WORD COMPLEXITY AND REPETITIONS IN WORDS[1]

LUCIAN ILIE[2], SHENG YU[3], and KAIZHONG ZHANG[4]
Department of Computer Science, University of Western Ontario
N6A 5B7, London, Ontario, CANADA
e-mails: `ilie|syu|kzhang@csd.uwo.ca`

ABSTRACT

With ideas from data compression and combinatorics on words, we introduce a complexity measure for words, called *repetition complexity*, which quantifies the amount of repetition in a word. The repetition complexity of w, $\textsc{r}(w)$, is defined as the smallest amount of space needed to store w when reduced by repeatedly applying the following procedure: n consecutive occurrences $uu \ldots u$ of the same subword u of w are stored as (u, n). The repetition complexity has interesting relations with well-known complexity measures, such as *subword complexity*, $\textsc{sub}$, and *Lempel-Ziv complexity*, $\textsc{lz}$. We have always $\textsc{r}(w) \geq \textsc{lz}(w)$ and could even be that the former is linear while the latter is only logarithmic; e.g., this happens for prefixes of certain infinite words obtained by iterated morphisms. An infinite word α being ultimately periodic is equivalent to: (i) $\textsc{sub}(\mathrm{pref}_n(\alpha)) = \mathcal{O}(n)$, (ii) $\textsc{lz}(\mathrm{pref}_n(\alpha)) = \mathcal{O}(1)$, and (iii) $\textsc{r}(\mathrm{pref}_n(\alpha)) = \lg n + \mathcal{O}(1)$. De Bruijn words, well known for their high subword complexity, are shown to have almost highest repetition complexity; the precise complexity remains open. $\textsc{r}(w)$ can be computed in time $\mathcal{O}(n^3(\log n)^2)$ and it is open, and probably very difficult, to find fast algorithms.

Keywords: repetitions in words, Lempel-Ziv complexity, subword complexity, iterated morphisms, de Bruijn words

1. Introduction

The repetitions in words is one of the properties of words studied longest. The study of repetitions dates back to the pioneering work of Thue [22, 23] at the beginning of the last century. He was concerned with infinite repetition-free words. Ever since, various aspects of repetitions in words were quite extensively investigated, see, e.g., [15], [4], [16] and the references therein.

In the present paper we investigate repetition in words from a new perspective: word complexity. Several measures of the complexity of words were proposed in the literature. The complexity of a word can be considered from different points of view: the shortest program to generate it (Kolmogorov [12], see also [19, 3]), the shortest compressed form (Lempel and Ziv [14]), the number of subwords ([15, 2]), the number of maximal repetitions ([13]), the highest order of repetitions ([9]). The new complexity measure we introduce concerns the amount of repetition in a word; we call it *repetition complexity*. The basic idea is that the more the repetitions, the less the complexity. However, measures related with classical properties of words, such as the number of repetitions or highest order of repetitions, seem to be less appropriate.

Our complexity measure takes ideas from both data compression theory and combinatorics on words. Essentially, from a repetition of a word we remember only

[1] An extended abstract of this paper has been presented at *The 8th Annual International Computing and Combinatorics Conference (COCOON'02)*, Singapore, 2002.

[2] cooresponding author; Research partially supported by NSERC grant R3143A01.

[3] Research partially supported by NSERC grant OGP0041630.

[4] Research partially supported by NSERC grant OGP0046373.

the base and the exponent; that is, we replace n consecutive occurrences of the same word $uu \cdots u$ by (u, n). The complexity is the minimum size to which a word is reduced by iteratively applying this procedure. As we shall see, the problem of doing optimally such reductions can be very intricate.

We investigate the repetition complexity from several points of view. It turns out that, aside from introducing challenging combinatorial problems, it is closely connected with well-known complexity measures, such as Lempel-Ziv complexity and subword complexity.

Due to the optimal compression it produces, [24, 11], the Lempel-Ziv complexity turns out to be always lower than the repetition complexity. Moreover, there are arbitrarily long words for which Lempel-Ziv complexity is much smaller. To prove this, we use prefixes of infinite words obtained by iterating exponential prolongable morphisms. The general result we use here, interesting in its own, says that prefixes of such infinite words have logarithmic Lempel-Ziv complexity.

