



HAL
open science

Avoiding squares over words with lists of size three amongst four symbols

Matthieu Rosenfeld

► **To cite this version:**

Matthieu Rosenfeld. Avoiding squares over words with lists of size three amongst four symbols. Mathematics of Computation, 2022, 91 (337), pp.2489-2500. 10.1090/mcom/3732 . hal-03583228

HAL Id: hal-03583228

<https://hal.science/hal-03583228>

Submitted on 17 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Avoiding squares over words with lists of size three amongst four symbols

Matthieu Rosenfeld*

University of Montpellier, LIRMM

May 12, 2021

Abstract

In 2007, Grytczuk conjectured that for any sequence $(\ell_i)_{i \geq 1}$ of alphabets of size 3 there exists a square-free infinite word w such that for all i , the i -th letter of w belongs to ℓ_i . The result of Thue of 1906 implies that there is an infinite square-free word if all the ℓ_i are identical. On the other hand, Grytczuk, Przybyło and Zhu showed in 2011 that it also holds if the ℓ_i are of size 4 instead of 3.

In this article, we first show that if the lists are of size 4, the number of square-free words is at least 2.45^n (the previous similar bound was 2^n). We then show our main result: we can construct such a square-free word if the lists are subsets of size 3 of the same alphabet of size 4. Our proof also implies that there are at least 1.25^n square-free words of length n for any such list assignment. This proof relies on the existence of a set of coefficients verified with a computer. We suspect that the full conjecture could be resolved by this method with a much more powerful computer (but we might need to wait a few decades for such a computer to be available).

1 Introduction

A *square* is a word of the form uu where u is a non-empty word. We say that a word is *square-free* (or avoids squares) if none of its factors is a square. For instance, *hotshots* is a square while *minimize* is square-free. In 1906, Thue showed that there are arbitrarily long ternary words avoiding squares [16, 17]. This result is often regarded as the starting point of combinatorics on words, and the generalizations of this particular question received a lot of attention.

Nonrepetitive colorings of graphs were introduced by Alon *et al.* [1]. A coloring of the vertices (or of the edges) of a graph is said to be *nonrepetitive* if there is no path of the graph whose color sequence is a square. The *nonrepetitive chromatic number* (resp. *nonrepetitive chromatic index*) of a graph is the minimal number of colors in a nonrepetitive coloring of the vertices (resp. the edges) of the graph. Alon *et al.* showed that nonrepetitive chromatic index is at most $O(\Delta^2)$ where Δ is the maximum degree of G [1]. Different authors

*Supported by the ANR project CoCoGro (ANR-16-CE40-0005).

successively improved the upper bounds on the nonrepetitive chromatic number and the nonrepetitive chromatic index and the best known bound for the nonrepetitive chromatic number is also in $O(\Delta^2)$ [2, 4, 10, 13]. Non-repetitive colorings have since been studied in many other contexts (see for instance [19] for a recent survey on this topic).

Most results regarding non-repetitive colorings of graphs of bounded maximal degree are based on the Lovász Local Lemma, entropy-compression, or related methods and they naturally hold in the stronger setting of list coloring. In this setting, each vertex is assigned a list of colors and the colorings of the graph must assign to each vertex a color from its list. The *nonrepetitive list chromatic number* is the smallest integer n such that the graph is nonrepetitively colorable as soon as all the lists contain at least n colors. This notion was studied in relation to many notions of colorings and contrary to the first intuition it is often not the case that the worst possible list assignment is the one that gives the same list to every vertex. For instance, every planar graph has nonrepetitive chromatic number at most 768, but can have an arbitrarily large nonrepetitive list chromatic number [3]. However, the best known bound on the nonrepetitive chromatic number in terms of the maximal degree also holds for the nonrepetitive list chromatic number [19]. It is unknown whether the optimal bounds in terms of the maximal degree also holds for the nonrepetitive list chromatic number are indeed identical or not. The simplest graph for which the question is non-trivial is the path. The result of Thue implies that the nonrepetitive chromatic number of any path over at least four vertices is 3 [16]. There are various simple proofs that the nonrepetitive list chromatic number of any path is at most 4 (see [8] for a proof based on the Lefthanded Local Lemma, [9] for a proof based on entropy compression and [13] for a proof based on a simple counting argument). It was first conjectured by Grytczuk [6] that the nonrepetitive list chromatic number of any path is in fact at most 3. This conjecture has been mentioned many times, but not much progress has been made in the direction of proving or disproving it (see for instance [5, 6, 7, 9, 13, 15, 19, 20] for some of the occurrences of this problem).

