Avoiding Redundancies in Words

Badkobeh, Golnaz

Awarding institution:
King's College London

Avoiding Redundancies in Words

Golnaz Badkobeh

A Thesis Submitted for the Degree of Doctor of Philosophy

Department of Informatics
King's College London

12.12.2012

Acknowledgements

I would like to thank my supervisor, Prof. Maxime Crochemore and my second supervisor, Prof. Costas Iliopoulos for their unyielding support, guidance and friendship throughout my PhD.

I am also grateful to my collaborators: Dr. Ochem and Dr. Rao. Also the following people whom I have been privileged to meet in various occasions and our discussions have made a positive impact on my work: Prof. Karumäki, Dr. Puglisi, Dr. Shur and Prof. Smyth, to name a few.

Thanks also goes to Carl, Farzaneh, Laura, Shekoofeh and Vida who have been always there to support me.

Many people in the Department of Informatics at King's have offered support in various forms. There are too many people to mention by name, but I am grateful to them all. I would like to thank in particular Dr. Bennett, Dr. Black, Dr. Modgil , Mr. Hampson and Ms. Abdelhadi for kindly reading chapters of my thesis.

I would like to mention my beloved son, Parsa. His smile and unconditional love is the greatest motivation.

Finally, this thesis is dedicated to Shahnaz S.Fard, my mum, who never stopped believing in me.

Abstract

The study of Combinatorics on words started at the beginning of the 20th century with the work of the Norwegian Mathematician Axel Thue, who published several articles in a relatively unknown journal. His work had primarily theoretical objectives, but ever since many of his results have been rediscovered independently by other researchers in relation to other problems. Although many questions have been studied and solved in the area, there are yet many open questions left to be studied. Among the basic discoveries of Thue are the existence of infinite words with no occurrence of squares (words of the form uu for a nonempty word u) on an alphabet of at least three symbols, and with no occurrence of cubes (and even overlaps) on a binary alphabet.

The constraints on repetitions in infinite words have been raised to optimality after Dejean's conjecture on the repetitive threshold associated with the alphabet size, which last cases have been proved recently by Rao after the works of Carpi , Pansiot , Moulin-Ollagnier, Mohammad-Noori and Currie, Currie and Rampersad. The first case says that the repetitive threshold of the binary alphabet is 2 (infinite binary words can avoid factor of exponent larger than 2 but cannot do more) and the second case, proved by Dejean, states that it is 7/4 for the three-letter alphabet.

The constraint studied later on by Fraenkel and Simpson is somewhat orthogonal to the previous notion. Their parameter to the complexity of binary infinite words is the number of squares occurring in them without any restriction on the number of occurrences.

The analysis of repetitions in strings is primarily of combinatorial interest in relation to the entropy of sequences. But repetitions or repeats are also of main concerns in the domains of text compression and of pattern matching. The knowledge of extreme situations or strongest constraints on words help analyse the behaviour of the corresponding algorithms.

In this document, we provide a new proof for the Fraenkel and Simpson result, we give a proof that there exists an infinite binary word which contains finitely many squares and simultaneously avoids words of exponent larger than 7/3, which leads us to the concept introduced hereafter. A chapter is dedicated to new notion of Finite-Repetition threshold and some results about it.

We give some new results on the trade-off between the number of squares and the number of maximal-exponent powers in infinite binary words.This is done in three cases where the maximal exponent is 7/3, 5/2, and 3, that is the only cases of interest. We show that there exists no infinite 3^+-free binary word avoiding squares of odd period. This study also reveals there exists no infinite binary word, simultaneously avoiding cubes and squares of even period. Moreover, we proof that there exists

an infinite 3^+-free binary word avoiding squares of even-period length. We investigate the trade-off between the maximal period length of repetitions contained and their number. Similarly we exhibit a trade-off between number of cubes and number of squares occurring in an infinite word avoiding even-period squares. All bounds provided in these cases are shown to be optimal.

Repetitions or repeats are also of main concern in the domains of text compression and of pattern matching. The knowledge of extreme situations or strongest constraints on words helps analyse the behaviour of the corresponding algorithms. In this document we mostly deal with the combinatorial aspects of the question. The algorithmic part is strongly linked and it is used to explore the words satisfying constraints on the repetitions they contain.

Publications

During the compilation of this thesis, the following related articles have been published by the author.

1. Golnaz Badkobeh. Fewest repetitions vs maximal-exponent powers in infinite binary words. *Theoretical Computer Science*, 412(48):6625–6633, 2011.

2. Golnaz Badkobeh. An infinite binary word containing only three distinct squares. 2012. *Submitted to Journal of Discrete Algorithms*.

3. Golnaz Badkobeh, Supaporn Chairungsee, and Maxime Crochemore. Hunting redundancies in strings. In *Developments in Language Theory*, pages 1–14, 2011.

4. Golnaz Badkobeh and Maxime Crochemore. Finite-Repetition threshold for infinite ternary words. In *WORDS*, pages 37–43, 2011.

5. Golnaz Badkobeh and Maxime Crochemore. Fewest repetitions in infinite binary words. *RAIRO-Theoretical Informatics and Applications*, 46(1):17–31, 2012.

6. Golnaz Badkobeh and Maxime Crochemore. Infinite binary word containing odd-period repetitions. 2012. *Submitted to The Electronic Journal of Combinatorics*.

7. Golnaz Badkobeh, Maxime Crochemore, and Chalita Toopsuwan. Computing the maximal-exponent repeats of an overlap-free string in linear time. In E. C. L. Calderon-Benavides, C. Gonzlez-Caro and N. Ziviani, editors, *Symposium on String Processing and Information Retrieval*, number 7608 in LNCS, pages 61–72. Springer, 2012. **Awarded the Best Paper of SPIRE 2012**

8. Golnaz Badkobeh and Pascal Ochem. Characterization of some binary words with few squares. 2012. *Submitted to Theoretical Computer Science*.

9. Golnaz Badkobeh, Michaël Rao, and Maxime Crochemore. Finite-repetition threshold for large alphabets. In *14th Mons Days of Theoretical Computer Science*, 2012. To appear.

Contents

1

Avoiding Redundancies in Words

The study of Combinatorics on words started at the beginning of the 20th century with the work of the Norwegian Mathematician Axel Thue [65, 66] (see [14]) who published several articles in a relatively unknown journal. His work had primarily theoretical objectives, but ever since many of his results have been rediscovered independently by other researchers in relation to other problems. Although many questions have been studied and solved in the area, there are yet many open questions left to be studied.

The analysis of repetitions in strings is primarily of combinatorial interest in relation to the entropy of sequences. But repetitions or repeats are also of main concern in the domains of text compression and of pattern matching. The knowledge of extreme situations or strongest constraints on words helps analyse the behaviour of the corresponding algorithms. In this document we mostly deal with the combinatorial aspects of the question. The algorithmic part is strongly linked and it is used to explore the words satisfying constraints on the repetitions they contain.

Among the basic discoveries of Thue are the existence of infinite words with no occurrence of squares (words of the form uu for a nonempty word u) on an alphabet of at least three symbols, and with no occurrence of cubes (and even overlaps) on a binary alphabet.

Avoiding repetitions was explored further by Bean, Ehrenfeucht and McNulty [11] in the form of avoidability of patterns. We discuss this topic in detail below.

Apart from their composition, repetitions are characterised by their length, their periods, and their exponent. This latter parameter is the most prominent in previous studies but others are also considered in this document.

A word x is a factor of y if y is uxv, where u and v are two words. A nonempty word x has period p if its letters at distance p are equal. The exponent of a word is the quotient of its length over its smallest period. For example `alfalfa` has period 3 and exponent 7/3. A string with exponent e is also called an e-power. The notion of maximal exponent is central in questions related to the avoidability of patterns in

infinite words. An infinite word is said to avoid e-powers (resp. e^+-powers) if the exponents of its finite factors are smaller than e (resp. no more than e).

The constraints on type of repetitions occurring in infinite words have been raised to optimality after Dejean's conjecture [32] on the repetitive threshold associated with the alphabet size. *The repetitive threshold* (Dejean's repetitive threshold) of order k is the infimum of maximal exponents of factors of all (infinite) words over a k-letter alphabet.

The first (interesting) case concerns the binary alphabet. The conjecture says that the repetitive threshold is 2 (that is, infinite binary words can avoid factors of exponent larger than 2 but cannot avoid squares). The second case, solved by Dejean [32], states that it is 7/4 for the three-letter alphabet. She then conjectured that $r_4 = 7/5$ and $r_k = \frac{k}{k-1}$ for $k \geq 5$. The conjecture remained open for about fourty years despite several partial results. The last cases have been eventually proved recently by Rao [56] after the works of Carpi [19], Pansiot [53], Moulin-Ollagnier [49], Mohammad-Noori and Currie [48], Currie and Rampersad [29]. All these results contribute to the proof of Dejean's conjecture.

The constraint studied later on by Entringer, Jackson and Schatz [34] is somewhat orthogonal to the previous notion. Their parameterisation of the complexity of binary infinite words is the number of squares occurring in them without any restriction on their number of occurrences. Let $g(n)$ be the length of a longest binary word containing at most n squares. They showed in 1974 that there exists an infinite word with 5 different squares, i.e. $g(5) = \infty$. Then, Fraenkel and Simpson [36] refined this result by showing that there exists an infinite binary word that has only the three squares 00, 11, and 0101, and thus $g(3) = \infty$. A somewhat simplified proof of this result was given by Rampersad, Shallit and Wang [55], using two uniform morphisms (a structure-preserving mapping from one word to another). Later, Harju and Nowotka [38] gave a simpler proof of the same result.

It is fairly straightforward to check that no infinite binary word can contain less than three squares, then $g(0) = 3$ (e.g. `010`). A simple checking shows that $g(1) = 7$ (e.g. `0001000`) and $g(2) = 18$ (e.g. `010011000111001101`).

Contribution [3]: In Chapter 3, we provide yet a new proof that the maximal length of binary words containing at most 3 squares is infinite. The proof is based on an iterated morphism and a translating morphism. The second morphism used in the proof is the simplest of its form, that can generate an infinite binary word with at most 3 squares. Simplest in the sense that the sum of all its codewords is 24, the smallest possible value amongst the existing morphisms satisfying the property.

Instead of avoiding squares, an interesting variation on the avoidability of repe-

titions is to omit large repetitions. Entringer, Jackson and Schatz [34] showed that there exist infinite binary words avoiding squares of period at least three. Following their work, avoiding large squares has been studied by Dekking [33], Rampersad et al. [55], Shallit [64], Ochem[50], and many others.

Contribution [7]: In Chapter 3, we provide some new results as an outcome of studying the pattern avoidance from a different point of view. We analyse the possibility of avoiding repetitions of even and odd periods, and further impose a constraint on their maximal exponent. We show that there exists no infinite 3^+-free binary word avoiding all squares of odd period. This study reveals there exists no infinite binary word, simultaneously avoiding cubes and squares of even period. Moreover, we prove that there exists an infinite 3^+-free binary word avoiding squares of even-period length.

We study a trade-off between the maximal period length and number of repetitions after a similar trade-off between number of cubes and number of distinct squares. We succeed in reducing the number of repetitions contained in infinite binary words without compromising the constraint on parity of their period. We conclude that in such words the minimal number of squares is 7 when only 1 cube occurs. The number reduces to 4 when 2 cubes are allowed in the word.

Avoiding large squares in words whose maximal exponent is constrained, has been studied by various combinatorists. To name a few, Karhumäki and Shallit [41] showed:

(i) Every infinite 7/3-free binary word contains arbitrarily large squares.

(ii) There exists an infinite $7/3^+$-free binary word such that each square factor ww satisfies $|w| \leq 13$.

Later, Shallit [64] refined the result and showed: there exists an infinite $7/3^+$-free binary word such that each square factor ww satisfies $|w| \leq 7$.

Contribution [6]: In Chapter 4, we introduce a new constraint on infinite words and give some results. We also state some conjectures that need further and deeper investigations. Looking at the maximal exponent of factors in words containing a bounded number of r_k-powers introduces a new type of threshold, that we call the *finite-repetition threshold*. For the alphabet of k letters, $\mathrm{FRt}(k)$ is defined as the smallest rational number for which there exists an infinite word avoiding $\mathrm{FRt}(k)^+$-powers and containing a finite number of r_k-powers, where r_k is Dejean's repetitive threshold. Associated with the *finite-repetition threshold* is the smallest number of r_k-powers (limit repetitions), $Rn(k)$, that an infinite Dejean's word can accommodate.

The results by Karhumäki and Shallit [41] can then be restated as $\mathrm{FRt}(2) = 7/3$.

Contribution [6, 5, 10]: In Chapter 4, we first provide a new proof for FRt(2) = 7/3 and we show that the associated number of squares is 12 ($Rn(2) = 12$). Second, we show that FRt(3) = r_3 = 7/4. Proofs provided in this chapter are two-fold, because they have to show the value of FRt(k) as well as the associated number of r_k-powers. We prove on ternary words the minimum number of associated r_k-powers is 2. The result completes Dejean's result on the 3-letter alphabet. Indeed, the only previous proof of the 7/4 repetition threshold is due to Dejean [32], where she has given an infinite word which is pure morphic word; generated by an endomorphism, which readily implies the number of 7/4-powers contained in her word is not bounded.

Moreover, we show that there exists an infinite word on 4 letters containing only 2 7/5-powers and no factor of exponent more than 7/5. The only known proofs of the 7/5 repetition threshold for 4 letters are due to Pansiot [53] and Rao [56]; both their words contain 24 7/5-powers.

Next, we turn to the 5-letter alphabet. The only proof of the 5/4 threshold is by Moulin-Ollagnier [49]. After showing that it provides a word with 360 5/4-powers of periods 4, 12 and 44, we show that the number of 5/4-powers can be reduced to 60 and conjecture that it can be reduced further to 45, the smallest possible number.

Finally, revisiting the existing morphisms and proofs of Dejean's conjecture we show for $k \geq 5$, FRt(k) = r_k. Now the question worth investigation becomes: what is the minimum number of associated r_k-powers, ($Rn(k)$), occurring in an infinite k-ary word complying with FRt(k)?

The idea of repetitive threshold was extended into the generalised repetition threshold by Ilie et al. in [39] as follows. There, the notion of (β, p)-freeness is introduced: a word is (β, p)-free if it contains no factor that is a (β', p')-repetition (it is a word w with period length p' and exponent β': $w = p^{\beta'}$) for $\beta' \geq \beta$ and $p' \geq p$. Then, a word is (β^+, p)-free if it is (β', p)-free for all $\beta' > \beta$ and the generalised repetition threshold $R(k, p)$ is defined for k-letter alphabet as the real number α such that either

(a) there exists an (α^+, p)-free infinite word and all (α, p)-free words are finite; or

(b) there exists an (α, p)-free infinite word and for all $\alpha > 0$, $(\alpha - \epsilon, p)$-free words are finite.

where p is the minimal avoided period.

A proof of boundary of this threshold for all alphabet sizes is also presented in [39]. Essentially $R(k, 1)$ is Dejean's repetitive threshold. Karhumäki and Shallit in [41] and Ochem in [50] have studied binary words under two constraints: maximal exponent and the longest period.

Contribution [2]: In Chapter 5, we provide some results that gives deeper insight into the question of avoidable patterns in infinite binary words, by introducing another point of view. We analyse the trade-off between the number of (distinct) squares and the number of maximal-exponent repetitions occurring in infinite binary words when the maximal exponent is constant. The interesting results show the behaviour of infinite binary words when the maximal exponent varies between 3 to 7/3. The value 7/3 is called the Finite-repetition threshold as mentioned above, and the value 3 of the maximal exponent is where the number of squares is the absolute minimum, i.e. 3. The next table summarises the results.

Maximal exponent e	Allowed number of e-powers	Minimum number of squares
7/3	2	12
	1	14
5/2	2	8
	1	11
3	2	3
	1	4

Proving that it is impossible to have less than a given number of squares when avoiding some e-powers occurring in binary words needs a simple computation. For every case, to generate the binary word with the desired property, we first use a pure morphic word and we translate its corresponding fixed point to a binary word with a second morphism.

Finding infinite words that avoid repetitions has its roots in the work of Thue, and has been pursued, in particular, in connection with problems of algebra. The general idea of unavoidable patterns was introduced independently by Bean, Ehrenfeucht and McNulty [11] and by Zimin [67].

A pattern is a finite word over the alphabet of capital letters $\{A, B, ...\}$. An occurrence of a pattern is obtained by replacing each alphabet letter with a non-empty word. For example, the word 0111010011 is an occurrence of the pattern $ABBA$ where $A \to 011$ and $B \to 10$; it also contains another occurrence of this pattern (i.e. 1001) as a factor. Formally, a word avoids a pattern P if it contains no occurrence of P as a factor. The avoidability index $\lambda(P)$ of the pattern P is the smallest alphabet size over which an infinite word avoiding P exists. Patterns such as $A, ABC, ABA, ABACBA$ cannot be avoided with any finite alphabet. These patterns are said to be unavoidable, denoted as $\lambda(P) = \infty$, and have been characterised by Zimin [67].

A pattern, P is said to be k-avoidable if there exists an infinite word on k letters avoiding P. Thue [65, 66] showed in fact that AA is 2-unavoidable but 3-avoidable,

and A^β for $\beta > 2$ is 2-avoidable. Schmidt [62] proved that every binary pattern of length at least 13 is 2-avoidable. Later, Roth refined the result in [58] by showing that every binary pattern of length 6 is 2-avoidable.

Thereafter,the remaining set to be examined is a finite set of patterns of length at most 5. Cassaigne [21] completed this study by considering all the patterns in this set and reached the conclusion:

- 2-unavoidable patterns : $\epsilon, A, AA, AB, AAB, ABA, AABA, ABBA, AABB, ABAB, AABAA, AABAB$;
- 2-avoidable patterns: $AAA, ABAAB, AABBA, ABABA$.

Note that if a pattern is unavoidable so are all its factors, therefore we can represent all 2-unavoidable patterns by the following minimal set

$$\{ABBA, AABB, AABAA, AABAB\}.$$

A factor of an infinite word is recurrent if it occurs infinitely often in that word. Given a finite set $\mathcal{P}$ of patterns and a finite set $\mathcal{F}$ of words over Σ_k, we say that $\mathcal{P} \cup \mathcal{F}$ *characterises* a morphic word $w \in \Sigma_k^*$ if every recurrent word occurring in an infinite word avoiding $\mathcal{P} \cup \mathcal{F}$ is a factor of w.

There is still no characterisation of k-unavoidable patterns, i.e. patterns that are unavoidable over k-letter alphabet. Thue [65, 66] (see [14]) gave the characterisation of overlap-free binary words: $\{ABABA\} \cup \{000, 111\}$. The set characterises the fixed point of the morphism $0 \mapsto 01$, $1 \mapsto 10$, the Thue-Morse word, which also avoids AAA. Roth [58] proved the pattern $ABAAB$ is avoidable and finally, Cassaigne [21] proved the only remaining pattern, $AABBA$ to be avoidable.

Although the avoidability of binary patterns on binary words is complete, Samsonov and Shur [61] started a variation of this study on cube-free binary words. As mentioned above the pattern $ABABA$ is avoided by the Thue-Morse word. This is the only pattern of length at most 5 that is avoidable by cube-free words. Here is the list of all eight cube-free patterns of length 6, excluding equivalent patterns by reversal and negation: $\{AABAAB, AABABA, AABABB, AABBAA, AABBAB, ABAABA, ABABBA, ABBAAB\}$

The first two patterns $AABAAB$ and $AABABA$ are obviously avoided by the Thue-Morse word. Samsonov and Shur show in a recent article [61] that patterns $AABBAB$, $ABAABA$, $ABABBA$ and $ABBAAB$ are not avoidable by binary cube-free words. And the pattern $AABBAA$ is avoidable by binary cube-free word. The only pattern with unclear avoidability status is $AABABB$; it is conjectured to be avoidable by cube-free words in the same article, but this has not yet been proven.

Cassaigne [22] partitioned all ternary patterns into 2-unavoidable patterns, 2-avoidable patterns, and patterns with unclear status. One of the patterns with unclear status, $ABCBABC$, was proved by Ilie et al. [39] to be 2-avoidable. The remaining cases were proved to be also 2-avoidable by Ochem [50].

Contribution [9]: In Chapter 6, we prove such characterisations mostly for the binary words considered in [2] that contain one or two 2^+-repetitions and as few squares as possible. The results are summarised in the following table. We use the notation SQ_t for the pattern corresponding to squares of words of length at least t, that is, $SQ_1 = AA$, $SQ_2 = ABAB$, $SQ_3 = ABCABC$, and so on. We denote a set of avoided factors in each infinite words by F_k, where k is the number of squares contained in that string.

Maximal exponent e	(Number of e-powers, Minimum number of squares)	Avoided patterns and factors
5/2	(2, 8)	$\{SQ_7\} \cup F_8$
7/3	(1, 14)	$\{SQ_9\} \cup F_{14}$
7/3	(2, 12)	$\{SQ_9\} \cup F_{12}$
3	(2, 3)	$\{SQ_5\} \cup F_3$
5/2	(1, 11)	$\{SQ_3\} \cup F_{11}$

We also give a characterisation for words avoiding the patterns $AABBCC$ (i.e., three consecutive squares) as the union of SQ_3 and a finite set of factors.

We now discuss the algorithmic part of the document. The exponent of a string can be calculated in linear time using basic string matching that computes the smallest period associated with the longest border of the string (see [25]). A naive consequence provides a $O(n^3)$-time solution to compute the maximal exponent of all factors of a string of length n since there are potentially of the order of n^2 factors. But a quadratic time solution is also a simple application of basic string matching. In contrast, the solution provided in Chapter 7 runs in linear time on a fixed-size alphabet.

When a string contains runs, that is, maximal occurrences of repetitions of exponent at least 2, computing their maximal exponent can be done in linear time by adapting the algorithm of Kolpakov and Kucherov [44] that computes all the runs occurring in the string. Their result relies on the fact that there exists a linear number of runs in a string [44] (see [60, 27] for precise bounds). But this does not apply to square-free strings.

Repeats are string of exponent at most 2. They refer to strings of the form uvu where u is its longest border (both a prefix and a suffix). The study of repeats in a string has to do with long-distance interactions between separated occurrences of the

same segment (the u part) in the string. Although occurrences may be far away from each others, they may interact when the string is folded as is the case for genomic sequences.

Contribution [8, 4]: In Chapter 7, we consider the question of computing the maximal exponent of repeats occurring in a given string. Thus, we are looking for factors w of the form uvu, where u is the longest border of w. To do so, we use two main tools: a factorisation of the string and the Suffix Automaton of some factors. Our solution works on overlap-free strings for which the maximal exponent of factors is at most 2.

The Suffix Automaton is used to search for maximal repeats in a product of two strings due to its ability to locate occurrences of all factors of a pattern. Here, we enhance the automaton to report the right-most occurrences of those factors. Using it alone in a balanced divide-and-conquer manner produces a $O(n \log n)$-time algorithm. To eliminate log factor we additionally use the f-factorisation of the string. It has now become common to use it to derive efficient or even optimal algorithms. The f-factorisation (see [25]), a type of LZ77 factorisation fit for string algorithms, allows to skip larger and larger parts of the strings during an online computation. For our purpose, it is composed of factors occurring before their current position with no overlap. The factorisation can be computed in $O(n \log a)$-time using a Suffix Tree or a Suffix Automaton, but also in linear time on an integer alphabet using a Suffix Array [28].

The running time of the proposed algorithm depends additionally on the repetitive threshold of the underlying alphabet of the string. The threshold restricts the context of the search for a second occurrence of u associated with a repeat uvu.

We show a very surprising property of repeats whose exponent is maximal in an overlap-free string: there are no more than a linear number of occurrences of them, although the number of occurrences of maximal (i.e. non extensible occurrences of) repeats can be quadratic. As a consequence, the algorithm can be upgraded to output all occurrences of maximal-exponent repeats of an overlap-free string in linear time.

The question would have a simple solution by computing MinGap on each internal node of the Suffix Tree of the input string. MinGap of a node is the smallest difference between the positions assigned to leaves of the subtree rooted at the node. Unfortunately, the best algorithms for MinGap computation, equivalent to MaxGap computation, run in time $O(n \log n)$ (see [12, 40, 16] and the discussion in [23]).

A remaining question to the present study is to unify the algorithmic approaches for repetitions of exponent at least 2 and for repeats of exponent at most 2.

2
Preliminaries

In this chapter we are defining some standard terms in combinatorics on words. Also a review is given of essential background and related work that underpins the technique used in thesis. This includes introducing the topic of morphisms and L-systems and reviewing properties of a few fundamental existing systems. Later the methodology used throughout this research is explained.

An alphabet is any set, the members of which are called letters. Let Σ_k denote the alphabet of size k, that is, $\Sigma_k = \{0, 1, \ldots, k-1\}$. For the sake of clarity we also denote the binary alphabet $\Sigma_2 = \{0, 1\}$ as B, and we use equivalently A $= \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ to represent the ternary alphabet $\Sigma_3 = \{0, 1, 2\}$.

A word or a string is a sequence of letters drawn from an alphabet. The empty word, denoted by ϵ is a string with no letters and is considered as a word on every alphabet. The concatenation of two words u and v is denoted by the mere juxtaposition uv. Equipped with the concatenation on words, the set of all finite words over the alphabet Σ_k, namely Σ_k^*, becomes a monoid since the operation is associative and ϵ is its neutral element. The length of the word w, denoted by $|w|$, is the number of occurrences of letters in w. Hence $|\mathtt{abaca}| = 5$.

The word v is called a factor of x if there exist words u and w such that $x = uvw$; in such case that $u = \epsilon$ (resp. $w = \epsilon$) then v is a prefix (resp. suffix) of x. Let x be a non-empty string. An integer p such that $0 < p \leq |x|$ is called a period of x if $x[i] = x[i + p]$ for $i = 0, 1, ..., |x|p1$. The exponent of x is the quotient of its length over its smallest period. For example `alfalfa` has period 3 and exponent 7/3. If the exponent is e and prefix period word is u (the prefix of length p), then $x = u^e$. Thus a *square* is any word with an even integer exponent. *Cubes* and *kth powers* are defined accordingly.

The maximum exponent of a word w is the supremum of $E(x)$, where $E(x)$ is the set of exponents of all factors of x. A word is *overlap-free* if it does not contain any factor of the form $xyxyx$ for a non-empty word x. In general a word is said to be *α-free* if it contains no factor of the form u^β for any rational number $\beta \geq \alpha$. It is

α^+-*free* if it contains no factor of the form u^β for any rational number $\beta > \alpha$.

A morphism is a map $f : \Sigma_n^* \to \Sigma_m^*$ compatible with the structure of monoid of both sets. This means that $f(uv) = f(u)f(v)$ for all $u, v \in \Sigma_n^*$ and that $f(\epsilon) = \epsilon$. As a consequence, the morphism is completely defined by couples $(a, f(a))$ for $a \in \Sigma_n$. If $f(a) = ax$ for some letter $a \in \Sigma_n$ we say that f is prolongable on a. A fixed point of a morphism f is an infinite word w such that $f(w) = w$. As $f^n(a)$ is a prefix of $f^{n+1}(a)$ then the limit of $f^i(a)$ exists, thus we can iterate f infinitely many times from a to get an infinite word: $f^\infty(a) := axf(x)f^2(x)f^3(x)\cdots$, that is a fixed point of the morphism. For $q \geq 2$ a morphism f is said to be q-uniform if $|f(a)| = q$ for all $a \in \Sigma_n$. An endomorphism is a special case of morphism for which $n = m$.

An endomorphism $h : \Sigma_n^* \to \Sigma_n^*$ is square-free (resp. cube-free, overlap-free) if $h(w)$ is square-free (resp. cube-free, overlap-free) whenever $w \in \Sigma_n^*$ is square-free (resp. cube-free, overlap-free). The morphism $h : \Sigma_n^* \to \Sigma_m^*$ is synchronising if for any $a, b, c \in \Sigma_n$ and $v, w \in \Sigma_m^*$, $h(ab) = vh(c)w$ implies either $v = \epsilon$ and $a = c$ or $w = \epsilon$ and $b = c$. Uniform morphisms have particularly nice properties. For example, the class of words generated by applying a coding to infinite iteration of q-uniform morphisms coincides with the class of q-automatic sequences generated by finite automata [1].

To exhibit infinite words we often iterate a morphism and then translate its infinite fixed point. The translation is done with a second morphism. The process can be translated in terms of L-systems (see [59] for example) as follows. A D0L-system is defined as a triple (Σ, f, w) where Σ is an alphabet, f is a morphism from Σ^* to itself, and w is a word over Σ. An HD0L-system is defined as a 5-tuple $(\Sigma, \Sigma', f, h, w)$ where Σ and Σ' are alphabets, (Σ, f, w) is a D0L system and h is a morphism from Σ to Σ'. In our HD0L systems $(\Sigma, \Sigma', f, h, a)$, a is a letter in Σ, f stands for the morphism prolongable on a that is iterated, h stands for the translation, and we are interested in the properties of the infinite word $h(f^\infty(a))$.

1 Binary Morphisms

In 1906, Thue [65] established that the squares are avoidable on 3-letter alphabets and cubes are avoidable on 2-letter alphabets. Iterating the endomorphism m defined by:

$$m(\mathtt{0}) = \mathtt{01},$$
$$m(\mathtt{1}) = \mathtt{10}.$$

starting with the word $\mathtt{0}$ gives Thue-Morse sequence:

$$\mathrm{TM} = \mathtt{0110100110010110100101 10\ldots}$$

which is overlap-free [65] (see also [47]). Pansiot [52] observed that the only morphisms generating the Thue-Morse word are powers of m. This was extended by Séébold as follows.

Theorem 1 ([63]). *Let x be an infinite overlap-free binary word that is generated by iterating some morphism h. Then h is a power of m.*

Let $u(n)$ be the number of overlap-free binary words of length n, Restivo and Salemi [57] gave a polynomial upper bound to $u(n)$.

Theorem 2 ([57]).

$$u(n) \leq C * n^r$$

with $C > 0$ and $r = log_2 15 \simeq 3.906$.

Kobayashi [43] improved these bounds:

$$C_1 * n^{1.155} \leq u(n) \leq C_2 * n^{1.587}$$

The lower bound is obtained by counting the overlap-free words that are infinitely extensible on the right.

Moreover, Cassaigne [20] gave an upper bounds and lower bounds for α and β where

$$C_1 * n^{\alpha} \leq u(n) \leq C_2 * n^{\beta}$$

He showed α and β are distinct and together with Kobayashi's result $1.155 < \alpha < 1.276 < 1.332 < \beta < 1.587$. Finally, Carpi [18] proved that a finite automaton can be used to compute $u(n)$ and demonstrated a method to find upper bounds $C * n^r$ with r arbitrarily close to the optimal value.

Later, Currie and Rampersad [30] showed there are k-uniform cube-free binary morphisms for $k \geq 1$. The empty morphism and the identity morphism are obviously cube-free and Thue-Morse morphism is cube-free since it is overlap-free. For all $k > 2$ Currie and Rampersad [30] demonstrated how to build k-uniform cube-free morphisms.

2 Ternary Morphisms

The infinite word of Thue-Morse contains squares and in fact its only binary square-free factors are: $\epsilon, a, b, ab, ba, aba, bab$. In Thue-Morse word between two consecutive `0`s there is either ϵ or `1` or `11`, therefore the following morphism, t is well defined.

The string derived from Thue-Morse sequence by taking its inverse image with the morphism t:

$$t(\mathtt{a}) = \mathtt{011},$$
$$t(\mathtt{b}) = \mathtt{01},$$
$$t(\mathtt{c}) = \mathtt{0}.$$

is square-free [66] (see also [47]), and is the fixed point of the morphism f defined from Σ_3^* to itself by:

$$f(\mathtt{a}) = \mathtt{abc},$$
$$f(\mathtt{b}) = \mathtt{ac},$$
$$f(\mathtt{c}) = \mathtt{b}.$$

Since the letter a is a prefix of $f(\mathtt{a})$, the infinite word $\mathbf{f} = f^\infty(\mathtt{a})$ is well defined. It is known that this word is square-free (see [47, Chapter 2]). However, the morphism f is not square-free since $f(\mathtt{aba}) = \mathtt{abcacabc}$ contains the square caca while aba is square-free. A morphism whose fixed point is square-free is called *weakly square-free*. It can additionally be checked that all square-free words of length 3 occur in $\mathbf{f}$ except aba and cbc.

These results were independently rediscovered by Arshon in 1937 and by Morse and Hedlund around 1940. Little is known about the characterisation of weakly square-free morphisms.

There are many square-free morphisms and in order to show that a morphism is square-free on 3-letter alphabet, it is sufficient to show that the images of the twelve words: {aba, abc, aca, acb, bab, bac, bca, bcb, cab, cac, cba, cbc} are square-free [11].

Corollary 1. *[66] The endomorphisms h and g defined respectively by:*

$$h(\mathtt{a}) = \mathtt{abcab},$$
$$h(\mathtt{b}) = \mathtt{acabcb},$$
$$h(\mathtt{c}) = \mathtt{acbcacb}.$$

and by

$$g(\mathtt{a}) = \mathtt{abacb},$$
$$g(\mathtt{b}) = \mathtt{abcbac},$$
$$g(\mathtt{c}) = \mathtt{abcacbc}.$$

are both square-free.

Indeed, these two morphisms are the simplest and only square-free morphisms whose codewords, images of letters, have lengths smaller than eight. Carpi [17] has shown that square-free morphisms over three letters must have size at least 18, where size of a morphism is the sum of the lengths of the images of its letters.

Later, Pleasants [54] established an identical map to the above morphism h and Leech in [46] introduced the 13-uniform morphism L from Σ_3^* to itself defined by:

$$L(\mathtt{a}) = \mathtt{abcbacbcabcba},$$
$$L(\mathtt{b}) = \mathtt{bcacbacabcacb},$$
$$L(\mathtt{c}) = \mathtt{cabacbabcabac}.$$

Their results are both independent of Thue's work.

Next, Brandenburg [15] gave an example of 11-uniform square-free ternary morphism and stated that there are no smaller uniform square-free ternary morphisms. Currie and Rampersad [30] stated this open problem: do there exist k-uniform square-free ternary morphisms for all $k \geq 11$?

The number of square-free words of length n on a 3-letter alphabet seems to grow as a polynomial, but the result due to Brandenburg shows that this is in fact an exponential growth.