Next, we give a result which relates all three complexities: subword, Lempel-Ziv, and repetition. Certain orders of these complexities for prefixes of infinite words turn out to be equivalent with ultimate periodicity. Precisely, an infinite word being ultimately periodic is equivalent to any of the following three properties, where n stands for the length of the prefixes:

(i) the subword complexity of its prefixes is linear in n,

(ii) the Lempel-Ziv complexity of prefixes is constant, and

(iii) the repetition complexity of prefixes is $\lg n$ plus a constant.

In particular, these provide new characterizations of ultimate periodicity of infinite words.

Another connection with the subword complexity is done via de Bruijn words. These are well known as having very high repetition complexity. We give a lower bound on their repetition complexity which is close to linear, that is, highest. We believe they have actually linear repetition complexity. However, this remains open.

Finally, we present an algorithm for computing the repetition complexity in time $\mathcal{O}(n^3(\log n)^2)$. Although this might seem slow, we give clear insight on why it seems a very difficult problem to find fast algorithms to compute the repetition complexity. Other open questions are proposed at the end.

2. Repetition complexity

We give here the basic definitions and notations we need in the paper. For basic results in combinatorics on words we refer to [15, 4, 16].

Let A be a finite alphabet and A^* the free monoid generated by A with the identity ε; $A^+ = A^* - \{\varepsilon\}$. For $u, v \in A^*$, we say that v is a *subword* of u if $u = u'vu''$, for some $u', u'' \in A^*$; v is a *prefix* (resp., *suffix*) of u if $u' = \varepsilon$ (resp., $u'' = \varepsilon$). The prefix relation is denoted $\leq_{\text{pref}}$ and the prefix of length n of w is denoted $\text{pref}_n(w)$.

For any word $w \in A^*$, the *length* of w is denoted by $|w|$. If $w = a_1 a_2 \cdots a_{|w|}$, where $a_i \in A$, then any integer $i, 1 \leq i \leq |w|$, is called a *position* in w; for $1 \leq i \leq |w|, i \leq j \leq |w|$, $w[i, j]$ denotes the subword $a_i a_{i+1} \cdots a_j$ of w; it has length $j - i + 1$. For $n \geq 0$, the nth power of w, denoted w^n, is defined inductively by $w^0 = \varepsilon$, $w^n = w^{n-1}w$. w is *primitive* if there is no $n \geq 2$ such that $w = u^n$, for some word u. The *primitive root* of w, denoted $\rho(w)$, is the unique primitive word

u such that $w \in u^+$. The *order* of w is $\mathrm{ord}(w) = \frac{|w|}{|\rho(w)|}$; we have $w = \rho(w)^{\mathrm{ord}(w)}$. A *period* of w is p such that $a_i = a_{i+p}$, for any $1 \leq i \leq |w| - p$.

A *repetition* in w is a subword of w of the form u^n for some nonempty word u and integer $n \geq 2$; n is the *order* and $|u|$ is the *period* of the repetition. For technical reasons, we formally define a repetition in w as a triple of positive integers (i, p, e) such that the word $(w[i, i + p - 1])^e$ is a subword of w starting at position i; that is, we have at position i a repetition of order e and period p.

We use repetitions to reduce the representation of a word, that is, we iteratively replace n consecutive occurrences of the same word $uu \cdots u$ by u and n. While the former takes $n|u|$ units of space to represent, we assume that the latter needs only $|u| + \lceil \lg(n + 1) \rceil$ (where lg stands for base 10 logarithm). We shall assume decimal representation for exponents but the results hold essentially unchanged for any base greater than or equal to two. Notice that, if n is in decimal then u^n is shorter than or equal to n consecutive u's, for any word u, as soon as $n \geq 2$; this helps avoiding special irrelevant cases in our reasoning.

We shall call this procedure a *reduction*; u^n is a reduced form of $uu \cdots u$. A word w can be iteratively reduced using the above procedure for repetitions inside w. However, some repetitions cannot be reduced simultaneously (because of overlapping), while further reductions can be applied inside already reduced repetitions. We formally define the repetition complexity below.