The question can be reformulated in terms of combinatorics on words.

Question 1. *Let \mathcal{A} be an infinite alphabet. Is it true that for any sequence $(\ell_i)_{i \geq 1}$ of subsets of \mathcal{A} of size 3 there exists an infinite square-free word w such that for all i , the i -th letter of w belongs to ℓ_i ?*

As already mentioned the answer is positive if 3 is replaced by 4. It was even shown in [13] that there are at least 2^n such words of length n for any list assignment. We first show in this article, that there are at least 2.45^n such words. We then show our main result: if \mathcal{A} is of size 4 then there is such a word. Our approach is similar to the idea of [13], that is, we show some strong bounds on the number of such words using an inductive argument. Our approach also relies on ideas of an approach of Shur to bound the number of words power-free languages [14] (this improves an older technique introduced by Kolpakov [11]). This idea is that instead of directly counting the words we associate a weight to each word and we count the total weight of the set of valid words. In the approach of Kolpakov, he had to deal with three different kinds of squares (short squares, long squares, and mid-length squares) and we borrow a trick from Shur that allows dealing with only two kinds of squares (*i.e.*, the mid-length squares do not need to be dealt with separately). In fact, the main

difference between our proof and the approach of Shur is that eigenvectors and eigenvalues are replaced by a more complicated notion (which does not seem to be well studied or even named in the literature). For our main result, we use a computer to verify the existence of a set of weights with the right property. This approach could probably be applied if \mathcal{A} is of size larger than 4 (and even for infinite \mathcal{A}), but the computational power required is larger than what modern computers offer. In particular, we suspect that this approach would work if \mathcal{A} was of size 21 and that would imply the conjecture.

The article is organized as follows. We first provide some notations and definitions in Section 2. In Section 3, we show that for any list assignment with lists of size 4, there are at least 2.45^n square-free words of size n . We use this proof to introduce our technique. In section 4, we use this technique to show our main result. In Section 5, we detail how we use the computer to verify the existence of the set of weights needed for our result. We conclude this article, in Section 6, by explaining why we suspect that the same approach could solve Question 1.

2 Definitions and notations

We use the standard definitions and notations of combinatorics on words introduced in Chapter 1 of [12].

For the sake of notations, all our alphabets are sets of integers. For any fixed alphabet \mathcal{A} , the *set of extensions of a word w* is the set of words $\{wa : a \in \mathcal{A}\}$. That is, a word u is an extension of another word w if u can be obtained by the concatenation of a letter at the end of w . For any words $w = w_1w_2 \dots w_n$ and any i, j such that $1 \leq i \leq j \leq n$, we write $w[i, j] = w_iw_{i+1} \dots w_j$.

A *square* is a word of the form uu with u a non-empty word. The *period* of the square uu is u and by abuse of notation we also call the length $|u|$ of u the period of uu . A word is *square-free* (or *avoids squares*) if none of its factors is a square.

A *list assignment* is a sequence $(\ell_i)_{i \geq 1}$ of subsets of the integers. A *k -list assignment* is a list assignment $(\ell_i)_{i \geq 1}$ such that each list is of size k , *i.e.* for all $i \geq 1$, $|\ell_i| = k$. We say that a word $w = w_1 \dots w_n$ *respects* a list assignment $(\ell_i)_{i \geq 1}$ if for all $i \in \{1, \dots, n\}$, $w_i \in \ell_i$.

3 The number of square-free words over 4 letters

We say that a word of length at least 3 is *perfect* if its suffix of length 3 contains 3 distinct letters (it is 012 up to a permutation of the alphabet). A word is *nice* if its suffix of length 3 is 010 up to a permutation of the alphabet. A square-free word is either nice or perfect. The idea is that since nice words are intuitively easier to extend than perfect words it is better to count them separately. However, since it is complicated to count them separately, we will count them together by weighting them.

For any set of words S , \widehat{S} is the quantity obtained by summing the number of perfect words in S together with $\sqrt{3} - 1$ times the number of nice words in S . For instance, if $S = \{021, 010, 101, 212\}$ then $\widehat{S} = 3\sqrt{3} - 2$.

Lemma 2. Let $\ell = (\ell_i)_{i \geq 1}$ be a 4-list assignment and let T_n be the number of square-free words of length n that respect ℓ . Let $\beta > 1$ be a real such that $1 + \sqrt{3} - \frac{1}{\beta(\beta-1)} \geq \beta$. Then, for any $n \geq 3$,

$$\widehat{T}_{n+1} \geq \beta \widehat{T}_n.$$

Proof. We proceed by induction on n . Suppose that for every $i < n$, $\widehat{T}_{i+1} \geq \beta \widehat{T}_i$. Then, for all i ,

$$\widehat{T}_n \geq \beta^i \widehat{T}_{n-i}. \quad (1)$$

We need to show $\widehat{T}_{n+1} \geq \beta \widehat{T}_n$.

