Application of entropy compression in pattern avoidance

Pascal Ochem Alexandre Pinlou[*]

LIRMM, Université Montpellier 2, CNRS
Montpellier, France
{pascal.ochem,alexandre.pinlou}@lirmm.fr

Submitted: Jan 7, 2013; Accepted: XX; Published: XX
Mathematics Subject Classifications: 68R15

Abstract

In combinatorics on words, a word w over an alphabet Σ is said to avoid a pattern p over an alphabet Δ if there is no factor f of w such that $f = h(p)$ where $h : \Delta^* \to \Sigma^*$ is a non-erasing morphism. A pattern p is said to be k-avoidable if there exists an infinite word over a k-letter alphabet that avoids p. We give a positive answer to Problem 3.3.2 in Lothaire's book "Algebraic combinatorics on words", that is, every pattern with k variables of length at least 2^k (resp. $3 \times 2^{k-1}$) is 3-avoidable (resp. 2-avoidable). This conjecture was first stated by Cassaigne in his thesis in 1994. This improves previous bounds due to Bell and Goh, and Rampersad.

Keywords: Word; Pattern avoidance.

1 Introduction

A pattern p is a non-empty word over an alphabet $\Delta = \{A, B, C, \dots\}$ of capital letters called *variables*. An *occurrence* of p in a word w is a non-erasing morphism $h : \Delta^* \to \Sigma^*$ such that $h(p)$ is a factor of w. The avoidability index $\lambda(p)$ of a pattern p is the size of the smallest alphabet Σ such that there exists an infinite word w over Σ containing no occurrence of p. Bean, Ehrenfeucht, and McNulty [1] and Zimin [16] characterized unavoidable patterns, i.e., such that $\lambda(p) = \infty$. We say that a pattern p is t-avoidable if $\lambda(p) \leqslant t$. For more informations on pattern avoidability, we refer to Chapter 3 of Lothaire's book [8].

[*] Second affiliation: Département MIAp, Université Paul-Valéry, Montpellier 3, Route de Mende, 34199 Montpellier, France

In this paper, we consider upper bounds on the avoidability index of long enough patterns with k variables. Bell and Goh [2] and Rampersad [12] used a method based on power series and obtained the following bounds. Let $v(p)$ be the number of distinct variables of the pattern p.

Theorem 1 ([2, 12])**.** *Let p be a pattern.*

(a) If p has length at least $2^{v(p)}$ then $\lambda(p) \leqslant 4$. [2]

(b) If p has length at least $3^{v(p)}$ then $\lambda(p) \leqslant 3$. [12]

(c) If p has length at least $4^{v(p)}$ then $\lambda(p) = 2$. [12]

Our main result improves these bounds:

Theorem 2. *Let p be a pattern.*

(a) If p has length at least $2^{v(p)}$ then $\lambda(p) \leqslant 3$.

(b) If p has length at least $3 \times 2^{v(p)-1}$ then $\lambda(p) = 2$.

Theorem 2 gives a positive answer to Problem 3.3.2 of Lothaire's book [8]. As noticed by Cassaigne [5, 8], both bounds of Theorem 2 are tight. The bound $2^{v(p)}$ in Theorem 2.(a) is tight in the sense that the patterns p in the family $\{A, ABA, ABACABA, ABACABADABACABA, \dots\}$ have length $2^{v(p)} - 1$ and are unavoidable. Similarly, the bound $3 \times 2^{v(p)-1}$ in Theorem 2.(b) is tight in the sense that the patterns in the family $\{AA, AABAA, AABAACAABAA, AABAACAABAADAABAACAABAA, \dots\}$ have length $3 \times 2^{v(p)-1} - 1$ and are not 2-avoidable. Hence, this shows that the upper bound 3 of Theorem 2.(a) is best possible.

The avoidability index of every pattern with at most 3 variables is known, thanks to various results in the literature. In particular, Theorem 2 is proved for every pattern p with $v(p) \leqslant 3$:

- For $v(p) = 1$, the famous results of Thue [14, 15] give $\lambda(AA) = 3$ and $\lambda(AAA) = 2$.
- For $v(p) = 2$, every binary pattern of length at least 4 contains a square, and is thus 3-avoidable. Moreover, Roth [13] proved that every binary pattern of length at least 6 is 2-avoidable.
- For $v(p) = 3$, Cassaigne [5] began and the first author [10] finished the determination of the avoidability index of every pattern with at most 3 variables. Every ternary pattern of length at least 8 is 3-avoidable and every binary pattern of length at least 12 is 2-avoidable.

So, there remains to prove Theorem 2 for every pattern p with $v(p) \geqslant 4$.

Section 2 is devoted to some preliminary results. We prove Theorem 2.(a) in Section 3 as a corollary of a result of Bell and Goh [2]. In Section 4, we prove Theorem 2.(b) using the so-called *entropy compression method*.