Let $D = \{\mathsf{0}, \mathsf{1}, \ldots, \mathsf{9}\}$ be the set of decimal digits, $D \cap A = \emptyset$, and let $\langle, \rangle, \hat{}$ be three new letters; put $T = A \cup D \cup \{\langle, \rangle, \hat{}\}$. For a positive integer n, $\mathrm{dec}(n) \in D^*$ is the decimal representation of n. (For a word w, $|\mathrm{dec}(|w|)| = \lceil \lg(|w| + 1) \rceil$ is the length of the decimal representation of the length of w.) Define the binary relation $\Rightarrow \subseteq T^* \times T^*$

$$u \Rightarrow v \quad \text{iff} \quad \begin{array}{l} u = u_1 x^n u_2, v = u_1 \langle x \rangle \hat{} \langle \mathrm{dec}(n) \rangle u_2, \\ \text{for some } u_1, u_2 \in T^*, x \in A^+, n \geq 2. \end{array}$$

Let $\Rightarrow^*$ be the reflexive and transitive closure of $\Rightarrow$; if $u \Rightarrow^* v$, then v is a reduced form of u. Define also a morphism $h : T^* \to (A \cup D)^*$ which simply erases all letters from $\{\langle, \rangle, \hat{}\}$ and keeps those in $A \cup D$ unchanged. The *repetition complexity* of a word $w \in A^*$, denoted $\mathrm{R}(w)$, is formally defined as

$$\mathrm{R}(w) = \min_{w \Rightarrow^* u} |h(u)|.$$

Such an u with $\mathrm{R}(w) = |h(u)|$ is called a *shortest reduced form* of w and $w \Rightarrow^* u$ is an *optimal reduction* of w.

We notice that our reduction relation $\Rightarrow$ is not confluent. For instance, if $w = \mathsf{ababcbc}$, then we have two reductions which cannot be continued any further: $w \Rightarrow \langle \mathsf{ab} \rangle \hat{} \langle 2 \rangle \mathsf{cbc}$ and $w \Rightarrow \mathsf{aba} \langle \mathsf{bc} \rangle \hat{} \langle 2 \rangle$. Actually, both are optimal reductions.

Example 1 Consider the word $w = \mathsf{ababaababababbbabb}$. Several possible reductions for w are shown below (the first is optimal and so $\mathrm{R}(w) = 10$):

$$\begin{array}{l} w \Rightarrow \langle \mathsf{ababa} \rangle \hat{} \langle 2 \rangle \mathsf{bbbabb} \Rightarrow \langle \mathsf{ababa} \rangle \hat{} \langle 2 \rangle \langle \mathsf{b} \rangle \hat{} \langle 3 \rangle \mathsf{abb} \Rightarrow \langle \langle \mathsf{ab} \rangle \hat{} \langle 2 \rangle \mathsf{a} \rangle \hat{} \langle 2 \rangle \langle \mathsf{b} \rangle \hat{} \langle 3 \rangle \mathsf{abb} \\ w \Rightarrow^* \langle \mathsf{ab} \rangle \hat{} \langle 2 \rangle \mathsf{aaba} \langle \mathsf{babb} \rangle \hat{} \langle 2 \rangle \\ w \Rightarrow^* \mathsf{a} \langle \mathsf{ba} \rangle \hat{} \langle 2 \rangle \langle \mathsf{ab} \rangle \hat{} \langle 2 \rangle \mathsf{a} \langle \mathsf{b} \rangle \hat{} \langle 3 \rangle \mathsf{a} \langle \mathsf{b} \rangle \hat{} \langle 2 \rangle \end{array}$$

The next lemma gives the bounds for the R-complexity.

Lemma 2 *For any $w \in A^*$ with $|w| \geq 2$, $1 + |\mathrm{dec}(|w|)| \leq \mathrm{R}(w) \leq |w|$.*

Proof. By induction on $|w|$ using the definition of $\mathrm{R}(w)$. The upper bound is clearly optimal. The lower bound can be shown to be optimal. Also, the only words which reach it are of the forms: a^n, ab^{10^n-1}, and $(\mathsf{ab})^k$ with $\lceil \lg(2k+1) \rceil = |\mathrm{dec}(k)| + 1$. □

The next result concerns words with highest repetition complexity. Using results from combinatorics of words we show that there are many of such words.