A word of length $n+1$ is *good* if

- it respects ℓ ,
- its prefix of length n is in T_n ,
- and it contains no square of period 1 or 2.

A word is *wrong*, if it is good, but not square-free (i.e., if one of its suffixes is a square of period longer than 2). Let G be the set of good words and F be the set of wrong words. Then $T_{n+1} = G \setminus F$ and

$$\widehat{T}_{n+1} \geq \widehat{G} - \widehat{F}. \quad (2)$$

We will now lower bound \widehat{G} and then upper bound F to reach our result.

Let u be a perfect word from T_n and let abc be the factor of size 3 at the end of u . Let us count the contributions to G of the extensions of u . Since u is perfect the only reason for a square of period 1 or 2 to appear when adding a letter to u is if this letter is c , so we need to forbid at most one letter of ℓ_{n+1} . Moreover, if b belongs to ℓ_{n+1} then there are at least 2 ways to extend u into a perfect word of G and one way to extend u to a nice word of G . In this case, the contribution of the extensions of u to \widehat{G} is at least $2 + (\sqrt{3} - 1) = 1 + \sqrt{3}$. If b does not belong to ℓ_{n+1} then there are at least 3 ways to extend u to a perfect word of G and the contribution of the extensions of u to \widehat{G} is at least 3. Thus the contribution to \widehat{G} of the extensions of any perfect word from T_n is at least $1 + \sqrt{3}$. So the contribution of the extensions of any perfect word to \widehat{G} is at least $1 + \sqrt{3}$ times as large as its contribution to \widehat{T}_n .

For any value of the list ℓ_{n+1} , any nice word from T_n can be extended in at least 2 perfect words of G . The contribution to \widehat{G} of the extensions of any nice word from T_n is at least 2. Since the contribution of a nice word to \widehat{T}_n is $\sqrt{3} - 1$, the contribution of the extensions of any nice word to \widehat{G} is at least $\frac{2}{\sqrt{3}-1} = 1 + \sqrt{3}$ times as large as its contribution to \widehat{T}_n .

Since the contribution of the extensions of any word to \widehat{G} is at least $1 + \sqrt{3}$ times as large as its contribution to \widehat{T}_n , we deduce

$$\widehat{G} \geq (1 + \sqrt{3}) \widehat{T}_n. \quad (3)$$

Let us now bound F . For all $i \geq 1$, let F_i be the set of words from F that end with a square of period i . Clearly, $F = \cup_{i \geq 1} F_i$ and

$$\widehat{F} \leq \sum_{i \geq 1} \widehat{F}_i. \quad (4)$$

By definition of G and F , $|F_1| = |F_2| = 0$. Let $i \geq 3$ and $u \in F_i$. Since u ends with a square of period i , the last i letters of u are uniquely determined by its prefix v of size $n+1-i$. Since v is a proper prefix of u , $v \in T_{n+1-i}$. And moreover, the last 3 letters of v are identical to the last 3 letters of u . Thus the contribution of v to $\widehat{T_{n+1-i}}$ is the same as the contribution of u to $\widehat{F_i}$. We deduce

$$\widehat{F_i} = \widehat{T_{n+1-i}}.$$

Using this last equation with (1) and (4) yields

$$\widehat{F} \leq \sum_{i \geq 3} \widehat{F_i} \leq \sum_{i \geq 3} \widehat{T_{n+1-i}} \leq \sum_{i \geq 3} \widehat{T_n} \beta^{1-i} \leq \frac{\widehat{T_n}}{\beta(\beta-1)}.$$

We can use this bound and (3) in equation (8) to finally upper-bound $\widehat{T_{n+1}}$ and we obtain

$$\widehat{T_{n+1}} \geq \left(1 + \sqrt{3} - \frac{1}{\beta(\beta-1)}\right) \widehat{T_n} \geq \beta \widehat{T_n}$$

where the second inequality is a consequence of the theorem hypothesis. \square

Since $\beta = 2.45$ satisfies the condition of Lemma 2, we deduce the following theorem.

