) + \sum_{1\le\ell\le n} d(\ell) + d(n)$, for $n \ge 2$. According to Statement 1, we obtain that $T(n) \ge T(n-1) + n\lg n + 2$. Applying iteratively this reasoning, we obtain that $T(n) \ge 2n + \sum_{1\le\ell\le n} \ell \lg \ell$.

Now, recall an elementary form of the Chebyshev inequality that says that if $(a_\ell)_{1\le\ell\le n}$ and $(b_\ell)_{1\le\ell\le n}$ are two increasing sequences of real numbers, then it is the case that $\sum_{1\le\ell\le n} a_\ell b_\ell \ge \frac{1}{n}(\sum_{1\le\ell\le n} a_\ell)(\sum_{1\le\ell\le n} b_\ell)$. We apply this inequality to obtain a lower bound for $T(n)$, taking $a_\ell = \ell$ and $b_\ell = \lg \ell$. Therefore, $T(n) \ge 2n + \sum_{1\le\ell\le n} \ell\lg\ell \ge 2n + \frac{1}{n}(\sum_{1\le\ell\le n}\ell)(\sum_{1\le\ell\le n}\lg\ell) \ge 2n + \frac{n(n+1)}{2n} \cdot \frac{n\lg(n/4)}{4}$. Thus, $T(n) \in \Omega(n^2 \lg n)$.

Now, $T(n) = \sum_{1\le\ell\le n}(n-\ell+1)d(\ell) \le n\sum_{1\le\ell\le n} d(\ell)$. According to Statement 1, once more, we obtain that $T(n) \in \mathcal{O}(n^2 \lg n)$.

Therefore, $T(n) \in \Theta(n^2 \lg n)$, and the proof of Statement 3 is concluded. □

Algorithm for testing $w \in \{t, f(t)\}\{t, f(t)\}^+$***:***

1. We first test whether there is a prefix t of w such that $w \in t\{t, f(t)\}^*$. If the test returns a positive answer, we halt and decide that w is in $\{t, f(t)\}\{t, f(t)\}^+$.
2. Then, for all the prefixes $x = w[1..i]$, with $i < n$, of w we do the following. Set $s = i + 1$.
   a. While x occurs at position s in w do $s = s + i$.
   b. We reached a position s where x does not occur.
   c. Check, as explained above, whether there exists y occurring at position $s - i$ such that $f(y) = x$; then, check whether $w[s-i..n] \in y\{y, f(y)\}^*$ exactly as in the case of the iterative instruction from the previous algorithm. If the check returns a positive answer then decide that $w \in f(y)\{y, f(y)\}^+$.
   d. Check, as explained above, whether there exists y occurring at position s such that $f(y) = x$; then, check whether $w[s..n] \in y\{y, f(y)\}^*$ exactly as in the case described before. If the check returns a positive answer then decide that $w \in f(y)\{y, f(y)\}^+$.
3. If we did not conclude that $w \in \{t, f(t)\}\{t, f(t)\}^+$ for some factor t of w, then decide that w is not in this set. □

Description and computation of the matrix M from the Solution of Problem 2:

For $1 \le i \le n$ and $1 \le d \le n$, we define $M[i][d] = (j, i_1, i_2)$ such that

- $d \mid j - i + 1$,
- $w[j..i] \in \{t, f(t)\}^k$ for some factor $t = w[j + \ell d..j + (\ell+1)d - 1]$ of $w[j..i]$ and $k = \frac{j-i+1}{d}$,
- $w[j-d..i] \notin \{t, f(t)\}^{k+1}$,
- $j \le i_1 \le i - d + 1$ such that $w[i_1..i_1 + d - 1] = t$, $d \mid i_1 - j$, and i_1 is maximal with this property,
- If there exists ℓ such that $w[j + \ell d..j + (\ell+1)d - 1] = f(t)$, then $j \le i_2 \le i - d + 1$ such that $w[i_2..i_2 + d - 1] = t$, $d \mid i_2 - j$, and i_1 is maximal with this property; otherwise, $i_2 = -1$.