Theorem 3 *The number of words over three (or more) letters of maximum repetition complexity is exponential in terms of the length.*

Proof. For any w, $\mathrm{R}(w) = |w|$ if and only if w is square-free or has only squares of a single letter. We have from [4] that the number $SF_n(3)$ of square-free words of length n over three letters verifies the inequalities $6 \cdot 1.032^n \leq SF_n(3) \leq 6 \cdot 1.38^n$. The claim follows. □

A property expected from a complexity measure is subadditivity. The complexity we introduced has it.

Lemma 4 *For any u, v, $\mathrm{R}(uv) \leq \mathrm{R}(u) + \mathrm{R}(v)$.*

3. The definition of repetition complexity

We discuss here our choice of defining the repetition complexity. Another choice could have been the highest order of a repetition. This is a local property which does not necessarily affect the whole word. If the highest order is very low, then it becomes more relevant. For instance, if it is less than 2, then we obtain our highest complexity, but for higher values we can have totally different words. For instance, the word $((\cdots(\mathsf{a}_1^2\mathsf{a}_2)^2\cdots\mathsf{a}_n)^2$ has highest order of repetition 2 but it clearly contains a lot more repetition than a prefix of a 2^--free word (see [4]). The difference with respect to the highest order of repetition is smallest but the repetition complexities are quite different: logarithmic for the former and linear for the latter.

The number of repetitions is another candidate. A good example here is the Fibonacci word defined by $\mathbf{f} = \lim_{n\to\infty} f^n(\mathsf{a})$, where $f(\mathsf{a}) = \mathsf{ab}$, $f(\mathsf{b}) = \mathsf{a}$. By [6], any prefix of length n of $\mathbf{f}$ has $\Theta(n \lg n)$ maximal repetitions (i.e., repetitions which cannot be extended). A much less complex word, a^n has only one maximal repetition. For further results concerning the number of repetitions in words, see [13] and the references therein.

Of course, the repetition in a^n is very long. We should therefore take into account both the number of repetitions and their lengths. But our complexity does it. Moreover, it takes implicitly into account overlappings among the repetitions by the fact that overlapping repetitions cannot be reduced simultaneously.

Finally, one could argue that the exponents should be counted as one unit of space each, as in a RAM model. But then, an infinite word like $aaa\ldots$ would have all prefixes reduced to size two which is unreasonable.

4. Subword and Lempel-Ziv complexities

We recall two basic complexity measures of words to which we compare the repetition complexity. These are the subword complexity and Lempel-Ziv complexity.

For a word w, the *subwords complexity* of w is the number of subwords of w, denoted by $\mathrm{SUB}(w)$. The next lemma gives the optimal range for the subword complexity.

Lemma 5 *For any w, we have $|w|+1 \leq \text{SUB}(w) \leq 1+\frac{1}{2}(|w|^2+|w|)$.*

Proof. The lower bound is attainable for powers of a single letter and the upper bound for words in which no letter occurs twice. □

Essential for us in the above lemma is the fact that, on fixed alphabets, $\text{SUB}(w)$ is at least linear and at most quadratic; e.g., for $w = \mathsf{a}^n\mathsf{b}^n$, $\text{SUB}(w) = \Theta(|w|^2)$.

One of the most famous complexity measures is the one introduced by Lempel and Ziv [14] in connection with their algorithm for data compression, see [24, 25, 7].

For a word w, we define the *e-decomposition*[a] of w as the (unique) decomposition $w = w_1w_2\cdots w_k$ such that, for any $1 \leq i \leq k$ (with the possible exception of $i = k$), w_i is the shortest prefix of $w_iw_{i+1}\cdots w_k$ which does not occur before in w; that is, w_i does not occur as a subword of $\pi(w_1w_2\cdots w_i)$, where the application π removes the last letter of its argument.