Very recently, Blanchet-Sadri and Woodhouse [4] independently proved Theorem 2 using completely different methods.

2 Preliminary results

Let p be a pattern over $\Delta = \{A, B, C, \ldots\}$. An *occurrence* of p in a word w over the alphabet Σ is a non-erasing morphism $h : \Delta^* \to \Sigma^*$ such that $h(p)$ is a factor of w. Note that two distinct occurrences of p may form the same factor. For example, if $p = ABA$, then the occurrence $h = (A \to 00; B \to 1)$ of p forms the factor $h(p) = h(ABA) = h(A)h(B)h(A) = 00100$; on the other hand, $h' = (A \to 0; B \to 010)$ is a distinct occurrence of p which forms the same factor $h'(p) = h'(ABA) = h'(A)h'(B)h'(A) = 00100$.

A pattern p is *doubled* if every variable of p appears at least twice in p. A pattern p is *balanced* if it is doubled and every variable of p appears both in the prefix and the suffix of length $\left\lfloor \frac{|p|}{2} \right\rfloor$ of p. Note that if the pattern has odd length, then the variable X that appears in the middle of p (i.e. in position $\left\lfloor \frac{|p|}{2} \right\rfloor + 1$) must appear also in the prefix and in the suffix in order to make p balanced.

Claim 3. *For every integer $f \geqslant 2$, every pattern p with length at least $f \times 2^{v(p)-1}$ contains a balanced pattern p' with length at least $f \times 2^{v(p')-1}$ as a factor.*

Proof. We prove this claim by induction on $v(p)$. If $v(p) = 1$, then p has size at least $f \geqslant 2$ and is clearly balanced. Suppose this is true for some $v(p) = n$, i.e. p with n variables and length at least $f \times 2^{n-1}$ contains a balanced pattern p' as a factor with length at least $f \times 2^{v(p')-1}$. Let $v(p) = n + 1$ and let p_1 (resp. p_2) be the prefix (resp. the suffix) of p of size $\left\lfloor \frac{|p|}{2} \right\rfloor$. If p is not balanced, then there exists a variable X in p that does not occur in p_i for some $i \in \{1, 2\}$. Thus, we have $v(p_i) \leqslant v(p) - 1 = n$ and $|p_i| \geqslant f \times 2^{n-1}$. Therefore, by induction hypothesis, p contains a balanced pattern p' with length at least $f \times 2^{v(p')-1}$ as a factor. □

In the following, we will only use the fact that the pattern p' in Claim 3 is doubled instead of balanced.

3 3-avoidable long patterns

We prove Theorem 2.(a) as a corollary of the following result of Bell and Goh [2]:

Lemma 4 ([2])**.** *Every doubled pattern with at least 6 variables is 3-avoidable.*

Proof of Theorem 2.(a). We want to prove that every pattern p with length at least $2^{v(p)}$ is 3-avoidable, or equivalently, that every pattern p with $v(p) \leqslant k$ and length at least 2^k is 3-avoidable. By Claim 3, every such pattern contains a doubled pattern p' as a factor with length at least $2^{v(p')}$. So there remains to show that every doubled pattern p with $v(p) \leqslant k$ and length at least 2^k is 3-avoidable. As discussed in the introduction, the case of patterns with at most 3 variables has been settled. Now, it is sufficient to prove that doubled patterns of length at least $2^4 = 16$ are 3-avoidable.

Suppose that p_1 is a doubled pattern containing a variable X that appears at least 4 times. Replace 2 occurrences of X with a new variable to obtain a pattern p_2. Example:

We replace the first and third occurrence of B in $p_1 = ABBCDBCABDDCB$ by a new variable E to obtain $p_2 = AEBCDECABDDCB$. Then p_2 is a doubled pattern such that $|p_1| = |p_2|$ and $\lambda(p_1) \leqslant \lambda(p_2)$, since every occurrence of p_1 is also an occurrence of p_2.

Given a doubled pattern p of length at least 16, we make such replacements as long as we can. We thus obtain a doubled pattern p' of length at least 16 such that $\lambda(p) \leqslant \lambda(p')$. Moreover, every variable in p' appears either 2 or 3 times and therefore p' contains at least $\lceil 16/3 \rceil = 6$ variables. So p' is 3-avoidable by Lemma 4. Thus p is 3-avoidable, which finishes the proof. □

4 2-avoidable long patterns

We want to prove that every pattern p with length at least $3 \times 2^{v(p)-1}$ is 2-avoidable, or equivalently, that every pattern p with $v(p) \leqslant k$ variables and length at least $3 \times 2^{k-1}$ is 2-avoidable. By Claim 3, every such pattern contains a doubled pattern p' as a factor with length at least $3 \times 2^{v(p')-1}$. So there remains to show that every doubled pattern p with $v(p) \leqslant k$ and length at least $3 \times 2^{k-1}$ is 2-avoidable.