Theorem 3. *Fix a 4-list assignment $(\ell_i)_{i \geq 1}$ and let T_n be the set of square-free words of length n that respect this list assignment. Then for all $n \geq 1$,*

$$|T_n| \geq 1.45^n.$$

Proof. It is easy to verify that T_3 is not empty and thus $\widehat{T_3} > 0$. One easily verifies that $\beta = 2.45$ satisfies the conditions of Theorem 5. Thus for all $n \geq 3$,

$$\widehat{T_n} \geq \frac{\widehat{T_3}}{2.45^3} \times 2.45^n.$$

Since the weight of any words is at most 1, for any $n \geq 3$,

$$|T_n| \geq \widehat{T_n} \geq \frac{\widehat{T_3}}{2.45^3} \times 2.45^n. \quad (5)$$

For the sake of contradiction suppose that $|T_n| < 2.45^n$ for some $n > 1$. Then, there exists an integer i such that $|T_n|^i < \frac{\widehat{T_3}}{2.45^3} \times 2.45^{ni}$. Any factor of a square-free word is square-free, so the sequence $(|T_n|)_{n \geq 0}$ is submultiplicative, that is, for all $i, j \geq 1$, $|T_{i+j}| \leq |T_i||T_j|$. In particular, $|T_{ni}| \leq |T_n|^i < \frac{\widehat{T_3}}{2.45^3} \times 2.45^{ni}$ which contradicts equation (5). We deduce that for all n , $|T_n| \geq 2.45^n$, as desired. \square

Let us briefly explain why this value of $1 + \sqrt{3}$ plays a particular role in this proof. If the lists are all identical then any perfect word can be extended in two ways into another perfect word and in one way into a nice word, while any nice word can only be extended in two ways into a perfect word. The matrix of the corresponding automaton is

$$\begin{pmatrix} 2 & 1 \\ 2 & 0 \end{pmatrix}.$$

The dominant eigenvalue of this matrix is $1 + \sqrt{3}$ and one possible eigenvector is $\begin{pmatrix} 1 \\ \sqrt{3} - 1 \end{pmatrix}$ which also explains the choice of the weights. We do not need for our weights to be an eigenvector, but we require a property similar to the statement of Lemma 4. It happens to be the case that the vector that has this property is also the eigenvector of this matrix, which corresponds to the intuition that the worst choice of list assignment is the one where all the lists are identical.

4 Proof of the main result

We fix $\mathcal{A} = \{0, 1, 2, 3\}$. An $(\mathcal{A}, 3)$ -list assignment is a sequence of subsets of size 3 of \mathcal{A} . We also fix $p = 21$.

A word is *normalized* if it is the smallest of all the words obtained by a permutation of the alphabet. Let Λ be the set of normalized prefixes of minimal squares of period at most p . For any w , we let $\Lambda(w)$ be the longest word from Λ that is a suffix of w up to a permutation of the alphabet. For any set of words S , and any $w \in \Lambda$, we let $S^{(w)}$ be the set of words from S whose longest prefix that belongs to Λ up to a permutation of the alphabet is w , that is $S^{(w)} = \{u \in S : \Lambda(u) = w\}$.

We denote the set of words that contain no square of period at most p by $S_{free}^{\leq p}$. We are now ready to state the following lemma.

Lemma 4. *There exist coefficients $(C_w)_{w \in \Lambda}$ such that $C_w > 0$ and for all $v \in \Lambda$,*

$$\alpha C_v \leq \min_{\substack{l \subseteq \mathcal{A} \\ |l|=3}} \sum_{\substack{a \in l \\ va \in S_{free}^{\leq p}}} C_{\Lambda(va)} \quad (6)$$

where $\alpha = 13948/10721 \approx 1.301$.

The proof of this lemma relies on a computer verification that we delay to section 5. For the rest of this section let us fix coefficients $(C_w)_{w \in \Lambda}$ and α that respect the conditions of Lemma 4. For each set S of words, we let

$$\widehat{S} = \sum_{w \in \Lambda} C_w |S^{(w)}|.$$

Whenever we mention *the weight of a word w* , we mean $C_{\Lambda(w)}$.

The main idea of the proof of Theorem 5 is essentially the same as in Lemma 2, but we have a few more technicalities to handle along the way. We are going to count inductively the total weight of the square-free words of size n that respect a fixed 3-list assignment. Intuitively, the set Λ plays the same role as $\{010, 012\}$ and α plays the same role as $1 + \sqrt{3}$. We are now ready to state our main Theorem.