- It may be the case that there exist t and t' such that $t \neq t'$ and $w[j..i] \in \{t, f(t)\}^k \cap \{t', f(t')\}^k$, as above. This could lead to the fact that the matrix M is not well defined. But, in this case, we clearly have $t = f(t')$ and $f(t) = t'$, and, when we define $M[i][d]$, it is either the triple (j, i_1, i_2) or the triple (j, i_2, i_1); we just choose one of these two triples, as it makes no differences for our algorithm, in this case.

In other words, $M[i][d]$ stores the beginning point of the longest word $w[j..i]$ contained in $\{t, f(t)\}^+$, for some t of length d, as well as the last occurrences of t and $f(t)$ in this word.

Next, we show that this matrix can be computed by dynamic programming in $\mathcal{O}(n^2)$ time. For all $d \in \{1, \ldots, n\}$ and $n \geq i \geq 2d$ we have:

- $M[d+\ell][d] = (1, 1, 1)$, when $0 \geq \ell < d$ and $w[\ell+1..d+\ell] = f(w[\ell+1..d+\ell])$.
- $M[d+\ell][d] = (1, 1, -1)$, when $0 \geq \ell < d$ and $w[\ell+1..d+\ell] \neq f(w[\ell+1..d+\ell])$.
- $M[i][d] = (j, i_1, i-d+1)$ when $M[i-d][d] = (j, i_1, i_2)$ and $w[i-d+1..i] = w[i_2..i_2+d-1]$.
- $M[i][d] = (j, i-d+1, i_2)$ when $M[i-d][d] = (j, i_1, i_2)$ and $w[i-d+1..i] = w[i_1..i_1+d-1]$.
- $M[i][d] = (i_2+d, i-d+1, i-2d+1)$ when $M[i-d][d] = (j, i_1, i_2)$, $i_1 = i-2d+1$ and $f(w[i-d+1..i]) = w[i_1..i_1+d-1]$, but $w[i-d+1..i] \neq w[i_2..i_2+d-1]$ and $w[i_1..i_1+d-1] \neq w[i_2..i_2+d-1]$.
- $M[i][d] = (i_1+d, i-d+1, i-2d+1)$ when $M[i-d][d] = (j, i_1, i_2)$, $i_2 = i-2d+1$ and $f(w[i-d+1..i]) = w[i_2..i_2+d-1]$, but $w[i-d+1..i] \neq w[i_1..i_1+d-1]$ and $w[i_1..i_1+d-1] \neq w[i_2..i_2+d-1]$.
- $M[i][d] = (i_1+d, i-2d+1, i-d+1)$ when $M[i-d][d] = (j, i_1, i_2)$, $i_2 = i-2d+1$ and $w[i-d+1..i] = f(w[i_2..i_2+d-1])$, but $w[i_1..i_1+d-1] \neq w[i_2..i_2+d-1]$.

It is clear that $M[i][d]$ can be computed from $M[i-d][d]$ in constant time using *LCPref* queries on the word $x = wf(w)$. □

"Given** $w \in V^*$**, decide whether there exists an increasing antimorphism** $f : V^* \to V^*$**, and a prefix** t **of** w **with** $|t| \geq 2$**, such that** $w \in f(t)\{t, f(t)\}^*t \cup t\{t, f(t)\}^*f(t)$**". Case** $w = a^n$**:

Note that $w \neq a^n$, for $n \geq 4$ a composite number, as, in this case, the initial problem has a solution. Indeed, if $n = pk$, for some prime number p, we have that $w = (a^p)^k$, thus we can take, for instance, $t = a^p$ and f any antimorphism that sets $f(a) = a$. Also, if $w = a^n$, where n is a prime number, the problem we consider has no solution. To show this, assume that $w \in \{f(t), t\}\{t, f(t)\}^+$ for some proper factor t of w, of length at least 2. It follows that $|t| \mid |w|$, thus $|t| = 1$ or $|t| = |w|$, a contradiction. □