As discussed in the introduction, the case of patterns with at most 3 variables has been settled. Now, it is sufficient to prove Theorem 2.(b) for doubled patterns with at least 4 variables.

Let $\Sigma = \{0, 1\}$ be the alphabet. For the remaining of this section, let $k \geqslant 4$ and $q(k) = 3 \times 2^{k-1}$.

Suppose by contradiction that there exists a doubled pattern p on k variables and length at least $q(k)$ that is not 2-avoidable. Then there exists an integer n such that every word $w \in \Sigma^n$ contains p. We put an arbitrary order on the k variables of p and call A_j the j-th variable of p.

4.1 The algorithm AvoidP

Let $V \in \{0, 1\}^t$ be a vector of length t. The algorithm AvoidP takes the vector V as input and returns a word w avoiding p and a data structure R that is called a *record* in the remaining of the paper.

The way we encode information in R at lines 5 and 7 will be explained in Subsection 4.2.

In the algorithm AvoidP, let w_i be the word w after i steps. Clearly, w_i avoids p at each step. By contradiction hypothesis, the resulting word w of the algorithm (that is w_t) has length less than n. We will prove that each output of the algorithm allows to determine the input. Then we obtain a contradiction by showing that the number of possible outputs is strictly smaller than the number of possible inputs when t is chosen large enough compared to n. This implies that every pattern p with at most k variables and length at least $q(k)$ is 2-avoidable.

To analyze the algorithm, we borrow ideas from graph coloring problems [6, 7]. These

```
Algorithm 1: AvoidP
  Input  : $V$.
  Output: $w$ (a word avoiding $p$) and $R$ (a data structure).

  $w \leftarrow \epsilon$
  $R \leftarrow \emptyset$
  for $i \leftarrow 1$ to $t$ do
      Append $V[i]$ (the $i$-th letter of $V$) to $w$
      Encode in $R$ that a letter has been appended to $w$
      if $w$ contains a factor of length $\ell$ corresponding to an occurrence of $p$ then
          Encode in $R$ the occurrence of $p$
          Erase the suffix of length $\ell$ of $w$
  return $R$, $w$
```

results are based on the Moser-Tardos [9] entropy-compression method which is an algorithmic proof of the Lovász Local Lemma.

4.2 The record R

An important part of the algorithm is to update the record R at each step of the algorithm. Let R_i be the record after i steps of the algorithm AvoidP. On one hand, given V as input of the algorithm, this produces a pair (R_t, w_t). On the other hand, given a pair (R_t, w_t), we will show in Lemma 6 that we can recover the entire input vector V. So, each input vector V produces a distinct pair (R_t, w_t).

Let $\mathcal{V}$ be the set of input vectors V of size t, let $\mathcal{R}$ be the set of records R produced by the algorithm AvoidP and let $\mathscr{O}$ be the set of different outputs (R_t, w_t). After the execution of the algorithm (t steps), w_t avoids p by definition and therefore $|w_t| < n$ by contradiction hypothesis. Hence, the number of possible final words w_t is independent from t (it is at most 2^n). We then clearly have $|\mathscr{O}| \leqslant 2^n \times |\mathcal{R}|$. We will prove that $|\mathcal{V}| \leqslant |\mathscr{O}|$ and that $|\mathcal{R}| = o(2^t)$ to obtain the contradiction $2^t = |\mathcal{V}| \leqslant |\mathscr{O}| \leqslant 2^n \times |\mathcal{R}| = o(2^t)$.

The record R is a triplet $R = (D, L, X)$ where D is a binary word (each element is 0 or 1), L is a vector of $(k-1)$-sets of non-zero integers and X is a binary word. At the beginning, D, L and X are empty. At step i of the algorithm, we append $V[i]$ to w_{i-1} to get w'_i.

If w'_i contains no occurrence of p, then we append 0 to D to get R_i and we set $w_i = w'_i$. Otherwise, suppose that w'_i contains an occurrence h of p that forms a factor $h(p)$ of length ℓ, that is, the suffix of length ℓ of w'_i is $h(p)$. Recall that A_j is the j-th variable of p. For $1 \leqslant j \leqslant k-1$, let $z_j = |h(A_1 \dots A_j)|$. Let $L' = \{z_1, z_2, \dots, z_{k-1}\}$ be a $(k-1)$-set of non-zero integers. To get R_i, we append the factor 01^ℓ to D; we add L' as the last element of L; and we append the factor $h(A_1 A_2 \dots A_k)$ to X.

Example 5.