Theorem 5. *Let $\ell = (\ell_i)_{i \geq 1}$ be a $(\Sigma, 3)$ -list assignment and for all $n \geq 0$, let S_n be the set of square-free words of length n that respect ℓ . Let $\beta > 1$ be a real number such that*

$$\alpha - \frac{\beta^{1-p}}{\beta - 1} \geq \beta.$$

Then for all $n \geq 0$,

$$\widehat{S_{n+1}} \geq \beta \widehat{S_n}.$$

Proof. We proceed by induction on n . Let n be an integer such that the lemma holds for any integer smaller than n and let us show that $\widehat{S}_{n+1} \geq \beta \widehat{S}_n$.

By induction hypothesis, for all i ,

$$\widehat{S}_n \geq \beta^i \widehat{S}_{n-i}. \quad (7)$$

A word of length $n+1$ is *good* if

- it respects ℓ ,
- its prefix of length n is in S_n ,
- and it contains no square of period at most p .

A word is *wrong*, if it is good, but not square-free (i.e., if one of its suffixes is a square of period longer than p). We let G be the set of good words and F be the set of wrong words. Then $S_{n+1} = G \setminus F$ and

$$\widehat{S}_{n+1} \geq \widehat{G} - \widehat{F}. \quad (8)$$

Let us first lower-bound $\widehat{G} = \sum_{w \in \Lambda} |G^{(w)}| C_w$.

By definition, $\Lambda(v)$ is the longest suffix of v that is prefix of a square of period of length at most p (up to permutation of the alphabet). This implies that for any square-free word v and for any word u , $vu \in \mathcal{S}_{free}^{\leq p}$ if and only if $\Lambda(v)u \in \mathcal{S}_{free}^{\leq p}$. For the same reason, for any square-free word v and letter a , $\Lambda(va) = \Lambda(\Lambda(v)a)$. We then deduce that the contribution of the extensions of any word $v \in S_n$ to \widehat{G} is

$$\sum_{\substack{a \in \ell_{n+1} \\ va \in \mathcal{S}_{free}^{\leq p}}} C_{\Lambda(va)} = \sum_{\substack{a \in \ell_{n+1} \\ \Lambda(v)a \in \mathcal{S}_{free}^{\leq p}}} C_{\Lambda(\Lambda(v)a)} \geq \min_{\substack{l \subseteq \mathcal{A} \\ |l|=3}} \sum_{\substack{a \in l \\ \Lambda(v)a \in \mathcal{S}_{free}^{\leq p}}} C_{\Lambda(\Lambda(v)a)}.$$

By Lemma 4, we deduce that the contribution of the extensions of any word $v \in S_n$ to \widehat{G} is at least $\alpha C_{\Lambda(v)}$. We sum the contributions over S_n

$$\widehat{G} \geq \sum_{v \in S_n} \alpha C_{\Lambda(v)} = \sum_{u \in \Lambda} \alpha C_u |S_n^{(u)}| = \alpha \widehat{S}_n. \quad (9)$$

Let us now bound F . For all i , let F_i be the set of words from F that end with a square of period i . Then $F = \cup_{i \geq 1} F_i$ and

$$\widehat{F} \leq \sum_{i \geq 1} \widehat{F}_i. \quad (10)$$

Let us now upper-bound the \widehat{F}_i separately depending on i .

Case $i \leq p$: By definition of G and F , any word from F avoids squares of period at most p . Hence, $i \leq p$ implies $|F_i| = 0$ and $\widehat{F}_i = 0$.

Case $i > p$: Let $w \in \Lambda$ and $u \in F_i^{(w)}$. For the sake of contradiction, suppose that $|w| > i$. Let v be the period of w . Then the factor of length $i - |v|$ that ends a position $n + 1 - |v|$ is identical to the last $i - |v|$ letters of the word, that is

$$u[n + 2 - i, n + 1 - |v|] = u[n + 2 - i + |v|, n + 1].$$

These two factors are well defined since $|v| \leq p < i$. Moreover $u \in F_i^{(w)}$, and the factor of length $i - |v|$ that ends at position $n + 1 - i$ is identical to the last $i - |v|$ letters of the word, that is

$$u[n + 2 - 2i + |v|, n + 1 - i] = u[n + 2 - i + |v|, n + 1].$$

The resulting equality

$$u[n + 2 - i, n + 1 - |v|] = u[n + 2 - 2i + |v|, n + 1 - i]$$

implies that there is a square of period $i - |v| \geq 1$ in u . Since w is square free, we have $2|v| > |w| \geq i$ and $i - |v| < |v| \leq p$. So the square is of period at most p which is a contradiction. Hence if $F_i^{(w)}$ is non-empty, then $|w| \leq i$.