Theorem 3 ([15, 13]). *Let $c(n)$ be the number of square-free words of length n on a three letter alphabet. Then*

$$6 * c_1^n \leq c(n) \leq 6 * c_2^n$$

where $c_1 = 1.032$ and $c_2 = 1.38$.

3 Morphisms on Larger alphabets

One of the basic questions is whether a given morphism from Σ_m^* to Σ_n^* is square-free. Several people have investigated the problem and derived conditions for morphism square-free-ness, the simplest and most precise one is due to Crochemore [24], who introduced the notion of a pre-square with respect to a morphism h:

Let w be a square-free word in Σ_m^* and u a factor of $h(w)$; an occurrence of u in $h(w)$ is given by words α, β in Σ_m^* such that $h(w) = \alpha u \beta$; that occurrence of u is called a *pre-square* if $u \neq \epsilon$ and if there exists a word w in Σ_m^* satisfying: ww is square-free and u is a prefix of $\beta h(w)$ or ww is square-free and u is a suffix of $h(w)\alpha$. In that case we also say that $h(w)$ contains a pre-square, and that w' duplicates the pre-square u of $h(w)$.

Theorem 4 ([24]). *Let h be a morphism from Σ_m^* into Σ_n^* with Σ_m having at least three letters. Then h is square-free iff the following conditions hold:*

- *$h(x)$ is square-free for square-free words $x \in \Sigma_m$ of length 3;*
- *No $h(a)$, for $a \in \Sigma_m$, contains an internal pre-square (occurs within the word).*

Crochemore introduces a boundary on the length of words that must be square-free in order to have a square-free morphism. Let us define:

$$m(h) = min\{|h(a)| : a \in \Sigma_m\},$$

$$M(h) = max\{|h(a)| : a \in \Sigma_m\}.$$

Theorem 5 ([24]). *Let h be a morphism from Σ_m^* into Σ_n^* then h is square-free iff $h(x)$ is square-free for all square-free words x of length $k = max\{3, \lceil (M(h)-3)/m(h)+1 \rceil\}$.*

If the morphism is uniform this bound is 3, the next Corollary is then a consequence of the previous theorem.

Corollary 2 ([24]). *Let h be a morphism from Σ_3^* into Σ_n^* Then h is square-free iff $h(x)$ is square-free for all words x of length 5.*

Bean et al. [11] generalise the existence of square-free morphisms from Σ_n^* to Σ_3^* and of cube-free morphisms from Σ_n^* to B^*. A different characterisation of square-free morphisms is also given in the same paper.

Theorem 6 ([11]). *Let h be a morphism from Σ_m^* into Σ_n^*. If $h(w)$ is square-free whenever $w \in \Sigma_m$ is square-free and of length no greater than three, and if $a = b$ whenever $a, b \in \Sigma_m$ with $h(a)$ a factor of $h(b)$, then $h(u)$ is square-free whenever u is a square-free word on Σ_m.*

Furthermore, they characterise morphisms which are kth power-free.

Theorem 7 ([11]). *Let h be a morphism from Σ_m^* into Σ_n^*, where Σ_m and Σ_n are alphabets. Let $k > 2$. If*

- *$h(w)$ is kth power-free whenever w is a kth power-free word on Σ_m with length no greater than $k + 1$,*
- *$a = b$ whenever $h(a)$ is a factor of $h(b)$, and*
- *if $a, b, c \in \Sigma_m$ and $uh(a)v{=}h(b)h(c)$ where u and v may be empty then either u is empty and $a = b$ or else v is empty and $a = c$,*

then h is kth power-free.

In addition, they showed that not all square-free morphisms are cube-free, then by the next theorem they gave sufficient conditions for a square-free morphism to be kth power-free morphism.

Theorem 8 ([11]). *Let h be a morphism from Σ_m^* into Σ_n^*. If*

- *h is square-free,*
- *$a = b$ whenever $h(a)$ is a factor of $h(b)$, and*
- *no proper prefix of $h(a)$ is a suffix of $h(a)$ for all $a \in \Sigma_m$,*

then h is kth power-free for all $k > 1$.

4 Methodology

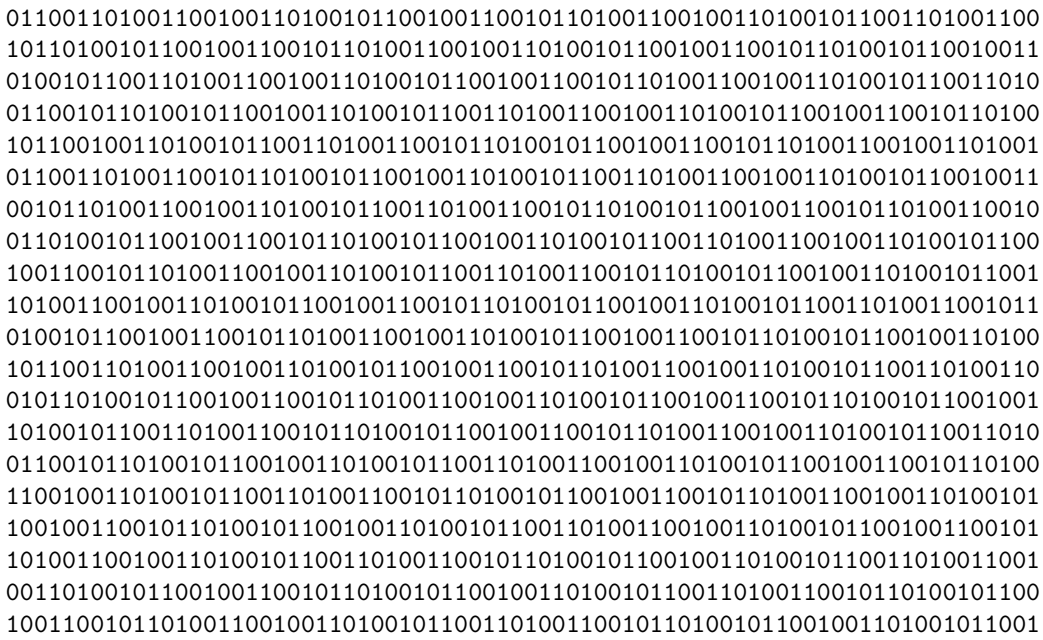
Throughout this document, in order to prove the existence of an infinite word complying with some properties, several methods are used. The main technique is to design an HD0L system, which means finding two morphisms generating an appropriate infinite word. One of the experimental techniques that we used consists of the following steps.

- First, we generate a long enough word satisfying the pre-defined constraints using a backtracking strategy.
- Second, we search for its most repetitive motifs.
- Third, using selective elements of the set of motifs, we try to decode the word to find its pre-image according to the morphism defined by the motifs.
- Fourth, we iterate the previous two steps with the new word (pre-image of the first word).

Backtracking is a general algorithm for finding all (or some) solutions to some computational problem. It incrementally builds candidates to the solutions, and abandons each partial candidate as soon as it determines it cannot possibly be completed to a valid solution [42].

For example, to prove there exists an infinite $7/3^+$-free binary word containing one 7/3-power and at most 14 squares, applying a backtracking technique we get the following 2000-bit long binary word satisfying the desired properties:

```
11010011001001101001011001001100101101001011001001101001011001101001100 1
00110100101100100110010110100110010011010010110011010011001011010010110010
01101001011001101001100100110100101100100110010110100101100100110100101100
11010011001011010010110010011001011010011001001101001011001001100101101001
```



One of the most frequent factors of this word is 101001:

Using the above factor we can see that the whole word can be decoded with the following factors 101001100101, 1010011001001, 101001011001, 101001011001001, and 101001011001001100101.

When renamed as a, b, ..., e we get the translation morphism h defined by:

$$h(\mathtt{a}) = \mathtt{101001100101}$$
$$h(\mathtt{b}) = \mathtt{1010011001001}$$
$$h(\mathtt{c}) = \mathtt{101001011001}$$
$$h(\mathtt{d}) = \mathtt{101001011001001}$$
$$h(\mathtt{e}) = \mathtt{10100101100100110010 1}$$

The pre-image of the above word by the morphism h is:

bedcbebcadcbedcaebecbebc
aebedcbebcadcbedcaebcadc
bebcaebedcbebcadcbedcaeb
edcbebcaebedcaebcadcbebc
aebedcbebcadcbedcaebcadc.

Iterating the same procedure on a long enough part of the last word, we find the following factors: adcbebc, adcbedc, aebc, aebedc, aebedcbebc. A brute-force try of the 5! permutations of possible mappings reveals the following morphism (see Chapter 5):

$$f_4(\mathtt{a}) = \mathtt{adcbebc},$$
$$f_4(\mathtt{b}) = \mathtt{adcbedc},$$
$$f_4(\mathtt{c}) = \mathtt{aebc},$$
$$f_4(\mathtt{d}) = \mathtt{aebedc},$$
$$f_4(\mathtt{e}) = \mathtt{aebedcbebc}.$$

This is the morphism that is iterated to get a fixed point. The two morphisms found by the method constitute the essential part of the HD0L system proving the existence of an infinite word satisfying the initial conditions.

Several other methods are exploited in this document, in order to make conjectures and also to discover morphisms. While some are based on clever computations and search techniques, others require deep analysis of the words and careful decomposition of longer words into their smaller factors.

In the last chapter, we exploit a type of data structure called suffix automaton and a type of string factorisation called L-Z factorisation. The definitions are given in Chapter 7 where some other well known pattern matching techniques are also employed.

3

Infinite binary words containing few squares

The number of repetitions in infinite words is a classic problem in combinatorics on words which, for the last century, has been studied in depth. Let $g(n)$ be the length of a longest binary word containing at most n squares. Then $g(0) = 3$ (e.g. `010`), $g(1) = 7$ (e.g. `0001000`) and $g(2) = 18$ (e.g. `010011000111001101`).

The question of behaviour of this function was posed by Erdös [35]. Entringer, Jackson, and Schatz [34] showed in 1974 that there exists an infinite word with 5 different squares, $g(5) = \infty$. Later Fraenkel and Simpson [36] showed that there exists an infinite binary word that has only three squares 00, 11, and 0101, and thus $g(3) = \infty$. A somewhat simplified proof of this result was given by Rampersad, Shallit and Wang [55], using two uniform morphisms. Later, in 2006, Harju and Nowotka [38] gave a simpler proof of this result. We give a new proof that the maximal length is infinite if 3 squares are allowed to appear in a binary word. Our proof is simpler than the original proof in [36] and uses a morphism simpler than the proofs of [55] and [38].

Instead of avoiding just squares, an interesting variation on this problem is to avoid large repetitions. Entringer, Jackson, and Schatz [34] showed that there exist infinite binary words avoiding squares of period at least three. Furthermore, avoiding large squares has been studied by Dekking [33], Rampersad et al. [55], Shallit [64], Ochem[50], and many others.

In this chapter we provide some new results as an outcome of studying the pattern avoidance from a different point of view. We analyse the possibility of avoiding repetitions of even and odd periods, and further impose a constraint on their maximal exponent.

We show that there exists no infinite 3^+-free binary word avoiding all squares of odd period. This study reveals there exists no infinite binary word, simultaneously avoiding cubes and squares of even period. Moreover, we prove that there exists an

infinite 3^+-free binary word avoiding squares of even-period length.

The trade-off as there is between the maximal period length and number of repetitions contained will follow a similar trade-off between number of cubes and number of distinct squares. We succeed in reducing the number of repetitions contained in infinite binary words without compromising the constraint on parity of their period. We conclude that in such words the minimal number of squares is 7 when only 1 cube occurs. The number reduces to 4 when 2 cubes are allowed in the word.

1 The infinite binary word by Fraenkel and Simpson

Here we recall the morphism f from Chapter 2: The morphism f is defined from Σ_3^* to itself by:

$$\begin{aligned} f(\mathtt{a}) &= \mathtt{abc},\\ f(\mathtt{b}) &= \mathtt{ac},\\ f(\mathtt{c}) &= \mathtt{b}. \end{aligned}$$

Since the letter a is a prefix of $f(\mathtt{a})$, the infinite word $\mathbf{f} = f^\infty(\mathtt{a})$ is well defined. It is well known [37] that $\mathbf{f}$ is square-free and avoids aba and cbc.

To construct an infinite binary word with the desired property, Fraenkel and Simpson first transform $\mathbf{f}$ into $\mathbf{q}$ by replacing every occurrence of cb in $\mathbf{f}$ by cdb, and every occurrence of bc by bec. Then they translate the infinite word $\mathbf{q} \in \Sigma_5^\infty$ to an infinite word with the following morphism:

$$\begin{aligned} h_{\mathrm{fs}}(\mathtt{a}) &= \mathtt{011000111001},\\ h_{\mathrm{fs}}(\mathtt{b}) &= \mathtt{011100011001},\\ h_{\mathrm{fs}}(\mathtt{c}) &= \mathtt{011001110001},\\ h_{\mathrm{fs}}(\mathtt{d}) &= \mathtt{01100010111001},\\ h_{\mathrm{fs}}(\mathtt{e}) &= \mathtt{01110010110001}. \end{aligned}$$

Lemma 1 ([36])**.** *The infinite word* $\mathbf{h}_{\mathrm{fs}} = h_{\mathrm{fs}}(\mathbf{q})$ *contains the* 3 *squares* 00*,* 11*, and* 0101 *only.*

The following theorem is a direct consequence of Lemma 1.

Theorem 9. *There exists an infinite binary word containing only three squares.*

2 The infinite binary word by Rampersad et al.

In 2005, Rampersad et al. [55] constructed an infinite binary word avoiding all squares except 00, 11, and 0101. They use two uniform morphisms, one on 5 letters to be

iterated in order to generate a fixed point and the second morphism to translate the 5-ary fixed point of the first morphism to a binary word. The 24-uniform morphism from Σ_5^* to itself is defined by f_{rsw}:

$$\begin{aligned}
f_{\mathrm{rsw}}(0) &= 012321012340121012321234,\\
f_{\mathrm{rsw}}(1) &= 012101234323401234321234,\\
f_{\mathrm{rsw}}(2) &= 012101232123401232101234,\\
f_{\mathrm{rsw}}(3) &= 012321234323401232101234,\\
f_{\mathrm{rsw}}(4) &= 012321234012101234321234.
\end{aligned}$$

In the article they first prove that if $w \in \Sigma_5^*$ is square-free and avoids the factors 02, 03, 04, 14, 20, 30, 41, then $f_{\mathrm{rsw}}(w)$ is square-free and avoids the factors 02, 03, 04, 13, 14, 20, 24, 30, 31, 41, 42, 010, 434.

Next they consider the following 6-uniform morphism, h_{rsw} from Σ_5^* to B^*:

$$\begin{aligned}
h_{\mathrm{rsw}}(0) &= 011100,\\
h_{\mathrm{rsw}}(1) &= 101100,\\
h_{\mathrm{rsw}}(2) &= 111000,\\
h_{\mathrm{rsw}}(3) &= 110010,\\
h_{\mathrm{rsw}}(4) &= 110001.
\end{aligned}$$

Lemma 2 ([55]). *If w is square-free and avoids the factors 02, 03, 04, 13, 14, 20, 24, 30, 31, 41, 42, 434, and 010 then the only squares in $h_{\mathrm{rsw}}(w)$ are 00, 11, 0101.*

The corollary is Theorem 9.

3 The infinite binary word by Harju and Nowotka

Later, in 2006, Harju and Nowotka [38] produced yet another proof for the result. Unlike the previous proofs, the pure morphic word $w \in \Sigma_3^*$ needs to be square-free and no other pattern need to be avoided.

They consider the morphism h_{hn} from Σ_3^* to B^* defined by

$$\begin{aligned}
h_{\mathrm{hn}}(\mathtt{a}) &= 101100011100101100111000,\\
h_{\mathrm{hn}}(\mathtt{b}) &= 101110010110001110010111000,\\
h_{\mathrm{hn}}(\mathtt{c}) &= 110010110001011100101100111000.
\end{aligned}$$

Lemma 3 ([38]). *For any square-free word w, $w \in \Sigma_3^\infty$, the infinite word $\mathbf{h}_{\mathrm{hn}} = h_{\mathrm{hn}}(w)$ contains the 3 squares 00, 11 and 0101 only. The cubes 000, and 111 are the only factors of exponent larger than 2 occurring in $\mathbf{h}_{\mathrm{hn}}$.*

Here we mention a few properties of this morphism:

1. the codewords $h_{\mathrm{hn}}(\mathtt{a}), h_{\mathrm{hn}}(\mathtt{b})$, and $h_{\mathrm{hn}}(\mathtt{c})$ are border-free or, equivalently, their smallest periods are their lengths;
2. they are pairwise overlap-free;
3. the longest common prefix (suffix, resp.) of two codewords is 1011 of length 4 (011100101100111000 of length 18, resp.);
4. the images by h_{hn} of all square-free words of length 3 contain only the three squares 00, 11, and 0101, and the two cubes 000 and 111.

Fact 1. *For letters $a_0, a_1, a_2 \in \Sigma_3$, $a_1 \neq a_2$, if $h_{\mathrm{hn}}(a_0)$ is a factor of $h_{\mathrm{hn}}(a_1a_2)$ then either $a_0 = a_1$ and the only occurrence of $h_{\mathrm{hn}}(a_0)$ in $h_{\mathrm{hn}}(a_0a_2)$ is as a prefix, or $a_0 = a_2$ and the only occurrence of $h_{\mathrm{hn}}(a_0)$ in $h_{\mathrm{hn}}(a_1a_0)$ is as a suffix.*

Fact 1 is a direct consequence of properties of morphisms h_{hn}, which leads to Lemma 3 and Theorem 9.

In the next section we present another morphism that can construct an infinite binary word containing only 3 squares. This morphism is the simplest of its form, that can generate an infinite binary word with at most 3 squares. Simplest in this sense that sum of all its codewords lengths is 24; shortest amongst the existing morphisms satisfying the property.

4 A simple morphism

This section is dedicated to the new proof of Theorem 9. The proof relies on two morphisms, f also used by Fraenkel and Simpson (see Section 1), and g_1 from Σ_3^* to B^* defined by

$$\begin{aligned} g_1(\mathtt{a}) &= \mathtt{01001110001101}, \\ g_1(\mathtt{b}) &= \mathtt{0011}, \\ g_1(\mathtt{c}) &= \mathtt{000111}. \end{aligned}$$

Lemma 4. *The infinite word $\mathbf{g_1} = g_1(\mathbf{f})$ contains the 3 squares 00, 11, and 1010 only. The cubes 000 and 111 are the only factors of exponent larger than 2 occurring in $\mathbf{g_1}$.*

The codewords of the morphism g_1 form a prefix code, which implies that the morphism itself is an injective function. Therefore the word $\mathbf{g_1}$ can be parsed uniquely

Table 3.1: The gaps between consecutive occurrences of $\mathbf{z} = \mathtt{000}$ are 1101, 11010011, 1110100111, and 11100110100111.

$g_1(\mathtt{ac})$	=	01001110001101 000111	4
$g_1(\mathtt{abc})$	=	01001110001101 0011 000111	8
$g_1(\mathtt{ca})$	=	000111 01001110001101	10
$g_1(\mathtt{cba})$	=	000111 0011 01001110001101	14

to recover the square-free word $\mathbf{f}$. Proof of Lemma 4 relies on parsing $\mathbf{g_1}$ by locating the occurrences of the triplet $\mathbf{z} = \mathtt{000}$ in $\mathbf{g_1}$. $\mathbf{z}$ occurs only within $g_1(\mathtt{a})$ or $g_1(\mathtt{c})$. Indeed, any occurrence of $\mathbf{z}$ preceded by 111 determines an occurrence of a in $\mathbf{f}$, and otherwise it is followed by 111 and determines an occurrence of c.

Gaps between occurrences of specific factors. For the purpose of the proof we define the gap function *gap* related to $\mathbf{g_1}$ as follows. For any factors u and v of $\mathbf{g_1}$:

$$gap(u, v) = \{|w| \mid uwv \text{ factor of } \mathbf{g_1} \text{ and only one occurrence of } v \text{ in } wv\}.$$

Although *gap* is only used in the proof in a very restricted way, note that the gap between any two factors of $\mathbf{g_1}$ is well defined. Table 3.1 shows all the incidents of two consecutive occurrences of $\mathbf{z}$s in $\mathbf{g_1}$ therefore $gap(\mathbf{z}, \mathbf{z}) = \{4, 8, 10, 14\}$.

Here, we present proof of Lemma 4:

Proof. We assume that w^2 occurs in $\mathbf{g_1}$ for some non-empty word w and distinguish three cases: where w^2 contains at most one occurrence of $\mathbf{z} = \mathtt{000}$, an even number of occurrences of it, or an odd number.

The square w^2 contains at most one occurrence of z. Under this hypothesis, the only factors of $\mathbf{g_1}$ that we have to consider are images by the morphism g_1 of square-free words in which sum of occurrences of a and c is at most three times, therefore images of these square-free words contain all factors containing at most one occurrenceof $\mathbf{z}$ (because $g_1(\mathtt{a})$ and $g_1(\mathtt{c})$ contain each an occurrence of $\mathbf{z}$). Table 3.2 shows these words, which contain only the three expected squares.

The square w^2 contains an even number of occurrences of z. In this situation and after the first case, w^2 contains $2k$ occurrences of $\mathbf{z}$ for an integer $k > 0$. These occurrences are evenly distributed between the first half and the second half of w^2; w contains exactly k occurrences of $\mathbf{z}$.

Table 3.2: Squares in short factors of $\mathbf{g_1}$.

$g_1(\mathtt{aca})$	=	01001110001101 000111 01001110001101
$g_1(\mathtt{acba})$	=	01001110001101 000111 0011 01001110001101
$g_1(\mathtt{abca})$	=	01001110001101 0011 000111 01001110001101
$g_1(\mathtt{abcba})$	=	01001110001101 0011 000111 0011 01001110001101
$g_1(\mathtt{cac})$	=	000111 01001110001101 000111
$g_1(\mathtt{cabc})$	=	000111 01001110001101 0011 000111
$g_1(\mathtt{cbac})$	=	000111 0011 01001110001101 000111
$g_1(\mathtt{cbabc})$	=	000111 0011 01001110001101 0011 000111

The word w can be decomposed as $u_0z_1u_1 \dots z_ku_k$ where $z_1 = \cdots = z_k = \mathbf{z}$ and $u_0, \dots, u_k \in \{\mathtt{0},\mathtt{1}\}^*$. In addition, z_k and z_1 are consecutive occurrences of $\mathbf{z}$ in w^2. Here, we examine these occurrences according to $gap(z_k, z_1)$, in other words, possibilities of $|u_ku_0|$ (see Table 3.1). It can be noted that the z_k and z_1 occur in w^2 in different contexts (if one indicates the occurrence of $g_1(\mathtt{a})$ the other indicates the occurrence of $g_1(\mathtt{c})$) and certainly not in the same context.

Gap of length 4**.** Here, $u_ku_0 = \mathtt{1101}$.

Using the Table 3.1 and knowing the fact that$gap(\mathbf{z}, \mathbf{z})$ is injective we deduce that $z_ku_ku_0z_1$ is a factor of $g_1(\mathtt{ac})$. Therefore, z_k indicates an occurrence of $g_1(\mathtt{a})$ and z_1 indicates an occurrence of $g_1(\mathtt{c})$. This implies that w^2 is a factor of $g_1(\alpha\mathtt{c}v\mathtt{ac}v\mathtt{a})$ for some word v in $\mathbf{f}$ and $\alpha \in \{\mathtt{a},\mathtt{b},\mathtt{c}\}$, which is impossible because $\mathbf{f}$ is square-free.

Gap of length 8**.** Here, $u_ku_0 = \mathtt{11010011}$. Therefore, $z_ku_ku_0z_1$ is a factor of $g_1(\mathtt{abc})$. Either $|u_k| > 6$ or $|u_0| > 1$, if the former is true then w^2 is a factor of $g_1(\alpha\mathtt{c}v\mathtt{abc}v\mathtt{ab})$, which indicates occurrenceof a square $\mathtt{c}v\mathtt{abc}v\mathtt{ab}$ in $\mathbf{f}$, contradiction. In the second case; ($|u_0| > 1$), w^2 is a factor of $g_1(\alpha\mathtt{bc}v\mathtt{abc}v\mathtt{a}\beta)$ for some word v in $\mathbf{f}$ and $\alpha, \beta \in \{\mathtt{a},\mathtt{b},\mathtt{c}\}$, which is impossible because $\mathbf{f}$ is square-free.

Gap of length 10**.** Here, $u_ku_0 = \mathtt{1110100111}$. Therefore, $z_ku_ku_0z_1$ is a factor of $g_1(\mathtt{ca})$. So, z_k indicates an occurrence of $g_1(\mathtt{c})$ and z_1 indicates an occurrence of $g_1(\mathtt{a})$. This implies that w^2 is a factor of $g_1(\alpha\mathtt{a}v\mathtt{ca}v\mathtt{c}\beta)$ for some word v in $\mathbf{f}$ and $\alpha, \beta \in \{\mathtt{a},\mathtt{b},\mathtt{c}\}$, which is impossible because $\mathbf{f}$ is square-free.

Gap of length 14**.** Here, $u_ku_0 = \mathtt{11100110100111}$. Therefore, $z_ku_ku_0z_1$ is a factor of $g_1(\mathtt{cba})$. The arguments of the previous cases apply similarly either $|u_k| > 4$ or $|u_0| > 9$ and derive to the same contradiction.

Therefore, w^2 cannot contain an even number of occurrences of $\mathbf{z}$.

The square w^2 contains an odd number of occurrences of z. Here, w^2 contains an odd number k, $k > 2$, of occurrences of $\mathbf{z}$, in which case w is of the form $\mathtt{0}y\mathtt{00}$ or $\mathtt{00}y\mathtt{0}$. From the gap analysis we also deduce that $|w| \geq 8$, which implies that the central occurrence of $\mathbf{z}$, overlapping the junction between the two occurrences of w, is followed and preceded in w^2 by at least 7 letters. We treat the form $\mathtt{0}y\mathtt{00}$, the other form can be dealt with symmetrically.

The central occurrence of $\mathbf{z}$ is either followed by $\mathtt{111}$ or preceded by it. In the first case $\mathtt{111}$ is not followed by $\mathtt{000}$ and so identifies $g_1(\mathtt{c})$. This implies that $\mathtt{00}$ precedes w^2 in $\mathbf{g_1}$ and that $\mathtt{00}ww(\mathtt{00})^{-1}$ occurs in $\mathbf{g_1}$. In the second case $\mathtt{111}$ and its preceding letters identify $g_1(\mathtt{a})$, which implies that $\mathtt{0}$ follows w^2 in $\mathbf{g_1}$ and that $\mathtt{0}^{-1}ww\mathtt{0}$ occurs in $\mathbf{g_1}$. In both cases, the central occurrence of $\mathbf{z}$ disappears in the resulting conjugate of w^2 and we are taken back to the case where $\mathbf{z}$ occurs an even number of times. Therefore, w^2 cannot contain an odd number > 1 of occurrences of $\mathbf{z}$.

From the above we deduce that the only squares in $\mathbf{g_1}$ are those occurring in the words of Table 3.2. They are $\mathtt{00}$, $\mathtt{11}$, and $\mathtt{1010}$. Factors having a rational exponent larger than 2 are prefixed by a square. Thus, the only factors of this type occurring in $\mathbf{g_1}$ are $\mathtt{000}$ and $\mathtt{111}$ of exponent 3. This concludes the proof of Lemma 4. □

We do not know if the morphism g_1 is the simplest possible to satisfy the desired property, but note that the morphism h' defined by

$$\begin{aligned} h'(\mathtt{a}) &= \mathtt{01},\\ h'(\mathtt{b}) &= \mathtt{0011},\\ h'(\mathtt{c}) &= \mathtt{000111}. \end{aligned}$$

does not work because $h'(\mathtt{bcacb}) = \mathtt{00110001110100011100 11}$ contains four squares: $\mathtt{0}^2$, $\mathtt{1}^2$, $\mathtt{10}^2$, and $(\mathtt{10001110})^2$.

Cubes are unavoidable. One can ask whether an infinite word can contain fewer factors of exponent at least 2. We show that an infinite binary word that contains only three squares cannot avoid cubes. This is done by exploring the trie of binary words satisfying the condition, which appears to be finite. Without loss of generality we consider words starting with $\mathtt{001}$. Branches of the trie end when adding a letter to their label produces an occurrence of a cube or of a fourth square. The trie is displayed in Figure 3.1 and answers the question. The trie also shows that the maximal length of binary words containing no cube and 2 squares is 12.

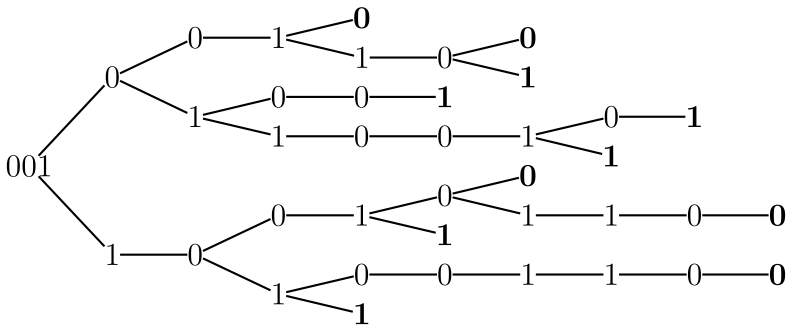


Figure 3.1: Trie of binary words starting with `001` and containing at most 3 squares and no cubes. Their maximal length is 12.

5 Words containing only squares of odd-length period

Here, we study further the infinite binary words and the squares they contain. Looking at the parity of the periods of the squares reveals interesting properties. A simple computer check similar to the method illustrated in Figure 3.1 verifies the following facts:

Fact 2. *There exist no infinite* 3^+*-free binary word avoiding all squares of odd period.*

Note that the only infinite binary words omitting 00 and 11 are $(01)^\infty$ or $(10)^\infty$ both contain 3^+-powers.

Fact 3. *There exist no infinite binary word, simultaneously avoiding cubes and square* xx *with* $|x| = 2k$ *for* $k > 0$*. The maximal length of a cube-free binary word containing only squares of odd period is 21.*

The remaining of this chapter is dedicated to demonstrate if the constraint on the maximal exponent is relaxed, so that the word may contain cubes, then avoiding squares of even-period becomes possible.

The following technique is used to proof all the theorems in the remaining of this chapter. To demonstrate how this technique works, a step by step proof is given for Proposition 1 as an example to make it easier for the reader.

Proof Technique. Suppose we are given a *synchronising* morphism $g : \Sigma_3^* \to B^*$, if s is an infinite square-free word on Σ_3^* then the only squares occurring in $g(s)$ also occur in the images of square-free words of length 3. In this section s is any infinite square-free ternary word.

Theorem 10. *There exists an infinite* 3^+*-free binary word avoiding squares of even-period length.*

The proof relies on the following *synchronising* 8-morphism g_2 from Σ_3^* to B^* defined by

$$g_2(\mathtt{a}) = \mathtt{11011001},$$
$$g_2(\mathtt{b}) = \mathtt{11001001},$$
$$g_2(\mathtt{c}) = \mathtt{00011000}.$$

Proposition 1. *The infinite word* $\mathbf{g_2} = g_2(s)$ *contains no repetition with exponent greater than* 3 *and no square* uu *with* $|u| = 2k$ *for* $k > 0$ *for any square-free* s*. Furthermore,* $\mathbf{g_2}$ *contains 12 squares only:* $Sq = \{\mathtt{00}, \mathtt{11}, (\mathtt{001})^2, (\mathtt{010})^2, (\mathtt{011})^2, (\mathtt{100})^2, (\mathtt{110})^2, (\mathtt{00011})^2, (\mathtt{00110})^2, (\mathtt{01100})^2, (\mathtt{10001})^2, (\mathtt{11001})^2\}$*. And 3 cubes:* $\mathtt{000}$, $\mathtt{111}$ *and* $(\mathtt{100})^3$*.*

Here, we demonstrate how the proof technique works for Proposition 1, the following step by step proofs in this chapter are omitted.

Proof. Let assume that $g_2(s)$ contains a square $uu \notin Sq$, then either $|u| > 16$ or uu is a factor of image of $w \in s$ for $w \leq 3$.

- Case $|u| > 16$:

  $$uu = \overbrace{u_1 \underbrace{\cdots} v_1}\overbrace{u_1 \underbrace{\cdots} v_1}$$

  where v_1u_1 is codeword. Then v_1 is not longer than the longest common prefix between two different codewords (otherwise it refers to the same letter in s and shows an existance of a square in s), that is, $|v_1| \leq 3$. Symmetrically, u_1 is not longer than the longest common suffix of two different codewords, that is, $|u_1| \leq 4$. But then $|v_1u_1| \leq 7$ and cannot be a complete codeword, a contradiction.

- Case $|u| \leq 16$: Here, it is enough to look at images of all $w \in s$ for $w \leq 3$ and a simple computer check verifies the fact that all squares contained in these images are in Sq. Since a cube is an extension of a square, one can easily count the number of cubes to be 3.

□

The proof of Theorem 10 is a direct consequence of the above proposition, since the set Sq contains only squares of odd period (1, 3 and 5).

6 Avoiding long repetitions

Looking at the length of periods of squares contained in $\mathbf{g_2}$ (Section 5), one may ask if it is possible to reduce the length of the longest squares in an infinite word without compromising the other conditions imposed on the word. It is trivial to see that there exist no infinite binary words containing only squares of period 1. However the next theorem shows that we can reduce the longest period to 3.

Theorem 11. *There exist an infinite 3^+-free binary word containing only squares of period 1 or 3.*

The proof relies on the following *synchronising* 11-morphism g_3 from Σ_3^* to B^* defined by

$$g_3(\mathtt{a}) = \mathtt{11001001101},$$
$$g_3(\mathtt{b}) = \mathtt{11001001110},$$
$$g_3(\mathtt{c}) = \mathtt{11001001000}.$$

Proposition 2. *The infinite word $\mathbf{g_3} = g_3(s)$ contains no repetitions with exponent greater than 3 and no squares uu with $|u| = 2k$ for $k > 0$. Furthermore, $\mathbf{g_3}$ contains only 7 squares: 00, 11, $(001)^2$, $(010)^2$, $(011)^2$, $(100)^2$ and $(110)^2$, and 3 cubes: 000, 111 and $(100)^3$.*

The proof is very similar to the proof of Proposition 1, therefore omitted and Theorem 11 follows.