"Given** $w \in V^*$ **decide whether there exist an increasing antimorphism** $f : V^* \to V^*$**, and a prefix** t **of** w **with** $|t| \geq 2$**, such that** $w \in t\{t, f(t)\}\{t, f(t)\}^+$**": This problem is, clearly, in NP. Once again, to show its hardness we give a polynomial time reduction from the pattern-description problem. Assume that we have an input instance of the pattern-description problem, namely two words x and y, over an alphabet V; we want to decide whether there exists an increasing morphism f such that $f(x) = y$. We construct the word $w = ea^nxb^nc^ny^Rd^nfc^ny^Rd^nf$, where $n = 2\max\{|x|, |y|\}$ and a, b, c, d, e, f are pairwise distinct symbols that do not appear in V. We show that there exists an increasing morphism f such that $f(x) = y$ if and only if there exists $f : V^* \to V^*$, an increasing antimorphism, and a prefix t of w, with $|t| \geq 2$, such that $w \in t\{t, f'(t)\}\{t, f'(t)\}^+$. The left to right implication is easy; we show the other one. If $t = e$ we easily obtain a contradiction, as $f'(e)$ should contain all the symbols a, b, c, d, f, and w has at least two factors $f'(e)$. If $t = ea^k$, with $k \geq 2$, we obtain, as before, a contradiction. Thus $t = ea^nx'$, $f'(t) = f'(x')(f'(a))^nf'(e)$, and, clearly, $f'(t)$ is a suffix of w. It follows easily that $f'(e) = f$ and $f'(a) = d$. Since $w \in t\{t, f'(t)\}\{t, f'(t)\}^+$ we obtain that, in fact, $w = tf'(t)f'(t)$ and $f'(t) = c^ny^Rd^nf$. From this it follows that $t = ea^nxb^n$ and $f'(b) = c$. Finally, we get that $f'(x) = y^R$. But this shows the existence of a morphism f that makes $f(x) = y$; this morphism is defined for the letters $s \in V$ as $f(s) = (f'(s))^R$. This concludes the proof of this implication. The equivalence that we have just shown exhibits a polynomial time reduction from the pattern-description problem to our problem. Therefore, our problem is also NP-complete. □

Proof of Lemma 9: Let us denote $w = a_1 \cdots a_n$ with $a_i \in V$, where $1 \leq i \leq n$. Since $f(w) = w$, it is straightforward that $f^2(w) = f(f(w)) = f(w)$, and it follows that $w = a_1 \cdots a_n = f^2(a_1) \cdots f^2(a_n)$. Thus, $a_i = f^2(a_i)$ with $1 \leq i \leq n$. □

Proof of Theorem 10: The proof is similar to the one in [17]. In the Fine and Wilf's case the proof follows by induction after the length of $|u| + |v|$. For the base case we consider $|u| = |v|$ (note that this case actually includes the case when $|u| + |v| = 2$), thus, we get that $u = v$, and the result follows.

Now assume without loss of generality that $|u| > |v|$. Then it must be the case that, for some word w we have $u = vw$. Observe that the prefix of length v of $v^{-1}\beta$ is an iteration of $f(v)$. Denoting this iteration by z, and changing appropriately all occurrences from α and β of iterations of f over v with iterations over z, one gets that we only need to look at the words $v^{-1}\alpha \in w\{w, f(w), \ldots, f^k(w), z, f(z), \ldots, f^k(z)\}^*$ and $v^{-1}\beta \in z\{w, f(w), \ldots, f^k(w), z, f(z), \ldots, f^k(z)\}^*$. The conclusion follows from a previous step of the induction. □

Proof of Proposition 12: Let us assume that $V = \{a_1, \ldots, a_n\}$. As we explained, f is seen as a permutation of V. Assume that f has t disjoint cycles, and let $c_i = (j_{i,1}, \ldots, j_{i,p_i})$, for $1 \leq i \leq t$, denote these cycles (we assume that the numbers in a cycle are ordered increasingly). Also let x_i be the word obtained by concatenating the letters $a_{i,j}$ of a cycle, for $1 \leq j \leq p_i$, and denote $x = x_1 \ldots x_t$.

Now take $u = xf^k(x)f^{k-1}(x) \cdots f(x)$, that basically contains all possible iterations of f, and $v = xf^k(x)f^{k-1}(x) \cdots f^2(x)$, having only k factors. Note that $\gcd(|u|, |v|) = |x|$ and that $|u| = |v| + |x|$. It is quite straightforward to check that $vf(v)$ is a prefix of length $|u| + |v| - |x|$ of u^2.