The suffix of length $2i$ of any word from $F_i^{(w)}$ is a square, so the last i letters of any word of $F_i^{(w)}$ are uniquely determined by the remaining prefix, and this prefix belongs to $S_{n+1-i}^{(w)}$ (since $|w| \leq i$). Hence,

$$\widehat{F}_i \leq \widehat{S_{n+1-i}} \leq \widehat{S}_n \beta^{1-i}$$

where the second inequality comes from (7). This bound with (10) yields

$$\widehat{F} \leq \sum_{i \geq p+1} \widehat{S}_n \beta^{1-i} \leq \widehat{S}_n \frac{\beta^{1-p}}{\beta - 1}.$$

We use this bound and (9) in equation (8) to finally upper-bound $\widehat{S_{n+1}}$,

$$\widehat{S_{n+1}} \geq \left(\alpha - \frac{\beta^{1-p}}{\beta - 1} \right) \widehat{S}_n \geq \beta \widehat{S}_n$$

where the second inequality is a consequence of the theorem hypothesis. \square

Since $\beta = 1.25$ satisfies the conditions of Theorem 5, we can deduce our main result.

Theorem 6. Fix a $(\Sigma, 3)$ -list assignment $(\ell_i)_{i \geq 1}$ and let S_n be the set of square-free words of length n that respect this list assignment. Then for all $n \geq 1$,

$$|S_n| \geq 1.25^n.$$

Proof. By definition, $\widehat{S}_0 = C_\varepsilon$. One easily verifies that $\beta = 1.25$ satisfies the conditions of Theorem 5. Thus for all n ,

$$\widehat{S}_n \geq C_\varepsilon \times 1.25^n.$$

Hence,

$$|S_n| \geq \frac{\widehat{S}_n}{\max_{w \in \Lambda} C_w} \geq \frac{C_\varepsilon}{\max_{w \in \Lambda} C_w} \times 1.25^n. \quad (11)$$

For the sake of contradiction suppose that $|S_n| < 1.25^n$ for some $n > 1$. By Lemma 4, $C_\varepsilon > 0$ so there exists an integer i such that $|S_n|^i < \frac{C_\varepsilon}{\max_{w \in \Lambda} C_w} \times 1.25^{ni}$. Any factor of a square-free word is square-free, so the sequence $(|S_n|)_{n \geq 0}$ is submultiplicative, that is, for all $i, j \geq 1$, $|S_{i+j}| \leq |S_i||S_j|$. In particular, $|S_{ni}| \leq |S_n|^i < C_\varepsilon \times 1.25^{ni}$ which contradicts equation (11). We deduce that for all n , $|S_n| \geq 1.25^n$, as desired. \square

5 Proof of Lemma 4

A classic procedure to compute Perron-Frobenius egeinvector (and eigenvalue) of a matrix is simply to iterate the matrix over some “random” starting vector. Our coefficients $(C_w)_{w \in \Lambda}$ plays a similar role to the role of an eigenvector and we use the same idea to compute the desired coefficients. In the case of Perron-Frobenius eigenvector, it is known that under general assumptions this procedure converges toward the desired vector. In our case, we suspect that the procedure is also convergent under rather general assumptions, but we did not try to prove anything in this direction. However, after enough iterations, we find a vector that respects the conditions of our Lemma. Before discussing the details of this procedure we restate the Lemma.

Lemma 7. *There exist coefficients $(C_w)_{w \in \Lambda}$ such that $C_\varepsilon > 0$ and for all $v \in \Lambda$,*

$$\alpha C_v \leq \min_{\substack{l \subseteq \mathcal{A} \\ |l|=3}} \sum_{\substack{a \in l \\ va \in \mathcal{S}_{free}^{\leq p}}} C_{\Lambda(va)} \quad (6)$$

where $\alpha = 13948/10721 \approx 1.301$.

It is enough to provide coefficients with the right property. Since $|\Lambda| = 854683883$, instead of providing the coefficients, it is more efficient to provide a computer program that computes the set Λ and the coefficients with the desired properties¹. This also has the advantage that the same program directly verifies that the coefficients do indeed have the desired property.