The following *synchronising* morphism g_4 also generates an infinite 3^+-free binary words containing squares of period 1 and 3 only. Therefore g_4 could also be exploited to prove Theorem 11. Furthermore the infinite binary word generated by g_4 omits the third cube in $\mathbf{g_3}$. Thus it contains less cubes.

The word with 7 squares and 2 cubes. To generate an infinite word with these properties we use the following *synchronising* 12-morphism g_4 from Σ_3^* to $\mathtt{B}^*$ defined by

$$g_4(\mathtt{a}) = \mathtt{110110001110},$$
$$g_4(\mathtt{b}) = \mathtt{110111000100},$$
$$g_4(\mathtt{c}) = \mathtt{110111001000}.$$

Proposition 3. *The infinite word $\mathbf{g_4} = g_4(s)$ contains no repetitions with exponent greater than 3 and no squares uu with $|u| = 2k$ for $k > 0$. Furthermore, $\mathbf{g_4}$ contains only 7 squares: 00, 11, $(001)^2$, $(011)^2$, $(100)^2$, $(101)^2$ and $(110)^2$, and 2 cubes: 000 and 111.*

The proof is very similar to the proof of Proposition 1 and as explained earlier the step by step proof is omitted.

7 Reducing the number of repetitions

It is natural to ask if there exists an infinite binary word avoiding squares of even-length period and containing less that 7 squares or 2 cubes.

Fact 4. *A binary word avoiding squares of even-length period that contains at most 6 squares and only one cube has length at most 57.*

Although this fact shows that simultaneously reducing the number of squares and cubes is not possible, the following two theorems show that there exist infinite binary words avoiding squares of even-length periods either containing only 1 cube and 7 squares, or 2 cubes and less than 7 squares.

Theorem 12. *There exists an infinite* 3^+*-free binary word with at most one cube, avoiding squares of even-length period and containing 7 squares only.*

The proof relies on the following *synchronising* 73-morphism g_5 from Σ_3^* to $\mathtt{B}^*$ defined by

$$\begin{aligned} g_5(\mathtt{a}) = {} & \mathtt{1101100010011011001000110110001001000011} \\ & \mathtt{011001001100010010001100100110110010000}, \\ g_5(\mathtt{b}) = {} & \mathtt{1101100010011011001001100010010001100 10} \\ & \mathtt{0110110001001000110110010011000100 1000}, \\ g_5(\mathtt{c}) = {} & \mathtt{110110001001101100100110001001000110110} \\ & \mathtt{0100110110001001000110010011000100 1000}. \end{aligned}$$

Proposition 4. *The infinite word* $\mathbf{g_5} = g_5(s)$ *contains no repetitions with exponent greater than* 3 *and no squares* uu *with* $|u| = 2k$, *for* $k > 0$.
Furthermore, $\mathbf{g_5}$ *contains only 7 squares:* `00`, `11`, $(\mathtt{001})^2$, $(\mathtt{010})^2$, $(\mathtt{011})^2$, $(\mathtt{100})^2$, *and* $(\mathtt{110})^2$, *and only one cube* `000`.

Theorem 13. *There exists an infinite* 3^+*-free binary word with at most two cubes, avoiding squares of even-length period and containing only 4 squares.*

The proof relies on the following *synchronising* 39-morphism g_6 from Σ_3^* to $\mathtt{B}^*$ defined by

$$\begin{aligned} g_6(\mathtt{a}) &= \mathtt{111000100110001110010001100100111001000}, \\ g_6(\mathtt{b}) &= \mathtt{111000100111001000110010011100011001000}, \\ g_6(\mathtt{c}) &= \mathtt{111000100111001001100010011100011001000}. \end{aligned}$$

Proposition 5. *The infinite word* $\mathbf{g_6} = g_6(s)$ *contains no repetitions with exponent greater than* 3 *and no squares* uu *with* $|u| = 2k$ *for* $k > 0$.
Furthermore, $\mathbf{g_6}$ *contains only 4 squares:* `00`, `11`, $(\mathtt{001})^2$, *and* $(\mathtt{100})^2$, *and only two cubes:* `000` *and* `111`.

The following result is verified by computer check:

Fact 5. *A binary word avoiding squares of even-length period that contains at most 3 squares has length at most 29.*

Here, it is worth mentioning if we remove the constraint on the parity of the squares period then in Chapter 5 we show:

- There exists a 3^+-free infinite binary word with only one cube that contains no more than 4 squares 24.
- There exists a 3-free infinite binary word with at most 8 squares 21.

We also in Chapter 5 show that these numbers are minimal.

8 Conclusion

In this chapter we presented a new proof of Fraenkel and Simpson's result [36] and further studied the infinite binary words whose squares have odd-length periods. The tables below summarises these results.

	Longest allowed period	Number of cubes	Number of squares	Length of the morphism
Proposition 1	5	3	12	8
Proposition 2	3	3	7	11
Proposition 3	3	2	7	12
Proposition 4	3	1	7	73
Proposition 5	3	2	4	39

Note that all the infinite binary words considered in these proofs are 3^+-free and avoid squares of even-length period. So all the propositions mentioned above imply Theorem 10. The morphisms in Propositions 2, 3 and 4 generate binary words containing 7 squares of period 1 or 3. The only differences between them are the number of cubes they contain and the morphisms length. The longer morphism contains fewer cubes.

Similar comparison was made between two morphisms used in Propositions 3 and 5. Both generate binary words with only 2 cubes, the longer one (length 39) contains fewer squares than the shorter one (length 12).

	Allowed number of cubes	Minimum number of squares
Theorem 13	2	4
Theorem 12	1	7

4

Finite-repetition threshold

The constraints on the exponents of repetitions in infinite words have been raised to optimality after Dejean's conjecture [32] on the repetitive threshold associated with the alphabet size. *The repetitive threshold* (Dejean's repetitive threshold) of order k is the infimum of maximum exponents of all (infinite) words over a k-letter alphabet.

The first case says that the repetitive threshold of the binary alphabet is 2 (infinite binary words can avoid factor of exponent larger than 2 but cannot avoid squares) and the second case states that it is 7/4 for the three-letter alphabet [32], Dejean conjectured that $r_4 = 7/5$ and $r_k = \frac{k}{k-1}$ for $k \geq 5$. The last cases have been proved recently by Rao [56] after the works of Carpi [19], Pansiot [53], Moulin-Ollagnier [49], Mohammad-Noori and Currie [48], Currie and Rampersad [29]. All these results contribute to the proof of Dejean's conjecture.

The idea of repetitive threshold was extended into the generalized repetition threshold in [39] as follows. There, the notion of (β, p)-freeness is introduced: a word is (β, p)-free if it contains no factor that is a (β', p')-repetition (it is a word w with period length p' and exponent β': $w = p^{\beta'}$) for $\beta' \geq \beta$ and $p' \geq p$. Therefore a word is (β^+, p)-free if it is (β', p)-free for all $\beta' > \beta$ and the generalized repetition threshold $R(k, p)$ is defined for k-letter alphabet as the real number α such that either

(a) there exists an (α^+, p)-free infinite word and all (α, p)-free words are finite; or

(b) there exists an (α, p)-free infinite word and for all $\alpha > 0$, $(\alpha - \epsilon, p)$-free words are finite.

where p is the minimal avoided period.

Proof of boundary of this threshold for all alphabet sizes is also presented in [39]. Essentially $R(k, 1)$ is Dejean's repetitive threshold.

In this chapter we introduce a new constraint on infinite words and give some results. We also state some conjectures that need further and deeper investigations. Looking at maximal exponent of words containing bounded number of r_k-powers

introduces a new type of threshold, that we call the *finite-repetitions threshold*. For the alphabet of k letters, $\mathrm{FRt}(k)$ is defined as the smallest rational number for which there exists an infinite word avoiding $\mathrm{FRt}(k)^+$-powers and containing a finite number of r_k-powers, where r_k is Dejean's repetitive threshold, and $\mathrm{FRt}(k)$ is ∞ if no such exponent exists. Associated with the *finite-repetitions threshold* is the smallest number of factors of r_k-powers (limit repetitions), $Rn(k)$, that an infinite Dejean's word can accommodate.

Karhumäki and Shallit [41] showed:

(i) Every infinite 7/3-free binary word contains arbitrarily large squares.

(ii) There exists an infinite $7/3^+$-free binary word such that each square factor ww satisfies $|w| \leq 13$.

These results imply that $\mathrm{FRt}(2) = 7/3$.

In Section 1 we provide a new proof for $\mathrm{FRt}(2) = 7/3$ and we show that the associated number of squares is 12 ($Rn(2) = 12$).

In Section 2, we show that, for $k = 3$, $\mathrm{FRt}(k) = r_k = 7/4$. Proofs provided in this chapter are two-fold, because they have to show the value of $\mathrm{FRt}(k)$ as well as the associated number of r_k-powers. We prove on ternary words minimum number of associated r_k-powers is 2. The only proof of 7/4 repetition threshold is due to Dejean [32], where she has used a pure morphic word, therefore the number of 7/4-powers contained in her infinite word is not bounded.

In Section 3, we show that there exists an infinite word on 4 letters containing only 2 7/5-powers and no factor of exponent more than 7/5. The only known proofs of the 7/5 repetition threshold for 4 letters are due to Pansiot [53] and Rao [56]; their both words contain 24 7/5-powers.

In Section 4, we show on 5 letters, the only proof of the 5/4 threshold by Moulin-Ollagnier [49] provides a word with 360 5/4-powers of periods 4, 12 and 44. We show that their number can be reduced to 60 and conjecture that it can be lower down to 45, the smallest possible number.

Finally, in Section 5, revisiting the existing morphisms and proofs of Dejean's conjecture we show for $k \geq 5$, $\mathrm{FRt}(k) = r_k$. Now the question worth investigation is what is the minimum number of associated r_k-powers, ($Rn(k)$), in infinite k-ary words complying with $\mathrm{FRt}(k)$.

1 Finite-repetition threshold for k=2

In this section we consider infinite binary words in which a small number of squares occur.

It is impossible to avoid 2^+-powers and keep a bounded number of squares. As proved by Karhumäki and Shallit [41], the exponent has to go up to 7/3 to allow the property. The constraint on the number of squares imposed on binary words in this section slightly differs from the constraint considered by Shallit [64]. The squares occurring in his word have period smaller than 7. Our word contains less squares but their maximal period is 8. Here, we define two morphisms and derive the properties that we need to prove the next statement.

Theorem 14. *There exists an infinite binary word whose factors have an exponent at most* 7/3 *and that contains* 12 *squares, the fewest possible.*

Our infinite binary word contains 12 squares: $\mathtt{0}^2$, $\mathtt{1}^2$, $(\mathtt{01})^2$, $(\mathtt{10})^2$, $(\mathtt{001})^2$, $(\mathtt{010})^2$, $(\mathtt{011})^2$, $(\mathtt{100})^2$, $(\mathtt{101})^2$, $(\mathtt{110})^2$, $(\mathtt{01101001})^2$, $(\mathtt{10010110})^2$, and two words 0110110 and 1001001 of exponent 7/3.

Proving that it is impossible to have less than 12 squares in the previous statement results from the next table. The table gives the maximal length of binary words whose factors have an exponent at most 7/3, for each s number of squares, $0 \leq s \leq 11$.

s	0	1	2	3	4	5	6	7	8	9	10	11
$\ell(s)$	3	5	8	12	14	18	24	30	37	43	83	116

The above table is constructed by generating all binary words containing at most s squares and recording the the maximal length of such words for all s such that $0 \leq s \leq 11$

A weakly square-free morphism on six letters: In order to prove the Theorem 14, we consider a specific morphism, f_1 defined from Σ_6^* to itself by:

$$\begin{aligned} f_1(\mathtt{a}) &= \mathtt{abac},\\ f_1(\mathtt{b}) &= \mathtt{babd},\\ f_1(\mathtt{c}) &= \mathtt{eabdf},\\ f_1(\mathtt{d}) &= \mathtt{fbace},\\ f_1(\mathtt{e}) &= \mathtt{bace},\\ f_1(\mathtt{f}) &= \mathtt{abdf}. \end{aligned}$$

We prove below that the morphism is weakly square-free in the sense that $\mathbf{f_1} = f_1^\infty(\mathtt{a})$ is an infinite square-free word, that is, all its finite factors have an exponent smaller than 2. However, f_1 is not a square-free morphism since for example

$f_1(\mathtt{cf}) = \mathtt{eabdfabdf}$ contains a square $(\mathtt{abdf})^2$. This fact prevents us from using characterisation of square-free morphisms, or equivalently of the fixed points of such morphisms. As far as we know only an ad hoc proof is possible.

The set of codewords $f_1(a)$'s ($a \in \Sigma_6$) is a prefix code and therefore a uniquely-decipherable code. Note also that any occurrence of $\mathtt{abac}$ in $f_1(w)$, for $w \in \Sigma_6^*$, uniquely corresponds to an occurrence of $\mathtt{a}$ in w. The proof below relies on the fact that not all doublets and triplets (words of length 2 and 3 respectively) occur in $\mathbf{f_1}$, as the next statements show.

Lemma 5. *The set of doublets occurring in* $\mathbf{f_1}$ *is*

$$D = \{\mathtt{ab}, \mathtt{ac}, \mathtt{ba}, \mathtt{bd}, \mathtt{cb}, \mathtt{ce}, \mathtt{da}, \mathtt{df}, \mathtt{ea}, \mathtt{fb}\}.$$

Proof. Note that all letters of Σ_6 appear in $\mathbf{f_1}$. Then doublets $\mathtt{ab}$, $\mathtt{ac}$, $\mathtt{ba}$, $\mathtt{bd}$, $\mathtt{ce}$, $\mathtt{df}$, $\mathtt{ea}$, $\mathtt{fb}$ appear in $\mathbf{f_1}$ because they appear in the images of one letter. The images of these doublets generate two more doublets, $\mathtt{cb}$ and $\mathtt{da}$, whose images do not create new doublets. □

Lemma 6. *The set of triplets in* $\mathbf{f_1}$ *is*

$$T = \{\mathtt{aba}, \mathtt{abd}, \mathtt{acb}, \mathtt{ace}, \mathtt{bab}, \mathtt{bac}, \mathtt{bda}, \mathtt{bdf}, \mathtt{cba}, \mathtt{cea}, \mathtt{dab}, \mathtt{dfb}, \mathtt{eab}, \mathtt{fba}\}.$$

Proof. Triplets appear in the images of a letter or of a doublet. Triplets found in images of one letter are: $\mathtt{aba}$, $\mathtt{abd}$, $\mathtt{ace}$, $\mathtt{bab}$, $\mathtt{bac}$, $\mathtt{bdf}$, $\mathtt{eab}$, $\mathtt{fba}$. The images of doublets occurring in $\mathbf{f_1}$, in set D of Lemma 5, contain the extra triplets: $\mathtt{acb}$, $\mathtt{bda}$, $\mathtt{cba}$, $\mathtt{cea}$, $\mathtt{dab}$, $\mathtt{dfb}$. □

To prove that the infinite word $\mathbf{f_1}$ is square-free we discard squares containing less than four occurrences of the word $f_1(\mathtt{a}) = \mathtt{abac}$.

The word $\mathtt{abac}$ is chosen because its occurrences in $\mathbf{f_1}$ correspond to $f_1(\mathtt{a})$ only, so they are used to synchronise the parsing of the word according to the codewords $f_1(\mathtt{a})$'s.

Lemma 7. *No square in* $\mathbf{f_1}$ *can contain less than four occurrences of* $\mathtt{abac}$.

Proof. Assume by contradiction that a square ww in $\mathbf{f_1}$ contains less than four occurrences of $\mathtt{abac}$. Let x be the shortest word whose image by f_1 contains ww.

Then x is a factor of $\mathbf{f_1}$ that belongs to the set $\mathtt{a}((\Sigma_6 \setminus \{\mathtt{a}\})^*\mathtt{a})^5$. Since two consecutive occurrences of $\mathtt{a}$ in $\mathbf{f_1}$ are separated by a string of length at most 4 (the largest such string is indeed $\mathtt{bdfb}$ as a consequence of Lemma 5), the set is finite.

The square-freeness of all these factors has been checked via an elementary implementation of the test, which proves the result. □

Table 4.1: Gaps of abac: words between consecutive occurrences of abac in $\mathbf{f_1}$. They are images of gaps between consecutive occurrences of a.

$f_1(\mathtt{b})$	=	babd	4
$f_1(\mathtt{cb})$	=	eabdfbabd	9
$f_1(\mathtt{bd})$	=	babdfbace	9
$f_1(\mathtt{ce})$	=	eabdfbace	9
$f_1(\mathtt{bdfb})$	=	babdfbaceabdfbabd	17

Proposition 6. *No square in* $\mathbf{f_1}$ *can contain at least four occurrences of* abac.

Proof. Let k be the maximal integer for which $f_1^k(\mathtt{a})$ is square-free and let ww be a square occurring in $f_1^{k+1}(\mathtt{a})$ and containing at least 4 occurrences of abac. Distinguishing several cases according to the words between consecutive occurrences of abac (see Table 4.1), we deduce that $f_1^k(\mathtt{a})$ is not square-free, the contradiction.

The square ww can be written

$$\underbrace{v_0(\mathtt{abac}\cdots\mathtt{abac})u_1}\,\underbrace{v_1(\mathtt{abac}\cdots\mathtt{abac})u_2}$$

where v_0, u_1, v_1, and u_2 contain no occurrence of abac. The central part of w starting and ending with abac is the image of a unique word U factor of $f_1^k(\mathtt{a})$ due to the code property:

$$f_1(U) = v_0^{-1}wu_1^{-1} = v_1^{-1}wu_2^{-1}.$$

We split the proof into two parts according to whether abac occurs in u_1v_1 or not.

No abac in u_1v_1. We consider five cases according to the value of u_1v_1, the gap of abac (see Table 4.1).

1. $u_1v_1 =$ babd corresponds to $f_1(\mathtt{b})$ only. If either u_1 or v_1 is empty, then v_0 or u_2 is $f_1(\mathtt{b})$, in either case we get $\mathtt{b}U\mathtt{b}U$ or $U\mathtt{b}U\mathtt{b}$ that are squares. Else v_0 has a suffix d so it belongs to $f_1(\mathtt{b})$, and again $\mathtt{b}U\mathtt{b}U$ is a square.

2. $u_1v_1 =$ eabdfbabd corresponds to $f_1(\mathtt{cb})$ only. An occurrence of cb always belongs to $f_1(\mathtt{ab})$ therefore U has a prefix abd and a suffix aba, and the letter after aba is c. If v_1 is empty, u_2 has a prefix eabdfbabd so it is $f_1(\mathtt{cb})$ and again $U\mathtt{cb}U\mathtt{cb}$ is a square. If v_1 is not empty then v_0 has a suffix d, suffix of $f_1(\mathtt{b})$, therefore $\mathtt{b}U\mathtt{cb}U\mathtt{c}$ is a square.

3. $u_1v_1 =$ babdfbace corresponds to f_1(bd). The word abda is a factor of f_1(ba) only so U has a prefix aba and a suffix ba. If $|u_1| = 0$, $v_0 =$ babdfbace can only be f_1(bd) so bd$U$bd$U$ is a square. Otherwise u_2 must have a prefix b; since U has a suffix ba the next letter after it is either b or c; as only f_1(b) is prefixed by b the letter is b so u_2 has a prefix or is a prefix of f_1(b), and we know that bab is always followed by d thus Ubd$U$bd is a square.

4. $u_1v_1 =$ eabdfbace corresponds to f_1(ce) only. If u_1 is empty, v_0 is f_1(ce) so ce$U$ce$U$ is a square. Otherwise, u_2 has a prefix or is a prefix of f_1(c); the next letter after f_1(c) is either b or e; (see Lemma 5); if it is b the right-most U has a suffix aba but the left-most U has a suffix fba, which cannot be. Therefore the letter after c is e and Uce$U$ce is a square.

5. $u_1v_1 =$ babdfbaceabdfbabd. If $|v_1| > 12$, v_0 has a suffix f_1(dfb) and the letter before it is b, so bdfb$U$bdfb$U$ is a square. If $0 < |v_1| \leq 12$, then $|u_1| \geq 5$, so u_2 has a prefix or is a prefix of f_1(bd) so the next letter is either a or f. If it is a the right-most U has a suffix ba but v_0 is a suffix of or has a suffix f_1(b); the letter before it is either f_1(c) or f_1(f); if it is c then U has a prefix abd and bdfbabd is from the concatenation of f_1(c) and f_1(b) or f_1(dfb); in either case the left occurrence of U will have ea as a suffix, a contradiction since fb$U$bdfb$U$bd and Ubdfb$U$bdfb are both squares.

An occurrence of abac in u_1v_1. Then the suffix of u_1 is either aba, ab or a while the respective prefix of v_1 is c, ac or bac.

Note that c is followed either by b or e (Lemma 5) and that cb occurs only in the image of ab. Then if the occurrence of abac is followed by b, the occurrence of cb in v_0 is preceded by aba, and then there is a square starting 1, 2 or 3 positions before the occurrence of ww, which brings us back to the first case. Therefore, abac is followed by e.

The occurrence of abace comes from f_1(ac), and by Lemma 6 u_1v_1 contains an occurrence of f_1(bac). So, the occurrence of abace is preceded by d, and since da occurs only in the image of ba, the occurrence of da in u_2 is followed by bac, which yields a square starting 1, 2 or 3 positions after the occurrence of ww. Again this takes us back to the first case.

In all cases we deduce the existence of a square in f_1^k(a), which is a contradiction with the definition of k. Therefore there is no square in $\mathbf{f_1}$ containing at least four occurrences of abac. □

The next corollary is a direct consequence of Lemma 7 and Proposition 6.

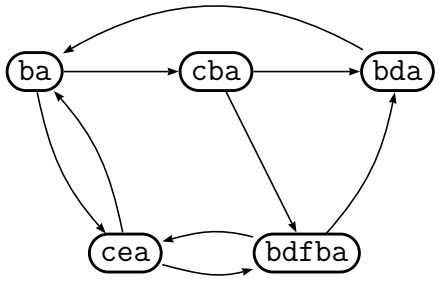


Figure 4.1: Graph showing immediate successors of gaps in the word $\mathbf{f_1}$: a suffix of it following an occurrence of a is the label of an infinite path.

Corollary 3. *The infinite word* $\mathbf{f_1}$ *is square-free, or equivalently, the morphism* f_1 *is weakly square-free.*

Binary translation: The second part of the proof of Theorem 14 consists in showing that the special infinite square-free word on 6 letters introduced earlier can be transformed into the desired binary word. This is done with a second morphism g_7 from Σ_6^* to B^* defined by

$$\begin{aligned} g_7(\mathtt{a}) &= \mathtt{10011},\\ g_7(\mathtt{b}) &= \mathtt{01100},\\ g_7(\mathtt{c}) &= \mathtt{01001},\\ g_7(\mathtt{d}) &= \mathtt{10110},\\ g_7(\mathtt{e}) &= \mathtt{0110},\\ g_7(\mathtt{f}) &= \mathtt{1001}. \end{aligned}$$

Note that the codewords of g_7 do not form a prefix code, nor a suffix code, nor even a uniquely-decipherable code! We have for example $f_1(\mathtt{ae}) = \mathtt{10011}\cdot\mathtt{0110} = \mathtt{1001}\cdot\mathtt{10110} = f_1(\mathtt{fd})$. However, parsing the word $g_7(y)$ when y is a factor of $\mathbf{f_1}$ is unique due to the absence of some doublets and triplets in $\mathbf{f_1}$ (see Lemmas 5 and 6). For example fd does not occur, which induces the unique parsing of 100110110 as $\mathtt{10011}\cdot\mathtt{0110}$.

Proposition 7. *The infinite word* $\mathbf{g_7} = g_7(f_1^\infty(\mathtt{a}))$ *contains no factor of exponent larger than* 7/3. *It contains only* 12 *squares* $\mathtt{0}^2$, $\mathtt{1}^2$, $(\mathtt{01})^2$, $(\mathtt{10})^2$, $(\mathtt{001})^2$, $(\mathtt{010})^2$, $(\mathtt{011})^2$, $(\mathtt{100})^2$, $(\mathtt{101})^2$, $(\mathtt{110})^2$, $(\mathtt{01101001})^2$, $(\mathtt{10010110})^2$. *Words* 0110110 *and* 1001001 *are the only factors with an exponent larger than* 2.

The proof is based on the fact that occurrences of 10011 in $\mathbf{g_7}$ identify occurrences of a in $\mathbf{f_1}$ and on the unique parsing mentioned above. It proceeds by considering several cases according to the gaps between consecutive occurrences of 10011 (see

Table 4.2: Gaps between consecutive occurrences of 10011 in $\mathbf{g_7}$.

$g_7(\mathtt{b})$	=	01100	5
$g_7(\mathtt{cb})$	=	0100101100	10
$g_7(\mathtt{bd})$	=	0110010110	10
$g_7(\mathtt{ce})$	=	010010110	9
$g_7(\mathtt{bdfb})$	=	0110010110100101100	19

Table 4.2), associated with gaps between consecutive occurrences of a in $\mathbf{f_1}$, which leads to path analyses in the graph of Figure 4.1.

Proof. We show that if $\mathbf{g_7}$ contains a square with at least two occurrences of 10011, it corresponds to a square in $\mathbf{f_1}$, which cannot be since $\mathbf{f_1}$ is square-free (Corollary 3).

Let w^2 be a potential square in $\mathbf{g_7}$. It is a factor of $g_7(f_1^k(\mathtt{a}))$, for some integer k. Since occurrences of 10011 correspond to $g_7(\mathtt{a})$, therefore w^2 can be written in the following form $\underbrace{v_0(g_7(\mathtt{a})\cdots g_7(\mathtt{a}))u_1}\,\underbrace{v_1(g_7(\mathtt{a})\cdots g_7(\mathtt{a}))v_2}$. The central part of w is the image of a unique square-free factor U of $f_1^k(\mathtt{a})$ due to the unique parsing mentioned above:

$$g_7(U) = (g_7(\mathtt{a})\cdots g_7(\mathtt{a})) = v_0^{-1}wu_1^{-1} = v_1^{-1}wu_2^{-1}.$$

We proceed through different cases as in the proof of Proposition 6. We are using tries to demonstrate how we find a factor of $\mathbf{f_1}$ whose image by g_7 contains w^2. The tries have at each branching part two possibilities according to Figure 4.1.

No $g_7(\mathtt{a})$ in u_1v_1.

1. $u_1v_1 = \mathtt{01100}$ corresponds to $g_7(\mathtt{b})$ only.

   If $|v_1| > 1$, then v_0 belongs to $g_7(\mathtt{b})$, $\mathtt{b}U\mathtt{b}U$ is a square. Else $|u_1| \geq 4$ so v_2 belongs to $g_7(\mathtt{b})$, it cannot belong to $g_7(\mathtt{e})$ since ae is not a factor of $\mathbf{f_1}$, therefore $U\mathtt{b}U\mathtt{b}$ is a square of $\mathbf{f_1}$.

2. $u_1v_1 = \mathtt{0110010110}$ corresponds to $g_7(\mathtt{bd})$.

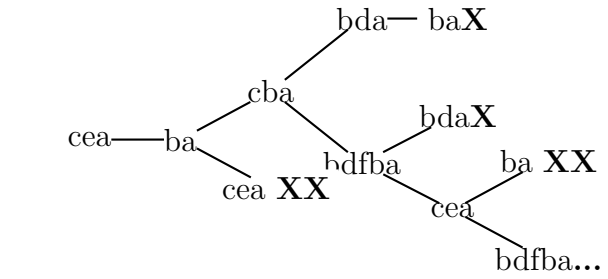
   $$v_0\,\underbrace{(g_7(\mathtt{a})\cdots g_7(\mathtt{a}))}\,g_7(\mathtt{bd})\,\underbrace{(g_7(\mathtt{a})\cdots g_7(\mathtt{a}))}\,v_2$$

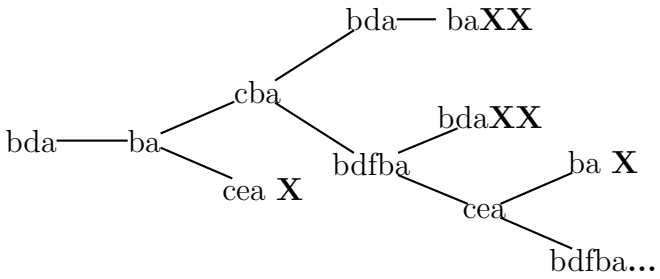
   the word abda is a factor of $f_1(\mathtt{ba})$ only, so U has a prefix abac and a suffix ba (Note that U cannot be aba since abababa is not a factor of $\mathbf{f_1}$).

   $$v_0\,\underbrace{(g_7(\mathtt{abac})\cdots g_7(\mathtt{ba}))}\,g_7(\mathtt{bd})\,\underbrace{(g_7(\mathtt{abac})\cdots g_7(\mathtt{ba}))}\,v_2$$

If v_2 comes from or has a prefix $g_7(\mathtt{b})$ then the letter after $\mathtt{bab}$ is always $\mathtt{d}$ so we have the square $U\mathtt{bd}U\mathtt{bd}$. Then v_2 is a prefix of or has a prefix $g_7(\mathtt{c})$, the LCP of $g_7(\mathtt{c})$ and $g_7(\mathtt{b})$ is $\mathtt{01}$, so v_0 has a suffix $\mathtt{10010110}$, which is a suffix of $g_7(\mathtt{bd})$ or $g_7(\mathtt{ce})$. If v_0 comes from $g_7(\mathtt{bd})$ then we have the square $\mathtt{bd}U\mathtt{bd}U$. So v_0 is a suffix of $g_7(\mathtt{ce})$

$$g_7(\mathtt{ce})\underbrace{(g_7(\mathtt{abac})\cdots g_7(\mathtt{ba}))}\, g_7(\mathtt{bd})\underbrace{(g_7(\mathtt{abac})\cdots g_7(\mathtt{ba}))}\, g_7(\mathtt{c}).$$





The sign XX shows that the particular branch of the trie terminates because either a square occurs or the sequence is not a factor of $\mathbf{f_1}$. The sign X on the other hand represents the termination of a particular branch as a consequence of the discontinuation of the corresponding branch in the other trie. If we continue these tries we will have:

$$\mathtt{ce}\,\underbrace{\mathtt{abac\,babd\,fbace\,abdf\,babd\,abac\,eabdf\,bace\,abac\,babd\,abac\,eabdf\,\ldots\,ba}}$$

$$\mathtt{bd}\,\underbrace{\mathtt{abac\,babd\,fbace\,abdf\,babd\,abac\,eabdf\,bace\,abac\,babd\,abac\,eabdf\,\ldots\,ba}}\,\mathtt{c}$$

which is the image of

$$\mathtt{eabdf\,bace}\,\underbrace{\mathtt{abac\ldots abac}}\,\mathtt{babd\,fbace}\,\underbrace{\mathtt{abac\ldots abac}}\,\mathtt{e}$$

itself image of

$$\mathtt{ce}\,\underbrace{\mathtt{a\ldots a}}\,\mathtt{bd}\,\underbrace{\mathtt{a\ldots a}}\,\mathtt{c}$$

so we have the same situation as at the starting point; but U is shorter in this case, therefore if we continue this process we should have

$$\texttt{ce abac babd fbace abdf babd abac babd fbace abdf bace a}$$

but $\texttt{abdf bace}$ is the image of $\texttt{fe}$ that is not in D (Lemma 5).

3. $u_1v_1 = \texttt{0100101100}$ corresponds to $g_7(\texttt{cb})$.

   The word $\texttt{acba}$ is a factor of $f_1(\texttt{ab})$ only, so U has a prefix $\texttt{abd}$ and a suffix $\texttt{aba}$:

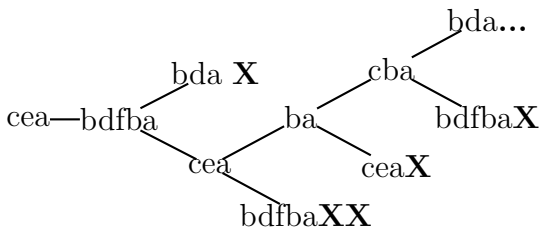
$$v_0\,\underbrace{(g_7(\texttt{abd})\ldots g_7(\texttt{aba}))}\,g_7(\texttt{cb})\,\underbrace{(g_7(\texttt{abd})\ldots g_7(\texttt{aba}))}\,v_2$$

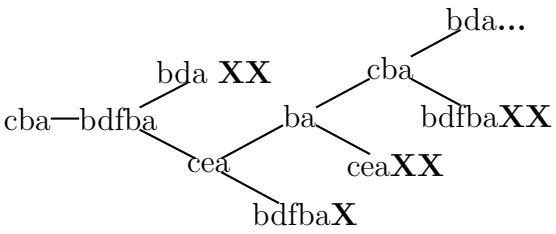
   The word v_2 comes from or has a prefix $g_7(\texttt{c})$. If the letter after it is $\texttt{b}$, we have the square $U\texttt{cb}U\texttt{cb}$.

   Otherwise v_2 comes from or has a prefix $g_7(\texttt{ce})$. If v_0 comes from or has a suffix $g_7(\texttt{b})$ then we have the square $\texttt{b}U\texttt{cb}U\texttt{c}$.

   Therefore the letter before U is $\texttt{e}$ preceded by $\texttt{c}$, i.e. the string before the left U is $\texttt{ce}$:

$$g_7(\texttt{ce})\,\underbrace{(g_7(\texttt{abd})\ldots g_7(\texttt{aba}))}\,g_7(\texttt{cb})\,\underbrace{(g_7(\texttt{abd})\ldots g_7(\texttt{aba}))}\,g_7(\texttt{ce}).$$





   Now we have the same situation as in the previous case

$$g_7(f_1(\texttt{ce}))\,\underbrace{(g_7(f_1(\texttt{abac}))\ldots g_7(f_1(\texttt{ba})))}\,g_7(f_1(\texttt{bd}))\,\underbrace{(g_7(f_1(\texttt{abac}))\ldots g_7(f_1(\texttt{ba})))}\,g_7(f_1(\texttt{c})).$$

4. $u_1v_1 = \texttt{010010110}$ corresponds to $g_7(\texttt{ce})$ only.