Let us now show that there does not exist a word t, such that $u \in t\{f^{i_1}(t), \ldots, f^{i_\ell}(t)\}^*$ for a set $\{i_1, \ldots, i_\ell\}$ strictly included in $\{1, \ldots, k\}$. Clearly, if such a t would exist, then its length is a divisor of n. If $|t| = n$ one would not be able to generate all the factors of length n of u using only the factors $f^{i_1}(t), \ldots, f^{i_\ell}(t)$, as the order of f is $k > \ell$. If $|t| < n$ then $x = tf^{j_1}(t) \ldots f^{j_p}(t)$, for $\{j_1, \ldots, j_p\}$ a set of numbers included in $\{i_1, \ldots, i_\ell\}$. It follows immediately that f is a cyclic permutation (thus, of order n) and that all the factors of length n of u begin with a different letter. Therefore, all the iterations of f must be used in writing u as the catenation of factors of the form $f^i(t)$. This concludes our proof. □

Proof of Proposition 14: Let v_1 be the prefix of length $gcd(|u|, |v|)$ of v; let $v = v_1v_2$. It is rather easy to see that $u \in v\{v, f(v)\}^*v_1$ or $u \in v\{v, f(v)\}^*f(v_1)$.

When u ends with v_1, it follows immediately that v_2 is a prefix of u or $f(u)$, since the first u of α is followed by either u or $f(u)$. In the first case, v_2 is a prefix of v and one immediately obtains that $v_1 = v_2$. In the second case, we obtain immediately that $v_2 = f(v_1)$. Moreover, looking at what follows v_2 in β, we have that either $f(v_2) = v_1$ or $f(v_2) = f(v_1)$. In both cases, one may take $t = v_1$ and obtain that $u, v \in t\{t, f(t)\}^*$.

Let us now analyse the case when u ends with $f(v_1)$. Here, we obtain as above, that $f(v_2)$ is either a prefix of u or of $f(u)$. First, we obtain that $f(v_2) = v_1$, and, looking what follows after the prefix $uf(v_2)$ of β, we once more get that, $v_2 \in \{v_1, f(v_1)\}$. Similarly, in the second case, $f(v_2) = f(v_1)$, thus, $v_2 = v_1$. This conclusion follows, since the optimality is obtained from Example 13. □

Proof of Proposition 16: Let us denote by u' the longest prefix of u with $u' \in v\{v, f(v)\}^*$ and by v_1 the prefix of v with $|v_1| = |u| - |u'|$. Obviously, $gcd(|v_1|, |v|) = d \neq |v|/2$. We analyse two main cases, depending on the length of $|v_1|$, namely $|v_1| < |v|/2$ and $|v_1| > |v|/2$.

Let us assume first that $|v_1| < |v|/2$ and denote $v = v_1v_2v_3$, where $|v_2| = |v_1|$.

Consider the case when $\alpha = u'v_1u\alpha' = u'v_1v_1v_2v_3u''\alpha'$, where $\alpha' \in \{u, f(u)\}^*$ and $u = vu''$. Note that u' is a prefix of β, such that $\beta = u'v\beta'$, with $\beta' \in \{v, f(v)\}^*$. The discussion follows now several cases.

If $\beta = u'vv\beta''$ then, by Lemma 1, we obtain that both v_1 and v are power of the same word t. Thus, we easily get that $u, v \in t\{t, f(t)\}^*$.