Let us give an example with $k = 3$, $p = ACBBCBBABCAB$ and $V = [0,0,1,0,0,1,1,0,0,1,1,1,0,0,1,1,0,1,1,1,0,0,0,1,1,0]$. The variables of p were initially ordered as (A,B,C). For the first 24 steps, no occurrence of p appeared, so at each step $i \leqslant 24$, we append $V[i]$ to w_{i-1} and we append one 0 to D. Hence, at step 24, we have:

- $w_{24} = 001001100111001101110001$
- $R_{24} = \begin{cases} D &= 000000000000000000000000 = 0^{24} \\ L &= [\,] \\ X &= \epsilon \end{cases}$

Now, at step 25, we first append $V[25] = 1$ to w_{24} to get w'_{25}. The word w'_{25} contains an occurrence $h = (A \to 01; B \to 1; C \to 100)$ of p which forms a factor of length 21 (the 21 last letters of w'_{25}). Then we set $L' = \{|h(A)|, |h(AB)|\} = \{2,3\}$. We obtain w_{25} from w'_{25} by erasing its suffix of length 21. To get R_{25}, we append the factor 01^{21} to D, we add L' as the last element of L, and we append the factor $h(ABC) = 011100$ to X. This gives:

- $w_{25} = 0010$
- $R_{25} = \begin{cases} D &= 0000000000000000000000000111111111111111111111 = 0^{25}1^{21} \\ L &= [\{2,3\}] \\ X &= 011100 \end{cases}$

Let V_i be the vector V restricted to its i first elements. We will show that the pair (R_i, w_i) at some step i allows to recover V_i.

Lemma 6. *After i steps of the algorithm* AvoidP*, the pair (R_i, w_i) permits to recover V_i.*

Proof. Before step 1, we have $w_0 = \epsilon$, $R_0 = (\epsilon, [\,], \epsilon)$, and $V_0 = \epsilon$. Let $R_i = (D, L, X)$ be the record after step i, with $1 \leqslant i \leqslant t$.

Suppose that 0 is a suffix of D. This means that at step i, no occurrence of p was found: the algorithm appended $V[i]$ to w_{i-1} to get w_i. Therefore $V[i]$ is the last letter of w_i, say x. Then the word w_{i-1} is obtained from w_i by erasing the last letter and the record R_{i-1} is obtained from R_i by removing the suffix 0 of D. We recover V_{i-1} from (R_{i-1}, w_{i-1}) by induction hypothesis and we obtain $V_i = V_{i-1} \cdot x$.

Suppose now that 01^ℓ is a suffix of D. This means that an occurrence h of p has been created during step i such that $|h(p)| = \ell$. Let L' be the last element of L which is a $(k-1)$-set $L' = \{z_1, z_2, \ldots, z_{k-1}\}$. By construction of L', we have $|h(A_1)| = z_1$ and $|h(A_s)| = z_s - z_{s-1}$ for $2 \leqslant s \leqslant k-1$. We know the pattern p, the total length of the factor $h(p)$ (that is ℓ) and the lengths of the $k-1$ first variables of p in $h(p)$, so we are able to compute $|h(A_k)|$. Now, we can parse the suffix of length $\sum_{1\leqslant j\leqslant k}|h(A_j)|$ of X, which is the factor $h(A_1\ldots A_k)$, to obtain the factors $h(A_1),\ldots,h(A_k)$. Thus, we have recovered the occurrence h of p.

Now, w_{i-1} is obtained by removing the last letter x of $w_i\cdot h(p)$. This letter x is $V[i]$, the letter appended to w_{i-1} at step i to get w'_i. The record R_{i-1} is obtained from R_i as follows: remove the suffix 01^ℓ from D, remove the last element of L, and remove the suffix $h(A_1\ldots A_k)$ of X. We recover V_{i-1} from (R_{i-1},w_{i-1}) by induction hypothesis and we obtain $V_i=V_{i-1}\cdot x$.

□

The previous lemma proves that distinct input vectors cannot correspond to the same pair (R_t,w_t). So we get $|\mathcal{V}|\leqslant|\mathscr{O}|$.

4.3 Analysis of $\mathcal{R}$

Now we compute $|\mathcal{R}|$. Let $R=R_t=(D,L,X)$ be a given record produced by an execution of AvoidP. Let $\mathcal{D}$ be the set of such binary words D. For a given $D\in\mathcal{D}$, let $\mathcal{L}_D$ be the set of such vectors of $(k-1)$-sets of non-zero integers L compatible with D. Let $\mathcal{X}$ be the set of such binary words X.

We thus have $|\mathcal{R}|\leqslant|\mathcal{D}|\times\max_{D\in\mathcal{D}}|\mathcal{L}_D|\times|\mathcal{X}|$.

Let us give some useful information in order to get upper bounds on $|\mathcal{D}|$, $|\mathcal{X}|$, and $|\mathcal{L}_D|$. The algorithm runs in t steps. At each step, one letter is appended to w, so t letters have been appended and therefore the number of erased letters during the execution of the algorithm is $t-|w_t|$. At some steps, an occurrence h of p appears and the factor $h(p)$ is immediately erased. Let m be the number of erased factors during the execution of the algorithm. Let $h_i(p)$, $1\leqslant i\leqslant m$, be the m erased factors. We have $|h_i(p)|\geqslant q(k)$ since each variable of p is a non-empty word and p has length at least $q(k)$. Moreover, we have $\sum_{1\leqslant i\leqslant m}|h_i(p)|=t-|w_t|\leqslant t$. Each time a factor $h_i(p)$ is erased, we add an element to L, so $|L|=m$.