The complexity measure introduced by Lempel and Ziv represents the *number of subwords in the e-decomposition* of w; we denote it by $\text{LZ}(w)$.

Example 6 Consider the word $w = \mathsf{aababbabbabb}$. The e-decomposition of w is $w = \mathsf{a.ab.abb.abbabb}$, where the subwords are marked by dots. Therefore, $\text{LZ}(w) = 4$.

We notice that the e-decomposition can be defined in the same way for infinite words; at each step we take the longest prefix of the remaining infinite suffix which does not appear before; this prefix may be the remaining suffix of the infinite word, in which case it is the last element of the decomposition. This definition will be used in Lemma 12.

The next lemma is a weak form of a result of [14] which states the bounds for the LZ-complexity (for fixed alphabet).

Lemma 7 $\text{LZ}(w) = \mathcal{O}\Big(\dfrac{|w|}{\lg|w|}\Big)$.

5. Relation with Lempel-Ziv complexity

We shall compare in this section our complexity with the Lempel-Ziv complexity. We start by investigating closer the R-complexity. As defined above, $\text{R}(w)$ is the size of $h(v)$ for an optimal reduction $u \Rightarrow^* v$. At each step in this reduction, we use a repetition (i,p,e) in w to decrease the size; denote the space saved by reducing w according to this repetition by $\text{red}(w,i,p,e) = (e-1)p - |\text{dec}(e)|$. When w is understood, we write simply $\text{red}(i,p,e)$. Of course, the saving in space does not depend on the position of the repetition in w, but we still keep i as an argument in order to be able to identify precisely what repetition we are talking about.

The next lemma shows how an optimal reduction is obtained.

Lemma 8 *For any word w, there is an ordered sequence of $m \geq 0$ repetitions in w*

$$(i_1,p_1,e_1),(i_2,p_2,e_2),\ldots,(i_m,p_m,e_m) \tag{1}$$

such that

$$\text{R}(w) = |w| - \sum_{k=1}^{m}\text{red}(i_k,p_k,e_k), \tag{2}$$

[a] 'e' comes from 'exhaustive'; Lempel and Ziv [14] called this decomposition the *exhaustive production history* of w; it is called *f-factorization* by [7] and *s-factorization* by [18].

and any two repetitions (i_k, p_k, e_k) *and* (i_l, p_l, e_l), $1 \le k < l \le m$, *are*

(i) either disjoint, i.e., $[i_k, i_k + p_k e_k - 1] \cap [i_l, i_l + p_l e_l - 1] = \emptyset$,

(ii) or the one appearing later in (1) is contained in the first period of the other, i.e., $i_k \le i_l$ *and* $i_l + p_l e_l \le i_k + p_k$.

Proof. According to the definition, there exists a word $u \in T^*$ such that $w \Rightarrow^* u$ and $\mathrm{R}(w) = |h(u)|$. The proof is by induction on the number of steps n of the reduction $w \overset{n}{\Rightarrow} u$. If $n = 0$, then $u = w$ and $\mathrm{R}(w) = |w|$. We may then choose the empty sequence for (1). Notice that (2) and (i), (ii) are fulfilled.

We assume the statement true for any reduction in $n - 1$ steps or less and prove it for reductions in $n \ge 1$ steps. Emphasising the first step, the reduction is

$$w = w_1 w_2^e w_3 \Rightarrow w_1 \langle w_2 \rangle^\wedge \langle \mathrm{dec}(e) \rangle w_3 \Rightarrow^* u = u_1 \langle u_2 \rangle^\wedge \langle \mathrm{dec}(e) \rangle u_3.$$

and we have also the following three reductions, each having at most $n - 1$ steps:

$$w_j \overset{\le n}{\Rightarrow} u_j, \text{ for any } 1 \le j \le 3.$$

Using the inductive hypothesis, we have the statement true for any of $w_j, 1 \le j \le 3$. If the ordered sequence (1) corresponding to w_j is $(i_l^{(j)}, p_l^{(j)}, e_l^{(j)})_{1 \le l \le m_j}$, then the ordered sequence (1) corresponding to w will be

$$\begin{array}{l}(i_1^{(1)}, p_1^{(1)}, e_1^{(1)}), \ldots, (i_{m_1}^{(1)}, p_{m_1}^{(1)}, e_{m_1}^{(1)}), (|w_1| + 1, |w_2|, e), \\ (i_1^{(2)}, p_1^{(2)}, e_1^{(2)}), \ldots, (i_{m_2}^{(2)}, p_{m_2}^{(2)}, e_{m_2}^{(2)}), (i_1^{(3)}, p_1^{(3)}, e_1^{(3)}), \ldots, (i_{m_3}^{(3)}, p_{m_3}^{(3)}, e_{m_3}^{(3)}).\end{array} \tag{3}$$