   Before $\texttt{c}$ is always $\texttt{ba}$ (Lemma 6) and after $\texttt{e}$ is $\texttt{ab}$ (Lemma 6), so $\texttt{ab}$ is a prefix of U and $\texttt{ba}$ is a suffix of U:

$$v_0 \underbrace{(g_7(\texttt{ab}) \dots g_7(\texttt{ba}))} g_7(\texttt{ce}) \underbrace{(g_7(\texttt{ab}) \dots g_7(\texttt{ba}))} v_2.$$

   (i): v_2 belongs to $g_7(\texttt{cb})$ since we cannot have $U\texttt{ce}U\texttt{ce}$ and the letter after $\texttt{c}$ is $\texttt{b}$ or $\texttt{e}$ (Lemma 5):

$$v_0 \underbrace{(g_7(\texttt{ab}) \dots g_7(\texttt{ba}))} g_7(\texttt{ce}) \underbrace{(g_7(\texttt{ab}) \dots g_7(\texttt{ba}))} g_7(\texttt{cb})$$

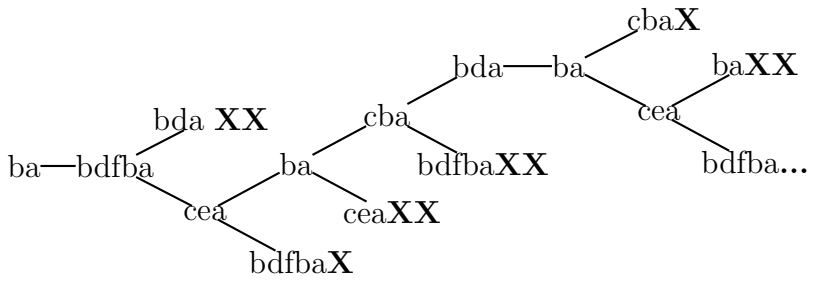
   The letter before $\texttt{bacb}$ is $\texttt{a}$ so:

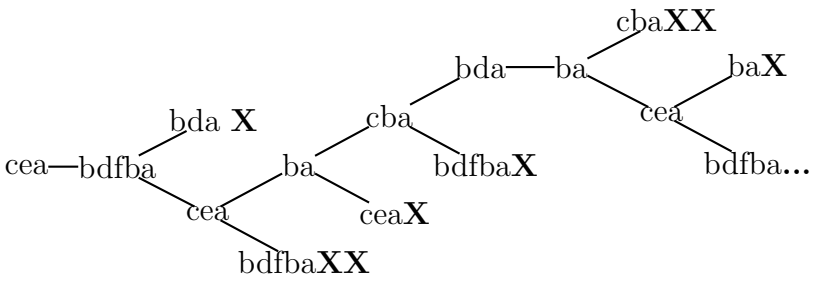
$$v_0 \underbrace{(g_7(\texttt{ab}) \dots g_7(\texttt{aba}))} g_7(\texttt{ce}) \underbrace{(g_7(\texttt{ab}) \dots g_7(\texttt{aba}))} g_7(\texttt{cb}).$$

   NOTE: U is not $\texttt{aba}$ since $\texttt{abaceabacb}$ is not a factor of $f_1^k(\texttt{a})$.

   Now $\texttt{abace}$ is a prefix of the image of $\texttt{ac}$ so U has a prefix $\texttt{abdf}$ and the word before it is either $\texttt{ce}$ or $\texttt{b}$; the first choice gives the square $\texttt{ce}U\texttt{ce}U$ and the second choice:

$$g_7(\texttt{b}) \underbrace{(g_7(\texttt{abdf}) \dots g_7(\texttt{aba}))} g_7(\texttt{ce}) \underbrace{(g_7(\texttt{abdf}) \dots g_7(\texttt{aba}))} g_7(\texttt{cb}).$$





Now if we continue the above tries we get:

$$\mathtt{b}\,\underbrace{\mathtt{abd\,fbace\,abac\,babd\,abac\,eabdf\,babd\,abac\,babd\,fbace\,abdf\,ba}\ldots\mathtt{ba}}$$

$$\mathtt{ce}\,\underbrace{\mathtt{abdf\,bace\,abac\,babd\,abac\,eabdf\,babd\,abac\,babd\,fbace\,abdf\,ba}\ldots\mathtt{ba}}\,\mathtt{cb}$$

which is the image of

$$\mathtt{bd}\,\underbrace{\mathtt{abac\,babd\,fbace\,abdf}\ldots\mathtt{ba}}\,\mathtt{ce}\,\underbrace{\mathtt{abac\,babd\,fbace\,abdf}\ldots\mathtt{ba}}\,\mathtt{b}.$$

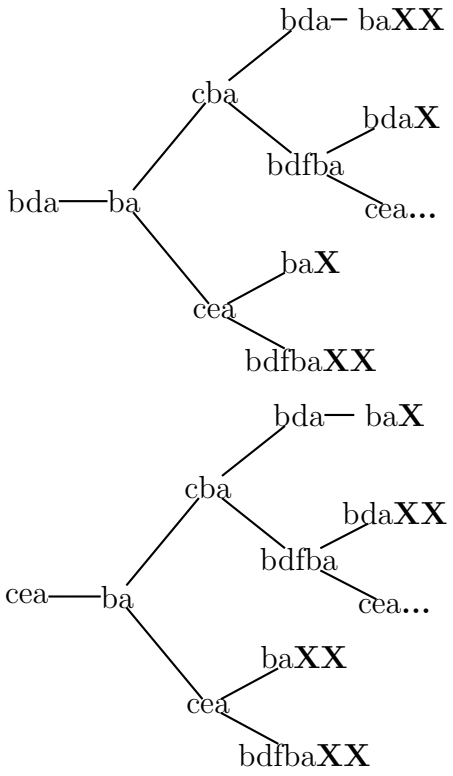
This is the same situation as the next case and we will see that after going one step back it brings us back to this case again. Now we are exactly in the same situation as at the beginning except that the length of the word $X = \mathtt{abdf}\ldots\mathtt{a}$ is shorter than U. Repeating this process enough times we should see that

$$\mathtt{babd\,fbace\,abac\,babd\,abac\,eabdf\,bace\,abac\,babd\,aba}$$

which is the image of $\mathtt{bdabaceaba}$ is not a factor of $f_1^k(\mathtt{a})$.

(ii): v_2 belongs to $g_7(\mathtt{b})$ (the LCP of $g_7(\mathtt{c})$ and $g_7(\mathtt{b})$ is $\mathtt{01}$) so v_0 must have a suffix $\mathtt{0010110}$, which belongs to $g_7(\mathtt{bd})$ because if it belongs to $g_7(\mathtt{ce})$ then $\mathtt{ce}U\mathtt{ce}U$ is a square.

$$g_7(\mathtt{bd})\,\underbrace{(g_7(\mathtt{ab})\ldots g_7(\mathtt{ba}))}\,g_7(\mathtt{ce})\,\underbrace{(g_7(\mathtt{ab})\ldots g_7(\mathtt{ba}))}\,g_7(\mathtt{b}).$$



Continuing this trie we have

$$\texttt{bd}\,\underbrace{\texttt{abac babd fbace a}\ldots\texttt{ba}}\,\texttt{ce}\,\underbrace{\texttt{abac babd fbace a}\ldots\texttt{ba}}\,\texttt{bd}.$$

This is factor of $f_1(\texttt{b}\,\underbrace{\texttt{abdf}\ldots\texttt{a}}\,\texttt{ce}\,\underbrace{\texttt{abdf}\ldots\texttt{a}}\,\texttt{cb})$ which is the previous case.

5. $u_1v_1 = \texttt{0110010110100101100}$ corresponds to $g_7(\texttt{bdfb})$ only. This case is dealt with by the same method.

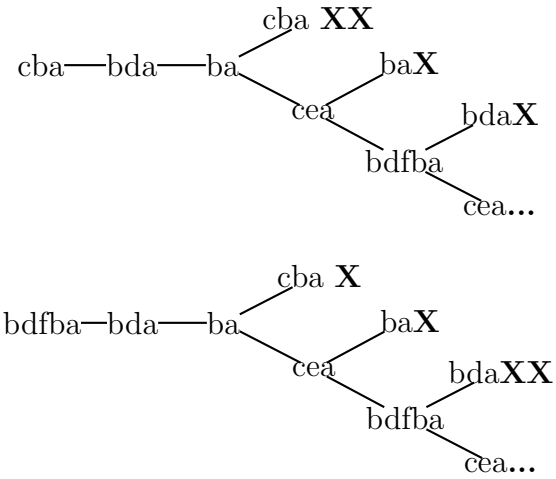
$$u_0\,\underbrace{(g_7(\texttt{a})\ldots g_7(\texttt{a}))}\,g_7(\texttt{bdfb})\,\underbrace{(g_7(\texttt{a})\ldots g_7(\texttt{a}))}\,v_2.$$

If v_2 belongs to $g_7(\texttt{c})$, the LCP of $g_7(\texttt{c})$ and $g_7(\texttt{b})$ is $\texttt{01}$ so u_0 must have a suffix $\texttt{10010110100101100}$, therefore u_0 belongs to $g_7(\texttt{bdfb})$. But $\texttt{bdfb}U\texttt{bdfb}U$ is a square and a factor of $f_1^k(a)$; a contradiction, so v_2 belongs to or has a prefix $g_7(\texttt{b})$. We have two choices here.

(i): the next word after the right occurrence of U is $\texttt{ba}$. The LCP of $g_7(\texttt{bd})$ and $g_7(\texttt{ba})$ is $\texttt{10}$, u_0 has suffix of $\texttt{110100101100}$, so it either belongs to $g_7(\texttt{dfb})$ or $g_7(\texttt{acb})$. The first case gives that $\texttt{dbf}U\texttt{bdbf}U\texttt{b}$ is a square and a factor of $f_1^k(a)$, a contradiction. So u_0 belongs to $g_7(\texttt{acb})$:

$$g_7(\texttt{acb})\,\underbrace{(g_7(\texttt{abda})\ldots g_7(\texttt{a}))}\,g_7(\texttt{bdf b})\,\underbrace{(g_7(\texttt{abda})\ldots g_7(\texttt{a}))}\,g_7(\texttt{ba}).$$

Prefixes and suffixes of U are determined only by looking at D and T.



We have:

$$\texttt{abac babd}\,\underbrace{\texttt{abac eabdf bace}\ldots\texttt{abac}}\,\texttt{babd fbace}$$

$$\texttt{abdf babd}\,\underbrace{\texttt{abac eabdf bace}\ldots\texttt{abac}}\,\texttt{babd fbace abac}$$

which is the image of

$$\texttt{ab}\underbrace{\texttt{ace}\ldots\texttt{a}}\texttt{bdfb}\underbrace{\texttt{ace}\ldots\texttt{a}}\texttt{bda}.$$

Now this is the next case so if we go back enough steps we should see that the length of U decreases and at the end we get

$$\texttt{ac babd abac eabdf babd abac eaba}$$

but this is not a factor of $f_1^k(a)$, a contradiction.

(ii): the word after U is bd. Now here the only possible letter after abd is a since if it is f it is a prefix of fb so we have Ubdfb$U$bdfb, a contradiction. As the LCP of g_7(bdfb) and g_7(bda) is 01100101101001 u_0 must have a suffix 01100 so it can belong to g_7(ab) or g_7(acb).

(I):

$$g_7(\texttt{ab})\underbrace{(g_7(\texttt{a})\ldots g_7(\texttt{a}))}g_7(\texttt{bdfb})\underbrace{(g_7(\texttt{a})\ldots g_7(\texttt{a}))}g_7(\texttt{bda}).$$

Only using D, T and the Figure 4.1 we can continue building U,

$$g_7(\texttt{ab})\underbrace{(g_7(\texttt{ace})\ldots g_7(\texttt{ba}))}g_7(\texttt{bdfb})\underbrace{(g_7(\texttt{acea})\ldots g_7(\texttt{ba}))}g_7(\texttt{bda}).$$

Continuing further we get:

$$g_7(\texttt{abac eabdf babd}\underbrace{\texttt{abac}\ldots\texttt{abac}}\texttt{babd fbace abdf babd}\underbrace{\texttt{abac}\ldots\texttt{abac}}\texttt{babda}).$$

This is the image of

$$g_7(f_1(\texttt{acb}\underbrace{\texttt{a}\ldots\texttt{a}}\texttt{bdfb}\underbrace{\texttt{a}\ldots\texttt{a}}\texttt{ba}))$$

and we are back to the case above.

(II):

$$g_7(\texttt{acb})\underbrace{(g_7(\texttt{a})\ldots g_7(\texttt{a}))}g_7(\texttt{bdfb})\underbrace{(g_7(\texttt{a})\ldots g_7(\texttt{a}))}g_7(\texttt{bda}).$$

Using the same method we build the word U:

$$\texttt{ac b}\underbrace{\texttt{abd}\ldots\texttt{ba}}\texttt{bd fb}\underbrace{\texttt{ace}\ldots\texttt{ba}}\texttt{bd a}.$$

Here we cannot go further as U cannot have abd nor ace as prefixes at the same time.

An occurrence of $g_7(\mathtt{a})$ in u_1v_1: Looking at Figure 4.1, the images of the concatenation of two connected nodes (distance 1 arrow) are the possibilities for $u_1v_1g_7(\mathtt{a})$, but note that the second period of the square must start within $g_7(a)$, starting point of the arrow, otherwise it is one of the cases above. If the lengths of both nodes are larger than 2 then by unique parsing we are bound to have a square in $f_1^k(a)$ and get a contradiction. So we have to consider only the four cases where one of the nodes is ba:

1. $u_1v_1 = g_7(\mathtt{bacb}) = 01100\mathbf{10011}0100101100$, so v_2 must have a prefix $g_7(\mathtt{b})$ and u_0 a suffix of $g_7(\mathtt{cb})$, before cb is always a, so $acbUbacbUb$ is a square in $f_1^k(a)$.

2. $u_1v_1 = g_7(\mathtt{bace}) = 01100\mathbf{10011}010010110$, so v_2 must have a prefix $g_7(\mathtt{b})$ and u_0 a suffix $g_7(\mathtt{ce})$, before ce is always a, so $aceUbaceUb$ is a square in $f_1^k(a)$.

3. $u_1v_1 = g_7(\mathtt{ceab}) = 0100101101\mathbf{10011}01100$, so v_2 must have a prefix of $g_7(\mathtt{ce})$ and u_0 a suffix of $g_7(\mathtt{b})$, after ce is always a, so $bUceabUcea$ is a square in $f_1^k(a)$.

4. $u_1v_1 = g_7(\mathtt{bdab}) = 0110010110\mathbf{10011}01100$, so using tries as before shows that after enough backward iteration we should have

   fbace abdf babd abac babd abac eabdf babd abac babd

   which contains a square.

In all cases the conclusion is that we get a square in $f_1^k(\mathtt{a})$, a contradiction with the definition of k. Here, we showed there is no square in $\mathbf{g_7}$ containing at least one 10011 Looking at the factors of $\mathbf{g_7}$ with no occurrences of 10011 shows the only squares in $\mathbf{g_7}$ are the ones listed in Proposition 7. □

Theorem 14 follows immediately from Proposition 7.

2 Finite-repetition threshold for k=3

The longest 7/4-free ternary word has length 38. Dejean's showed that *repetition threshold* for ternary words is 7/4 using the following 19-uniform morphism:

$$D(0) = 0120212012102120210,$$
$$D(1) = 1201020120210201021,$$
$$D(2) = 2012101201021012102,$$

then $D^\infty(0)$ is $(7/4)^+$-free [32]. However, since it is a pure morphic word, it contains infinitely many 7/4-powers.

Computation shows that the maximal length of $(7/4)^+$-free ternary word with only one 7/4-repetition is 102. Here, we show this length is infinite with two 7/4-powers.

Theorem 15. *The finite-repetition threshold of the 3-letter alphabet is its Dejean's repetition threshold, that is,* $7/4$.
The smallest number of $7/4$*-powers occurring in a* $7/4^+$*-power free infinite ternary word is* 2.

Since the repetition threshold for a 3-letter alphabet is 7/4, to prove this ratio is also its finite-repetition threshold it is sufficient to show (contrary to the binary case) that there exists a $7/4^+$-free infinite ternary word with finitely many 7/4-powers. To do so, we use the fact that the repetition threshold of 4-letter alphabets is 7/5 and provide a translation morphism from 4 letters to 3 letters with suitable conditions.

Proposition 8. *The following 160-uniform morphism* g_8 *maps any infinite* $7/5^+$*-free word* s *on 4-letter alphabet to infinite* $7/4^+$*-free ternary word containing only two* $7/4$*-powers, the fewest possible. The* $7/4$*-powers occurring* $g_8(s)$ *are* $\{(1020)^{7/4}, (2101)^{7/4}\}$.

$g_8(\mathtt{a}) =$ 0102101202102010210121020120210120102120121020120210121021201021012102010212012102012021012102010210120210201210212010210121020120210120102120121020102101210212,
$g_8(\mathtt{b}) =$ 0102101202102010210121020120210120102120121020120210121020102101202102012102120102101210201021201210201202101210212010210121020120210120102120121020102101210212,
$g_8(\mathtt{c}) =$ 0102101202102010210121020120210120102120121020102101202102012102120102101210201021201210201202101210201021012021020102120121020120210120102120121020102101210212,
$g_8(\mathtt{d}) =$ 0102101202102010210121020120210120102120121020102101202102010212012102012021012102010210120210201210212010210121020102120121020120210120102120121020102101210212.

Another presentation of the morphism g_8 is:

$g_8(\mathtt{a}) = uv$02120121020120210121020102101202102012102120102101yz,
$g_8(\mathtt{b}) = uv$21021201021012102010212012102012021012102010210120yz,
$g_8(\mathtt{c}) = uw$01021012021020121021201021012102010212012102012021xz,
$g_8(\mathtt{d}) = uw$12010210121020102120121020120210121020102101202102xz,

where u, v, w, x, y and z are:
$u =$ 01021012021020102101210201202101201021201210201,
$v =$ 2021012102, $w =$ 0210120210201, $x =$ 2102010212, $y =$ 0121021201021,

$z = \texttt{0121020120210120102120121020102101210212}$.
The word u is the longest common prefix of the codewords, $|u| = 47$, and z is their longest common suffix, $|z| = 40$.

The codewords of the above morphisms were determined by generating long ternary words and partitioning and testing whether they satisfy and preserve the properties required. This has been achieved by computer programming.

Direct proof of Proposition 8: Let us assume that $g_8(s)$ contains a non-extensible repetition, excluding the two 7/4-powers $\texttt{0121012}$ and $\texttt{2010201}$, with exponent at least 7/4. The repetition can be written pq where $|p|$ is the period. Then $|pq|/|p| \geq 7/4$. A simple computation verifies that no image of a $7/5^+$-free word with length at most 3 contains such repetition. Therefore the repetition is long and occurs in the image by g_8 of a word of length at least 4.

We consider two cases.

- Case $|p| \leq |q|$. The word pq is of the form

  $$pq = \overbrace{u_1 \underbrace{\cdots} v_1}\overbrace{u_1 \underbrace{\cdots} v_1}\cdots$$

  where v_1u_1 is a codeword. Indeed pq starts with the square pp of the form

  $$\overbrace{u_1 g_8(s') v_1}\overbrace{u_1 g_8(s') v_1}$$

  where $s' \in \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}\}^*$.

  Note that s' cannot be the empty word because pq does not occur in the image of a triplet.

  Let $\alpha \in \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}\}$ be such that $g_8(\alpha) = v_1u_1$. Therefore $s'\alpha s'$ is a factor of s. The letter occurring before s' in s and the letter occurring after it must differ from α to avoid the squares $\alpha s'\alpha s'$ or $s'\alpha s'\alpha$ (since s is $7/5^+$-power free).

  Then u_1 is not longer than the longest common prefix between two different codewords, that is, $|u_1| \leq |uw| = 60$. Symmetrically, v_1 is not longer than the longest common suffix of two different codewords, that is, $|v_1| \leq |yz| = 53$. But then $|v_1u_1| \leq 113$ and cannot be a complete codeword, a contradiction.

- Case $|p| > |q|$. The word pq is of the form

  $$pq = \overbrace{u_1 \underbrace{\cdots} v_1}\cdots\overbrace{u_1 \underbrace{\cdots} v_1}$$

  Let a_0 be the letter before p and b_1 the letter after q, then a_0pqb_1 is of the form

  $$a_0 \underbrace{u_1 g_8(s') v_1} a_1 \cdots b_0 \underbrace{u_1 g_8(s') v_1} b_1$$

where $s' \in \{\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d}\}^*$, $a_0, a_1, b_0, b_1 \in \{\mathtt{0},\mathtt{1},\mathtt{2}\}$ and $a_0 \neq b_0$ and $a_1 \neq b_1$, because pq is non-extensible. It rewrites as

$$a_0 u_1 g_8(s') g_8(s'') g_8(s') v_1 b_1$$

where $g_8(s'') = v_1 q^{-1} p u_1$ because the morphism is synchronizing (no codeword occurs in the concatenation of two codewords). Therefore $g_8(s')g_8(s'')g_8(s')$ is a factor of $g_8(s)$ thus $s's''s'$ is a factor of s and since s is $7/5^+$-free we get

$$\frac{|s's''s'|}{|s's''|} \leq \frac{7}{5}$$

and

$$3|s'| \leq 2|s''|$$

and eventually

$$3|g_8(s')| \leq 2|g_8(s'')| \tag{4.1}$$

because the morphism g_8 is uniform.

Furthermore $pq = u_1 g_8(s') g_8(s'') g_8(s') v_1$ and $p = u_1 g_8(s') g_8(s'') {u_1}^{-1}$ so its exponent satisfies

$$\frac{|u_1 g_8(s') g_8(s'') g_8(s') v_1|}{|g_8(s') g_8(s'')|} \geq \frac{7}{4}$$

which rewrites as

$$|g_8(s')| + 4|u_1 v_1| \geq 3|g_8(s'')| \tag{4.2}$$

Using Equations 4.1 and 4.2 we get

$$\begin{aligned} 9|g_8(s')| &\leq 6|g_8(s'')| \\ &\leq 2(|g_8(s')| + 4|u_1 v_1|) \end{aligned}$$

and then

$$|g_8(s')| \leq \frac{8}{7}|u_1 v_1|.$$

But since $|u_1 v_1| \leq 113$ as in the first case, this implies that s' is empty. Therefore the repetition pq is a factor of the image of a triplet, a contradiction.

This completes the direct proof of Proposition 8.

Proof based on Ochem's result:

Lemma 8 ([50]). *Let $\alpha, \beta \in \mathbb{Q}$, $1 < \alpha < \beta < 2$ and $p \in \mathbb{N}^*$. Let $h : \Sigma_s^* \to \Sigma_e^*$ be synchronizing q-uniform morphism (with $q \geq 1$). If $h(w)$ is (β^+, p)-free for every α^+-free word w such that $|w| < \max\{\frac{2\beta}{\beta-\alpha}, \frac{2(q-1)(2\beta-1)}{q(\beta-1)}\}$, then $h(t)$ is (β^+, p)-free for every (finite or infinite) word α^+-free word t.*

Here we split the proof into two parts, first we show the $g_8(s)$ is $7/4^+$-free then we show the only 7/4-powers are the ones mentioned above.

Let $\beta = 26/15$ and $p = 5$, here $\alpha = 7/5$ and $q = 160$. Therefore, we can show that the morphism is $(26/15^+, 5)$-free if $g_8(w)$ is $(26/15^+, 5)$-free for all $|w| < \max\{\frac{2\beta}{\beta-\alpha}, \frac{2(q-1)(2\beta-1)}{q(\beta-1)}\}$, $|w| < 11$, this set is finite and a simple computation can verify this claim.
since every $(7/4^+, 5)$- repetition is also $(26/15^+, 5)$ then we can claim the morphism is $(7/4^+, 5)$-free. So the only possible 7/4-repetitions with period less that 5 are:
$\{(0121)^{7/4}, (0212)^{7/4}, (1020)^{7/4}, (1202)^{7/4}, (2010)^{7/4}, (2101)^{7/4}\}$. Any of which must be either a factor of a codeword or a factor of the image of a doublet, it immediately concludes the existence of only $\{(1020)^{7/4}, (2101)^{7/4}\}$ as factors of $g_8(s)$.

2.1 Infinite ternary word containing one square

If we increase the threshold to $e < 2$, the maximal length of words containing only one e-power stays at 102 . However, if we relax the maximal exponent constraint further, it can be shown that there exists an infinite ternary word in which occur only one square, namely 00 up to a permutation of letters, and no e-power with $7/4 \leq e < 2$.

Proposition 9. *The following 12-uniform morphism, g_9, is such that for any $7/5^+$-free word s on 4-letter alphabet $g_9(s)$ is overlap-free ternary word with only one square,* 00 *and no e-power with $7/4 \leq e < 2$*

$$\begin{aligned} g_9(\mathtt{a}) &= 002012021012, \\ g_9(\mathtt{b}) &= 002010210012, \\ g_9(\mathtt{c}) &= 002102012001, \\ g_9(\mathtt{d}) &= 002120102001. \end{aligned}$$

Proof. The proof is based on Ochem's result (Lemma 8). Let $\beta = 17/10$ and $p = 2$, here $\alpha = 7/5$ and $q = 12$ therefore we can show that the morphism is $(17/10^+, 2)$-free if $g_9(w)$ is $(17/10^+, 2)$-free for all $|w| < \max\{\frac{2\beta}{\beta-\alpha}, \frac{2(q-1)(2\beta-1)}{q(\beta-1)}\}$, $|w| < 12$, this set is finite and a simple computation can verify this claim.
So the only possible $17/10^+$-repetitions with period less that 2 have one of the followings as a prefix:

$\{00, 11, 22\}$, any of which must be either a factor of a codeword or a factor of the image of a doublet, it immediately concludes the existence of only 00 as a factor of $g_9(s)$. □

3 Finite-repetition threshold for k=4

Pansiot proved that the *repetition threshold* for 4-letter alphabet is 7/5. In order to prove the result, Pansiot used a construction that codes a$\frac{k-1}{k-2}$-free word over the alphabet Σ_k into a binary word. Let $k \geq 3$ and w be a $\frac{k-1}{k-2}$-free word over Σ_k, of length at least $k-1$. Then every factor of length $k-1$ consists of $k-1$ different letters. The *Pansiot code* of w is the binary word $\mathtt{P}_k(w)$ such that for all $i \in \{1, ..., |w| - k + 1\}$ (for all $i \geq 1$ if w is infinite):

$$\mathtt{P}_k(w)[i] = \begin{cases} 0 & w[i+k-1] = w[i] \\ 1 & w[i+k-1] \notin \{w[i], ..., w[i+k-2]\} \end{cases}$$

Note that w is uniquely defined by $P_k(w)$ and $w[1..k-1]$. One can define an inverse operation: for a binary word w, $M_k(w)$ is the word on the alphabet Σ_k such that:

$$\mathtt{M}_k(w)[i] = \begin{cases} i & i < k \\ \mathtt{M}_k(w)[i-k+1] & i \geq k \text{ and } w[i-k+1] = 0 \\ \alpha & \text{otherwise} \end{cases}$$

where $\{\alpha\} = \Sigma_k \setminus \{\mathtt{M}_k(w)[i-k+1], ..., \mathtt{M}_k(w)[i-1]\}$. Note that if $w[i] = i$ for all $i < k$, then $\mathtt{M}_k(P_k(w)) = w$.

We shall denote by $\mathbb{S}_k$ the *symmetric group* on k elements, therefore the elements of this set are the permutations of the set $\Sigma_k = \{1, 2, ..., k\}$.

Let $\Psi : \Sigma^* \to \mathbb{S}_k$ be a morphism. We identify sometimes the repetition by (p, e), where p is non empty, and e is a prefix of pe. A repetition (p, e) in w over the alphabet Σ_k is a *short repetition* if $|e| < k - 1$, otherwise it is a *kernel repetition*. A repetition (p, e) is a Ψ-*kernel repetition* if $p \in \ker(\Psi)$.

Let $\varphi : \{0, 1\} \to \mathbb{S}_k$ be the morphism such that $\varphi(0) = (1...k-1)$ and $\varphi(1) = (1...k)$. The following Lemma by Moulin-Ollagnier gives a strong relation between kernel repetitions in a word on a k-letter alphabet and φ-kernel repetitions in its Pansiot code.

Lemma 9 ([49]). *Let w be a $\frac{k-1}{k-2}$-free word w on a k-letter alphabet. Then w has a kernel-repetition (p, e) if and only if $P_k(w)$ has a φ-kernel-repetition (p', e') with*

$|p'| = |p|$, $p'e' = P_k(pe)$ *and* $|e'| = |e'| - k + 1$.

Since the *repetition threshold* for 4-letter alphabet is 7/5, it suffices to show that there exists a $7/5^+$-free infinite word on Σ_4 with finitely many 7/5-powers. There are two proofs of Dejean's conjecture for $k = 4$, by Pansiot [53] and Rao [56]. In both cases the number of $\frac{7}{5}$-powers contained in the infinite words is 24. This proves that the finite-repetition threshold of 4-letters is $\frac{7}{5}$. Here, we prove the following:

Theorem 16. *The finite-repetition threshold of 4-letter alphabets is* $\frac{7}{5}$ *and the minimal number of* $\frac{7}{5}$*-powers is* 2.

A computer check showed that a word on a 4-letter alphabet for which the maximal exponent of factors is 7/5 and that contains at most one 7/5-power has maximal length 230. We give a construction of an infinite $\frac{7}{5}^+$-free word with only two $\frac{7}{5}$-powers, consequently we prove Theorem 16.

Let:

$$f_2(\mathtt{a}) = \mathtt{abc},$$
$$f_2(\mathtt{b}) = \mathtt{cda},$$
$$f_2(\mathtt{c}) = \mathtt{adc},$$
$$f_2(\mathtt{d}) = \mathtt{cba}.$$

$$g_{10}(\mathtt{a}) = \mathtt{aacbbaaccbaabcabc},$$
$$g_{10}(\mathtt{b}) = \mathtt{aacbacbaabbcaabbc},$$
$$g_{10}(\mathtt{c}) = \mathtt{cbaaccbbaccabcabc},$$
$$g_{10}(\mathtt{d}) = \mathtt{aacbaccaabbcaabbc}.$$

$h_4(\mathtt{a}) =$ 10110101011011010110110101011010101101101010110110101011010101
01101101010110110101011010101101101010101

$h_4(\mathtt{b}) =$ 10110101011011010110110101011010101101101010110110101011010110
10101101010110110101011011010101101010101

$h_4(\mathtt{c}) =$ 10110101011011010110110101011010101101101010110110101011010110
10101101101010110101011011010101101101010

Theorem 17. $w_0 = M_4(h_4(g_{10}(f_2^\infty(\mathtt{a}))))$ *is* $7/5^+$*-free infinite and it contains only two* $\frac{7}{5}$*-powers:* $(3421432412, 3421)$ *and* $(1423412432, 1423)$.

A computer check shows that the Pansiot code of every infinite Dejean word with at most two limit repetitions(repetitions of exponent repetitive threshold) contains a $h_4(x)$ as factor, for a $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, Moreover, every Pansiot code of an infinite Dejean word with at most two limit repetitions starting with a $h_4(x)$ (for $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$) must be followed by a $h_4(y)$, for a $y \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Thus the morphism h_4 in our construction

is unavoidable, *i.e.* for every Dejean word w which proves Theorem 16, $P_4(w)$ must be the image by h_4 of a ternary word w'.

From now on, we say that a repetition (p, e) is *forbidden* if its exponent is greater that r_k, or if it is a limit repetition different from $(3421432412, 3421)$ and $(1423412432, 1423)$. Thus a φ-kernel repetition in a Pansiot code is forbidden if $\frac{|pe|+k-1}{|p|} \geq r_k$. A computer check showed that w_0 has no small forbidden repetition. We show now that $w_1 = h_4(g_{10}(f_2^\infty(\mathtt{a})))$ has no forbidden φ-kernel repetition. The following properties come from simple checks:

- f_2 is 3-uniform, g_{10} is 17-uniform and h_4 is 99-uniform. Thus $g_{10} \circ h_4$ is 1683-uniform.
- f_2, g_{10}, h_4 and $g_{10} \circ h_4$ are synchronizing.
- The longest common prefix in $\{g_{10} \circ h_4(\mathtt{a}), g_{10} \circ h_4(\mathtt{b}), g_{10} \circ h_4(\mathtt{c}), g_{10} \circ h_4(\mathtt{d})\}$ has size 635 and the longest common suffix has size 990.

The following proposition is easily checked by computer:

Fact 6. *$\varphi(h_4(x)) = (13)$ for every $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, thus $\varphi(h_4(g_{10}(x)) = (13)$ and $\varphi(h_4(g_{10}(f_2(x))) = (13)$ for every $x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$.*

Let $\varphi' : \{0, 1, 2, 3\}^* \to \mathbb{S}_4$ such that $\varphi'(u) = (13)^{|u|}$. Note that $\varphi'(u) = \varphi(h_4(g_{10}(u)) = \varphi(h_4(g_{10}(f_2(u))))$ since f_2 and g_{10} are uniform and of odd size. Thus (p, q) is a φ'-kernel repetition if (p, q) is a repetition, and $|p|$ is even. Applying Lemma 9, we get:

Corollary 4. *Let (p_0, e_0) be a repetition in w_0. If $|e_0| \geq 3$, then $w_1 = h_4(g_{10}(f_2^\infty(\mathtt{a})))$ has a φ-kernel-repetition (p_1, e_1), with $|e_1| = |e_0| - 3$.*

Lemma 10. *Let (p_1, e_1) be a φ-kernel-repetition of $w_1 = h_4(g_{10}(f_2^\infty(\mathtt{a})))$. If $|e_1| \geq 3365$, then $w_2 = f_2^\infty(\mathtt{a})$ has a φ'-kernel-repetition (p_2, e_2) with $|e_2| \geq \left\lceil \frac{|e_1|-1625}{1683} \right\rceil$ and $|p_1| = 1683 \cdot |p_2|$.*

Proof. Suppose w.l.o.g. that (p_1, e_1) is a maximal repetition, *i.e.* there is an occurrence of p_1e_1 in w_1 which cannot be extended to the left or to the right without loosing the property of being a repetition with the same period. If $|e_1| \geq 3365$, either $g_{10} \circ h_4(\mathtt{a})$, $g_{10} \circ h_4(\mathtt{b})$ or $g_{10} \circ h_4(\mathtt{c})$ appear as a factor in e_1. Since $g_{10} \circ h_4$ is synchronizing and 1683-uniform, $|p_1|$ is a multiple of 1683. Let $|p_1| = 1683 \times k$. Then there is a factor $u = a_1 \dots a_l$ in w_2 such that $g_{10} \circ h_4(u) = vp_1e_1v'$, v is a proper prefix of $g_{10} \circ h_4(a_1)$ and v' is a proper suffix of a_l. Since (p_1, e_1) is a repetition of size $k \times 1683$, for every $k < i < l$, $a_i = a_{i-k}$. Since p_1e_1 is maximal on the left, if $|v| < 693$, then

$a_1 = a_k$, and since $p_1 e_1$ is maximal on the right, if $|v'| < 1048$, then $a_l = a_{l-k}$. If $a_1 \neq a_k$ and $a_l \neq a_{l-k}$, then $(a_2 \dots a_k, a_{k+1} \dots a_{l-1})$ is a repetition of w_2 of period k, and $l - k - 1 \geq \left\lceil \frac{|e_1|-1625}{1683} \right\rceil$. If $a_1 = a_k$ and $a_l \neq a_{l-k}$, then $(a_1 \dots a_{k-1}, a_k \dots a_{l-1})$ is a repetition of w_2 of period k, and $l - k \geq \left\lceil \frac{|e_1|-635}{1683} \right\rceil$. If $a_1 \neq a_k$ and $a_l = a_{l-k}$, then $(a_2 \dots a_k, a_{k+1} \dots a_l)$ is a repetition of w_2 of period k, and $l - k \geq \left\lceil \frac{|e_1|-990}{1683} \right\rceil$. If $a_1 = a_k$ and $a_l = a_{l-k}$, then $(a_1 \dots a_{k-1}, a_k \dots a_l)$ is a repetition of w_2 of period k, and $l - k + 1 \geq \left\lceil \frac{|e_1|}{1683} \right\rceil$. In all cases, w_2 has a repetition (p_2, e_2) of period $|p_2| = k = \frac{|p_1|}{1683}$ and with $|e_2| \geq \left\lceil \frac{|e_1|-1625}{1683} \right\rceil$. □

The proof of the following Lemma is also similar, and is omitted.