The first step is to compute the set Λ of prefixes of minimal squares of period at most p . This set is stored inside a trie (also called prefix tree). It is efficient (in particular in terms of memory consumption) since the set Λ is prefixed closed (*i.e.* Λ contains any prefix of any word from Λ). Each word of Λ is given a unique integer as an identifier. In a second step, we compute the directed multi-graph G over the vertices $\{0, \dots, |\Lambda| - 1\}$ and such that there is an arc from i to j , if $\Lambda(ua)$ is the word corresponding to j where u is the word corresponding to i and a is any letter. More precisely, if u is the word associated with the integer i and v is the word associated with j then the multiplicity of the number of arcs from i to j in our graph is given by $|\{a \in \mathcal{A} : u = \Lambda(va)\}|$. There are not many arcs with multiplicity larger than 1, but because of the normalization, this may happen (for instance, there are three arcs from the vertex of the word 0 to the vertex of the word 01, since 01 is the normalization of 02 and 03 and similarly there are four arcs from the empty word to the word 0). This graph

¹The C++ implementation can be found in the ancillary file on the arXiv. Running this program took approximately 2 hours of computation and occupied 76.4 Go of RAM.

is useful to efficiently compute for any $v \in \Lambda$, the quantity $\min_{\substack{l \subseteq \mathcal{A} \\ |l|=3}} \sum_{\substack{a \in l \\ va \in \mathcal{S}_{free}^{\leq p}}} C_{\Lambda(va)}$.

It is simply the sum of the weights of its out-neighbors minus the weight of the one of largest weight².

We consider the procedure that takes coefficients $(C_w)_{w \in \Lambda}$ as input and produces the coefficients $(C'_w)_{w \in \Lambda}$ such that for each $v \in \Lambda$

$$C'_v = \min_{\substack{l \subseteq \mathcal{A} \\ |l|=3}} \sum_{\substack{a \in l \\ va \in \mathcal{S}_{free}^{\leq p}}} C_{\Lambda(va)}.$$

For every w , we call the quantity C'_w/C_w the *growth associated to w* . If we let α be the minimum of the growth over every $w \in \Lambda$ then α and the set of coefficients C_w respect the condition of equation 6. Our goal is then simply to find coefficients $(C_w)_{w \in \Lambda}$ that gives the largest value of α .

To find, our coefficient we simply start by setting all the C_w to the same value (100000 in our implementation), and then we iterate our procedure. Between two iterations we renormalize the coefficients by dividing every coefficient by the same constant to keep the average value at some fixed value (100000 in our implementation). After 50 iterations, we find coefficients $(C_w)_{w \in \Lambda}$ and $\alpha = 13948/10721 \approx 1.301$ such that the conditions of the Lemma are verified.

We suspect that there are good reasons for which this procedure seems to converge toward the optimal. However, it is enough that we verified that after 50 iterations this deterministic procedure produces coefficients $(C_w)_{w \in \Lambda}$ with the desired property.

Let us finally mention that, every computation is carried out using integers (or pairs of integers for rational numbers) so that there are no issues of precision. It is important, since with $\alpha > 1.3$ we can take $p = 21$ and $\beta = 1.25$ in Theorem 6, but with $\alpha = 1.295$ there is no β that satisfy the condition of Theorem 5.

6 Conclusion

We showed something slightly stronger than Theorem 6. Indeed, our proof of Theorem 5 also holds if an adversary chooses each list only right before we chose the letter from the corresponding list (instead of fixing the lists from the start we have). Our proof of Theorem 3 also holds in this stronger setting. We suspect that the answer to Question 1 is positive in this stronger setting.

We showed that for any choice of lists of size 3 amongst $\{0, 1, 2, 3\}$ the number of square-free word of length n is at least 1.25^n . By pushing the computation slightly further, we can replace 1.25 with 1.28. We suspect that the number of square-free words is minimal when all the lists are identical (the growth rate of the number of square-free word of length n is known to be approximatively 1.3017 [14]). This might even be true in the stronger context where an adversary chose the next list right after we chose the next letter. Note that, it is also really simple to use the same technique to improve the bounds of Theorem 3 by computing coefficients with the aid of a computer (although looking at suffixes of length 5 or 7 instead of 3 by hand is doable and would already improve the bounds).

²A vertex u is said to be an *out-neighbor* of a vertex v if there is an arc from v to u .

Finally, let us conclude by mentioning that we believe that this approach could be used to answer positively Question 1. The size of the set Λ depends on the size of \mathcal{A} . However, since we only consider normalized words for $p = 21$ the set Λ is the same for any \mathcal{A} such that $|\mathcal{A}| \geq 21$. Let us provide a crude upper bound on the size of this set.