Since $\mathrm{R}(w) = \sum_{j=1}^{3} \mathrm{R}(w_j) + |\mathrm{dec}(e)|$, it follows from the inductive hypothesis that the relation (2) is satisfied for w and (3). Also from the inductive hypothesis, we get that (3) satisfies (i) and (ii). □

We give next an example of an application of Lemma 8.

Example 9 For the word

$$w = \mathsf{aabbbabbbbaaababbbabbbbaaabbababab},$$

an ordered sequence (1) can be $(2, 13, 2), (2, 4, 2), (3, 1, 3), (11, 1, 3), (29, 2, 3)$. The space saved by each of them is, in order, 12, 3, 1, 1, 3. Finally, $\mathrm{R}(w) = |w| - (12 + 3 + 1 + 1 + 3) = 34 - 20 = 14$; w can be written as $w = \mathsf{a}((\mathsf{ab}^3)^2\mathsf{ba}^3\mathsf{b})^2\mathsf{b}(\mathsf{ab})^3$.

We establish next the connection in one direction with the Lempel-Ziv complexity and give some non-trivial examples showing the optimality of the result.

Theorem 10 *For any word* w, $\mathrm{R}(w) \ge \mathrm{LZ}(w)$.

Proof. By Lemma 8, we may assume a sequence (1) of repetitions in w such that $\mathrm{R}(w)$ is given by (2) and (1) fulfills (i) and (ii) in the lemma. Starting from (1), we construct the following decomposition of w:

(a) - for any repetition (i_j, p_j, e_j) in (1), consider $w[i_j + p_j, i_j + p_j e_j - 1]$ as subword in the decomposition;

(b) - for any position l in w which is contained in no interval $[i_j + p_j, i_j + p_j e_j - 1], 1 \le j \le m$, consider the letter at position l, $w[l, l]$, as a subword in the decomposition.

It is worth noticing that the conditions (i) and (ii) in Lemma 8 show that there is no contradictory decomposition at (a); this follows because for any $1 \le j \ne l \le m$, the two intervals $[i_j + p_j, i_j + p_j e_j - 1]$ and $[i_l + p_l, i_l + p_l e_l - 1]$ are disjoint.

Denoting the number of subwords in this decomposition (let us call it, for this proof only, d-decomposition) by d, we have $\mathrm{R}(w) \geq d$. Indeed, the set of subwords in the d-decomposition can be partitioned into two subsets, according to the step at which they were obtained. Now, those obtained at (b) are in one-to-one correspondence with the letters which actually occur in the reduced form of w of size $\mathrm{R}(w)$. Those obtained at (a) are in one-to-one correspondence with the exponents $e_1, e_2, \ldots, e_m$. Since any exponent counts at least one for $\mathrm{R}(w)$, the claim follows.

We show next that $d \geq \mathrm{LZ}(w)$. Assume $d < \mathrm{LZ}(w)$. Then, there will be one subword in the d-decomposition, $u = w[p,q]$, and another in the e-decomposition, $v = w[l,r]$, such that $p \leq l < r < q$. So $|u| \geq 2$ and, by our construction of the d-decomposition, u occurs before in w, i.e., starting at a position before $p-1$. Therefore, $\pi(v)$, which is a subword of u, occurs also before $l-1$, contradicting the definition of the e-decomposition. Thus, $d \geq \mathrm{LZ}(w)$ and the theorem is proved. □