Lemma 11. *If (p_2, e_2) is a φ'-kernel repetition of $w_2 = f_2^\infty(\mathtt{a})$ with $|e_2| \geq 5$, then w_2 has a φ'-kernel-repetition (p_2', e_2') with $|e_2'| \geq \left\lceil \frac{|e_2|-2}{3} \right\rceil$ and $|p_2| = 3 \cdot |p_2'|$.*

Lemma 12. *Suppose that w_2 has a φ'-kernel-repetition (p_2, e_2) with $|e_2| \geq 5$ and $\frac{|e_2|+1}{|p_2|} \geq \frac{2}{5}$. Then there exists a (p_2', e_2') φ'-kernel-repetition with $|p_2| = 3 \cdot |p_2'|$ and $\frac{|e_2'|+1}{|p_2'|} \geq \frac{2}{5}$.*

Proof. By Lemma 11,

$$\frac{2}{5} \leq \frac{|e_2|+1}{|p_2|} \leq \frac{3 \cdot |e_2'|+3}{3 \cdot |p_2'|} = \frac{|e_2'|+1}{|p_2'|}.$$

□

One can check by computer that:

Fact 7. *There is no φ'-kernel-repetition (p_2, e_2) with $2 \leq |e_2| < 5$ and $\frac{|e_2|+1}{|p_2|} \geq \frac{2}{5}$ in w_2.*

Thus by Lemma 12:

Corollary 5. *There is no φ'-kernel-repetition (p_2, e_2) with $2 \leq |e_2|$ and $\frac{|e_2|+1}{|p_2|} \geq \frac{2}{5}$ in w_2.*

Lemma 13. *w_1 has no φ-kernel-repetition (p_1, e_1) with $|e_1| \geq 3 \cdot 1683$ and $\frac{|e_1|}{|p_1|} \geq \frac{2}{5}$.*

Proof. Suppose that w_1 has a φ-kernel-repetition (p_1, e_1) with $|e_1| \geq 3 \cdot 1683$ and $\frac{|e_1|}{|p_1|} \geq \frac{2}{5}$. By Lemma 10, w_2 has a φ'-kernel repetition (p_2, e_2) with $|e_2| \geq 2$ and

$$\frac{2}{5} \leq \frac{|e_1|}{|p_1|} \leq \frac{1683 \cdot |e_2| + 1625}{1683 \cdot |p_2|} < \frac{|e_2|+1}{|p_2|}.$$

By Corollary 5, w_2 has no such φ'-kernel repetition. Contradiction. □

To show that w_0 has no forbidden kernel repetition, it suffice to show that w_1 has no forbidden φ-kernel repetition (p_1, e_1) with $|p_1| \leq 12622$, which has been done by a computer check.

4 Finite-repetition threshold for k=5

The only proof of Dejean's conjecture for $k = 5$ is by Moulin-Ollagnier [49]:

$$h_{\mathrm{m}}(\mathtt{0}) = \mathtt{010101101101010110110}$$
$$h_{\mathrm{m}}(\mathtt{1}) = \mathtt{101010101101101101101}$$

then $M_5({h_{\mathrm{m}}}^{\infty}(\mathtt{0}))$ is $\frac{5}{4}^+$-free, however it contains 360 of such powers, of which a third have period 4, a third period 12 and the remaining period 44. This proves that the finite-repetition threshold of 5-letter alphabets is $\frac{5}{4}$.

This section is devoted to the minimum number of limit repetitions in a Dejean word on k-letters. We construct a Dejean word with only 60 limit repetitions, and we conjecture that the minimal number of limit repetitions in a Dejean word is 45. Similarly to the proof of Theorem 16, here we are looking for a morphic word w_1 such that $w_0 = M_5(w_1)$ has the desired property. This can be done by the following construction:

$$f_3(\mathtt{a}) = \mathtt{aaabbababbaaabbaabb}$$
$$f_3(\mathtt{b}) = \mathtt{aabbbaababaabbbaabb}$$

$$g_{11}(\mathtt{a}) = \mathtt{aaaababbbbababaaaababbb}$$
$$g_{11}(\mathtt{b}) = \mathtt{bbbbabaaaababababbbbabaaa}$$

$$h_5(\mathtt{a}) = \mathtt{1101101010101101101010101101101010110110101011011010101101101010110}$$
$$\mathtt{1101101101010101101101010110110101011011010110101010110}$$
$$h_5(\mathtt{b}) = \mathtt{1101101010110110101010110110101010110110110110101011011011010101}$$
$$\mathtt{0110110101011011010101011011010101011011010110101010110}$$

Theorem 18. *$w_0 = M_5(h_5(g_{11}({f_3}^{\infty}(\mathtt{a}))))$ is $\frac{5}{4}^+$-free and it contains only 60 of such powers, all of which have period 4.*

The following properties will help the proof of the Theorem 18:

- f_3 is 19-uniform, g_{11} is 29-uniform and h_5 is 113-uniform. Thus $g_{11} \circ h_5$ is 3277-uniform.
- f_3, g_{11}, h_5 and $g_{11} \circ h_5$ are synchronizing.
- The longest common prefix in $\{g_{11} \circ h_5(\mathtt{a}), g_{11} \circ h_5(\mathtt{b})\}$ has size 11 and the longest common suffix has size 24.

- $\varphi(h_5(x)) = (12)(354)$ for every $x \in \{\mathtt{a}, \mathtt{b}\}$, thus $\varphi(h_5(g_{11}(x)) = (12)(345)$ and $\varphi(h_5(g_{11}(f_3(x))) = (12)(345)$ for every $x \in \{\mathtt{a}, \mathtt{b}\}$.

Let $\varphi' : \{0,1,2,3,4\}^* \to \mathbb{S}_5$ such that $\varphi'(u) = [(12)(345)]^{|u|}$. Thus (p,q) is a φ'-kernel repetition if and only if (p,q) is a repetition, and $|p|$ is divisible by 6.

Lemma 14. *Let (p_1, e_1) be a φ-kernel-repetition of $w_1 = h_5(g_{11}(f_3^\infty(\mathtt{a})))$. If $|e_1| \geq 6553$, then $w_2 = f_3^\infty(\mathtt{a})$ has a φ'-kernel-repetition (p_2, e_2) with $|e_2| \geq \left\lceil \frac{|e_1|-34}{3277} \right\rceil$ and $|p_1| = 3277 \cdot |p_2|$.*

Lemma 15. *If $|e_2| \geq 37$, then $w_2 = f_3^\infty(\mathtt{a})$ has a φ'-kernel-repetition (p'_2, e'_2) with $|e'_2| \geq \left\lceil \frac{|e_2|-8}{19} \right\rceil$ and $|p_2| = 19 \cdot |p'_2|$.*

Here we adapt the same approach as the Section 3 (Lemma 12 and Proposition 7) with cooperating the size of the morphism f and the exponent $5/4$, the next Corollary follows:

Corollary 6. *There is no φ'-kernel-repetition (p_2, e_2) with $6 \leq |e_2|$ and $\frac{|e_2|+1}{|p_2|} \geq \frac{1}{4}$ in w_2.*

Lemma 16. *w_1 has no φ-kernel-repetition (p_1, e_1) with $|e_1| \geq 6 \cdot 3277$ and $\frac{|e_1|}{|p_1|} \geq \frac{1}{4}$.*

The proof is a direct consequence of Lemma 14 and 15, therefore it is sufficient to check that w_0 has no forbidden repetition (p_0, e_0) with $|p_0| \leq (6 \cdot 3277 \cdot 4) = 78648$. This claim can be verified by a basic computation which also reveals that there are only 60 limit repetitions (p_0, e_0) in w_0, and for every limit repetition, $|e_0| = 1$.

The following facts have been verified by computer check.

Fact 8.
- *A Dejean word on a 5-letter alphabet that contains at most 44 limit repetitions has size at most 4648.*
- *A Dejean word on a 5-letter alphabet that contains at most 45 limit repetitions, and such that every limit repetition has period 4, has size at most 7330.*

Still based on computer checks, we conjecture the following:

Conjecture 1.
- *There exists an infinite Dejean word on a 5-letter alphabet with only 45 limit repetitions.*
- *There exists an infinite Dejean word on a 5-letter alphabet with only 46 limit repetitions, and such that every limit repetition has period 4.*

5 Finite-repetition threshold for $k > 5$

Looking at the existing proofs for Dejean's conjecture shows in fact $\mathrm{FRt}(k) = r_k$ for $k \geq 6$, that is, the known constructions of Dejean's words for $k \geq 5$ have finitely many limit repetitions.

- $6 \leq k \leq 11$ (cases are by Moulin-Ollagnier [49]), and $12 \leq k \leq 38$ (cases are by Rao [56]). In both proofs, authors show that if the Pansiot's code of the constructed word w contains a φ-kernel repetition (p, e) with e markable, then the word has a φ-kernel repetition of smaller period (p', e') with $\frac{|e|}{|p|} \leq \frac{|e'|}{|p'|}$ ([49, Section 3.5], [56, Corollary 9]). By Lemma 9, (p, e) (resp. (p', e')) corresponds to a kernel repetition of period $|p|$ and size $|pe| + k - 1$ in w (resp. $|p'|$ and size $|p'e'| + k - 1$). Since $\frac{|pe|+k-1}{|p|} < \frac{|p'e'|+k-1}{|p'|} \leq RT(t)$, (p', e') does not correspond to a limit repetition. Thus w cannot have arbitrary long limit kernel repetitions, and we have $\mathrm{FRt}(k) = r_k$. Moreover, a simple computer check reveals that in each of these cases, all limit repetitions have period $k - 1$, and thus there are at most $k!$ of limit repetitions.

- $k > 38$. Theses cases are done by Carpi. A close inspection of [19, Proposition 8.2] shows this proposition remains valid if the factor is a long enough limit repetition. Thus Carpi's construction cannot have arbitrary long limit repetitions, and we have $\mathrm{FRt}(k) = r_k$.

6 Conclusion

In this chapter, we introduced the notion of *finite-repetition threshold*. This notion is a bound the maximal exponent of infinite words on k-letter alphabet containing bounded number of r_k-powers, where r_k is Dejean's repetitive threshold.

For all k we studied this threshold, and concluded that $\mathrm{FRt}(k) = r_k$ for $k > 2$. For the case where $k = 2$ this threshold is $7/3$, we presented a new proof of this result.

With this constraint comes a function $Rn(n)$, the minimum number of r_k-powers (limit repetitions) in infinite words on k letters complying with $\mathrm{FRt}(k)$.

In this chapter we showed that $Rn(2) = 12$, $Rn(3) = 2$, $Rn(4) = 2$ and $Rn(5) \leq 60$. We conjecture $Rn(5) = 45$.

We finish this chapter by two straightforward open questions.

- Is it possible to construct Dejean's words such that the only allowed limit repetitions have period $k - 1$, for every $k > 38$? Maybe a closer inspection of Carpi's construction will give the result.

- Can we find a lower or an upper bound for $Rn(k)$ when $k > 5$?

5

Fewest repetitions vs maximal-exponent powers in infinite binary words

In this chapter, we provide some results that deepen the question of avoidable patterns in infinite binary words by introducing another point of view. We analyse the trade-off between the number of (distinct) squares and the number of maximal-exponent repetitions occurring in infinite binary words when the maximal exponent is constant. The interesting results show the behaviour of infinite binary words when the maximal exponent varies between 3 to 7/3. The value 7/3 is called the finite-repetition threshold in Chapter 4. And the value 3 of the maximal exponent is where the number of squares is the absolute minimum. The next table summarises the results.

Maximal exponent e	Allowed number of e-powers	Minimum number of squares	
7/3	2	12	Theorem 14
	1	14	Theorem 20
5/2	2	8	Theorem 21
	1	11	Theorem 22
3	2	3	Lemma 1
	1	4	Theorem 24

Proving that it is impossible to have less than 12 squares when avoiding 5/2 powers of binary words needs a simple computation. The next table shows the maximal length $\ell(s)$ of binary words that both avoid 5/2 powers and contain at most s squares, $0 \leq s \leq 11$.

s	0	1	2	3	4	5	6	7	8	9	10	11
$\ell(s)$	3	5	8	12	14	18	24	30	37	43	83	116

This leads to the following fact.

Fact 9. *There is no* 5/2*-free infinite binary word containing less than* 12 *squares.*

Similarly, proving that it is impossible to have less than 8 squares when avoiding cubes needs another simple computation. The next table displays the maximal length $\ell(s)$ of binary words that simultaneously avoid cubes and contain at most s squares, $0 \leq s \leq 7$.

s	0	1	2	3	4	5	6	7
$\ell(s)$	3	5	8	12	29	41	57	73

Consequence of this, results in the following fact.

Fact 10. *There is no cube-free infinite binary word containing less than* 8 *squares.*

The next table summarises the minimum number s of squares that an infinite e-free binary words should contain for the significant values of the exponent e.

e	3	5/2	7/3
s	8	12	∞

Each section is devoted to showing that each of the exponents, 7/3, 5/2 and 3, are truly thresholds. Proofs are two fold; first we show that there exists an infinite binary word complying with the threshold and the claimed number of squares. Second, we show this claimed number is in fact minimal.

For each case in order to generate the binary word with the desired property, we first use a pure morphic word and by a second morphism, translate the corresponding fixed point to a binary word. In order to show the number of squares is minimal, simple computations are exhibited. We generate all binary words containing less squares. For all of the following cases in this chapter these sets are final which proves the minimality of the corresponding repetitions.

1 Binary words with Maximum Exponent 7/3

In this section, we recall for completeness Theorem 14, from Chapter 4 that the Finite-repetition threshold of binary alphabet, FRt(2), is 7/3 and that its associated minimal number of squares is 12. We then show that number of squares goes up to 14 if the number of maximal-exponent powers is reduced to 1.

Theorem 19 ([6])**.** *There exists an infinite binary word whose factors have an exponent at most* 7/3 *and that contains* 12 *squares, the fewest possible.*

As defined in Chapter 4, 7/3 is the Finite-repetition threshold for the binary alphabet since there is no infinite binary word that avoids 7/3 powers and simultaneously contains finitely many squares [64]. To show that there exists an infinite binary whose factors have maximum exponent 7/3 and that contains only 12 squares we used two morphisms f_1 and g_7 in Chapter 4. The first morphism f_1 is defined from Σ_6^* to itself by:

$$\begin{array}{ll} f_1(\mathtt{a}) = \mathtt{abac}, & f_1(\mathtt{b}) = \mathtt{babd}, \\ f_1(\mathtt{c}) = \mathtt{eabdf}, & f_1(\mathtt{d}) = \mathtt{fbace}, \\ f_1(\mathtt{e}) = \mathtt{bace}, & f_1(\mathtt{f}) = \mathtt{abdf}. \end{array}$$

And the second morphism g_7 from Σ_6^* to B^* is defined by:

$$\begin{array}{ll} g_7(\mathtt{a}) = \mathtt{10011}, & g_7(\mathtt{b}) = \mathtt{01100}, \\ g_7(\mathtt{c}) = \mathtt{01001}, & g_7(\mathtt{d}) = \mathtt{10110}, \\ g_7(\mathtt{e}) = \mathtt{0110}, & g_7(\mathtt{f}) = \mathtt{1001}. \end{array}$$

Then the infinite word $\mathbf{g_7} = g_7({f_1}^{\infty}(\mathtt{a}))$ has the desired property. Finally, that 12 is the fewest number of squares is a consequence of Fact 9.

Our infinite binary word $\mathbf{g_7}$ contains 12 squares $\{\mathtt{0}^2,\ \mathtt{1}^2,\ (\mathtt{01})^2,\ (\mathtt{10})^2,\ (\mathtt{001})^2,\ (\mathtt{010})^2,\ (\mathtt{011})^2,\ (\mathtt{100})^2,\ (\mathtt{101})^2,\ (\mathtt{110})^2,\ (\mathtt{01101001})^2,\ (\mathtt{10010110})^2\}$. It also contains only two words 0110110 and 1001001 of exponent 7/3.

Under the same constraint on the maximal exponent (7/3) of factors in infinite words, but allowing only one factor of that exponent, the number of squares jumps to 14. This is the smallest possible number of squares as a consequence of a computation displayed in the table below, which gives the maximal length $\ell(s)$ of $7/3^+$-free binary words that contain only one 7/3-power and at most s squares, $0 \leq s \leq 13$.

s	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$\ell(s)$	3	5	8	12	14	18	24	30	36	39	50	70	100	167

Theorem 20. *There exists a $7/3^+$-free infinite binary word with only one 7/3-power and that contains no more than 14 squares.*

The proof is a corollary of Proposition 10 stated after a series of lemmas. As for previous proof, we generate the infinite word by morphism iteration and translation. We consider the specific morphism, f_4, defined from Σ_5^* to itself by:

$$f_4(\mathtt{a}) = \mathtt{adcbebc},$$
$$f_4(\mathtt{b}) = \mathtt{adcbedc},$$
$$f_4(\mathtt{c}) = \mathtt{aebc},$$
$$f_4(\mathtt{d}) = \mathtt{aebedc},$$
$$f_4(\mathtt{e}) = \mathtt{aebedcbebc}.$$

Then we translate $f_4^\infty(\mathtt{a})$ to binary using the second morphism g_{12} from Σ_5^* to B^* defined by:

$$g_{12}(\mathtt{a}) = \mathtt{101001100101},$$
$$g_{12}(\mathtt{b}) = \mathtt{1010011001001},$$
$$g_{12}(\mathtt{c}) = \mathtt{101001011001},$$
$$g_{12}(\mathtt{d}) = \mathtt{101001011001001},$$
$$g_{12}(\mathtt{e}) = \mathtt{10100101100100110010 1},$$

and denote $\mathbf{f_4} = {f_4}^\infty(\mathtt{a})$, $\mathbf{g_{12}} = g_{12}({f_4}^\infty(\mathtt{a}))$.

Lemma 17. *The set of doublets occurring in* $\mathbf{f_4}$ *is*

$$D = \{\mathtt{ad}, \mathtt{ae}, \mathtt{bc}, \mathtt{be}, \mathtt{ca}, \mathtt{cb}, \mathtt{dc}, \mathtt{eb}, \mathtt{ed}\}.$$

Proof. Note that doublets appear in the images of single letters or of doublets. Then doublets ad, ae, bc, be, cb, dc, eb, ed appear in $\mathbf{f_4}$ because they appear in the images of single letter, and ca appears in the image of any doublets. □

Lemma 18. *The set of triplets in* $\mathbf{f_4}$ *is*

$$T = \{\mathtt{adc}, \mathtt{aeb}, \mathtt{bca}, \mathtt{beb}, \mathtt{bed}, \mathtt{cad}, \mathtt{cae}, \mathtt{cbe}, \mathtt{dca}, \mathtt{dcb}, \mathtt{ebc}, \mathtt{ebe}, \mathtt{edc}\}.$$

Proof. Triplets appear in the images of single letter or of a doublet. Triplets found in images of one letter are: adc, aeb, beb, bed, cbe, dcb, ebc, ebe, edc. The images of doublets occurring in $\mathbf{f_4}$, in set D of Lemma 17, contain the extra triplets: bca, cad, cae, dca. □

Lemma 19. *Let* $P = \{\alpha s \mathtt{a} s \mathtt{b}, \mathtt{d} s \mathtt{b} s \mathtt{d}, \mathtt{a} s \mathtt{b} s \mathtt{a}, \mathtt{b} s \mathtt{d} s \alpha, \mathtt{c} s \mathtt{e} s \alpha, \mathtt{a} s \mathtt{e} s \alpha\}$ *where* s *is a factor of* $\mathbf{f_4}$ *and* $\alpha \in \Sigma_5$*. If* $f_4^k(\mathtt{a})$ *is square-free, it avoids the set* P*.*

Proof. The basis of the proof is to exhibit the letters of s. This is done from both ends, from left and from right, using the sets D and T iteratively and also by looking at the codewords. In any of the six cases below, only one s is possible and we show here that considering the word $f_4^k(\mathtt{a})$ is a finite word and square-free the existence of such s is impossible.

1. Assume that $\alpha s \mathtt{a} s \mathtt{b}$ is not avoidable in $f_4^k(\mathtt{a})$ then:

$$\alpha\cdots\texttt{ca}\cdots\texttt{cb}$$
$$\alpha\cdots\texttt{dca}\cdots\texttt{dcb}$$
$$\alpha\cdots\texttt{edca}\cdots\texttt{edcb}$$
$$\alpha\cdots\texttt{ebedca}\cdots\texttt{ebedcb}$$

Notice that we have not yet exhibited s fully, because α`ebedcaebedcb` is a factor of f_4(`de`) and `de` $\notin D$. Therefore, we continue completing s. f_4(`c`) must follow f_4(`d`):

$$\alpha\texttt{ebc}\cdots\texttt{aebedcaebc}\cdots\texttt{aebedcb}$$

therefore α is `b`, we continue

$$\texttt{bebc}\cdots\texttt{aebedcaebc}\cdots\texttt{aebedcb}$$

therefore it's a factor of image of $\alpha_1 s_1$`dc`s_1`e` where s_1 is not empty and α_1 is either `a` or `e`. Similarly we try to build s_1

$$\alpha_1\cdots\texttt{adc}\cdots\texttt{ae}$$
$$\alpha_1\texttt{be}\cdots\texttt{adcbe}\cdots\texttt{ae}$$

therefore α_1 is `e`

$$\texttt{ebedc}\cdots\texttt{adcbedc}\cdots\texttt{ae}$$

Note that $\cdots$ is not empty, so we continue:

$$\texttt{ebedcaebc}\cdots\texttt{aebedcbebcadcbedcaebc}\cdots\texttt{aebedcbebcae}$$

is a factor of image of `dc`s_2`ebc`s_2`ed` and s_2 is not empty, which is case 2.

2. Assume that `d`s`b`s`d` is not avoidable in f_4^k(`a`) then:

$$\texttt{dca}\cdots\texttt{bca}\cdots\texttt{d}$$
$$\texttt{dca}\cdots\texttt{dcbebca}\cdots\texttt{dcbed}$$

This is a factor of image of αs`a`s`b` which is case 1, note that $\cdots$ is not empty. Now we have a loop where $\cdots$ decreases at least 4 times each time, thus at some point this word should be a factor of image of a triplet in T. There exist no $w \in T$ such that image of w has a factor in the form of case 1 or 2.

3. Assume that `a`s`b`s`a` is not avoidable in f_4^k(`a`) then:

$$\begin{array}{c}\mathtt{aeb}\cdots\mathtt{cbeb}\cdots\mathtt{ca}\\ \mathtt{aebc}\cdots\mathtt{aebedcbebc}\cdots\mathtt{aebedca}\end{array}$$

this is a factor of image of $\mathtt{c}s_1\mathtt{e}s_1\mathtt{dc}$ where s_1 is not empty, look at case 5.

4. Assume that $\mathtt{bsd}s\alpha$ is not avoidable in $f_4^k(\mathtt{a})$ then:

$$\begin{array}{c}\mathtt{bca}\cdots\mathtt{dca}\cdots\alpha\\ \mathtt{bcaebedcbebc}\cdots\mathtt{adcbedcaebedcbebc}\cdots\mathtt{adcbe}\alpha\end{array}$$

therefore α is $\mathtt{b}$. The word above is a factor of image of $\mathtt{a}s_1\mathtt{b}s_1\mathtt{a}$ case 3.

5. Assume that $\mathtt{cse}s\alpha$ is not avoidable in $f_4^k(\mathtt{a})$ then:

$$\begin{array}{c}\mathtt{cbe}\cdots\mathtt{ebe}\cdots\alpha\\ \mathtt{cbedca}\cdots\mathtt{aebedca}\cdots\mathtt{a}\alpha\end{array}$$

now this is a factor of image of $\mathtt{bsd}s\alpha$, case 4.

6. Assume that $\mathtt{ase}s\alpha$ is not avoidable in $f_4^k(\mathtt{a})$ then:

$$\begin{array}{c}\mathtt{adcbe}\cdots\mathtt{edceb}\cdots\alpha\\ \mathtt{adcbebc}\cdots\mathtt{aebedcebc}\cdots\mathtt{aeb}\alpha\end{array}$$

the $\cdots$ is not empty, this is a factor of image of $\mathtt{a}s_1\mathtt{e}s_1\alpha$ where we started case 6, but each time $|s_1| < 4|s|$ and at no point $\cdots$ is empty. Contradiction to finiteness of $f_4^k(\mathtt{a})$.

Therefore we show here at no point the missing part of s is empty and this contradicts the finiteness of $f_4^k(\mathtt{a})$, thus if $f_4^k(\mathtt{a})$ is square-free, it avoids the set P. □

Lemma 20. *The morphism f_4 is weakly square-free, i.e. $f_4^\infty(\mathtt{a})$ is square-free.*

Proof. Letter $\mathtt{a}$ appears in $f_4^\infty(\mathtt{a})$ only as a prefix of the codewords therefore any factor of $f_4^\infty(\mathtt{a})$ starting and ending with $\mathtt{a}$ is uniquely decipherable. Set of all factors of $f_4^\infty(\mathtt{a})$ containing at most 3 occurrences of $\mathtt{a}$ is finite. A simple computation can verify that no word in this set contains a square.

Let k be the maximal integer such that $f_4^k(\mathtt{a})$ is square-free and ww contains at least 4 occurrences of $\mathtt{a}$ and it is a factor of $f_4^{k+1}(\mathtt{a})$, so the square ww can be written as:

$$\underbrace{u_0\overbrace{\mathtt{a}\cdots\mathtt{a}}v_0}\underbrace{u_1\overbrace{\mathtt{a}\cdots\mathtt{a}}v_1}$$

where $\mathtt{a}$ does not occur in v_0u_1, therefore $\mathtt{a}v_0u_1$ is one of the codewords. Distinguishing several cases according to the possibilities of v_0u_1 we deduce that $f_4^k(\mathtt{a})$ is not square-free or ww that is not a factor of $f_4^{k+1}(\mathtt{a})$ for any k, contradiction.

Case $\mathtt{a}v_0u_1 = f_4(\mathtt{a})$**:**

$$\underbrace{u_0\overbrace{\mathtt{a}\cdots\mathtt{a}}\mathtt{dcbebc}\overbrace{\mathtt{a}\cdots\mathtt{a}}v_1}$$

u_0 has $\mathtt{bc}$ as a suffix therefore it's either a suffix of $f_4(\mathtt{c})$ or $f_4(\mathtt{e})$ so ww is a factor of $f_4(\mathtt{c}s\mathtt{a}s\mathtt{b})$. By Lemma 19 case 1, $\mathtt{c}s\mathtt{a}s\mathtt{b}$ does not occur in $f_4^k(\mathtt{a})$, Contradiction.

Case $\mathtt{a}v_0u_1 = f_4(\mathtt{b})$**:**

$$\underbrace{u_0\overbrace{\mathtt{a}\cdots\mathtt{a}}\mathtt{dcbedc}\overbrace{\mathtt{a}\cdots\mathtt{a}}v_1}$$

u_0 has a suffix $\mathtt{dc}$, otherwise v_1 has $\mathtt{dcbed}$ as a prefix and therefore ww is a factor of $f_4(s\mathtt{b}s\mathtt{b})$, contradiction to square-freeness of $f_4^k(\mathtt{a})$. So u_0 is a suffix of $f_4(\mathtt{d})$ and similarly v_1 has $\mathtt{ad}$ as a prefix therefore it's a prefix of $f_4(\mathtt{a})$, thus ww is a factor of $f_4(\mathtt{d}s\mathtt{b}s\mathtt{a})$ since the only letter proceeded by $\mathtt{a}$ is $\mathtt{c}$ and also the only letter after $\mathtt{d}$ is $\mathtt{c}$ therefore s has suffix and prefix $f_4(\mathtt{c})$

$f_4(\mathtt{dc}\overbrace{\cdots}\mathtt{cbc}\overbrace{\cdots}\mathtt{ca})$ but $\mathtt{cbc}$ is not a factor of $f_4^k(\mathtt{a})$.

Case $\mathtt{a}v_0u_1 = f_4(\mathtt{c})$**:**

$$\underbrace{u_0\overbrace{\mathtt{a}\cdots\mathtt{a}}\mathtt{ebc}\overbrace{\mathtt{a}\cdots\mathtt{a}}v_1}$$

Looking at the set of triplets we can see that $\mathtt{c}$ is always followed by $\mathtt{a}$ or proceeded by $\mathtt{d}$. Therefore if the first case is true the only letter proceeded by $\mathtt{a}$ is $\mathtt{c}$ so we have ww is a factor of $f_4(\mathtt{ca}s\mathtt{ca}s\alpha)$ and $\mathtt{ca}s\mathtt{ca}s$ is a square in $f_4^k(\mathtt{a})$, a contradiction. In the second case the only letter followed by $\mathtt{d}$ is $\mathtt{c}$ so we have ww is a factor of $f_4(\alpha s\mathtt{dc}s\mathtt{dc})$ and $s\mathtt{dc}s\mathtt{dc}$ is a square in $f_4^k(\mathtt{a})$, a contradiction.

Case $\mathtt{a}v_0u_1 = f_4(\mathtt{d})$**:**

$$\underbrace{u_0\overbrace{\mathtt{a}\cdots\mathtt{a}}\mathtt{ebedc}\overbrace{\mathtt{a}\cdots\mathtt{a}}v_1}$$

so ww is a factor of $f_4(\alpha_0s\mathtt{d}s\alpha_1)$ where $\alpha_0, \alpha_1 \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{e}\}$, $\mathtt{d}$ is followed by $\mathtt{c}$ therefore α_0 is $\mathtt{b}$ and $\mathtt{bc}$ is followed by $\mathtt{a}$ always, therefore $\mathtt{dca}$ is preceded by $\mathtt{e}$ thus α_1 is $\mathtt{b}$. So we have ww a factor of $f_4(\mathtt{bca}\cdots\mathtt{edca}\cdots\mathtt{eb})$ but this string could not contain any squares as no concatenation of no prefix and suffix of $f_4(\mathtt{b})$ is the same as $f_4(\mathtt{d})$.

Case $\mathtt{a}v_0u_1 = f_4(\mathtt{e})$**:**

$$\underbrace{u_0\overbrace{\mathtt{a}\cdots\mathtt{a}}\mathtt{ebedcbebc}\overbrace{\mathtt{a}\cdots\mathtt{a}}v_1}$$

we have ww a factor of $f_4(\mathtt{a}s\mathtt{e}s\mathtt{d})$ the common letter followed by both $\mathtt{e}$ and $\mathtt{a}$ is $\mathtt{d}$ and the only letter proceeded by $\mathtt{ed}$ is $\mathtt{b}$ so we must have $f_4(\mathtt{ad}\cdots\mathtt{bed}\cdots\mathtt{bd})$ but $\mathtt{bd}$

is not in the set of doublets.

Now if ww contains at most 3 occurrences of $\mathtt{a}$: therefore ww is a factor of $f_4(s)$ where $|s| < 5$ this is simple to investigate and confirm image of all s with length at most 4 is square-free. □

Proposition 10. *The infinite word* $\mathbf{g_{12}} = g_{12}(f_4^{\infty}(\mathtt{a}))$ *contains no factor of exponent larger than* 7/3*. It contains* 14 *squares* $\{\mathtt{0}^2, \mathtt{1}^2, (\mathtt{01})^2, (\mathtt{10})^2, (\mathtt{001})^2, (\mathtt{010})^2, (\mathtt{100})^2, (\mathtt{101})^2, (\mathtt{0110})^2, (\mathtt{1001})^2, (\mathtt{100110})^2, (\mathtt{0100110})^2, (\mathtt{0110010})^2, (\mathtt{10010110})^2\}$*, and only one* 7/3*-power,* $\mathtt{1001001}$*.*

Proof. The factor $\mathtt{101001}$ appears in $g_{12}({f_4}^k(\mathtt{a}))$ only as a prefix of the codewords, therefore any factor starting and ending with $\mathtt{101001}$ is uniquely decipherable. If there is a square ww that contains $2n$ occurrences of $\mathtt{101001}$, where $n \geq 2$ and it is a factor of $g_{12}({f_4}^k(\mathtt{a}))$ where ${f_4}^k(\mathtt{a})$ is square-free, so the square ww can be written as:

$$\underbrace{u_0 \overbrace{\alpha_1 \cdots \alpha_n} v_0}\underbrace{u_1 \overbrace{\alpha_1 \cdots \alpha_n} v_1}$$

where $\alpha_1, ...\alpha_n$ are occurrences of $\mathtt{101001}$, $n \geq 2$ and v_0u_1 contains no $\mathtt{101001}$ as a factor therefore $\alpha_n v_0 u_1$ is one of the codewords.

Similar to the proof of Lemma 20, we study different cases of possible v_0u_1.

