Theoretical Computer Science 571 (2015) 50–57

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

# Searching for Zimin patterns

Wojciech Rytter [a,1], Arseny M. Shur [b,*,2]

[a] Institute of Informatics, Warsaw University, Poland
[b] Institute of Mathematics and Computer Science, Ural Federal University, Ekaterinburg, Russia

ARTICLE INFO

*Article history:*
Received 22 May 2014
Received in revised form 12 November 2014
Accepted 10 January 2015
Available online 16 January 2015
Communicated by R. Giancarlo

*Keywords:*
Zimin word
Unavoidable pattern
On-line algorithm
Fibonacci word

ABSTRACT

In the area of pattern avoidability the central role is played by special words called Zimin patterns. The symbols of these patterns are treated as variables and the rank of the pattern is its number of variables. Zimin type of a word $x$ is introduced here as the maximum rank of a Zimin pattern matching $x$. We show how to compute Zimin type of a word on-line in linear time. Consequently we get a quadratic time, linear-space algorithm for searching Zimin patterns in words. Then we demonstrate how the Zimin type of the length $n$ prefix of the infinite Fibonacci word is related to the representation of $n$ in the Fibonacci numeration system. Using this relation, we prove that Zimin types of such prefixes and Zimin patterns inside them can be found in logarithmic time. Finally, we give some upper bounds on the function $f(n,k)$ such that every $k$-ary word of length at least $f(n,k)$ has a factor that matches the rank $n$ Zimin pattern.



## 1. Introduction

Pattern avoidability is a well-established area studying the problems involving words of two kinds: "usual" words over the alphabet of constants and *patterns* over the alphabet of variables.[3] A pattern $X$ *embeds* in a word $w$ if $w$ has a factor of the form $h(X)$ where $h$ is a non-erasing morphism. An *unavoidable* pattern is a pattern that embeds in any long enough word over any finite alphabet. In the problem of pattern (un)avoidability the crucial role is played by Zimin words [10]. The Zimin word (or *Zimin pattern*) of rank $k$ is defined as follows:

$$\forall k > 1 \quad Z_k = Z_{k-1} \cdot x_k \cdot Z_{k-1}, \quad \text{and} \quad Z_1 = x_1.$$

Hence

$$Z_2 = x_1x_2x_1, \qquad Z_3 = x_1x_2x_1\, x_3\, x_1x_2x_1, \qquad Z_4 = x_1x_2x_1x_3x_1x_2x_1\, x_4\, x_1x_2x_1x_3x_1x_2x_1.$$

The seminal result in the area is the unavoidability theorem by Bean, Ehrenfeucht, McNulty, and Zimin ([2,10]; see [9] for an optimized proof). The theorem contains two conditions equivalent to unavoidability of a pattern $X$ with $k$ variables. The first condition is the existence of a successful computation in some nondeterministic reduction procedure on $X$, and the second, more elegant, condition says that $X$ embeds in the word $Z_k$. On the other hand, it is still a big open problem whether

* Corresponding author.
*E-mail addresses:* rytter@mimuw.edu.pl (W. Rytter), arseny.shur@urfu.ru (A.M. Shur).

1 Supported by the grant NCN2014/13/B/ST6/00770 of the Polish Research Center.
2 Supported under the Agreement 02.A03.21.0006 of 27.08.2013 between the Ministry of Education and Science of the Russian Federation and Ural Federal University, and by the grant 13-01-00852 of the Russian Foundation of Basic Research.

3 In a more general setting, which is not discussed in this paper, patterns may contain constants along with variables.

http://dx.doi.org/10.1016/j.tcs.2015.01.004
0304-3975/

unavoidability of a pattern can be checked in the time polynomial in its length [4, Problem 17]. This problem belongs to NP and is tractable for a fixed $k$. The general case is strongly suspected to be NP-complete, though no proof has been given.

Another natural computationally hard problem concerning avoidability is the embedding problem: given a word and a pattern, decide whether the pattern embeds in the word. This problem is NP-complete; Angluin [1] proved this fact for patterns with constants, but his proof can be adjusted for the patterns without constants as well. Note that the unavoidability problem does not refer to a (potentially long) Zimin word as a part of the input, and hence is not a particular case of the embedding problem. On the other hand, the inverse problem of embedding a Zimin pattern in a given word is a particular case of the embedding problem. Here we show that this particular case is quite simple.