Now let us take $\beta = u'vf(v)\beta''$. We get that $v_3 = v_1^\ell x$, for some positive number ℓ and $x \in V^*$ a prefix of v_1, such that $|x| < |v_1|$ with x possibly empty. Denoting $v_1 = xy$, we obtain immediately that $yx = f(v_1)$. If u'' starts with v, we obtain that the prefix yxv_1 of yxu'' is equal to the prefix $f(v_1)f(v_1)$ of β'. Therefore, $f(v_1) = v_1$. It follows that f is the identity on the alphabet of the words u and v, and the conclusion of our proposition follows from Theorem 2. If u'' starts with $f(v)$ it follows that $u'' = (f(v_1))^{\ell-1}f(xyx)$. But the suffix $f(yx)$ matches either a factor $f(v_1)$ of β or a factor v_1 of β. In the first case we get that f is the identity on the alphabet of u and v, and the conclusion follows from Theorem 2, while in the second case we get that $f^2(v_1) = v_1$, and, thus, f is an involution on the alphabet of u and v, and the conclusion follows from Theorem 3.

Next, we analyse the case when $\alpha = u'f(v_1)u\alpha' = u'f(v_1)v_1v_2v_3u''\alpha'$, where $\alpha' \in \{u, f(u)\}^*$ and $u = vu''$. Note that u' is a prefix of β, such that $\beta = u'f(v)\beta'$ with $\beta' \in \{v, f(v)\}^*$. In this case we obtain immediately that $f(v_2) = v_1$. But the suffix $f(v_1)$ of the u factor occurring before α' in α matches an $f(v_2)$ or a v_2 factor from β. In the first case we obtain that f is the identity on all letters of u and v, and the conclusion follows from Theorem 2, while, in the second case, we get that $f^2(v_1) = v_1$, and, thus, f is an involution on the alphabet of u and v, and the conclusion follows from Theorem 3.

We move now to the case when $|v_1| > |v|/2$, and set $v = v_1v_2$ with $|v_2| < |v_1|$.

Assume first that $\alpha = u'v_1u\alpha' = u'v_1v_1v_2u''\alpha'$, where $\alpha' \in \{u, f(u)\}^*$ and $u = vu''$. Note that u' is a prefix of β, such that $\beta = u'v\beta'$, with $\beta' \in \{v, f(v)\}^*$. Clearly, we get that v_2 is a prefix of v_1. If β' starts with v we obtain immediately, by Lemma 1, that both v_1 and v are powers of some t, and, therefore, u and v are in $t\{t, f(t)\}^*$. So, we consider the case when β' starts with $f(v)$. It follows immediately that $f(v_1)$ has v_2 as a suffix.

If u'' starts with v we obtain that the suffix $f(v_2)$ of the prefix $f(v)$ of β' matches the prefix v_2 of the prefix v of u''. Thus, f is the identity on the symbols of v_2. It is easy to see that the symbols of v_1 are those of v_2 and $f(v_2)$, and so, f is the identity also for the symbols of v_1, and, consequently, for the symbols of u and v. The conclusion follows from Theorem 2.

Now, consider the case when u'' starts with $f(v)$. If β' starts with $f(v)f(v)$ we obtain that $f(v_2)$ is a suffix of $f(v_1)$, and, thus, it is equal to v_2. As in the previous case, this leads to the conclusion that f is the identity on the alphabet of u and v, and the conclusion follows from Theorem 2. If β' starts with $f(v)v$ we obtain that $f(v_2)$ is a suffix of v_1, and, thus, $f^2(v_2)$ is a suffix of $f(v_1)$. Therefore, f is an involution on the alphabet of v_2, and it easily follows that it is also an involution on the alphabet of u and v. The conclusion follows from Theorem 3.

Assume now that $\alpha = u'f(v_1)u\alpha' = u'f(v_1)v_1v_2u''\alpha'$, where $\alpha' \in \{u, f(u)\}^*$ and $u = vu''$. Note that u' is a prefix of β, such that $\beta = u'f(v)\beta'$, with $\beta' \in \{v, f(v)\}^*$, and we obtain that $f(v_2)$ is a prefix of v_1.

Assume first that β' starts with $f(v)$. If u'' starts with $f(v_1)$, then $f(v_2)$ is a prefix of $f(v_1)$. But $f^2(v_1)$ is a prefix of $f(v_1)$ as well, so f is the identity on v_2. As in the previous cases, we obtain that f is the identity on all letters of u and v, and with the help of Theorem 2 reach the conclusion.