Case $\alpha_n v_0 u_1 = g_{12}(\mathtt{a})$**:** So ww is a factor of $g_{12}(\alpha s \mathtt{a} s \beta)$ where s is not empty and $\alpha, \beta \in \Sigma_5$. The letter before $\mathtt{a}$ is always $\mathtt{c}$ so s has $\mathtt{a}$ as a suffix. Since β is a letter occurring after s, therefore β must be $\mathtt{b}$. Thus ww is a factor of $g_{12}(\alpha s \mathtt{a} s \mathtt{b})$ where $\alpha s \mathtt{a} s \mathtt{b}$ is a factor of square-free ${f_4}^k(\mathtt{a})$. Contradiction to Lemma 19 Case 1.

Case $\alpha_n v_0 u_1 = g_{12}(\mathtt{b})$**:** So ww is a factor of $g_{12}(\alpha s \mathtt{b} s \beta)$ where s is not empty and $\alpha, \beta \in \Sigma_5$. If $|v_0| > 6$ then v_1 is a prefix of $g_{12}(\mathtt{a})$ and β is $\mathtt{a}$. So the last letter of s is a possible letter before $\mathtt{a}$ and $\mathtt{b}$; (see set D) so s has $\mathtt{c}$ as a suffix. Similarly the first letter of s is a possible letter after $\mathtt{cb}$ so it's $\mathtt{e}$. Therefore α is $\mathtt{a}$, however, $\mathtt{a}s\mathtt{b}s\mathtt{a}$ is not a factor of square-free $f_4^k(\mathtt{a})$, by Lemma 19 Case 3.

If $|v_0| \leq 6$ then u_0 is a suffix of $g_{12}(\mathtt{d})$, very similar to the proof above, we deduce that α is $\mathtt{d}$ and β is $\mathtt{d}$. However, $\mathtt{d}s\mathtt{b}s\mathtt{d}$ is not a factor of square-free ${f_4}^k(\mathtt{a})$, by Lemma 19 Case 2.

Case $\alpha_n v_0 u_1 = g_{12}(\mathtt{c})$**:** So ww is a factor of $g_{12}(\alpha s \mathtt{c} s \beta)$ where s is not empty and $\alpha, \beta \in \Sigma_5$. If the letter after $\mathtt{c}$ is $\mathtt{a}$ consequently α is bound to be letter $\mathtt{c}$. Then

$\mathtt{ca}\cdots\mathtt{ca}\cdots$ is a square and contradiction to square-freeness of ${f_4}^k(\mathtt{a})$. So the only possible letter after $\mathtt{c}$ is $\mathtt{b}$ and considering that the only possible letter before $\mathtt{cb}$ is $\mathtt{d}$, and β is a letter after $\mathtt{d}$ we deduce that β is $\mathtt{c}$. Then $\mathtt{b}\cdots\mathtt{dcb}\cdots\mathtt{dc}$ is a square and contradiction to square-freeness of ${f_4}^k(\mathtt{a})$.

Case $\alpha_n v_0 u_1 = g_{12}(\mathtt{d})$**:** So ww is a factor of $g_{12}(\alpha s\mathtt{d}s\beta)$ where s is not empty and $\alpha, \beta \in \Sigma_5$. The only possible letter after $\mathtt{d}$ is $\mathtt{c}$, so s has $\mathtt{c}$ as a prefix. In addition α occurs before s, so α must be $\mathtt{b}$. However, $\mathtt{b}s\mathtt{d}s\beta$ is not a factor of square-free ${f_4}^k(\mathtt{a})$, by Lemma 19 Case 4.

Case $\alpha_n v_0 u_1 = g_{12}(\mathtt{e})$**:** So ww is a factor of $g_{12}(\alpha s\mathtt{e}s\beta)$ where s is not empty and $\alpha, \beta \in \Sigma_5$. if the letter after $\mathtt{e}$ is $\mathtt{b}$, then s has $\mathtt{b}$ as a prefix. Thus α is $\mathtt{c}$, however, $\mathtt{c}s\mathtt{e}s\beta$ is not a factor of square-free $f_4^k(\mathtt{a})$, by Lemma 19 Case 5.

If the letter after $\mathtt{e}$ is $\mathtt{d}$, then s has $\mathtt{d}$ as a prefix. So α is $\mathtt{a}$, however, $\mathtt{a}s\mathtt{e}s\beta$ is not a factor of square-free ${f_4}^k(\mathtt{a})$, by Lemma 19 Case 6.

If ww contains odd number of $\mathtt{101001}$, it means one occurrence of $\mathtt{101001}$ is overlapping between the two ws. Note that $\mathtt{1010}$ also occurs in $g_{12}({f_4}^k(\mathtt{a}))$ as a prefix of the codewords only, thus $\mathtt{1010}$ is also overlapping the junction between the two occurrences of w. We can easily deduce that this central occurrence of $\mathtt{101001}$ belongs to image of $\mathtt{a}$. Since no other codeword has a common suffix with other codewords having length at most 3 letters less than itself. So ww is a factor of $g_{12}(\mathtt{d}s\alpha\mathtt{a}s\beta)$ where s is not empty and $\alpha, \beta \in \Sigma_5$. Immediately we conclude that α is $\mathtt{c}$. However, β is a letter whose image has $g_{12}(\mathtt{c})1$ as a prefix, such letter does not exist.

Now, if ww contains at most three $\mathtt{101001}$ then ww is a factor of $g_{12}(s)$ where s is a factor of ${f_4}^k(\mathtt{a})$ and $|s| \leq 4$, it is computationally verifiable to confirm that images of all s factors of ${f_4}^k(a)$ and $|s| \leq 4$ do not contain a square that is not in the list. □

2 Binary words with Maximum Exponent 5/2

In this section we show the minimum number of squares drops from 12 if we relax the constraint on the maximal exponent and allow 5/2-powers in infinite binary words. Additionally the number of squares varies according to the number of maximal-exponent powers: if there is only one 5/2-power the minimum number of squares is 11, and if there are two 5/2-powers, it becomes 8. The next table shows that this number is minimum by giving the maximal length $\ell(s)$ of $5/2^+$-free binary

words that contain at most s squares, $0 \leq s \leq 7$.

s	0	1	2	3	4	5	6	7
$\ell(s)$	3	5	8	12	29	41	55	72

Theorem 21. *There exists a $5/2^+$-free infinite binary word with only two $5/2$-powers that contains no more than 8 squares.*

The proof is a consequence of Proposition 11 below, which states a property of the infinite word $\mathbf{g_{13}} = g_{13}(f_4^\infty(\mathtt{a}))$ where g_{13} is defined by:

$$\begin{aligned} g_{13}(\mathtt{a}) &= \mathtt{001100101},\\ g_{13}(\mathtt{b}) &= \mathtt{0011001011},\\ g_{13}(\mathtt{c}) &= \mathtt{001101},\\ g_{13}(\mathtt{d}) &= \mathtt{001101011},\\ g_{13}(\mathtt{e}) &= \mathtt{00110101100101}. \end{aligned}$$

Proposition 11. *The infinite word $\mathbf{g_{13}}$ contains no factor of exponent larger than $5/2$. It contains 8 squares $\{\mathtt{0}^2, \mathtt{1}^2, (\mathtt{01})^2, (\mathtt{10})^2, (\mathtt{0110})^2, (\mathtt{1001})^2, (\mathtt{011001})^2, (\mathtt{100110})^2\}$, and two $5/2$-powers $\mathtt{01010}$, $\mathtt{10101}$.*

Proof. The method we used to prove Proposition 10 is also valid for this proof, as we go through the cases for v_0u_1 we will see that the only case that is slightly different is when $v_0u_1 = g_{13}(\mathtt{b})$. Here, we change the boundary on the length of the common prefix, $|v_0| > 5$, the rest follows identically to the proof of Proposition 10. □

Continuing with the same constraint on the maximal exponent, we consider the situation when only one $5/2$-power is permitted. Then the number of squares is at least 11 as a result of the computation reported in the next table, which displays the maximal length $\ell(s)$ of $5/2^+$-free binary words that contain only one $5/2$ power and at most s squares, $0 \leq s \leq 10$.

s	0	1	2	3	4	5	6	7	8	9	10
$\ell(s)$	3	5	8	12	19	23	31	40	59	90	109

Theorem 22. *There exists a $5/2^+$-free infinite binary word with only one $5/2$-power that contains no more than 11 squares.*

For the proof, corollary of Proposition 12, we consider the morphisms f (see Chapter 3) and g_{14}. The morphism f is defined from Σ_3 to itself by

$$\begin{aligned} f(\mathtt{a}) &= \mathtt{abc},\\ f(\mathtt{b}) &= \mathtt{ac},\\ f(\mathtt{c}) &= \mathtt{b}. \end{aligned}$$

It is known that this morphism is weakly square-free (see [47, Chapter 2]).

Here, we translate $f^\infty(\mathtt{a})$ to binary using the second morphism g_{14} from Σ_3^* to B^* defined by:

$$\begin{aligned} g_{14}(\mathtt{a}) &= \mathtt{1001001101011001101001011001001101100} \\ &\quad\ \mathtt{1011010011011001001101001011001101 01}, \\ g_{14}(\mathtt{b}) &= \mathtt{100100110100101}, \\ g_{14}(\mathtt{c}) &= \mathtt{1001001101100101101001101}. \end{aligned}$$

and denote $\mathbf{g_{14}} = g_{14}(f^\infty(\mathtt{a}))$.

Proposition 12. *The infinite word* $\mathbf{g_{14}}$ *is* $5/2^+$*-free. It contains only* 11 *squares* $\{\mathtt{0}^2$, $\mathtt{1}^2$, $(\mathtt{01})^2$, $(\mathtt{10})^2$, $(\mathtt{001})^2$, $(\mathtt{010})^2$, $(\mathtt{011})^2$, $(\mathtt{100})^2$, $(\mathtt{101})^2$, $(\mathtt{110})^2$, $(\mathtt{0110})^2\}$*, and only one* $5/2$*-power,* $\mathtt{10101}$*.*

Proof. If there is a square ww in $\mathbf{g_{14}}$ and it's not one of the 11 squares listed above and each w contains at least 1 complete codeword we can write ww as:

$$\underbrace{u_0\overbrace{g_{14}(\alpha_1)\cdots g_{14}(\alpha_n)}v_0}\underbrace{u_1\overbrace{g_{14}(\alpha_1)\cdots g_{14}(\alpha_n)}v_1}$$

where $n \geq 1$ and $\alpha_i \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $1 \leq i \leq n$. And v_0u_1 is one of the codewords:

- If it is $g_{14}(\mathtt{a})$, note that the longest common prefix of $g_{14}(\mathtt{a})$ and other codewords has length 11 ($\mathtt{10010011010}$) and longest common suffix of $g_{14}(\mathtt{a})$ and other codewords has length 4 ($\mathtt{0101}$). And length of $g_{14}(\mathtt{a})$ is 73, thus, we would either have ww as a factor of $g_{14}(\mathtt{a}s\mathtt{a}s\alpha_{n+1})$ or $g_{14}(\alpha_{n+1}s\mathtt{a}s\mathtt{a})$. Either way it's a contradiction to square-freeness of $\mathbf{f}$.

- If v_0u_1 is $g_{14}(\mathtt{b})$ then using the longest common prefix and suffix with other codewords we must have $u_0 = \mathtt{10010011010}$ and $v_1 = \mathtt{0101}$, thus ww is

  $$\underbrace{\mathtt{0101}\overbrace{g_{14}(\alpha_1)\cdots g_{14}(\alpha_n)}\mathtt{10010011010}}\underbrace{\mathtt{0101}\overbrace{g_{14}(\alpha_1)\cdots g_{14}(\alpha_n)}\mathtt{10010011010}}$$

  which is a factor of $g_{14}(\mathtt{a}\alpha_1\cdots\alpha_n\mathtt{b}\alpha_1\cdots\alpha_n\mathtt{a})$ then α_1 and α_n are $\mathtt{c}$ but $\mathtt{cbc}$ is not a factor of $\mathbf{f}$.

- If v_0u_1 is $g_{14}(\mathtt{c}) = \mathtt{1001001101100101101001101}$ then using the longest common prefix and suffix with other codewords we would either have ww is a factor of $g_{14}(\mathtt{c}s\mathtt{c}s\alpha_{n+1})$ or $g_{14}(\alpha_{n+1}s\mathtt{c}s\mathtt{c})$. Either way it is a contradiction to square-freeness of $\mathbf{f}$.

If ww occurs in $\mathbf{g_{14}}$ and not each w contains at least 1 complete codeword then ww belongs to the list. Since the set of words with this property is bounded simple computation can confirm this. Furthermore, any 2^+-repetition contains a square therefore the proof of non existence of squares that are not in the set is also valid for 2^+-repetitions. □

3 Binary words with Maximum Exponent 3

In this section we deal with infinite binary words whose factors have maximal exponent 3. We first recall Fraenkel and Simpson theorem [36] that shows the existence of infinite binary word containing only 3 squares and 2 cubes. Chapter 3 is dedicated to this result and proofs by various morphisms. Next we show the number of squares increases to 4 if only one cube is allowed in the infinite word.

Theorem 23 ([36])**.** *There exists a 3^+-free infinite binary word with only two cubes that contains no more than 3 squares.*

The proof given in [3] and discussed in detail in Chapter 3. We build the infinite binary word by iterating the weakly square-free morphism f from $\mathtt{a}$ and translating the obtained word with the morphism g_1 from Σ_3 to B defined by:

$$g_1(\mathtt{a}) = \mathtt{01001110001101},$$
$$g_1(\mathtt{b}) = \mathtt{0011},$$
$$g_1(\mathtt{c}) = \mathtt{000111}.$$

The infinite word $\mathbf{g_1} = g_1(f^\infty(\mathtt{a}))$ contains only 3 squares $\mathtt{00}$, $\mathtt{11}$ and $\mathtt{1010}$. The cubes $\mathtt{000}$ and $\mathtt{111}$ are the only factors of exponent larger than 2 occurring in $\mathbf{g_1}$.

When only one cube is allowed to appear, the minimum number of squares becomes 4, the smallest possible value as shown by the computation reported in the next table. The table shows the maximal length $\ell(s)$ of 3^+-free binary words that contain only one cube and at most s squares, $0 \leq s \leq 3$.

s	0	1	2	3
$\ell(s)$	3	7	12	21

Theorem 24. *There exists a 3^+-free infinite binary word with only one cube that contains no more than 4 squares.*

Proof. Immediate consequence of Proposition 13, relies on the next result stated as a Lemma 8 by Ochem in [50]. □

The lemma is used to prove that the morphism g_{15} defined by:

$$g_{15}(\mathtt{a}) = \mathtt{1100010110010100},$$
$$g_{15}(\mathtt{b}) = \mathtt{1101000110010100},$$
$$g_{15}(\mathtt{c}) = \mathtt{0110101100010100},$$

produces a 3^+-free binary word from any $7/4^+$-free word on Σ_3.

Proposition 13. *The infinite word* $\mathbf{g_{15}} = g_{15}(w)$*, where* w *is any infinite* $7/4^+$*-free ternary word, is* 3^+*-free and contains* 4 *squares* $\{00, 11, 0101, 1010\}$*, one cube* 000*.*

Proof. Using Lemma 8, if $\beta = 1.99$, $n = 3$ and $\alpha = 7/4$ then it's sufficient to look at $g_{15}(t)$ for all $t \leq 16$ to verify that $\mathbf{g_{15}}$ is $(1.99^+, 3)$-free, consequently, it does not contain any square that is not in the list. Furthermore, the cubes that we could have as a factor of $\mathbf{g_{15}}$ are: $000, 111, 010101, 101010$ because of their period length they must be a factor of $g_{15}(w')$ for $|w'| \leq 2$ this could simply be done by computation, deducing the existence of 000 only. □

4 Conclusion

In this chapter we analysed the behaviour of infinite binary words under a constraint on their maximal exponent. The study revealed the minimum number of squares contained in these words reduces as the maximal exponent increases.

This chapter showed the key thresholds are 7/3, 5/2 and 3. As each case was studied the concluding remark was made that if one repetition of maximal exponent is permitted, then the number of squares are more than when 2 maximal exponent repetitions are permitted. The summary of results for each case is shown in the next table:

Maximal exponent e	Allowed number of e-powers	Minimum number of squares
7/3	2	12
	1	14
5/2	2	8
	1	11
3	2	3
	1	4

The fact that these thresholds are the only key numbers to be studied is a direct consequence of a simple computation. A $7/3^+$-free word is 5/2-free, however 5/2-free

word may contain many e-powers with $e > 7/3$. Computation shows that if the infinite word is e-free, where e is between $7/3^+$ and $5/2$, then the word contains at least 12 squares. Similarly, if the infinite word is e-free, where e is between $5/2^+$ and 3, then the word contains at least 8 squares.

6

Characterising binary words with few squares

The concept of avoidable patterns was introduced by Bean, Ehrenfeuch and McNulty [11] and independently by Zimin [67]. A pattern is a finite word over the alphabet of capital letters $\{A, B, ...\}$. An occurrence of a pattern is obtained by replacing each alphabet letter with a non-empty word. For example, the word 0111010011 is an occurrence of the pattern $ABBA$ where $A \to 011$ and $B \to 10$; it also contains another occurrence of this pattern (i.e. 1001) as a factor. A word avoids a pattern P if it contains no occurrence of P as a factor. The avoidability index $\lambda(P)$ of the pattern P is the smallest alphabet size over which an infinite word avoiding P exists. Patterns such as $A, ABC, ABA, ABACBA$ cannot be avoided with any finite alphabet. These patterns are said to be unavoidable, denoted as $\lambda(P) = \infty$, and have been characterised by Zimin [67].

A pattern, P is said to be k-avoidable if there exists an infinite word on k letters avoiding P. Thue [65, 66] showed that AA is 2-unavoidable but 3-avoidable, and A^β for $\beta > 2$ is 2-avoidable. Schmidt [62] proved that every binary pattern of length at least 13 is 2-avoidable. Later, Roth [58] refined the result by showing that every binary pattern of length at least 6 is 2-avoidable.

Thereafter, remained a finite set of patterns of length at most 5 to be studied. Cassaigne [21] completed this study by considering all the patterns in this set:

- 2-unavoidable patterns : $\epsilon, A, AA, AB, AAB, ABA, AABA, ABBA, AABB, ABAB, AABAA, AABAB$;
- 2-avoidable patterns: $AAA, ABAAB, AABBA, ABABA$.

To prove a pattern is unavoidable it is sufficient to compute the longest word avoiding the pattern.

Given a finite set $\mathcal{P}$ of patterns and a finite set $\mathcal{F}$ of words over Σ_k, we say that $\mathcal{P} \cup \mathcal{F}$ *characterises* a morphic word $w \in \Sigma_k^*$ if and only if every recurrent factor of an infinite word avoiding $\mathcal{P} \cup \mathcal{F}$ is a factor of w.

There is still no characterisation of k-unavoidable patterns, that is, patterns that are unavoidable over a k-letter alphabet. Thue [14, 65, 66] gave the characterisation of overlap-free binary words: $\{ABABA\} \cup \{000, 111\}$ characterises the fixed point of the morphism $0 \mapsto 01$, $1 \mapsto 10$. Thue-Morse word also avoids AAA. Roth [58] proved the pattern $ABAAB$ to be avoided in $h_{\mathrm{r}}(f^{\infty}(\mathtt{0}))$ where f and h_{r} are defined by:

$$\begin{aligned} f(\mathtt{0}) &= \mathtt{012}, & h_{\mathrm{r}}(\mathtt{0}) &= \mathtt{000}, \\ f(\mathtt{1}) &= \mathtt{02}, & h_{\mathrm{r}}(\mathtt{1}) &= \mathtt{111}, \\ f(\mathtt{2}) &= \mathtt{1}. & h_{\mathrm{r}}(\mathtt{2}) &= \mathtt{010101}. \end{aligned}$$

Finally, Cassaigne [21] proved the only remaining pattern, $AABBA$ to be avoided in $h_{\mathrm{c}}(f^{\infty}(\mathtt{0}))$ where h_{c} is defined by:

$$\begin{aligned} h_{\mathrm{c}}(\mathtt{0}) &= \mathtt{00}, \\ h_{\mathrm{c}}(\mathtt{1}) &= \mathtt{010}, \\ h_{\mathrm{c}}(\mathtt{2}) &= \mathtt{0111}. \end{aligned}$$

Although the avoidability of binary patterns on binary words is complete, Samsonov and Shur [61] started a variation of this study on cube-free binary words. As mentioned above, the pattern $ABABA$ is avoided by the Thue-Morse word. This is the only pattern of length at most 5 which is avoidable by cube-free words. Here is the list of all eight cube-free patterns of length 6, excluding equivalent patterns by reversal and negation: $\{AABAAB, AABABA, AABABB, AABBAA, AABBAB, ABAABA, ABABBA, ABBAAB\}$

The first two patterns $AABAAB$ and $AABABA$ are obviously avoided by the Thue-Morse word. Samsonov and Shur show that patterns $AABBAB, ABAABA$, $ABABBA$, and $ABBAAB$ are not avoidable by binary cube-free words. They also show the pattern $AABBAA$ is avoidable by the binary cube-free word generated by iterating the following morphism.

$$\begin{aligned} f_{\mathrm{ss}}(\mathtt{0}) &= \mathtt{001}, \\ f_{\mathrm{ss}}(\mathtt{1}) &= \mathtt{011}. \end{aligned}$$

The only pattern with unclear avoidability status is $AABABB$; it is conjectured to be avoidable by cube-free words in the same article [61], but have not yet been proven.

Cassaigne [22] partitioned all ternary patterns to 2-unavoidable, 2-avoidable and unclear status. One of the patterns with unclear status, $ABCBABC$ was proved

by Ilie et al. [39] to be 2-avoidable, and the remaining cases were proved to be also 2-avoidable by Ochem [50].

Concerning ternary square-free words, Thue proved that

- $\{AA\} \cup \{010, 212\}$ characterises the fixed point of f (Chapter 2),
- $\{AA\} \cup \{010, 020\}$ characterises the morphic word $T_1(f_{\mathrm{T}}^{\infty}(0))$,
- $\{AA\} \cup \{121, 212\}$ characterises the morphic word $T_2(f_{\mathrm{T}}^{\infty}(0))$,

where the morphisms f_{T}, T_1, and T_2 are given below.

$$
\begin{array}{lll}
f_{\mathrm{T}}(1) = \mathtt{0432}, & T_1(0) = \mathtt{01210212}, & T_2(0) = \mathtt{021012}, \\
f_{\mathrm{T}}(2) = \mathtt{0134}, & T_1(1) = \mathtt{01210120212}, & T_2(1) = \mathtt{02102012}, \\
f_{\mathrm{T}}(3) = \mathtt{013432}, & T_1(2) = \mathtt{01210212021}, & T_2(2) = \mathtt{02101201}, \\
f_{\mathrm{T}}(4) = \mathtt{0434}. & T_1(3) = \mathtt{012102120210120212}, & T_2(3) = \mathtt{0210120102012}, \\
 & T_1(4) = \mathtt{0121012021}. & T_2(4) = \mathtt{0210201}.
\end{array}
$$

To obtain the last two results, Thue first proved that $\{AA\} \cup \{02, 03, 10, 14, 21, 23, 24, 30, 31, 41, 42, 040, 132, 404, 1201, 2012\}$ characterises $f_T^{\infty}(0)$.

Another characterisation has been obtained by Ochem [51]: $\{AABBCABBA\} \cup \{0011, 1100\}$ characterises $h_{\mathrm{o}}(f^{\infty}(0))$, where h_{o} is given below.

$$
\begin{array}{l}
h_{\mathrm{o}}(0) = \mathtt{0010110111011101001}, \\
h_{\mathrm{o}}(1) = \mathtt{00101101101001}, \\
h_{\mathrm{o}}(2) = \mathtt{00010}.
\end{array}
$$

Here, we prove such characterisations mostly for the binary words considered in [2] (Chapter 5) that contain one or two 2^+-repetitions and as few squares as possible. The results are summarised in the following table. We use the notation SQ_t for the pattern corresponding to squares of words of length at least t, that is, $SQ_1 = AA$, $SQ_2 = ABAB$, $SQ_3 = ABCABC$, and so on.

Maximal exponent e	(Number of e-powers, Minimum number of squares)	Avoided patterns and factors	
5/2	(2, 8)	$\{SQ_7\} \cup F_8$	Proposition 15
7/3	(1, 14)	$\{SQ_9\} \cup F_{14}$	Proposition 16
7/3	(2, 12)	$\{SQ_9\} \cup F_{12}$	Proposition 20
3	(2, 3)	$\{SQ_5\} \cup F_3$	Proposition 22
5/2	(1, 11)	$\{SQ_3\} \cup F_{11}$	Proposition 21

We also give, in Proposition 18, a characterisation of words avoiding the patterns $AABBCC$ (i.e. three consecutive squares), SQ_3, and a finite set of factors.

The proofs are obtained by computer using the technique described in the next section. An example of proof by hand is given for Proposition 15.

1 Proof technique

Two types of characterisations are obtained in this section.

- For a given morphism $f : \Sigma_k^* \to \Sigma_k^*$ and a finite set of factors $\mathcal{F}_p \subset \Sigma_k^*$, the pure morphic word $f^\infty(0)$ is characterised by $\{AA\} \cup \mathcal{F}_p$
- For a given morphism $g : \Sigma_k^* \to \Sigma_{k'}^*$ and a finite set of factors $\mathcal{F}_m \subset \Sigma_{k'}^*$, the morphic word $g(f^\infty(0))$ is characterised by $\{SQ_t\} \cup \mathcal{F}_m$.

Here, we omit the avoidability part, since for all cases this part has been proven in previous chapters; that the morphic word actually avoids the corresponding patterns and factors (Chapter 5). We now explain how to show the characterisation part, that every finite recurrent factor in an infinite word avoiding both the patterns and the factors is a factor of the morphic word.

1.1 Characterising a pure morphic word

We compute the set S_4 of binary words u such that there exists a word $pus \in \Sigma_k$ avoiding squares and F_p with $|u| = l$ and $|p| = |s| = 4l$, where $l \geq \max\{|f|, f \in \mathcal{F}_p\} \times \max\{|f(a)|, a \in \Sigma_k\}$. So S_4 contains the set S_3 of k-ary words of length l that are prolongable into an infinite word avoiding squares and $\mathcal{F}_p$. We also compute the set S_1 of factors of length l of a long enough prefix of $f^\infty(0)$. So S_1 is a subset of the set S_2 of all factors of length l of $f^\infty(0)$. Since we assume the avoidability part, we have $S_1 \subset S_2 \subset S_3 \subset S_4$.

"Long enough" means that S_1 and S_4 are actually identical, so that $S_1 = S_4$ and $S_2 = S_3$. This means that prolongable k-ary words avoiding squares and $\mathcal{F}_p$ are factors of the f-image of some k-ary word w, and moreover that w avoids squares and $\mathcal{F}_p$ too. Thus, prolongable k-ary words avoiding squares and $\mathcal{F}_p$ are factors of $f^\infty(0)$.

1.2 Characterising a morphic word

We assume the fact $\{AA\} \cup \mathcal{F}_p$ characterises $f^\infty(0)$ and we similarly prove that $\{SQ_t\} \cup \mathcal{F}_m$ characterises $g(f^\infty(0))$.

We compute the set S'_4 of binary words u such that there exists a word $pus \in \Sigma^*_{k'}$ avoiding $\{SQ_t\} \cup \mathcal{F}_m$ with $|u| = l$ and $|p| = |s| = 4l$, where $l \geq \max\{|f|, f \in \mathcal{F}_p\} \times \max\{|g(a)|, a \in \Sigma_{k'}\}$. So S'_4 contains the set S'_3 of binary words of length l that are prolongable into an infinite binary word avoiding $\{SQ_t\} \cup \mathcal{F}_m$. We have also computed the set S'_1 of factors of length l of a long enough prefix of $g(f^\infty(0))$. So S'_1 is a subset of the set S'_2 of factors of length l of $g(f^\infty(0))$.

Again, we notice that S'_1 and S'_4 are identical, which implies that $S'_2 = S'_3$. This means that prolongable words avoiding $\{SQ_t\} \cup \mathcal{F}_m$ are factors of the g-image of some k-ary word w that avoids $\{AA\} \cup \mathcal{F}_p$. We have to check that the g-images of small squares are forbidden if $t > \min\{|g(a)|, a \in \Sigma_{k'}\}$. Thus, prolongable k'-ary words avoiding $\{SQ_t\} \cup \mathcal{F}_m$ are factors of $g(f^\infty(0))$.

2 A 5-ary pure morphic word

The following morphism is a power of the morphism f_4 in Chapter 5, therefore the fixed point of f_4 can be labelled by: $\mathtt{a} \to 0$, $\mathtt{b} \to 3$, $\mathtt{c} \to 2$, $\mathtt{d} \to 1$ and $e \to 4$ to be the fixed point of the following morphism. We therefore use the same name f_4. Let $\mathbf{f_4}$ be $f_4^\infty(\mathtt{0})$ where f_4 is defined by

$$\begin{aligned} f_4(\mathtt{0}) &= \mathtt{012},\\ f_4(\mathtt{1}) &= \mathtt{34},\\ f_4(\mathtt{2}) &= \mathtt{32},\\ f_4(\mathtt{3}) &= \mathtt{04},\\ f_4(\mathtt{4}) &= \mathtt{3412}. \end{aligned}$$

Lemma 21. *The word* $\mathbf{f_4}$ *avoids squares and none of its factors belong to the following set:*

$$F = \{\mathtt{02}, \mathtt{03}, \mathtt{10}, \mathtt{13}, \mathtt{14}, \mathtt{21}, \mathtt{24}, \mathtt{30}, \mathtt{31}, \mathtt{40}, \mathtt{42}, \mathtt{041}, \mathtt{232}, \mathtt{323}, \mathtt{0120}, \mathtt{1201}\}.$$

Proposition 14. *If x is an infinite recurrent square-free 5-ary word, then x avoids F if and only if it has the same set of factors as* $\mathbf{f_4}$.

Proof. We exploit the technique explained in detail in subsection 1.1. □

2.1 Words containing two 5/2-repetitions and 8 squares

The following morphism has been defined and studied in Chapter 5, here we recall this morphism. Let g_{13} be the morphism from Σ_5^* to B^* defined by:

$$\begin{aligned} g_{13}(0) &= 100110010,\\ g_{13}(1) &= 100110101,\\ g_{13}(2) &= 100110,\\ g_{13}(3) &= 1001100101,\\ g_{13}(4) &= 10011010110010. \end{aligned}$$

and denote $\mathbf{g_{13}} = g_{13}({f_4}^{\infty}(0))$.

In Chapter 5 $\mathbf{g_{13}}$ is studied in depth and shown to avoid SQ_7. In addition, by a simple computation, it is verifiable that $\mathbf{g_{13}}$ also avoids the following set of factors:

$$\begin{aligned} F_{13} = \{&000, 111, 00100, 11011, 010010, 010101, 101010, 101101, 00110011,\\ &11001100, 0010110010, 1101001101\} \end{aligned}$$

Proposition 15. *Every infinite recurrent binary word G_{13} avoiding SQ_7 and F_{13} has the same set of factors as $\mathbf{g_{13}}$.*

The step by step proof of this proposition is given here, as an example to demonstrate how the proof technique explained in Subsection 1.2 for characterising a morphic word works.

Proof. As explained above $\mathbf{g_{13}}$ avoids SQ_7 and F_{13}.

Now, we prove the other direction of Proposition 15, that is, every factor of G_{13} is a factor of $\mathbf{g_{13}}$. First we check that every factor of G_{13} is a factor of $g_{13}(t)$, where t is a 5-ary word. We compute the set of factors of G_{13} of length $|g_{13}(4)| + |g_{13}(1)| = 24$ and remove the ones that are not prolongable in G_{13}. This set is equal to the set of all factors of $\mathbf{g_{13}}$ of length 24, where in this set every factor with prefix $g_{13}(i)$ for some $i \in \Sigma_5$ is followed by a factor $g_{13}(j)$. So, a factor of G_{13} is a factor of the g_{13}-image of a 5-ary word.

Let $L \subset \Sigma_5^*$ denote the language of words whose g_{13}-image is a factor of G_{13}. Since $g_{13}(2) = 100110$ is a common prefix of $g_{13}(s)$, $s \in \Sigma_5$, we note $g_{13}(s) = 100110r$ with $r \in \mathrm{B}^*$. Moreover, notice that $g_{13}(3) = g_{13}(0)1$ and $g_{13}(4) = g_{13}(1)10010$.

We assume that L contains a square uu for some $u \in \Sigma_5^+$, and $|g_{13}(uu)| \leq 12$ implying u must be 2. Now by prolongability, L contains $p22s$ and $g_{13}(22s) = 1001101001101r$ contains $1101001101 \in F_{13}$.

Thus L is square-free, and now we check that L cannot contain any element of the set F:

- L contains 02: by prolongability, L contains $02s$ for $s \in \Sigma_5$. $g_{13}(02s) = 100110010100110100110r$, but 0100110100 is not prolongable.
- L contains 03: $g_{13}(03) = g_{13}(0)g_{13}(0)1$, but $g_{13}(0)g_{13}(0)$ is a square with period 9.
- L contains 10: $g_{13}(10) = 100110101100110010$, but $11001100 \in F_{13}$.
- L contains 13: $g_{13}(13) = 1001101011001100101$, but $11001100 \in F_{13}$.
- L contains 14: $g_{13}(14) = g_{13}(1)g_{13}(1)10010$, but $g_{13}(1)g_{13}(1)$ is a square with period 9.
- L contains 21: $g_{13}(21) = 10011010011010$, but $1101001101 \in F_{13}$.
- L contains 24: $g_{13}(24) = 10011010011010110010$, but $1101001101 \in F_{13}$.
- L contains 30: $g_{13}(30s) = g_{13}(0)1g_{13}(0)10011r$, but $g_{13}(0)1g_{13}(0)1$ is a square with period 10.
- L contains 31: $g_{13}(31s) = 1001100101100110101100110r$, but $(010110011)^2$ is a square with period 9.
- L contains 40s: $g_{13}(40s) = 1001101011001010011001010011 0r$, but $(011001010)^2$ is a square with period 9.
- L contains 42s: $g_{13}(42s) = 1001100101001101011001010011010011 0r$ contains 0100110100 which is not prolongable: by simple computation the longest word with prefix 0100110100 avoiding SQ_7 and F_{13} has length 36.
- L contains 041: $g_{13}(041) = 100110010100110101100101001101 01$, but $(10010100110101)^2$ is a square with period 14.
- L contains 323: $g_{13}(323s) = g_{13}(32)g_{13}(32)r$, but $g_{13}(32)g_{13}(32)$ is a square with period 16.
- L contains 232: by prolongability, L contains $1232s$ for $s \in \Sigma_5$. $g_{13}(1232s) = 1001101011001101001100101100110100110r$, but $(0101100110100110)^2$ is a square with period 16.
- L contains 0120: by prolongability, L contains 201204. Now $g_{13}(201204) = g_{13}(201)g_{13}(201)10010$. But $g_{13}(201)g_{13}(201)$ is a square with period 24.
- L contains 1201: by prolongability, L contains 120123. Then $g_{13}(120123) = g_{13}(120)g_{13}(120)1$. But $g_{13}(120)g_{13}(120)$ is a square with period 24.