In the first part of the paper (Section 2) we address the following decision problem:

**Searching Zimin patterns**

*Input:* a word $w$ and an integer $k$;
*Output: yes* if $Z_k$ embeds in $w$.

We give an algorithm solving this problem in quadratic time and linear space. The main step of the algorithm is an online linear-time computation of the characteristic we call *Zimin type* of a word. Zimin type of a finite word $w$ is the maximum number $k$ such that $w$ is an image of the Zimin word $Z_k$ under a non-erasing morphism. By definition, the empty word has Zimin type 0. In Lothaire's book, Zimin types of words are referred to as the orders of sesquipowers [6, Ch. 4].

**Example 1.1.** Zimin type of $u = adbadccccadbad$ is 3, because $u$ is the image of $Z_3$ under the morphism

$$x_1 \to ad, \qquad x_2 \to b, \qquad x_3 \to cccc.$$

The Zimin decomposition of $u$ is: $u = ad\, b\, ad\, cccc\, ad\, b\, ad$.

The answer of **Searching Zimin patterns** for $k = 3$ and the word $w = ccccadbadccccadbadccccc$ is *yes* ($w$ has the word $u$ of Zimin type 3 as a factor), but for $k = 3$ and $w = aaabbaabbaa$ the answer is *no*.

In the second part of the paper (Section 3) we study Zimin types and the embeddings of Zimin patterns for Fibonacci words. First we relate the type of the length $n$ prefix of the infinite Fibonacci word to the representation of $n$ in the Fibonacci numeration system (Theorem 3.2). This result and the fact that for Fibonacci words Zimin types of prefixes dominate Zimin types of other factors (Theorem 3.5) allow us to solve **Searching Zimin patterns** for this particular case in logarithmic time (Theorem 3.8).

In the last part of the paper (Section 4) we consider a couple of combinatorial problems. In Section 4.1 we analyze the fastest possible growth of the sequence of Zimin types for the prefixes of an infinite word. Finally, in Section 4.2 we give some results on the length such that the given Zimin pattern embeds in any word of this length over a given alphabet.

## 2. Algorithmic problems

### 2.1. Recurrence for Zimin types of prefixes of a word

Recall that a factor (prefix, suffix) of a word is called *proper* if it does not coincide with the word itself. A *border* of a word $w$ is any word that is both a proper prefix and a proper suffix of $w$. Thus, any border of a border of $w$ is a border of $w$ as well. We call a border *short* if its length is $< \frac{|w|}{2}$. The notation $Bord(w)$ and $ShortBord(w)$ stand for the longest border of $w$ and the longest short border of $w$, respectively. Clearly, any of these borders can coincide with the empty word.

**Example 2.1.** For $w = aabaabcaab\,aabaabcaab\,aab$ we have:

$$Bord(w) = aabaabcaab\,aab, \qquad ShortBord(w) = aabaab.$$

Observe that in this particular example $ShortBord(w)$ is the second longest border of $w$, but for any $k \geq 1$ there are examples where $ShortBord(w)$ is the $k$th longest border of $w$ (the word $a^{2k-1}$ is such an example).

For a given word $x$ denote by $\mathsf{Ztype}[i]$ the Zimin type of $x[1..i]$.

**Lemma 2.2.** *Zimin type of a non-empty word can be computed iteratively through the equation*

$$\mathsf{Ztype}[i] = 1 + \mathsf{Ztype}[j], \quad \textit{where } j = \big|ShortBord\big(x[1..i]\big)\big| \tag{2.1}$$

**Proof.** Since $u = ShortBord(w)$ implies $w = uvu$ for some non-empty word $v$, the left-hand part of (2.1) majorizes the right-hand part. At the same time, $\mathsf{Ztype}[i] = \mathsf{Ztype}[j] + 1$, where $j$ is the length of *some* short border of $x[1..i]$. Hence, it suffices to show that increasing the length of the border within the interval $(0; i/2)$ cannot decrease its Zimin type.