4.3.1 Analysis of $\mathcal{D}$

In the binary word D, each 0 corresponds to an appended letter during the execution of the algorithm and each 1 corresponds to an erased letter. Therefore, D has length $2t-|w_t|$. Observe that every prefix in D contains at least as many 0's as 1's. Indeed, since a 1 corresponds to an erased letter x, this letter x had to be added first and thus there is a 0 before that corresponds to this 1. The word D is therefore a partial Dyck word. Since any erased factor $h_i(p)$ has length at least $q(k)$, any maximal sequence of 1's (which is called a *descent* in the sequel) in D has length at least $q(k)$. So D is a partial Dyck words with t 0's such that each descent has length at least $q(k)$.

Let $C_{t,r,d}$ (resp. $C_{t,d}$) be the number of partial Dyck words with t 0's and $t-r$ 1's (resp. Dyck words of length $2t$) such that all descents have length at least d.

Lemma 7. $C_{t,r,d}\leqslant C_{t+d,d}$.

Proof. We map every partial Dyck word y with t 0's and $t-r$ 1's to the Dyck word $y0^d1^{d+r}$, which has $t+d$ 0's and $t+d$ 1's. Since d is fixed, this mapping is injective. This proves the lemma. $\square$

If $q(k) \geqslant d$, then we have $|\mathcal{D}| \leqslant C_{t,|w_t|,q(k)} \leqslant C_{t,|w_t|,d} \leqslant C_{t+d,d}$ by Lemma 7. Let $\phi_d(x) = 1 + \sum_{j \geqslant d} x^j = 1 + \frac{x^d}{1-x}$. The radius of convergence of ϕ_d is 1. The following lemma comes from a more general statement of Esperet and Parreau [7] and gives an upper bound on $|\mathcal{D}|$.

Lemma 8. *[7] Let d be an integer such that the equation $\phi_d(x) - x\phi_d'(x) = 0$ has a solution τ with $0 < \tau < 1$. Then τ is the unique solution of the equation in the open interval $(0,1)$. Moreover, there exists a constant c_d such that $C_{t,d} \leqslant c_d \gamma_d^t t^{-\frac{3}{2}}$ where $\gamma_d = \phi_d'(\tau) = \frac{\phi_d(\tau)}{\tau}$.*

The equation $\phi_d(x) - x\phi_d'(x) = 0$ is equivalent to $P(x) = (1-x)^2 + (1-d)x^d + (d-2)x^{d+1} = 0$. Since $P(0) = 1$ and $P(1) = -1$, $P(x) = 0$ has a solution τ in the open interval $(0,1)$. By Lemma 8, this solution is unique and, for some constant c_d, we have $C_{t+d,d} \leqslant c_d \gamma_d^{t+d} (t+d)^{-\frac{3}{2}}$ with $\gamma_d = \phi_d'(\tau)$. We clearly have $C_{t+d,d} = o(\gamma_d^t)$. So, we can compute γ_d for d fixed. We will use the following bounds: $\gamma_{24} \leqslant 1.27575$ and $\gamma_{48} \leqslant 1.15685$.

So, by Lemmas 7 and 8, when t is large enough, we have $|\mathcal{D}| < 1.27575^t$ (resp. $|\mathcal{D}| < 1.15685^t$) if the length of any descent is at least 24 (resp. 48).

4.3.2 Analysis of $\mathcal{X}$

Each erased factor $h_i(p)$ adds $|h_i(A_1 \ldots A_k)|$ letters to X. Since p is doubled, we have $|h_i(p)| \geqslant 2|h_i(A_1 \ldots A_k)| + q(k) - 2k \geqslant 2|h_i(A_1 \ldots A_k)| + 24 - 2 \times 4$. This gives $|h_i(A_1 \ldots A_k)| \leqslant \frac{|h_i(p)|}{2} - 8$. Since $\sum_{1 \leqslant i \leqslant m} |h_i(p)| \leqslant t$, we have $|X| = \sum_{1 \leqslant i \leqslant m} |h_i(A_1 \ldots A_k)| \leqslant \sum_{1 \leqslant i \leqslant m} \left(\frac{|h_i(p)|}{2} - 8 \right) \leqslant \frac{t}{2} - 8m$. Therefore $|\mathcal{X}| \leqslant 2^{\frac{t}{2} - 8m + 1} \leqslant (\sqrt{2})^t$.