Therefore L is square-free and does not contain a factor in F, thus it has the same set of factors as $\mathbf{g_{13}}$ by Proposition 14. □

Here, we should mention that the morphism g_{13} is the composition of f_4 and the morphism g'_{13} from Σ_5^* to B^* is:

$$\begin{aligned} g'_{13}(0) &= 0110,\\ g'_{13}(1) &= 01,\\ g'_{13}(2) &= 010,\\ g'_{13}(3) &= 011,\\ g'_{13}(4) &= 010110. \end{aligned}$$

However, to prove Proposition 15, it is simpler to use the longer morphism g_{13}.

2.2 Words containing one 7/3-repetition and 14 squares

The following morphism has been defined and studied in Chapter 5. Let g_{12} be the morphism from Σ_5^* to B^* defined by:

$$\begin{aligned} g_{12}(0) &= 101001100101,\\ g_{12}(1) &= 1010011001001,\\ g_{12}(2) &= 101001011001,\\ g_{12}(3) &= 101001011001001,\\ g_{12}(4) &= 101001011001001100101, \end{aligned}$$

and denote $\mathbf{f_4} = {f_4}^{\infty}(0)$, $\mathbf{g_{12}} = g_{12}({f_4}^{\infty}(0))$.

In Chapter 5 $\mathbf{g_{12}}$ is studied in depth and shown to avoid SQ_9. In addition, by a simple computation, it is verifiable that $\mathbf{g_{12}}$ avoids the following set of minimal forbidden factors:

$$\begin{aligned} F_{12} = \{&000, 111, 11011, 010101, 101010, 0010010, 0100100, 00110011, 11001100,\\ &101001101, 101100101, 0100101101, 1100101100, 001001100100, 010011010011,\\ &0011001001100, 1011010010110011\} \end{aligned}$$

Proposition 16. *Every infinite recurrent binary word G_{12} avoiding SQ_9 and F_{12} has the same set of factors as $\mathbf{g_{12}}$.*

Proof. We exploit the technique explained in detail in Subsection 1.2. □

2.3 Words avoiding AABBCC

Ochem [50] proved that the pattern $AABBCC$, i.e. three consecutive squares, can be avoided over the binary alphabets. In particular, the proof shows that there exists exponentially many binary words avoiding both $AABBCC$ and SQ_3. We now show, that among such words, the word $\mathbf{g_{cs}}$ defined below admits a characterisation.

Let g_{cs} be the morphism from Σ_5^* to B* defined by:

$$\begin{aligned}
g_{cs}(\mathtt{0}) &= \mathtt{110100111001011000},\\
g_{cs}(\mathtt{1}) &= \mathtt{11010011100011010001011000011100101},\\
g_{cs}(\mathtt{2}) &= \mathtt{11010011100101},\\
g_{cs}(\mathtt{3}) &= \mathtt{11010011100011010001011000},\\
g_{cs}(\mathtt{4}) &= \mathtt{11010011100101100011010001011000011100101},
\end{aligned}$$

and denote $\mathbf{g_{cs}} = g_{cs}({f_4}^{\infty}(\mathtt{0}))$.

Where the morphism f_4 is weakly square-free defined and studied in Chapter 5.

Proposition 17. *The infinite word* $\mathbf{g_{cs}} = g_{cs}(f_4^{\infty}(\mathtt{a}))$ *contains only* 4 *squares* $\{\mathtt{0}^2$, $\mathtt{1}^2$, $(\mathtt{01})^2$, $(\mathtt{10})^2$. *It contains no pattern of the form* $AABBCC$.

Proof. The factor `11010011100` appears in $g_{cs}({f_4}^k(\mathtt{a}))$ only as a prefix of the codewords, therefore any factor starting and ending with `11010011100` is uniquely decipherable.

If there is a square ww that contains $2n$ occurrences of `11010011100`, where $n \geq 2$, and it is a factor of $g_{cs}({f_4}^k(\mathtt{a}))$ where ${f_4}^k(\mathtt{a})$ is square-free so the square ww can be written as:

$$\underbrace{u_0\overbrace{\alpha_1\cdots\alpha_n}v_0}\underbrace{u_1\overbrace{\alpha_1\cdots\alpha_n}v_1}$$

where $\alpha_1, ...\alpha_n$ are occurrences of `11010011100`, $n \geq 2$ and v_0u_1 contains no `11010011100` as a factor therefore $\alpha_n v_0 u_1$ is one of the codewords.

Similar to proof of Proposition 10 we study different cases of possible v_0u_1.

Case $\alpha_n v_0 u_1 = g_{cs}(\mathtt{a})$**:** So ww is a factor of $g_{cs}(\alpha s\mathtt{a}s\beta)$ where s is not empty so the letter before `a` is always `c` so ww is a factor of $g_{cs}(\alpha\cdots\mathtt{ca}\cdots\mathtt{c}\beta)$ therefore β is `b` thus ww is a factor of $g_{cs}(\alpha s\mathtt{a}s\mathtt{b})$ where$\alpha s\mathtt{a}s\mathtt{b}$ is a factor of square-free ${f_4}^k(a)$ this is Lemma 19 Case 1.

Case $\alpha_n v_0 u_1 = g_{cs}(\mathtt{b})$**:** If $|v_0| > 6$ then v_1 is a prefix of $g_{cs}(\mathtt{a})$ so ww is a factor of $g_{cs}(\alpha\cdots\mathtt{cb}\cdots\mathtt{ca})$ further it's a factor of ww is a factor of $g_{cs}(\alpha\mathtt{e}\cdots\mathtt{cbe}\cdots\mathtt{ca})$

therefore α is $\mathtt{a}$, we have $\mathtt{a}s\mathtt{b}s\mathtt{a}$ factor of square-free $f_4^k(a)$, Lemma 19 Case 3. If $|v_0| < 6$ then u_0 is a suffix of $g_{cs}(\mathtt{d}$ therefore ww is a factor of $g_{cs}(\mathtt{dc}\cdots\mathtt{bc}\cdots\alpha)$ further more it's a factor of $g_{cs}(\mathtt{dc}\cdots\mathtt{ebc}\cdots\mathtt{e}\alpha)$ therefore α is $\mathtt{d}$ now we have $\mathtt{d}s\mathtt{b}s\mathtt{d}$ is a factor of square-free ${f_4}^k(a)$ this is Lemma 19 Case 2.

Case $\alpha_n v_0 u_1 = g_{cs}(\mathtt{c})$**:** So ww is a factor of $g_{cs}(\alpha s\mathtt{c}s\beta)$ if the letter after $\mathtt{c}$ is $\mathtt{a}$ we have $\alpha\mathtt{a}\cdots\mathtt{ca}\cdots\beta$ so α is $\mathtt{c}$ then $\mathtt{ca}\cdots\mathtt{ca}\cdots$ is a square so the letter after $\mathtt{c}$ is $\mathtt{b}$ and as a consequence the only letter before $\mathtt{cb}$ is $\mathtt{d}$. Therefore β is a letter after $\mathtt{d}$ that can be only $\mathtt{c}$ so $\mathtt{b}\cdots\mathtt{dcb}\cdots\mathtt{dc}$ is a square.

Case $\alpha_n v_0 u_1 = g_{cs}(\mathtt{d})$**:** So ww is a factor of $g_{cs}(\alpha s\mathtt{d}s\beta)$ if the letter after $\mathtt{d}$ is $\mathtt{c}$ we have$\alpha\mathtt{c}\cdots\mathtt{dc}\cdots\beta$ so α is $\mathtt{b}$ we have $\mathtt{b}s\mathtt{d}s\beta$ is a factor of square-free ${f_4}^k(a)$ this is Lemma 19 Case 4.

Case $\alpha_n v_0 u_1 = g_{cs}(\mathtt{e})$**:** So ww is a factor of $g_{cs}(\alpha s\mathtt{e}s\beta)$ if the letter after $\mathtt{e}$ is $\mathtt{b}$ we have $\alpha\mathtt{b}\cdots\mathtt{eb}\cdots\beta$ so α is $\mathtt{c}$ then $\mathtt{c}s\mathtt{e}s\beta$ is a factor of square-free $f_4^k(a)$ this is Lemma 19 Case 5. If the letter after $\mathtt{e}$ is $\mathtt{d}$ we have $\alpha\mathtt{dc}\cdots\mathtt{edc}\cdots\beta$ so α is $\mathtt{a}$ then $\mathtt{a}s\mathtt{e}s\beta$ is a factor of square-free ${f_4}^k(a)$ this is Lemma 19 case 6.

If ww contains odd number of $\mathtt{11010011100}$, it means one occurrence of $\mathtt{11010011100}$ is overlapping between the two ws. Note that $\mathtt{1010}$ also occurs in $g_{12}({f_4}^k(\mathtt{a}))$ as a prefix of the codewords only, thus $\mathtt{1010}$ is also overlapping the junction between the two occurrences of w. We can easily deduce that this central occurrence of $\mathtt{101001}$ belongs to image of $\mathtt{a}$. Since no other codeword has a common suffix with other codewords having length at most 3 letters less than itself. So ww is a factor of $g_{12}(\mathtt{d}s\alpha\mathtt{a}s\beta)$ where s is not empty and $\alpha, \beta \in \Sigma_5$. Immediately we conclude that α is $\mathtt{c}$. However, β is a letter whose image has $g_{12}(\mathtt{c})1$ as a prefix, such letter does not exist.

Now if ww contains at most three $\mathtt{11010011100}$ then ww is a factor of $g_{cs}(s)$ where s is a factor of ${f_4}^k(a)$ and $|s| \leq 4$, it is computationally verifiable to confirm that images of all s factors of ${f_4}^k(a)$ and $|s| \leq 4$ do not contain a square that is not in the list. □

In addition by a simple computation, it's verifiable that $\mathbf{g_{cs}}$ avoids the following set of minimal forbidden factors:

$$F_{cs} = \{\mathtt{0000}, \mathtt{1111}, \mathtt{01010}, \mathtt{10101}, \mathtt{011001}, \mathtt{100110}, \mathtt{0011101}, \mathtt{1011100}, \mathtt{1100010}, \mathtt{00010111}, \mathtt{11101000}, \mathtt{0001110010110}, \mathtt{0110100111000}, \mathtt{1001011000111}, \mathtt{1110001101001}\}$$

Proposition 18. *Every infinite recurrent binary word G_{cs} avoiding SQ_3 and F_{cs} has the same set of factors as* $\mathbf{g_{cs}}$.

Proof. We exploit the technique explained in detail in Subsection 1.2. □

3 A 6-ary pure morphic word

The following morphism has been defined and studied in Chapter 5. Let $\mathbf{f_1}$ be $f_1^\infty(0)$ where f_1 is defined from Σ_6^* to itself by:

$$\begin{aligned} f_1(0) &= 0102,\\ f_1(1) &= 1013,\\ f_1(2) &= 40135,\\ f_1(3) &= 51024,\\ f_1(4) &= 1024,\\ f_1(5) &= 0135. \end{aligned}$$

Then $\mathbf{f_1}$ avoids squares (see Chapter 5 for the proof and properties of the fixed point of this morphism) and the following set of minimal forbidden factors:

$$\begin{aligned} F_1 = \{&03, 04, 05, 12, 14, 15, 20, 23, 25, 31, 32, 34, 41, 42, 43, 45, 50, 52, 53, 54, 213,\\ &302, 402, 513, 40130, 51021, 01024010, 10135101\} \end{aligned}$$

Proposition 19. *If y is an infinite recurrent square-free 6-ary word, then y avoids F_1 if and only if y has the same set of factors as* $\mathbf{f_1}$.

Proof. We exploit the technique explained in detail in Subsection 1.1. □

3.1 Words containing two 7/3 repetitions and 12 squares

The following morphism has been defined and studied in Chapter 5. Let morphism g_7 from Σ_6^* to B^* is defined by

$$\begin{aligned} g_7(0) &= 10011,\\ g_7(1) &= 01100,\\ g_7(2) &= 01001,\\ g_7(3) &= 10110,\\ g_7(4) &= 0110,\\ g_7(5) &= 1001. \end{aligned}$$

and denote $\mathbf{g_7} = g_7({f_1}^{\infty}(0))$.
See Chapter 5 for properties of $\mathbf{g_7}$. Furthermore, the word $\mathbf{g_7}$ avoids SQ_9 and the following set of minimal forbidden factors:

$$F_7 = \{000, 111, 01010, 10101, 001100, 110011, 0010010, 0100100, 1011011, 1101101, 0011010011, 0101100101, 1010011010, 1100101100, 01001011010010\}$$

Proposition 20. *Every infinite recurrent binary word G_7 avoiding SQ_9 and F_7 has the same set of factors as $\mathbf{g_7}$.*

Proof. Once again, we exploit the technique explained in detail in Subsection 1.2. □

4 Thue's ternary pure morphic word

Thue [14, 65, 66] proved that $\{AA\} \cup \{010, 212\}$ characterises the fixed point of

$$\begin{aligned} f(0) &= 012, \\ f(1) &= 02, \\ f(2) &= 1. \end{aligned}$$

In this section, we give characterisations of two words that are morphic images of $f^{\infty}(0)$.

It is not surprising that this word appears in the context of characterisations: as soon as a morphism m is such that $m(0) = 0x1$ and $m(1) = 01$, the m-image of words of the form $0u1u0$, $u \in \Sigma_3^*$, contains a large square: $m(0u1u0) = 0x1m(u)01m(u)0x1$ contains $(1m(u)0)^2$. Moreover, a ternary square-free word avoids factors of the form $0u1u0$ with $u \in \Sigma_3^*$, if and only if it avoids $\{010, 212\}$ [51].

4.1 Words containing one 5/2-repetition and 11 squares

The following morphism has been defined and studied in Chapter 5. Let g_{14} be the morphism from Σ_3^* to B^* defined by

$$\begin{aligned} g_{14}(0) &= 10010011010110011010010110010011011001011001101101, \\ g_{14}(1) &= 100100110100101, \\ g_{14}(2) &= 10010011011001011010011011. \end{aligned}$$

and denote $\mathbf{g_{14}} = g_{14}(f^{\infty}(0))$.
See Chapter 5 for properties of $\mathbf{g_{14}}$.and the fact that $\mathbf{g_{14}}$ avoids SQ_5. Furthermore, it is verifiable that $\mathbf{g_{14}}$ also avoids the following set of minimal forbidden factors:

$$F_{14} = \{000, 111, 01010, 001100, 0010010, 0100100, 1011011, 1101101\}$$

Proposition 21. *Every infinite recurrent binary word G_{14} avoiding SQ_5 and F_{14} has the same set of factors as* $\mathbf{g_{14}}$.

Proof. We exploit the technique explained in detail in Subsection 1.2. □

4.2 Words containing 3 squares

The following morphism has been defined and studied in Chapter 3, here we recall this morphism. Let the morphism g_1 from Σ_3^* to B^* defined by

$$\begin{aligned} g_1(0) &= 01001110001101, \\ g_1(1) &= 0011, \\ g_1(2) &= 000111. \end{aligned}$$

and denote $\mathbf{g_1} = g_1(f^\infty(0))$.
See Chapter 3 for properties of $\mathbf{g_1}$ and the fact that $\mathbf{g_1}$ avoids SQ_3 . Furthermore, by a simple computation it is verifiable that $\mathbf{g_1}$ avoids the following set of minimal forbidden factors:

$$F_1 = \{0000, 0010, 0101, 1111, 01000110, 10011101, 1001101000, 1110100110\}$$

Proposition 22. *Every infinite recurrent binary word G_1 avoiding SQ_3 and F_1 has the same set of factors as* $\mathbf{g_1}$.

Proof. We exploit the technique explained in detail in Subsection 1.2. □

5 Conclusion

Some of the morphic words characterised in this chapter are the morphic image of a same pure morphic word. We understand why the fixed point of f appears often in this context (see Section 4). We also know why Thue's words avoiding $\{AA\} \cup \{010, 020\}$ and $\{AA\} \cup \{121, 212\}$ use the same pure morphic word: the latter is obtained from the former by deleting the letter immediately after each occurrence of the letter 0. On the other hand, we don't yet know why the 5-ary pure morphic word of Section 2 is so useful.

Notice that a (pure) morphic word might be characterised in more than one way. For example, the word $\mathbf{g_{13}}$ in Section 2.1 is characterised by $\{SQ_7\} \cup \{000, 111,$ $00100, 11011, 010010, 010101, 101010, 101101, 00110011, 11001100, 0010110010,$ $1101001101\}$ and can be equivalently characterised by $\{SQ_7, AAA, AABBAABB\} \cup$ $\{00100, 11011, 010010, 101101, 0010110010, 1101001101\}$. It is also easy to check that

$\{AA\} \cup \{\texttt{010,212}\}$ characterises the same ternary word as $\{AA\} \cup \{\texttt{1021,1201}\}$.

An interesting open question is the following: suppose that P is an avoidable pattern with avoidability index $\lambda(P) = k$. Is it possible to find a finite set $\mathcal{P}$ of patterns and a finite set $\mathcal{F}$ of factors such that $P \in \mathcal{P}$ and $\mathcal{P} \cup \mathcal{F}$ characterises a morphic word over Σ_k ? Notice that this would be a strengthening of Cassaigne's conjecture [21] that there exists a morphic word avoiding P over Σ_k.

In particular, we now know that such characterisations exists for the patterns AA, $ABCABC$, $AABBCC$, and $AABBCABBA$. In all these characterisations, we essentially avoid large squares and some factors. We have indeed checked that $\{AABBCABBA\} \cup \{0011, 1100\}$ characterises the same binary word as $\{SQ_5\} \cup \{0000, 0011, 1100, 1111, 01010, 10101, 010111, 101000, 0001001, 1110110, 00100100, 01011010, 10100101, 11011011, 0110111010, 1001000101\}$

It would be interesting to obtain characterisations that cannot be expressed as $\{SQ_t\} \cup \mathcal{F}$.

7

Computing the maximal-exponent repeats of an overlap-free string in linear time

We consider the question of computing the maximal exponent of factors (substrings) of a given string. Repeats considered in this chapter are strings of exponent at most 2. They refer to strings of the form uvu where u is its longest border (both a prefix and a suffix). The study of repeats in a string is to do with long-distance interactions between separated occurrences of the same segment (the u part) in the string. Although occurrences may be far away from each others, they may interact when the string is folded as it is the case for genomic sequences.

The exponent of a string can be calculated in linear time using basic string matching that computes the smallest period associated with the longest border of the string (see [25]). A naive consequence provides a $O(n^3)$-time solution to compute the maximal exponent of all factors of a string of length n since there are potentially of the order of n^2 factors. However, a quadratic time solution is also a simple application of basic string matching. In contrast, our solution runs in linear time on a fixed-size alphabet.

When a string contains runs, that is, maximal occurrences of repetitions of exponent at least 2, computing their maximal exponent can be done in linear time by adapting the algorithm of Kolpakov and Kucherov [44] that computes all the runs occurring in the string. Their result relies on the fact that there exists a linear number of runs in a string [44] (see [60, 27] for precise bounds). However, this does not apply to square-free strings.

The solution presented in this chapter works on overlap-free strings for which the maximal exponent of factors is at most 2. Thus, we are looking for factors w of the form uvu, called repeats, where u is the longest border of w. In order to achieve our goal, we exploit two main tools: a factorisation of the string and the Suffix Automaton of some factors.

The Suffix Automaton is used to search for maximal repeats in a product of two

strings due to its ability to locate occurrences of all factors of a pattern. Here, we enhance the automaton to report the right-most occurrences of those factors. Using it alone in a balanced divide-and-conquer manner produces a $O(n \log n)$-time algorithm. To eliminate the log factor, we additionally use the f-factorisation of the string. It has now become common to employ this factorisation in order to derive efficient or even optimal algorithms. The f-factorisation (see [25]), a type of LZ77 factorisation fit for string algorithms, allows us to skip larger and larger parts of the strings during an online computation. The factorisation can be computed in $O(n \log a)$-time using a Suffix Tree or a Suffix Automaton, where a is alphabet size, but also in linear time on an integer alphabet using a Suffix Array [28].

The running time of the proposed algorithm depends additionally on the repetitive threshold of the underlying alphabet of the string. The threshold restricts the context of the search for a second occurrence of u associated with a repeat uvu.

We show a very surprising property of repeats whose exponent is maximal in an overlap-free string: there are no more than a linear number of occurrences of them, although the number of occurrences of maximal (i.e. non extensible occurrences of) repeats can be quadratic. As a consequence, the algorithm can be upgraded to output all occurrences of maximal-exponent repeats of an overlap-free string in linear time.

The question would have had a simple solution by computing MinGap on each internal node of the Suffix Tree of the input string. MinGap of a node is the smallest difference between the positions assigned to leaves of the subtree rooted at the node. Unfortunately, the best algorithms for MinGap computation, equivalent to MaxGap computation, run in time $O(n \log n)$ (see [12, 40, 16] and the discussion in [23]).

A remaining question to the present study is to unify the algorithmic approaches for repetitions of exponent at least 2 and for repeats of exponent at most 2.

The plan of this chapter is as follows. After defining the problem in the next section we present the general scheme of the algorithm that relies on the f-factorisation of the input string in Section 2. The sub-function operating a Suffix Automaton is described in Section 3 and the complexity of the whole algorithm is studied in Section 4. In Section 5 we count occurrences of maximal-exponent repeats followed by a conclusion in Section 6.

1 Maximal-exponent repeats

We consider a fixed overlap-free string y of length n and deal with the repeats occurring within y. A repeat w in y is a factor of the form uvu. We often consider the decomposition uvu for which u is the longest border of w (longest factor that is both a prefix and a suffix of w). Then $\text{period}(w) = |uv|$ and $\exp(w) = |uvu|/|uv| =$

$1 + |u|/\text{period}(w)$. By convention, in the following we allow a border-free factor to be considered as a repeat of exponent 1, though this is not a repeat in the common sense since the repeating element u is empty.

A repeat in y is said to be a **maximal-exponent repeat**, an MER for short, if its exponent is maximal among all repeats occurring in y. An occurrence of a repeat is said to be a maximal, a **maximal repeat** for short and abuse of terms, if it cannot be extended to the left nor to the right with the same period. Note an occurrence of an MER is a maximal repeat but the converse is obviously false.

2 Computing the maximal exponent of repeats

The core result of this chapter is an algorithm, MaxExpRep, that computes the maximal exponent of factors of the overlap-free string y. The algorithm has to look for factors that are repeats of the form uvu, for two strings u and v, u being the longest border of uvu. The aim of this algorithm is accomplished with the help of Algorithm MaxExp, designed in the next section, which detects those repeats occurring within the concatenation of two strings.

Algorithm MaxExpRep relies on the f-factorisation of y (see [25]), a type of LZ77 factorisation [68] defined as follows. It is a sequence of non-empty strings, z_1, z_2, ..., z_k, called phrases satisfying $y = z_1 z_2 \cdots z_k$ and where z_i is the longest prefix of $z_i z_{i+1} \cdots z_k$ occurring in $z_1 z_2 \cdots z_{i-1}$. When this longest prefix is empty, z_i is the first letter of $z_i z_{i+1} \cdots z_k$, thus it is a letter that does not occur previously in y. We adapt the factorisation to the purpose of our problem by defining z_1 as the longest prefix of y in which no letter occurs more than once. Then, $|z_1| \leq a$ and $\text{MaxExpRep}(z_1) = 1$. Note that $\text{MaxExpRep}(z_1 z_2) > 1$ if $z_1 \neq y$.

When the factorisation of y is computed, Algorithm MaxExpRep processes the phrases sequentially, from z_2 to z_k. After z_1, z_2, ..., z_{i-1} have been processed, the variable e stores the maximal exponent of factors of $z_1 z_2 \cdots z_{i-1}$. Then, the next repeats to be considered are those involving phrase z_i. Such a repeat uvu can either be internal to z_i or involve other phrases. However, the crucial property of the factorisation is that the second occurrence of u is only to be searched for in $z_{i-1} z_i$ because it cannot contain a phrase as this would contradict the definition of the factorisation. The reason is that if a phrase is a proper factor of second occurrence of u then since there is another occurrence of u prior to the starting point of this phrase, this implied that the phrase is extendible which is contradiction to the definition of factorisation.

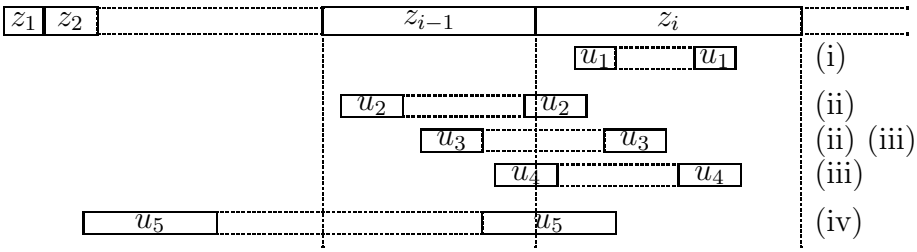


Figure 7.1: The only four possible locations of a repeat uvu involving phrase z_i of the factorisation of the string: (i) internal to z_i; (ii) the first occurrence of u is internal to z_{i-1}; (iii) the second occurrence of u is internal to z_i; (iv) the second occurrence of u is internal to $z_{i-1}z_i$.

We further distinguish four possible cases according to the position of the repeat uvu as follows (see Figure 7.1):

(i) The two occurrences of u are contained in z_i.

(ii) The first occurrence of u is contained in z_{i-1} and the second ends in z_i.

(iii) The first occurrence of u starts in z_{i-1} and the second is contained in z_i.

(iv) The first occurrence of u starts in $z_1 \cdots z_{i-2}$ and the second is contained in $z_{i-1}z_i$.

Case (i) needs no action since the phrase z_i is the longest prefix of $z_i z_{i+1}...z_k$ that occurred in $z_1...z_{i-1}$, so two occurrences of u or vuu has been already processed in previous stages. Other cases are dealt with calls to Algorithm MaxExp as described in the code below where $\widetilde{x}$ denotes the reverse of string x. For any two strings z and w and a positive rational number e, MaxExp(z, w, e) is the maximal exponent of repeats in zw whose occurrences start in z and end in w, and whose exponent is at least e; it is e itself if there is no such repeat.

MaxExpRep(y)
1 $(z_1, z_2, \ldots, z_k) \leftarrow$ f-factorisation of y
2 $\triangleright$ z_1 is the longest prefix of y in which no letter repeats
3 $e \leftarrow 1$
4 **for** $i \leftarrow 2$ **to** k **do**
5 $\quad e \leftarrow \max\{\text{MaxExp}(z_{i-1}, z_i, e), e\}$
6 $\quad e \leftarrow \max\{\text{MaxExp}(\widetilde{z_i}, \widetilde{z_{i-1}}, e), e\}$
7 $\quad$ **if** $i > 2$ **then**
8 $\quad\quad e \leftarrow \max\{\text{MaxExp}(\widetilde{z_{i-1}z_i}, z_1\widetilde{\cdots z_{i-2}}, e), e\}$
9 **return** e

Note that variable e can be initialised to the repetitive threshold $\mathrm{RT}(a)$ of the alphabet of string y if the string is long enough. The maximal length of words containing no repeat of exponent at least $\mathrm{RT}(a)$ is 3 for $a = 2$, 38 for $a = 3$, 121 for $a = 4$, and $a + 1$ for $a \geq 5$ (see [32]).

Another technical remark is that the instruction at line 6 can be tuned to deal only with type (iii) repeats of the form u_4vu_4 (see Figure 7.1), i.e. repeats for which the first occurrence of the border starts in z_{i-1} and ends in z_i, because line 5 finds those of the form u_3vu_3. However, this has no influence on the asymptotic runtime.

Theorem 25. *For any input overlap-free string,* MaxExpRep *computes the maximal exponent of repeats occurring in it.*

Proof. We consider a run of MaxExpRep(y). Let $e_1, e_2, \ldots, e_k$ be the successive values of the variable e, where e_i is the value of e just after the execution of lines 5–8 for index i. The initial value $e_1 = 1$ is the maximal exponent of repeats in z_1 as a consequence of its definition. We show that e_i is the maximal exponent of repeats occurring in $z_1z_2 \cdots z_i$ if e_{i-1} is that of $z_1z_2 \cdots z_{i-1}$, for $2 \leq i \leq k$.

To do so, since e_i is at least e_{i-1} (use of max at lines 5–8), all repeats occurring in $z_1z_2 \cdots z_{i-1}$ are taken into account and we only have to consider repeats coming from the concatenation of $z_1z_2 \cdots z_{i-1}$ and z_i, that is, repeats of the form uvu where the second occurrence of u ends in z_i. As discussed above and illustrated in Figure 7.1, only four cases are to be considered because the second occurrence of u cannot start in $z_1z_2 \cdots z_{i-2}$ without contradicting the definition of z_{i-1}.

Line 5 deals with Case (ii) by the definition of MaxExp. Similarly, line 6 is for Case (iii), and line 8 for Case (iv).

If a repeat occurs entirely in z_i, Case (i), by the definition of z_i it occurs also in $z_1z_2 \cdots z_{i-1}$, which is reported by e_{i-1}.

Therefore, all relevant repeats are considered in the computation of e_i, which is then the maximal exponent of repeats occurring in $z_1z_2 \cdots z_i$. This implies that e_k, returned by the algorithm, is that of $z_1z_2 \cdots z_k = y$ as stated. □

3 Locating repeats in a product

In this section, we describe Algorithm MaxExp for computing the maximal exponent of repeats in zw that end in w, whose left border occurs in z, and whose exponent is at least e. MaxExp is called in the main algorithm of previous section.

To locate repeats under consideration, the algorithm examines positions j on w and computes for each the longest potential border of a repeat, a longest suffix u of

$zw[0\,..\,j]$ occurring in z. The algorithm is built upon an algorithm that finds all of them using the Suffix Automaton of string z and described in [25, Section 6.6]. After u is found, some of its suffixes may lead to a repeat with a higher exponent, but the next lemmas show we can discard many of them.

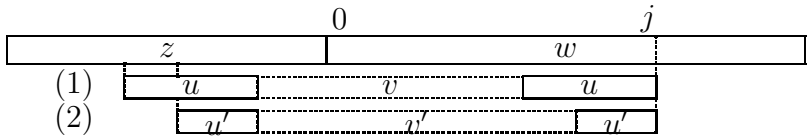


Figure 7.2: When u and its suffix u' ends at the same rightmost position on z, repeat (1) has a larger exponent than repeat (2).

Lemma 22. *Let u' be a suffix of u. If they are both associated with the same state of $\mathcal{S}(z)$ the maximal exponent of a $u'v'u'$ repeat is not greater than the maximal exponent of its associated uvu repeats.*

Note that a suffix u' of u may end at a position larger than the rightmost end position of u and lead to a repeat having a larger exponent. For example, let $z =$ `abadba` and $w =$ `cdaba`. The repeat `abadbacdaba` with border `aba` has exponent 11/8 while the suffix `ba` of `aba` implies the repeat `bacdaba` of greater exponent 7/5.

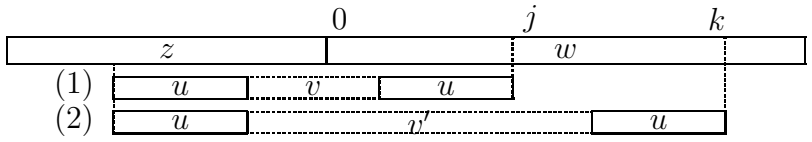


Figure 7.3: Repeat (1) ending at position j has a larger exponent than repeat (2) ending at position $k > j$.

Lemma 23. *If u occurs at end positions j and k on w with $k > j$, the repeat $uv'u$ ending at k cannot be a MER.*

The above properties are used by Algorithm MaxExp to avoid some exponent calculations as follows. Let uvu a repeat ending at j on $zw[0\,..\,j]$ for which u is the longest string associated with state $q = \text{goto}(\text{initial}(\mathcal{S}), u)$. Then next occurrences of u and of any of its suffixes cannot produce repeats with an exponent larger than that of uvu. State q is then marked to inform the next steps of the algorithm that it has been visited.

We use the Suffix Automaton of z (minimal automaton that recognizes the set of all suffixes of z), denoted $\mathcal{S}(z)$, to locate borders of repeats. The structure contains

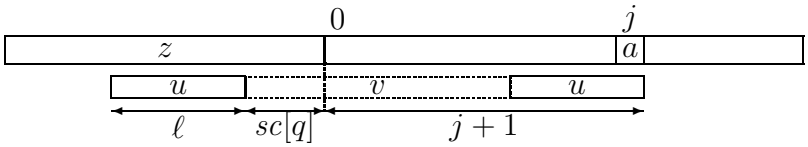


Figure 7.4: The maximal exponent of all repeats in question bordered by u, longest factor of z ending at j, is $(\ell + sc[q] + j + 1)/(sc[q] + j + 1)$.

the failure link F_z and the length function L_z both defined on the set of states. The link is defined as follows: let $p = \text{goto}(\text{initial}(\mathcal{S}(z)), x)$ for $x \in A^+$; then $F_z(p) = \text{goto}(\text{initial}(\mathcal{S}(z)), x')$, where x' is the longest suffix of x for which this latter state is not p. As for the length function, $L_z(p)$ is the maximal length of strings x for which $p = \text{goto}(\text{initial}(\mathcal{S}(z)), x)$.

We need another function, sc_z, defined on states of $\mathcal{S}(z)$ as follows: $sc_z(p)$ is the minimal length of paths from p to a terminal state; in other terms, if $p = \text{goto}(\text{initial}(\mathcal{S}(z)), x)$, then $sc_z(p) = |x'|$ where x' is the shortest string for which xx' is a suffix of z. With this precomputed extra element, computing an exponent is a mere division (see Figure 7.4).