When u'' starts with v, if β' starts with $f(v)v$ we get that v_1 has the suffix v_2. Thus, $f(v_2) = v_2$ and f is the identity for the alphabet of u and v. The conclusion follows again from Theorem 2. If β' starts with $f(v)f(v)$ we get that u'' starts with either vv or with $vf(v_1)$. In the latter case the conclusion follows as in the case when u'' starts with $f(v_1)$. In the first case, the analysis is restarted, ending up with either a solution as in the case when β' starts with $f(v)v$, or the case when u'' starts with $f(v_1)$, as u ends with $f(v_1)$. Hence, we conclude that this case leads also to what we wanted to prove.

Finally, assume that β' starts with v. If u'' starts with v_1 we obtain that both $f(v_2)$ and v_2 are prefixes of v_1, so f is the identity on the alphabet of u and v. If u'' starts with $f(v_1)$ we obtain that $f(v_1)$ starts with v_2, so $f^2(v_2) = v_2$. It follows that f is an involution on the alphabet of u and v. The conclusion of our proposition follows from Theorem 3.

This conclusion follows, since the optimality is obtained from Example 15. □

Proof of Proposition 18: As in the proof of Proposition 16 denote by u' the longest prefix of u with $u' \in v\{v, f(v)\}^*$. Moreover, for some factorization $v = v_1 \cdots v_m$ with $|v_i| = \gcd(|u|, |v|) = d$ for all $1 \leq i \leq m$, we denote by $v' = v_1 \cdots v_i$ the prefix of v, for which $|v'| = |u| - |u'|$. It is straightforward that $\gcd(|v'|, |v|) = \gcd(|u|, |v|) = d \neq |v|/2$, so $\gcd(i, m) = 1$.

Assume first that $\alpha = u'v_1 \ldots v_i f(u)\alpha'$, where $\alpha' \in \{u, f(u)\}^*$, and note that, since u' is a prefix of β, we have a factorization $\beta = u'v\beta'$, with $\beta' \in \{v, f(v)\}^*$. It is rather plain that $v \in \{v_1, f(v_1), \ldots, f^k(v_1)\}$, where $ord(f) = k + 1$. Indeed, we obtain first that $f(v_1) = v_{i+1}$, and then, we obtain that $f(v_{i+1}) = v_{(2i+1) \mod m}$ or $f(v_{i+1}) = f(v_{(2i+1) \mod m})$; this holds as we look at the factor of β' matching the factor $f(v_{i+1})$ from the prefix $f(u)$ of $f(u)\alpha'$, and the choice depends on the starting word of β', namely v or $f(v)$). So $v_{(2i+1) \mod m} \in \{f(v_1), f^2(v_1)\}$, and so on: we always look at the factor $f(v_{(\ell i+1) \mod n})$ from $f(u)$ and see what word of β' matches it. Basically, we get that $v_{(\ell i+1) \mod m} \in \{v_1, f(v_1), \ldots, f^k(v_1)\}$, for all $\ell \geq 1$. Since i and m are coprime, it follows that $\{(\ell i+1) \mod n \mid \ell \in \mathbb{N}\} = \{1, 2, \ldots, m\}$, thus, $v_j \in \{v_1, f(v_1), \ldots, f^k(v_1)\}$ for all $1 \leq j \leq m$.

Let us first look at the case when $i < m/2$. It follows that $f(v_1) = v_{i+1}$ and $f^2(v_1) = f(v_{i+1}) = v_{2i+1}$. Looking at the prefix of α' we may have v_1 or $f(v_1)$, depending if either v or $f(v)$ occur at that position.

When we have $f(v_1)$, this may match a word v_{2i+1} or $f(v_{2i+1})$ from β'. In the first case we obtain that f is the identity on the letters of v_1, and the conclusion follows from Theorem 2, while in the latter we obtain that $v_1 = v_{2i+1}$, and so f is an involution on the letters of v_1 and the conclusion follows from Theorem 3.