The number of square-free words of size n that use k (by that we mean that the k letters appear in the word) letters is less than $k \times (k-1)^{n-1}$ (we forbid square of period 1). So the number of normalized words of size n that use k letters is less than $k \times (k-1)^{n-1}/k! = (k-1)^{n-2}/(k-2)!$. The number of normalized minimal square of period n that use k letter is less than $(k-1)^{n-2}/(k-2)!$ and the number of proper prefixes of length at least n of such words is then at most $n \times (k-1)^{n-2}/(k-2)!$. By summing over n and k , the number of prefixes of minimal square-free words of period at most p over s letters is at most

$$\sum_{n=1}^p \sum_{k=1}^s n \frac{(k-1)^{n-2}}{(k-2)!}.$$

Evaluating this expression with $p = 21$ and $s = 21$ tells us that ,with $|\mathcal{A}| = 21$, the set Λ has size at most 3.4×10^{15} . This is less than 10^7 times larger than the set that required 70Go of RAM to be computed. Our bound being crude we suspect that Λ is in fact smaller than that (a more careful computation taking into account squares of period 2 divides this bounds by approximatively 4). We might be able to solve this question with this approach in a few decades. It might also be possible to exploit some other symmetries of the problem to reduce the number of words considered.

Let us finally mention that the base idea of the counting technique was recently used in more general context [13, 18]. In particular, Wanless and Wood provided a general result based on a similar idea and applied it to graph colorings, hypergraph colorings and SAT-formula [18]. It is not clear, how the more advanced argument used here (*i.e.*, counting the total weights of the solutions instead of simply counting the number of solutions) can be used in a context more general than combinatorics on words or even whether it can be used in a framework similar to the one developed by Wanless and Wood.

References

- [1] N. Alon, J. Grytczuk, M. Haluszczka, and O. Riordan. Nonrepetitive colorings of graphs. *Random Structures & Algorithms*, 21:336–346, 2002.
- [2] V. Dujmović, G. Joret, J. Kozik, and D. R. Wood. Nonrepetitive colouring via entropy compression. *Combinatorica*, 36(6):661–686, Dec 2016.
- [3] V. Dujmović, L. Esperet, G. Joret, B. Walczak, and D.R. Wood. Planar graphs have bounded nonrepetitive chromatic number. *Advances in Combinatorics*, 2020:5.
- [4] D. Gonçalves, M. Montassier, and A. Pinlou. Entropy compression method applied to graph colorings. *arXiv e-prints*, arXiv:1406.4380, 2014.
- [5] A. Gałol, G. Joret, J. Kozik and P. Micek Pathwidth and Nonrepetitive List Coloring, *Electronic Journal of Combinatorics*, 23(4), 2016.

- [6] J. Grytczuk, Nonrepetitive colorings of graphs — A survey, *Int J Math Math Sci* (2007), ArtID 74639, 10.
- [7] S. Czerwiński and J. Grytczuk. Nonrepetitive colorings of graphs. *Electronic Notes in Discrete Math.*, 28:453–459, 2007.
- [8] J. Grytczuk, J. Przybyło and X. Zhu. Nonrepetitive list colourings of paths. *Random Struct. Alg.*, 38: 162-173, 2011.
- [9] J. Grytczuk, J. Kozik and P. Micek. New approach to nonrepetitive sequences. *Random Struct. Alg.*, 42: 214-225, 2013.
- [10] J. Harant and S. Jendrol. Nonrepetitive vertex colorings of graphs. *Discrete Mathematics*, 312(2):374–380, 2012.
- [11] R. M. Kolpakov. On the number of repetition-free words. *Journal of Applied and Industrial Mathematics*, 1(4):453–462, 2007.
- [12] M. Lothaire. Algebraic combinatorics on words, volume 90 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2002
- [13] M. Rosenfeld. Another approach to non-repetitive colorings of graphs of bounded degree. *Electronic Journal of Combinatorics*, 27(3), 2020.
- [14] A.M. Shur Two-Sided Bounds for the Growth Rates of Power-Free Languages. In: *Developments in Language Theory*, vol 5583 of *Lecture Notes in Comput. Sci.*, (2009)
- [15] E. Škrabuláková. The Thue choice number versus the Thue chromatic number of graphs. *arXiv e-prints*, arXiv:1508.02559, August 2015.
- [16] A. Thue. Über unendliche Zeichenreihen. *Norske Vid. Selsk. Skr. I. Mat. Nat. Kl. Christiania*, 7:1–22, 1906.
- [17] A. Thue. Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen. *Norske Vid. Selsk. Skr. I. Mat. Nat. Kl. Christiania.*, 10:1–67, 1912.
- [18] I. M. Wanless and D. R. Wood. A general framework for hypergraph colouring. *arXiv e-prints*, arXiv:2008.00775, 2020.
- [19] D. R. Wood. Nonrepetitive Graph Colouring. *arXiv e-prints*, arXiv:2009.02001, 2020.
- [20] H. Zhao and X. Zhu. $(2 + \varepsilon)$ -nonrepetitive list colouring of paths. *Graphs Combin.*, 32(4):1635–1640, 2016.