Figure 7.6 illustrates a computation done by the algorithm using the Suffix Automaton of Figure 7.5.

```
MaxExp(z, w, e)
 1  S ← Suffix Automaton of z
 2  mark initial(S)
 3  (q, ℓ) ← (F[last(S)], L[F[last(S)]])
 4  for j ← 0 to min{⌊|z|/(e − 1) − 1⌋, |w| − 1} do
 5       while goto(q, w[j]) = NIL and q ≠ initial(S) do
 6            (q, ℓ) ← (F[q], L[F[q]])
 7       if goto(q, w[j]) ≠ NIL then
 8            (q, ℓ) ← (goto(q, w[j]), ℓ + 1)
 9            (q′, ℓ′) ← (q, ℓ)
10            while q′ unmarked do
11                 e ← max{e, (ℓ′ + sc[q′] + j + 1)/(sc[q′] + j + 1)}
12                 if ℓ′ = L[q′] then
13                      mark q′
14                 (q′, ℓ′) ← (F[q′], L[F[q′]])
15  return e
```

Note the potential overflow when computing $\lfloor |z|/(e-1) - 1 \rfloor$ can easily be fixed in the algorithm implementation.

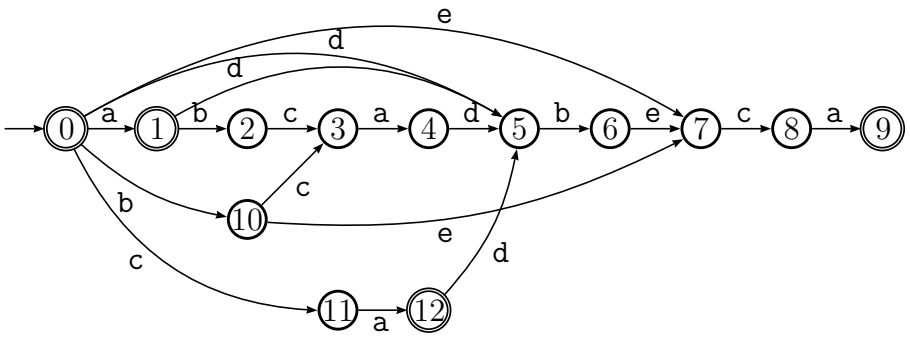


Figure 7.5: Suffix Automaton of abcadbeca. Suffix links: $F[1] = 0$, $F[2] = 10$, $F[3] = 11$, $F[4] = 1$, $F[5] = 0$, $F[6] = 10$, $F[7] = 0$, $F[8] = 11$, $F[9] = 12$, $F[10] = 0$, $F[11] = 0$, $F[12] = 1$. Maximal incoming string lengths: $L[0] = 0$, $L[1] = 1$, $L[2] = 2$, $L[3] = 3$, $L[4] = 4$, $L[5] = 5$, $L[6] = 6$, $L[7] = 7$, $L[8] = 8$, $L[9] = 9$, $L[10] = 1$, $L[11] = 1$, $L[12] = 2$. Minimal extension lengths: $sc[0] = 0$, $sc[1] = 0$, $sc[2] = 7$, $sc[3] = 6$, $sc[4] = 5$, $sc[5] = 4$, $sc[6] = 3$, $sc[7] = 2$, $sc[8] = 1$, $sc[9] = 0$, $sc[10] = 3$, $sc[11] = 1$, $sc[12] = 0$.

j		0	1	2	3	4	5	6	7	8	9
$w[j]$		d	e	c	a	d	b	e	c	a	d
q	12	5	7	8	9	5	6	7	8	9	5
ℓ	2	3	1	2	3	3	4	5	6	7	3
exp		8/5	5/4	3/2	7/4	4/3	13/9	14/9	5/3	16/9	17/14
				5/4			10/9				

Figure 7.6: Computing exponents when searching zw for repeats uvu. The first occurrence of u is in z and the second ends in zw. The Suffix Automaton of $z =$ abcadbeca with function sc is in Figure 7.5. The search is done by parsing $w =$ decadbecad with the automaton. Exponents of repeats are given by the expression $(\ell + sc[q] + j + 1)/(sc[q] + j + 1)$. The last line is for exponents corresponding to suffixes of u. The maximal exponent all repeats is 7/4.

Theorem 26. *Algorithm* MaxExp*, applied to strings z and w and to the rational number e, produces the maximal exponent of repeats in zw that end in w, whose left border occurs in z and exponent is at least e.*

Proof. In the algorithm, position j on w stands for a potential ending position of a relevant repeat. First, we show that the algorithm does not require to examine more values of j but those specified at line 4. The exponent of a repeat uvu is uvu/vu. Since we are looking for repeats satisfying $uvu/vu \geq e$, the longest possible such repeat has period $j+1$ and border z. Then $(z+j+1)/(j+1) > e$ implies $j < z/(e-1) - 1$ (which is $+\infty$ if $e = 1$). Since j is a position on w, $j < w$, which completes the first statement.

Second, given a position j on w, we show that the algorithm examines all the possible concerned repeats having an exponent at least e and ending at j. The following property related to variables q, state of $\mathcal{S}$, and ℓ is known from [25, Section 6.6]: let u be the longest suffix of $zw[0\,..\,j]$ that is a factor of z, then $q = \text{goto}(\text{initial}(\mathcal{S}), u)$ and $\ell = |u|$. The property is also true just after execution of line 3 for z alone due to the initialisation of the two variables.

Then string u is the border of a repeat ending in w and whose left border occurs in z. Lines 9 to 14 check the exponents associated with u and its suffixes. If q' is unmarked, the exponent is computed as explained before (see Figure 7.4). If the condition at line 11 is met, which means that u is the longest string satisfying $q' = \text{goto}(\text{initial}(\mathcal{S}), u)$, due to Lemma 23 the algorithm does not need to check the exponent associated with next occurrences of u, nor with the suffixes of u since they have been checked before. Due to Lemma 22, suffixes of u ending at the same rightmost position on z do not have a larger exponent. Therefore the next suffix whose associated exponent has to be checked is the longest suffix leading to a different state of $\mathcal{S}$: it is $F(q')$ and the length of the suffix is $L(F(q'))$ by definition of F and L.

Finally note the initial state of $\mathcal{S}$ is marked because it corresponds to an empty string u, that is a repeat of exponent 1, which is not larger than the values of e.

This proves the algorithm runs through all possible relevant repeats, which ends the proof. □

4 Complexity analysis

In this section, we analyse the running time and memory space of the previous algorithms.

Proposition 23. *Applied to strings z and w and to the rational number e, Algorithm* MaxExp *requires $O(|z|)$ space in addition to inputs and runs in total time $O(|z| +$*

$\min\{\lfloor |z|/(e-1)-1 \rfloor, |w|-1\})$ *on a fixed size alphabet. It performs less than* $2|z| + \min\{\lfloor |z|/(e-1)-1 \rfloor, |w|-1\}$ *exponent computations.*

Proof. The space is used mostly for storing the automaton, which is known to have no more $2|z|$ states and $3|z|$ edges (see [25]). It can be stored in linear space if edges are implemented by successor lists, which adds a multiplicative $\log a$ factor on transition time.

It is known from [25, Section 6.6] that the algorithm runs in linear time on a fixed alphabet, including the automaton construction with elements F, L and sc, if we exclude the time for executing lines 9 to 14.

So, let us count the number of times line 11 is executed. It is done once for each position j associated with an unmarked state. If it is done more than once for a given position, then the second value of q' comes from the failure link. A crucial observation is that condition at line 12 holds for such a state. Therefore, since $\mathcal{S}(z)$ has no more than $2|z|$ states, the total number of extra executions of line 11 is at most $2|z|$. Which gives the stated result. □

The proof of the linear running time of Algorithm MaxExpRep additionally relies on a combinatorial property of strings (see Chapter 4). It is Dejean's statement [32] proved in [56, 31] that gives for each alphabet size k, its repetitive threshold $\mathrm{RT}(a)$, i.e. the maximal exponent unavoidable in infinite strings over the alphabet. Thresholds are: $\mathrm{RT}(2) = 2$, $\mathrm{RT}(3) = 7/4$, $\mathrm{RT}(4) = 7/5$, and $\mathrm{RT}(a) = a/(a-1)$ for $a \geq 5$. Thus, if the string y is long enough the maximal exponent of its factors is at least $\mathrm{RT}(a)$ where a is its alphabet size (see the note following Algorithm `MaxExpRep`).

Theorem 27. *Applied to any overlap-free string of length n on a fixed-size alphabet, Algorithm* ***MaxExpRep*** *runs in time $O(n)$ and requires $O(n)$ extra space.*

Proof. Computing the f-factorisation $(z_1, z_2, \ldots, z_k)$ of the input takes time and space $O(n)$ on a fixed-size alphabet using any suffix data structure. (It can even be done in time $O(n)$ on an integer alphabet, see [28].)

Next instructions execute in linear extra space from Proposition 23. Line 5 takes time $O(|z|+\min\{\lfloor |z_{i-1}|/(e-1)-1 \rfloor, |z_i|-1\})$, which is bounded by $O(|z_{i-1}|+|z_{i-1}|/(e-1)-1)$, for $i = 2, \ldots, k$. For a long enough input, e is eventually at least $\mathrm{RT}(a)$ where a is the input alphabet. The time is then bounded by $O(|z_{i-1}|+|z_{i-1}|/(\mathrm{RT}(a)-1)-1)$, then $O(|z_{i-1}|)$ because $\mathrm{RT}(a) > 1$. The contribution of Line 5 to the total runtime is then $O(\Sigma_{i=2}^{k}|z_{i-1}|)$.

Similarly it is $O(\Sigma_{i=2}^{k}|z_i|)$ for Line 6 and $O(\Sigma_{i=2}^{k}|z_{i-1}z_i|)$ for Line 8. Thus the overall runtime is bounded by $O(\Sigma_{i=1}^{k}|z_i|)$, which is $O(n)$ as expected. □

5 Counting maximal-exponent repeats

In this section, we show there is a finite number of MERs in an overlap-free string. Note that on the alphabet $\{\mathtt{a}, \mathtt{a}_1, \ldots, \mathtt{a}_n\}$ the string $\mathtt{a}\mathtt{a}_1\mathtt{a}\mathtt{a}_2\mathtt{a}\ldots\mathtt{a}\mathtt{a}_n\mathtt{a}$ of length $2n+1$ has a quadratic number of maximal repeats. Indeed all occurrences of repeats of the form $\mathtt{a}w\mathtt{a}$ for a word w are non extensible. But only the n repeats of the form $\mathtt{a}c\mathtt{a}$ for a letter c have the maximal exponent 3/2.

We start with a simple property of MER, which does not lead to their linear number, but is used below to tune the upper bound.

Lemma 24. *Consider two occurrences of MERs with the same border length b starting at respective i and j on y, $i < j$. Then, $j - i > b$.*

Proof. The two MERs having the same border length, since they have the same exponent, they have also the same period and the same length. Let b their border length and p their period.

Assume ab absurdo $j-i \leq b$. The word $y[i\,..\,i+b-1] = y[i+p\,..\,i+p+b-1]$ is the border of the first MER. The assumption implies that $y[i+b] = y[i+p+b]$ because these letters belong to the border of the second MER. It means the first MER can be extended with the same period, a contradiction because it has the largest exponent. Therefore, $j - i > b$ as stated. □

If we count the occurrences of MERs by their border lengths after Lemma 24 we get an initial part of the harmonic series, quantity that is not linear with respect to the length y.

To refine the previous lemma and get a linear upper bound on the number of occurrences of MERs we introduce the notion of δ-MERs, for a positive real number δ, as follows. An MER uvu is a δ-MER if its border length $b = |u| = |uvu| - \mathrm{period}(uvu)$ satisfies $3\delta \leq b < 4\delta$. Then any MER is a δ-MER for some $\delta \in \Delta$, where $\Delta = \{1/3, 2/3, 1, 4/3, (4/3)^2, (4/3)^3, \ldots\}$. This is the technique used for example in [60, 27] to count runs in strings.

The proof of the next lemma is illustrated by Figure 7.7.

Lemma 25. *Let uvu and $u'v'u'$ be two δ-MERs starting at respective i and j on y, $i < j$. Then, $j - i \geq \delta$.*

Proof. Assume ab absurdo $j - i < \delta$ (see Figure 7.7).

Since both $|u| \leq 3\delta$ and $|u'| \leq 3\delta$, the two occurrences overlap. Let w be the overlap. It can be a suffix of u and a prefix of u' as in Figure 7.7, or w can be the

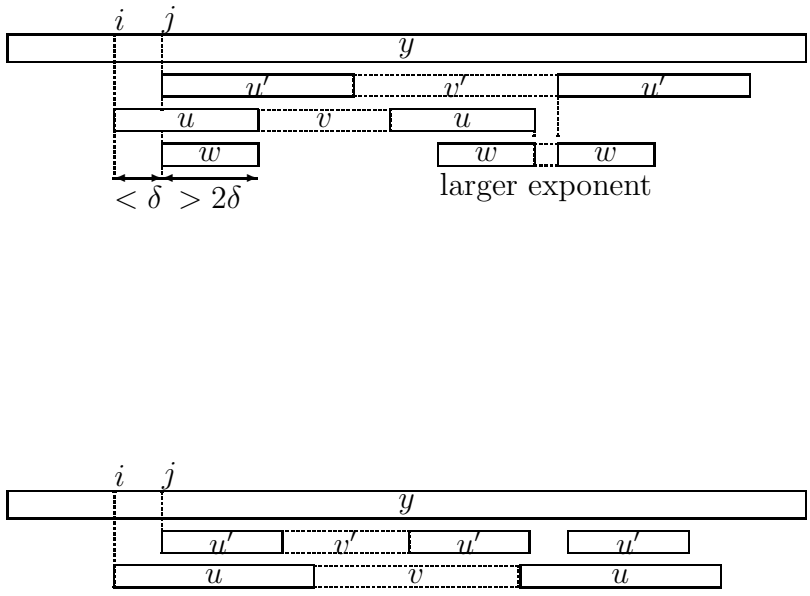


Figure 7.7: Top: two δ-MERs, uvu and $u'v'u'$, starting at close positions induce a repeat with a larger exponent, a contradiction. Bottom: the last two occurrences of u' are closer than the first two, leading to a larger exponent than $u'v'u'$, a contradiction. Indeed, the case is possible only if $|u'| \leq |u|/2$.

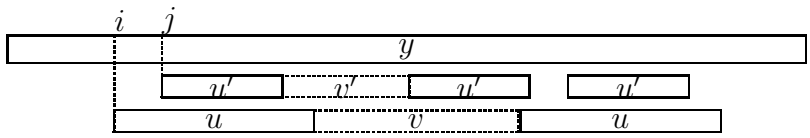


Figure 7.8: Second case of two δ-MERs, uvu and $u'v'u'$, starting at close positions: the last two occurrences of u' are closer than the first two, leading to a larger exponent than $u'v'u'$, a contradiction. Indeed, the case is possible only if $|u'| \leq |u|/2$.

shorter of u and u' when it occurs in the longer, see Figure 7.8. In both cases we have $|w| > 2\delta$.

Let $p = |uv|$ be the period of uvu and $p' = |u'v'|$ be that of $u'v'u'$. Note that the exponent of the two repeats is $e = 1 + |u|/p = 1 + |u'|/p'$, which implies $p' - p = (|u'| - |u|)/(e-1)$.

Due to the periodicity of the two repeats, w occurs at both positions $j + p$ and $j + p'$. Assume for example that $j + p < j + p'$ (we cannot have $j + p = j + p'$). The factor $y[j + p \,..\, j + p + |w| - 1]$ has exponent

$$1 + \frac{|w|}{p' - p} = 1 + \frac{|w|(e-1)}{(|u'| - |u|)}.$$

However since $|w| > 2\delta$ and $|u'| - |u| < 2\delta$, the exponent is larger than e, a contradiction with the definition of uvu and $u'v'u'$. Therefore $j - i \geq \delta$ as stated. □

A direct consequence of the previous lemma is the linear number of MER occurrences.

Theorem 28. *There is a constant α for which the number of occurrences of maximal-exponent repeats in a string of length n is less than αn.*

Proof. Lemma 25 implies the number of δ-MER occurrences in y is no more than n/δ. Since values of δ in Δ cover all border lengths, the total number of occurrences

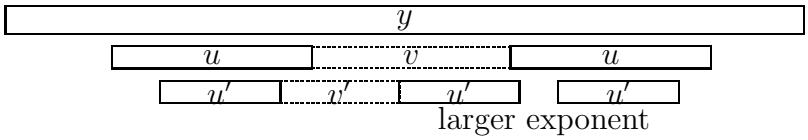


Figure 7.9: The left occurrence of u' from the MER $u'v'u'$ falls inside the left occurrence of u from the MER uvu. Then $|u'| \leq |u|/2$ because the contrary induces a repeat with a larger exponent, a contradiction.

of MERs is bounded by

$$\sum_{\delta \in \Delta} \frac{n}{\delta} = n \left(3 + \frac{3}{2} + 1 + \frac{3}{4} + \left(\frac{3}{4} \right)^2 + \dots \right) < 8.5\, n.$$

□

The next statement refines the upper bound given in the proof of the previous theorem.

Corollary 7. *There are less than* $3.11\, n$ *occurrences of MERs in a string of length* n*.*

Proof. According to Lemma 24 there are less than

$$\sum_{b=1}^{b=11} \frac{n}{b+1} = 2.103211\, n$$

occurrences of MERs with border length at most 11.

We then apply Lemma 25 with values of $\delta \in \Gamma$ that allow to cover all remaining border lengths of MERs: $\Gamma = \{4, 4(4/3), 4(4/3)^2, \dots\}$, we get the upper bound

$$\sum_{\delta \in \Gamma} \frac{n}{\delta} = \frac{1}{4} \left(1 + \frac{3}{4} + \left(\frac{3}{4} \right)^2 + \dots \right) n = n$$

for the number of occurrences of MER with border length at least 12.

Thus the global upper bound we get is $3.11\,n$. □

Note that the border length 11 (or 12) minimises the expression

$$\left(\sum_{b=1}^{b=k} \frac{n}{b+1}\right) + \frac{3}{k+1}\left(1 + \frac{3}{4} + \left(\frac{3}{4}\right)^2 + \dots\right) n = \left(\sum_{b=1}^{b=k} \frac{n}{b+1}\right) + \frac{12\,n}{k+1}$$

with respect to k, which means the technique is unlikely to produce a smaller bound. By contrast, experiments show that the number of occurrences of MERs is in fact smaller than n and not even close to n, at least for small values of n. The following table displays the maximal number of MERs for overlap-free string lengths $n = 5, 6, \dots, 20$ and for alphabet sizes 2, 3 and 4. It also displays (second element of pairs) the associated maximal exponent. In the binary case we already know that it is 2 since squares are unavoidable in strings whose length is greater than 3.

n	5	6	7	8	9	10	11	12
binary	2	3	4	5	5	6	6	8
ternary	(2, 1.5)	(3, 1.5)	(4, 2)	(5, 2)	(5, 2)	(6, 1.5)	(6, 2)	(8, 2)
4−ary	(2, 1.5)	(3, 1.5)	(4, 2)	(5, 2)	(5, 2)	(6, 1.5)	(7, 1.5)	(8, 2)

13	14	15	16	17	18	19	20
8	9	9	11	11	12	12	14
(8, 2)	(9, 2)	(9, 2)	(11, 2)	(11, 2)	(12, 2)	(12, 2)	(14, 2)
(8, 1.5)	(9, 1.5)	(10, 1.5)	(11, 2)	(12, 1.5)	(12, 1.5)	(13, 1.5)	(14, 1.5)

6 Conclusion

The result of Section 5 implies that Algorithm MaxExpRep can be upgraded to output all the MERs occurring in the input string in the same asymptotic time. Indeed, the only occurrences of MERs that are skipped by the algorithm when computing the maximal exponent are those occurring inside a phrase of the f-factorisation (Case (i) of Section 2). However, storing their previous occurrences and listing them can be done in time proportional to their number, which does not affect the asymptotic runtime of the algorithm and yields the next statement.

Corollary 8. *All the occurrences of maximal-exponent factors of a string can be listed in linear time with respect to its length.*

The present work triggers the study of a uniform solution to compute both repetitions and repeats. However, exponent 2 seems to reflect a transition phase in the

combinatorics of studied objects. For instance, the number of repetitions in a string can be of the order of $n \log n$, the number of runs linear, while the number of repeats and of their maximal occurrences can be quadratic.

An interesting question would be to select repeats which occur only a linear number of times or slightly more. An attempt has been achieved in [45] where it is shown that the number of maximal repetitions of any exponent more than $1 + \epsilon$ is bounded by $\frac{1}{\epsilon} n \ln n$. See also the discussions at the end of [44] and of [26].

We have left for a future work the calculation of the number of (distinct) MERs occurring in a string, as well as the lower bounds on these quantities.

8

Future Work

We summarise in this chapter a series of open questions that are found in previous chapters.

In Chapter 2, we mentioned a few well known square-free morphisms. However, little is known about the characterisation of weakly square-free morphisms, therefore it would be interesting to study this class of morphisms. The question raised by Currie and Rampersad is still open: does there exist k-uniform square-free ternary morphisms for all $k \geq 11$?

In Chapter 4, we proved that there exists an infinite overlap-free ternary word with only one square, 00 and no e-power with $7/4 \leq e < 2$. We conjecture that there exists an infinite word on 4-letter alphabet containing 010 and no e-power with $7/5 \leq e < 3/2$ or $e > 3/2$. This conjecture is based on an extensive computation. It remains to investigate infinite words on a higher alphabet and see if there exists a class of infinite words containing only one N_k-power and no e-power with $r_k \leq e < N_k$ or $e > N_k$, where N_k is a rational number and r_k is the repetitive threshold of the k-letter alphabet.

Remaining conjectures and questions stated in Chapter 4 are:

- There exists an infinite Dejean's word on a 5-letter alphabet with only 45 limit repetitions.
- There exists an infinite Dejean's word on a 5-letter alphabet with only 46 limit repetitions, and such that every limit repetition has period 4.
- Is it possible to construct Dejean's words such that the only allowed limit repetitions have period $k-1$, for every $k > 38$?
- Can we find a lower or an upper bound for $Rn(k)$ when $k > 5$, where $Rn(k)$ is the minimum number of limit repetitions in infinite Dejean's word ?

In Chapter 6, we studied some pattern avoidance. The study by Samsonov and

Shur on cube-free binary words avoiding some binary patterns can be extended. The interesting questions that remain are:

- What are the avoidability status of the binary patterns in 5/2-free binary words? The only cases are $AABBAA$ and $AABABB$.
- Is there any pattern that is unavoidable by cube-free binary words but avoidable by 3^+-free binary words?

Another interesting open question is the following:
Suppose that P is an avoidable pattern with avoidability index $\lambda(P) = k$. Is it possible to find a finite set $\mathcal{P}$ of patterns and a finite set $\mathcal{F}$ of factors such that $P \in \mathcal{P}$ and $\mathcal{P} \cup \mathcal{F}$ characterises a morphic word over Σ_k? Notice that this would be a strengthening of Cassaigne's conjecture [21] that there exists a morphic word avoiding P over Σ_k.

In Chapter 7, we provided a linear-time algorithm to compute the maximal exponent of factors occurring in a word. Our solution works on overlap-free strings for which the maximal exponent of factors is at most 2. Computing the maximal exponent of factors with exponent at least 2 can be done in linear time by adapting the algorithm of Kolpakov and Kucherov [44]. A remaining question to the present study is to unify the algorithmic approaches for repetitions of exponent at least 2 and for repeats of exponent at most 2.

The results presented in this thesis have made a significant contribution to extending previous work. However, as described throughout the thesis, there remain many open questions and directions in which this work can continue.

Bibliography

[1] J.-P. Allouche and J. O. Shallit. *Automatic Sequences - Theory, Applications, Generalizations*. Cambridge University Press, 2003.

[2] G. Badkobeh. Fewest repetitions vs maximal-exponent powers in infinite binary words. *Theoretical Computer Science*, 412(48):6625–6633, 2011.

[3] G. Badkobeh. An infinite binary word containing only three distinct squares. 2012. Submitted.

[4] G. Badkobeh, S. Chairungsee, and M. Crochemore. Hunting redundancies in strings. In *Developments in Language Theory*, pages 1–14, 2011.

[5] G. Badkobeh and M. Crochemore. Finite-Repetition threshold for infinite ternary words. In *WORDS*, pages 37–43, 2011.

[6] G. Badkobeh and M. Crochemore. Fewest repetitions in infinite binary words. *RAIRO-Theoretical Informatics and Applications*, 46(1):17–31, 2012.

[7] G. Badkobeh and M. Crochemore. Infinite binary word containing odd-period repetitions. 2012. Submitted.

[8] G. Badkobeh, M. Crochemore, and C. Toopsuwan. Computing the maximal-exponent repeats of an overlap-free string in linear time. In E. C. L. Calderon-Benavides, C. Gonzlez-Caro and N. Ziviani, editors, *Symposium on String Processing and Information Retrieval*, number 7608 in LNCS, pages 61–72. Springer, 2012.

[9] G. Badkobeh and P. Ochem. Characterization of some binary words with few squares. 2012. Submitted.

[10] G. Badkobeh, M. Rao, and M. Crochemore. Finite-repetition threshold for large alphabets. In *14th Mons Days of Theoretical Computer Science*, 2012. To appear.

[11] D. R. Bean, A. Ehrenfeucht, and G. F.McNulty. Avoidable patterns in strings of symbols. *Pacific Journal of Mathematics*, 85(2):1–22, 1979.

[12] O. Berkman, C. S. Iliopoulos, and K. Park. The Subtree Max-Gap Problem with Application to Parallel String Covering. *Information and Computation*, 123(1):127–137, 1995.

[13] J. Berstel. Some Recent Results on Squarefree Words. In *STACS*, pages 14–25, 1984.

[14] J. Berstel. Axel Thue's Papers on Repetitions in Words: a Translation. Technical report, Publications du Laboratoire de Combinatoire et d'Informatique Mathématique. Université du Québec à Montréal, 1995.

[15] F. J. Brandenburg. Uniformly Growing k-TH Power-Free Homomorphisms. *Theoretical Computer Science*, 23:69–82, 1983.

[16] G. S. Brodal and C. N. S. Pedersen. Finding Maximal Quasiperiodicities in Strings. In R. Giancarlo and D. Sankoff, editors, *Combinatorial Pattern Matching*, number 1848 in LNCS, pages 397–411. Springer, 2000.

[17] A. Carpi. On the Size of a Square-Free Morphism on a Three Letter Alphabet. *Information Processing Letters*, 16(5):231–235, 1983.

[18] A. Carpi. Overlap-Free Words and Finite Automata. *Theor. Comput. Sci.*, 115(2):243–260, 1993.

[19] A. Carpi. On Dejean's conjecture over large alphabets. *Theoretical Computer Science*, 385(1-3):137–151, 2007.

[20] J. Cassaigne. Counting Overlap-Free Binary Words. In P. Enjalbert, A. Finkel, and K. W. Wagner, editors, *STACS*, volume 665 of *Lecture Notes in Computer Science*, pages 216–225. Springer, 1993.

[21] J. Cassaigne. Unavoidable Binary Patterns. *Acta Informatica*, 30(4):385–395, 1993.

[22] J. Cassaigne. *Motifs évitables et régularités dans les mots*. PhD thesis, Université Paris 6, 1994.

[23] M. Christou, M. Crochemore, C. S. Iliopoulos, M. Kubica, S. P. Pissis, J. Radoszewski, W. Rytter, B. Szreder, and T. Walen. Efficient Seeds Computation Revisited. In R. Giancarlo and G. Manzini, editors, *Combinatorial Pattern Matching*, number 6661 in LNCS, pages 350–363, Berlin, 2011. Springer.

[24] M. Crochemore. Sharp Characterizations of Squarefree Morphisms. *Theoretical Computer Science*, 18:221–226, 1982.

[25] M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on Strings*. Cambridge University Press, 2007. 392 pages.

[26] M. Crochemore and L. Ilie. Maximal repetitions in strings. *Journal of Computer and System Sciences*, 74:796–807, 2008. DOI: 10.1016/j.jcss.2007.09.003.

[27] M. Crochemore, L. Ilie, and L. Tinta. The "runs" conjecture. *Theoretical Computer Science*, 412(27):2931–2941, 2011.

[28] M. Crochemore and G. Tischler. Computing Longest Previous non-overlapping Factors. *Information Processing Letters*, 111:291–295, 2011.

[29] J. D. Currie and N. Rampersad. Dejean's conjecture holds for $n \geq 30$. *Theoretical Computer Science*, 410(30-32):2885–2888, 2009.

[30] J. D. Currie and N. Rampersad. There are k-uniform cubefree binary morphisms for all $k >= 0$. *Discrete Applied Mathematics*, 157(11):2548–2551, 2009.

[31] J. D. Currie and N. Rampersad. A proof of Dejean's conjecture. *Mathematics of Computation*, 80(274):1063–1070, 2011.

[32] F. Dejean. Sur un Théorème de Thue. *Journal of Combinatorial Theory, Ser. A*, 13(1):90–99, 1972.

[33] F. M. Dekking. On Repetitions of Blocks in Binary Sequences. *Journal of Combinatorial Theory, Ser. A*, 20(3):292–299, 1976.

[34] R. C. Entringer and D. E. Jackson. On Nonrepetitive Sequences. *Journal of Combinatorial Theory, Ser. A*, 16(2):159–164, 1974.

[35] P. Erdös. Some Unsolved Problems. 4:291–300, 1957.

[36] A. S. Fraenkel and J. Simpson. How Many Squares Must a Binary Sequence Contain? *Electronic Journal of Combinatorics*, 2, 1995.

[37] M. Hall. Generators and relations in groups- the Burnside problem. *In T. L. Saaty (editor), Lectures on Modern Mathematics*, 2:42–92, Wiley, 1964.

[38] T. Harju and D. Nowotka. Binary Words with Few Squares. *Bulletin of the EATCS*, 89:164–166, 2006.

[39] L. Ilie, P. Ochem, and J. Shallit. A generalization of repetition threshold. *Theoretical Computer Science*, 345(2-3):359–369, 2005.

[40] C. S. Iliopoulos, D. W. G. Moore, and K. Park. Covering a String. *Algorithmica*, 16(3):288–297, 1996.

[41] J. Karhumäki and J. Shallit. Polynomial versus exponential growth in repetition-free binary words. *Journal of Combinatorial Theory, Ser. A*, 105(2):335–347, 2004.

[42] D. E. Knuth. *The Art of Computer Programming, Volume I: Fundamental Algorithms*. Addison-Wesley, 1968.

[43] Y. Kobayashi. Enumeration of irreducible binary words. *Discrete Applied Mathematics*, 20(3):221–232, 1988.

[44] R. Kolpakov and G. Kucherov. On Maximal Repetitions in Words. *Journal of Discret Algorithms*, 1(1):159–186, 2000.

[45] R. Kolpakov, G. Kucherov, and P. Ochem. On maximal repetitions of arbitrary exponent. *Information Processing Letters*, 110(7):252–256, 2010.

[46] J. Leech. A problem on strings of beads. *Mathematical Gazette*, 41:277–278, 1957.

[47] M. Lothaire, editor. *Combinatorics on Words*. Cambridge University Press, second edition, 1997.

[48] M. Mohammad-Noori and J. D. Currie. Dejean's conjecture and Sturmian words. *European Journal of Combinatorics*, 28(3):876–890, 2007.

[49] J. Moulin-Ollagnier. Proof of Dejean's Conjecture for Alphabets with 5, 6, 7, 8, 9, 10 and 11 Letters. *Theoretical Computer Science*, 95(2), 1992.

[50] P. Ochem. A generator of morphisms for infinite words. *RAIRO - Theoretical Informatics and Applications*, 40(3):427–441, 2006.

[51] P. Ochem. Binary words avoiding the pattern AABBCABBA. *RAIRO - Theoretical Informatics and Applications*, 44(1):151–158, 2010.

[52] J. J. Pansiot. The Morse Sequence and Iterated Morphisms. *Information Processing Letters*, 12(2):68–70, 1981.

[53] J. J. Pansiot. A Propos d'une Conjecture de F. Dejean sur les Répétitions dans les Mots. In J. Díaz, editor, *Automata, Languages and Programming*, volume 154 of *LNCS*, pages 585–596. Springer, 1983.

[54] P. A. B. Pleasants. Nonrepetitive sequences. *Mathematical Proceedings of the Cambridge Philosophical Society*, 68:267–274, 1970.

[55] N. Rampersad, J. Shallit, and M.-W. Wang. Avoiding large squares in infinite binary words. *Theoretical Computer Science*, 339(1):19–34, 2005.

[56] M. Rao. Last cases of Dejean's conjecture. *Theoretical Computer Science*, 412(27):3010–3018, 2011.

[57] A. Restivo and S. Salemi. Overlap-free words on two symbols. In M. Nivat and D. Perrin, editors, *Automata on Infinite Words*, volume 192 of *Lecture Notes in Computer Science*, pages 198–206. Springer, 1984.

[58] P. Roth. Every Binary Pattern of Length Six is Avoidable on the Two-Letter Alphabet. *Acta Informatica*, 29(1):95–107, 1992.

[59] G. Rozenberg. *Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology*. Springer-Verlag, 1992.

[60] W. Rytter. The number of runs in a string. *Information and Computation*, 205(9):1459–1469, 2007.

[61] A. V. Samsonov and A. M. Shur. Binary Patterns in Binary Cube-Free Words: Avoidability and Growth. 2012. To appear.

[62] U. Schmidt. Avoidable Patterns on Two Letters. *Theoretical Computer Science*, 63(1):1–17, 1989.

[63] P. Séébold. Sur les morphismes qui engendrent des mots infinis ayant des facteurs prescrits. In *Theoretical Computer Science*, pages 301–311, 1983.

[64] J. Shallit. Simultaneous avoidance of large squares and fractional powers in infinite binary words. *International Journal of Foundations of Computer Science*, 15(2):317–327, 2004.

[65] A. Thue. Uber unendliche Zeichenreihen. *Norske vid. Selsk. Skr. I. Mat. Nat. Kl. Christiana*, 7:1–22, 1906.

[66] A. Thue. Uber die gegenseitigen Lage gleicher Teile gewisser Zeichenreilren. *Norske vid. Selsk. Skr. I. Mat. Nat. Kl. Christiana*, 1:1–67, 1912.

[67] A. I. Zimin. Blocking sets of terms. *Math. USSR Sbornik*, 47(2):353–364, 1984.

[68] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. 23:337–343, 1977.