4.3.3 Analysis of $\mathcal{L}_D$

For a given $R = (D, L, X)$, the vector L is dependent on the partial Dyck word D. Indeed, by construction, the i-th element of L is a $(k-1)$-set of integers smaller than $\frac{\ell}{2}$ where ℓ is the length of the i-th descent of D. In this subsection, we compute an upper bound on the number of vectors L compatible with D for a given $D \in \mathcal{D}$ and thus we give an upper bound on $|\mathcal{L}_D|$.

Each element $L_i = \{z_1, z_2, \ldots, z_{k-1}\}$ of L corresponds to the erased factor $h_i(p)$ and by construction we have $|h_i(A_1 \ldots A_j)| = z_j$. By construction of D, $|h_i(p)|$ is the length of the i-th descent of D. Since D is fixed, $|h_i(p)|$ is fixed for every $1 \leqslant i \leqslant m$.

Let $s_k(\ell)$ be the number of such $(k-1)$-sets L_i that correspond to factors of length ℓ. Recall that $|h_i(p)| \geqslant q(k)$, so $s_k(\ell)$ is defined for $k \geqslant 4$ and $\ell \geqslant q(k)$. Each of the m elements of L corresponds to an erased factor, so $|\mathcal{L}_D| \leqslant s_k(|h_1(p)|) \times s_k(|h_2(p)|) \times \ldots \times s_k(|h_m(p)|)$. Let $g_k(\ell) = s_k(\ell)^{\frac{1}{\ell}}$ be defined for $k \geqslant 4$ and $\ell \geqslant q(k)$. Then $|\mathcal{L}_D| \leqslant$

$g_k(|h_1(p)|)^{|h_1(p)|} \times g_k(|h_2(p)|)^{|h_2(p)|} \times \ldots \times g_k(|h_m(p)|)^{|h_m(p)|}$. So, if we are able to upper-bound $g_k(\ell)$ by some constant c for all $\ell \geqslant q(k)$, then we get $|\mathcal{L}_D| \leqslant c^{|h_1(p)|} \times c^{|h_2(p)|} \times \ldots \times c^{|h_m(p)|} \leqslant c^t$.

Now we bound $g_k(\ell)$ using two different methods depending on the number k of variables in p and the length $q(k)$ of p.

4.3.3.1 Bound on $g_k(\ell)$ for $k = 4$, $\ell \geqslant 96$ or $k \geqslant 5$, $\ell \geqslant q(k)$

As shown in Section 4.3.2, we have $|h_i(A_1 \ldots A_k)| \leqslant \frac{|h_i(p)|}{2} - 8$. For a given $L_i = \{z_1, z_2, \ldots, z_{k-1}\}$ that corresponds to $h_i(p)$, we thus have $z_{k-1} = |h_i(A_1 \ldots A_{k-1})| \leqslant \frac{|h_i(p)|}{2} - 9$. Therefore, L_i is a set of $(k-1)$ distinct integers between 1 and $\frac{|h_i(p)|}{2} - 9$. So $s_k(\ell) \leqslant \binom{\lfloor \ell/2 \rfloor}{k-1}$ and $g_k(\ell) \leqslant \binom{\lfloor \ell/2 \rfloor}{k-1}^{\frac{1}{\ell}}$. We can upper-bound $g_k(\ell)$ by $\overline{g_k}(\ell) = \left(\frac{(\ell/2)^{k-1}}{(k-1)!}\right)^{\frac{1}{\ell}}$ for $\ell \geqslant q(k)$.

Let us show that when k is fixed, $\overline{g_k}(\ell)$ is a decreasing function of ℓ for $\ell \geqslant q(k)$. The derivative $(\overline{g_k}(\ell))' = \overline{g_k}(\ell) \times \frac{1}{\ell^2} \times \left(k - 1 - \ln\left(\frac{(\ell/2)^{k-1}}{(k-1)!}\right)\right)$ is negative if and only if $k - 1 < \ln\left(\frac{(\ell/2)^{k-1}}{(k-1)!}\right)$, that is, if and only if $(k-1)!e^{k-1} < (\ell/2)^{k-1}$. This inequality holds since $(k-1)!e^{k-1} < ((k-1)e)^{k-1} < \left(3 \times 2^{k-2}\right)^{k-1} \leqslant (\ell/2)^{k-1}$.

We also have that $\overline{g_k}(q(k))$ is a decreasing function of k for $k \geqslant 4$ since we have checked using Maple that the only zero of its derivative is at $k \approx 3.37$ and that its derivative is negative for $k \geqslant 3.38$.

Thus, we get $g_k(\ell) < \overline{g_k}(\ell) \leqslant \overline{g_k}(q(k)) \leqslant \overline{g_5}(48) < 1.21973$ for all $k \geqslant 5$ and $\ell \geqslant q(k)$, and we get $g_4(\ell) < \overline{g_4}(\ell) \leqslant \overline{g_4}(96) < 1.10773$ for all $\ell \geqslant 96$. We chose the value 96 to distinguish between the cases, because it is the smallest value such that the argument holds.