In the second case, the word v_1 may match a word v_{2i+1} or a word $f(v_{2i+1})$ from β'. In the first case, we get that f is an involution on the letters of v_1, and the conclusion follows from Theorem 3. In the second case, a more careful analysis is needed. Let us denote $u = vu''$. If both $f(u'')$ and β' begin with $f(v)$, then the conclusion follows from Lemma 1, since from $f(v_1 \ldots v_{m-i})$ and $f(v)$ being powers of some word t', we get that $v_1 \ldots v_{m-i}$ and v are powers of a word t with $f(t) = t'$, and the conclusion follows immediately. If $f(u'')$ begins with $f^2(v)$ and β' begins with v we get that $f^2(v_1) = v_{i+1}$, and so f is the identity on the letters of v_1 and the conclusion follows from Theorem 2. Finally, if $f(u'')$ begins with $f^2(v)$ and β' with $f(v)$, or $f(u'')$ begins with $f(v)$ and β' with v we continue the discussion exactly as above but looking at the words that follow in $f(u'')$ and β', respectively. However, $f(u'')$ has the suffix $f(v_1 \ldots v_i)$ and this matches the beginning of a factor $f(v)$ of β' (as $f(v_{2i+1})$ appears at position $|u|+1$ in $v_{i+1} \ldots v_m \beta'$). Thus, $f(v_1) = f(v_{i+1})$, and we obtain that f is the identity on v_1, and the conclusion follows from Theorem 2.

When $i > m/2$, we have $2i+1 > m$, and so we obtain that $f^2(v_1) = f(v_{i+1}) \in \{v_{(2i+1) \bmod m}, f(v_{(2i+1) \bmod m})\}$. Both cases can be treated analogously to the previously presented ones, and so the conclusion follows in the same manner.

Finally, assume that $\alpha = u' f(v_1 \ldots v_i) f(u) \alpha'$, where $\alpha' \in \{u, f(u)\}^*$ and $u = vu''$. Note that u' is a prefix of β, such that $\beta = u' f(v) \beta'$, with $\beta' \in \{v, f(v)\}^*$. As in the previous case, it is rather plain that $v \in \{v_1, f(v_1), \ldots, f^k(v_1)\}$, where $ord(f) = k+1$. Now, we only have to look what is the prefix of β'. If this prefix is $f(v)$ the conclusion follows from Lemma 1. Otherwise, v is a prefix of β'. In this case we obtain that $f(v_1) = f(v_{i+1})$, and so $v_1 = v_{i+1}$, and $v_{i+1} \in \{f(v_1), f^2(v_1)\}$. Hence, f is either the identity or an involution on v_1. Therefore, the conclusion follows in this case, as well.

The conclusion follows, since the optimality is obtained from Example 17. □

Proof of Proposition 20: Since $|v| = 2\gcd(|u|, |v|)$, it follows that there exists a factorization of $v = v_1 v_2$ with $|v_1| = |v_2| = \gcd(|u|, |v|)$.

Assume first that $u \in v\{v, f(v)\}^* v_1$. If uu is a prefix of α, it follows that both v_1 and v_2 are prefixes of u. Since they have the same length, the conclusion follows. If $uf(u)$ is a prefix of α, since f is an antimorphism and v_1 is a suffix of u, it follows that $f(v_1)$ is a prefix of $f(u)$. Thus, both $f(v_1)$ and v_2 are prefixes of $f(u)$, and so $v_2 = f(v_1)$ and $f(v_2) = f^2(v_1)$. We get that $v = v_1 f(v_1)$ and $u \in v_1\{v_1, f(v_1), f^k(v_1)\}$, and the conclusion follows.

Now let us assume that $u \in v\{v, f(v)\}^* f(v_2)$. If uu is a prefix of α, it follows that both v_1 and $f(v_1)$ are prefixes of u. Looking at what follows $v_1 = f(v_1)$ in the second occurrence of u, we have that either $v_2 = f(v_2)$, or $v_2 = v_1$. Using Lemma 9 we get in both cases that f is an involution, and the conclusion follows from Theorem 4.

If $uf(u)$ is a prefix of α, since f is an antimorphism and $f(v_2)$ is a suffix of u, it follows that $f^2(v_2)$ is a prefix of $f(u)$. Thus, both $f(v_1)$ and $f^2(v_2)$ are prefixes of $f(u)$, and so $v_2 = f^k(v_1)$ and $f(v_2) = v_1$. We get that $v = v_1 f^k(v_1)$ and $u \in v_1\{v_1, f(v_1), f^k(v_1)\}$, and the conclusion follows. □

Proof of Lemma 22: Assume first that $w = w_1 \cdots w_n$ is a proper factor of $wf(w)$, where n is the length of w. It follows that for some j with $1 < j \leq n$, we have that $w_j \cdots w_n = f(w_n) \cdots f(w_j)$, and using Lemma 9 we get that for the alphabet of this factor, f is an involution. Looking now at the equality $w_1 \cdots w_{j-1} = w_{n-j+1} \cdots w_n$, one can easily prove that the alphabet of this factor is the same as the one of $w_j \cdots w_n$, and so f is an involution for all letters in w.

If w is a proper factor of $f(w)w$, then $w_1 \cdots w_j = f(w_j) \cdots f(w_1)$, and again by Lemma 9, it follows that for the alphabet of this factor, f is an involution. Looking at the equality $w_{j+1} \cdots w_n = w_1 \cdots w_{n-j}$, one concludes once again that f is an involution for w.

If w is a proper factor of $f(w)f(w)$, then $w_1 \cdots w_j = f(w_j) \cdots f(w_1)$ and $w_{j+1} \cdots w_n = f(w_n) \cdots f(w_{j+1})$, and again by Lemma 9, we conclude that f is an involution for w.

Assume now that $f(w)$ is a proper factor of $wf(w)$. It follows that for some j with $1 < j \leq n$, we have that $f(w_n) \cdots f(w_j) = w_j \cdots w_n$, and by Lemma 9, it follows that for the alphabet of this factor, f is an involution. Looking now at the equality $f(w_{j-1}) \cdots f(w_1) = f(w_n) \cdots f(w_{n-j+1})$, one can easily prove that the alphabet of this factor is the same as the one of $w_j \cdots w_n$, and so f is again an involution for all letters in w.

If $f(w)$ is a proper factor of $f(w)w$, then $f(w_j) \cdots f(w_1) = w_1 \cdots w_j$, and again by

Lemma 9, it follows that for the alphabet of this factor, f is an involution. Looking at the equality $f(w_n)\cdots f(w_{j+1}) = f(w_{n-j})\cdots f(w_1)$, one concludes once again that f is an involution for the entire alphabet of w.

Finally, consider the case when $f(w)$ is a proper factor of ww. Since $f(w_n)\cdots f(w_j) = w_j\cdots w_n$ and $f(w_{j-1})\cdots f(w_1) = w_1\cdots w_{j-1}$, by Lemma 9 we conclude that f is an involution for the alphabet w is defined on. □

Proof of Proposition 23: The proof of this statement is based on the key remark that the prefix u in α is followed by either v, the prefix of u, or $f(u)$, which has $f(v)$ as a suffix. In both cases, since $|v| \geq 3\gcd(|u|,|v|)$, we get that either v or $f(v)$ are proper factors of some word in $\{v, f(v)\}^2$. If not all factors are equal, the proof follows from Lemma 22.

It is important to note here that, in the case when α has $uf(u)$ as a prefix, the suffix $f(v)$ of $f(u)$ is a proper factor of $\{v, f(v)\}^2$. This is true since, otherwise we would have that for some coprime integers k, k' with $|u| = kd$ and $|v| = k'd \geq 3d$, there would exist an integer h such that $2kd = hk'd$. Thus, from $2k = hk'$ and the fact that k and k' are coprime, we get that $k = h$ and $k' = 2$, which is a contradiction.

In the other case, denoting $u \in v\{v, f(v)\}^*v'$, or $u \in v\{v, f(v)\}^*f(v'')$, for some appropriate factorization $v = v'v''$, and using Lemma 1, we have that v' and v'' are powers of the same word. The conclusion easily follows since, then, also u is an f-power of the same word.

Remark that when $f(v)$ is a proper factor of $f(v)f(v)$, we get that $f(v) = t^j$, for some integer j with $1 < t \leq k+1$. In this case we have $v = f^{k-j+1}(t)$ and $f(v) = f^{k-j+2}(t)$. □