4.3.3.2 Bound on $g_4(\ell)$ for $24 \leqslant \ell \leqslant 95$

The second method to bound the size of $g_4(\ell)$ is based on ordinary generating functions (OGF). Here, $k = 4$, so let A_1, A_2, A_3, A_4 be the four variables of p and let a_i be the number of instances of A_i in p. Therefore, $a_1 + a_2 + a_3 + a_4 = |p|$. Recall that each variable appears at least twice in p since p is doubled, so $a_i \geqslant 2$. Moreover, a factor of length ℓ, with $24 \leqslant \ell \leqslant 95$, necessarily corresponds to an occurrence of a pattern of length between 24 and 95. So we just have to consider patterns p with $24 \leqslant |p| \leqslant 95$.

Given $L_i = \{z_1, z_2, z_3\}$ an element of L corresponding to $h_i(p)$, we have $|h_i(A_1)| = z_1$, $|h_i(A_2)| = z_2 - z_1$, $|h_i(A_3)| = z_3 - z_2$ and $|h_i(A_4)| = \frac{|h_i(p)| - (a_1|h_i(A_1)| + a_2|h_i(A_2)| + a_3|h_i(A_3)|)}{a_4}$. Let $\mathcal{A}_p = \sum_{j \geqslant |p|} b_j\ x^j$ be the OGF of such sets L', i.e. b_j is the number of 3-sets $\{z_1, z_2, z_3\}$ that corresponds to a factor of length j formed by an occurrence of p. In other words, b_j is the number of 4-tuples $(\ell_1, \ell_2, \ell_3, \ell_4)$ such that $a_1 \times \ell_1 + a_2 \times \ell_2 + a_3 \times \ell_3 + a_4 \times \ell_4 = j$ and with $\ell_i \geqslant 1$ (since each variable of p corresponds to a non-empty word). So by definition of h_4, we have $h_4(\ell) = b_\ell$ and thus $g_4(\ell) = b_\ell^{\frac{1}{\ell}}$.

This kind of OGF has been studied and is similar to the well-known problem of counting the number of ways you can change a dollar [11]: you have only five types of coins (pennies, nickels, dimes, quarters, and half dollars) and you want to count the number of ways you can change any amount of cents. So, let $\mathcal{C} = \sum_{j\geqslant 1} c_j \; x^j$ be the OGF of the problem and thus any c_j is the number of ways you can change j cents. Then, for example, c_{100} corresponds to the number of ways you can change a dollar. Here, $\mathcal{C} = \frac{1}{1-x} \times \frac{1}{1-x^5} \times \frac{1}{1-x^{10}} \times \frac{1}{1-x^{25}} \times \frac{1}{1-x^{50}}$.

In our case, we have four coins with value a_1, a_2, a_3, and a_4 respectively (so we can have different types of coins with the same value) and each type of coins appears at least once (since $\ell_i \geqslant 1$). Thus we get $\mathcal{A}_p = \sum_{j\geqslant |p|} b_j \; x^j = \frac{x^{a_1}}{1-x^{a_1}} \times \frac{x^{a_2}}{1-x^{a_2}} \times \frac{x^{a_3}}{1-x^{a_3}} \times \frac{x^{a_4}}{1-x^{a_4}}$. We use Maple for our computation. For each $24 \leqslant |p| \leqslant 95$, for each 4-tuple (a_1, a_2, a_3, a_4) such that $\sum a_i = |p|$, we consider the associated OGF $\mathcal{A}_p$ and we compute, using Maple, the truncated series expansion up to the order 95, that gives $\mathcal{A}_p = b_{24}x^{24} + b_{25}x^{25} + \ldots + b_{95}x^{95} + O(x^{96})$ with explicit values for the coefficients b_j. So, for any $24 \leqslant \ell \leqslant 95$, $g_4(\ell)$ is upper-bounded by the maximum of $b_\ell^{\frac{1}{\ell}}$ taken over all $\mathcal{A}_p$. Maple gives that $b_\ell^{\frac{1}{\ell}}$ is maximal for $|p| = 24$, $(a_1, a_2, a_3, a_4) = (2, 2, 2, 18)$, and $\ell = 46$: in this case, $b_{46} = 84$ (i.e. there exist 84 distinct 3-sets L_i that correspond to some factor of length 46 formed by an occurrence of a pattern of length 24 where three variables appear twice and one variable appears 18 times). So, $g_4(\ell) \leqslant 84^{\frac{1}{46}} < 1.10112$ for all $24 \leqslant \ell \leqslant 95$.

4.3.3.3 Bound on $g_k(\ell)$ for all $k \geqslant 4$

We can deduce from Paragraphs 4.3.3.1 and 4.3.3.2 the following.

If $k = 4$, then $g_4(\ell) < 1.10112$ for $24 \leqslant \ell \leqslant 95$ and $g_4(\ell) < 1.10773$ for $\ell \geqslant 96$. So for $k = 4$, we have $|\mathcal{L}_D| < (1.10773)^t$.

If $k \geqslant 5$, then $g_k(\ell) < 1.21973$ for $\ell \geqslant q(k)$. So for $k \geqslant 5$, we have $|\mathcal{L}_D| < (1.21973)^t$.

4.4 End of the proof

The bounds on $|\mathcal{L}_D|$ obtained in Subsection 4.3.3 hold for any fixed $D \in \mathcal{D}$. So they also hold for $\max_{D\in\mathcal{D}} |\mathcal{L}_D|$.

Aggregating the above analysis, we get the following. For $k \geqslant 5$, we have $q(k) \geqslant 48$: then $|\mathcal{R}| \leqslant |\mathcal{D}| \times \max_{D\in\mathcal{D}} |\mathcal{L}_D| \times |\mathcal{X}| \leqslant (1.15685 \times 1.21973 \times \sqrt{2})^t = o(2^t)$. For $k = 4$, we have $q(k) \geqslant 24$: then $|\mathcal{R}| \leqslant |\mathcal{D}| \times \max_{D\in\mathcal{D}} |\mathcal{L}_D| \times |\mathcal{X}| \leqslant (1.27575 \times 1.10773 \times \sqrt{2})^t = o(2^t)$.

Thus for all $k \geqslant 4$, $|\mathcal{R}| = o(2^t)$ and so we obtain the desired contradiction:

$$2^t = |\mathcal{V}| \leqslant |\mathscr{O}| \leqslant 2^n \times |\mathcal{R}| = 2^n \times o(2^t) = o(2^t).$$

5 Conclusion

In our results, we heavily use the fact that the patterns are doubled. The fact that the patterns are long is convenient for our proofs but does not seem so important. So we ask whether every doubled pattern is 3-avoidable. By the remarks in Section 1 and by Lemma 4, the only remaining cases are doubled patterns with 4 and 5 variables. Also, does there exist a finite k such that every doubled pattern with at least k variables is

2-avoidable ? Using the standard backtracking algorithm, we have checked by computer that ABCCBADD is not 2-avoidable. So we know that such a k is at least 5.

Acknowledgments

We are grateful to the anonymous referees for their helpful suggestions to improve the presentation of the paper.

References

[1] D.R. Bean, A. Ehrenfeucht, and G.F. McNulty, Avoidable Patterns in Strings of Symbols, *Pacific J. of Math.* **85** (1979) 261–294.

[2] J. Bell, T. L. Goh. Exponential lower bounds for the number of words of uniform length avoiding a pattern, *Inform. and Comput.* **205** (2007), 1295-1306.

[3] J. Berstel. Axel Thue's work on repetitions in words. Invited Lecture at the 4th Conference on Formal Power Series and Algebraic Combinatorics, Montreal, 1992, June 1992. Available at `http://www-igm.univ-mlv.fr/~berstel/index.html`.

[4] F. Blanchet-Sadri, B. Woodhouse. Strict Bounds for Pattern Avoidance. *Theor. Comput. Sci.* **506** (2013), 17–27.

[5] J. Cassaigne. Motifs évitables et régularité dans les mots, Thèse de Doctorat, Université Paris VI, Juillet 1994.

[6] V. Dujmović, G. Joret, J. Kozik, and D. R. Wood. Nonrepetitive Colouring via Entropy. *Combinatorica*, to appear, 2013+ (Also available on arXiv:1112.5524).

[7] L. Esperet and A. Parreau. Acyclic edge-coloring using entropy compression. *European Journal of Combinatorics* **36(4)** (2013), 1019–1027.

[8] M. Lothaire. Algebraic Combinatorics on Words. *Cambridge Univ. Press* (2002).

[9] R. A. Moser, G. Tardos. A constructive proof of the general Lovasz local lemma. *J. ACM*, **57(2)** (2010), p. 11:1-11:15.

[10] P. Ochem. A generator of morphisms for infinite words. *RAIRO: Theoret. Informatics Appl.* **40** (2006) 427–441.

[11] G. Pólya, R. E. Tarjan, D. R. Woods. Notes on Introductory Combinatorics. *Progress in Computer Science*, Birkhäuser (1983).

[12] N. Rampersad. Further applications of a power series method for pattern avoidance. *Electron. J. Combinatorics.* **18(1)** (2011), #P134.

[13] P. Roth. Every binary pattern of length six is avoidable on the two-letter alphabet. *Acta Inform.* **29** (1992), 95–107.

[14] A. Thue. Über unendliche Zeichenreihen. *Norske Vid. Selsk. Skr. I. Mat. Nat. Kl. Christiania*, 7:1–22, 1906.

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

[16] A.I. Zimin. Blocking sets of terms. *Math. USSR Sbornik* **47(2)** (1984) 353–364. English translation.