Example 11 Consider $n \geq 1$ and n different letters $\mathsf{a}_i, 1 \leq i \leq n$, and construct the word

$$w_n = ((\ldots(\mathsf{a}_1^9\mathsf{a}_2)^9\mathsf{a}_3)^9\cdots\mathsf{a}_{n-1})^9\mathsf{a}_n.$$

We have $|w_n| = \frac{9^n-1}{8}$ and $\mathrm{R}(w_n) = 2n-1 = \Theta(\lg|w_n|)$. Denoting $x_n = w_{n-1}^8\mathsf{a}_n$, we have $w_n = w_{n-1}x_n$. The e-decomposition of w_n is $w_n = \mathsf{a}_1.x_2.x_3.x_4.\cdots.x_n$ and so $\mathrm{LZ}(w_n) = n = \Theta(\lg|w_n|)$. Therefore the result in Theorem 10 cannot be improved by more than a constant.

We now consider the relation in the opposite direction. We show that we have the opposite case: there are words of high R-complexity but low LZ-complexity. We shall need a result about infinite words which is interesting in itself. We denote by A^ω the set of infinite words over A. For $w \in A^*$, we denote $w^\omega = www\cdots$.

A morphism $\varphi : A^* \to A^*$ is called *prolongable* on $a \in A$ if $\varphi(a) \in aA^+$. If φ is prolongable on a, then $\lim_{n\to\infty}\varphi^n(a) \in A^\omega$ exists and we denote it by $\varphi^\infty(a)$. φ is called *exponential* if there are an integer $n_0 \geq 1$ and a real $c > 1$ such that, for any $w \in A^*$, $|\varphi^{n_0}(w)| \geq c|w|$.

Lemma 12 *Let $\varphi : A^* \to A^*$ be an exponential morphism prolongable on $a \in A$ and denote $\alpha = \varphi^\infty(a)$. Then,* $\mathrm{LZ}(\mathrm{pref}_n(\alpha)) = \mathcal{O}(\lg n)$.

Proof. If the e-decomposition of α has finitely many elements, then we have $\mathrm{LZ}(\mathrm{pref}_n(\alpha)) = \mathcal{O}(1)$ and we are done. Assume this is not the case. Since φ is exponential, there is a subword w in the e-decomposition such that $w = w''w'a, a \in A, w' \neq \varepsilon$, and the image of w' by φ occurs in α after w, that is, $\alpha = u.w''w'a.x\varphi(w')\varphi(a)\alpha'$, $x \in A^*, \alpha' \in A^\omega$. (We have marked by dots only the subword w in the e-decomposition of α.) We may also assume that any letter occurring in $ax\varphi(w')\varphi(a)\alpha'$ occurs also in uw'.

Consider the decomposition of $x\varphi(w')$ induced by the e-decomposition of α, say $x\varphi(w') = w_1.w_2.\cdots.w_k$, $k \geq 1$; in case there is no decomposition position inside $x\varphi(w')$, then $k = 1$. Put $w_i = w_i'a_i, a_i \in A$. Consider then the following decomposition of α, which we call (for this proof only) d-decomposition:

$$\alpha = u.w.w_1'a_1.w_2'a_2.\cdots.w_k'a_k.\varphi(a).\varphi(w_1').\varphi(a_1).\cdots.\varphi(w_k').\varphi(a_k).$$
$$\varphi^2(a).\varphi^2(w_1').\varphi^2(a_1).\cdots.\varphi^2(w_k').\varphi^2(a_k).\varphi^3(a).\varphi^3(w_1').\varphi^3(a_1).\cdots$$

The decomposition of u is the one induced by the e-decomposition of α and not shown. We should say also that we consider only the nonempty words among the subwords in the above d-decomposition. In fact, $w_i \neq \varepsilon$, for all $1 \leq i \leq k-1$.

The d- and e-decompositions for the prefix $uww_1w_2\cdots w_{k-1}$ of α coincide, while w_k might be a subword or only a prefix of a subword in the e-decomposition. For the remaining part $\varphi(a)\alpha'$, we claim that any subword in the d-decomposition occurs also before in α. This is true for any $\varphi(a), \varphi(a_i), 1 \le i \le k$, since a and a_i occur also in u. Therefore, it is also true for any $\varphi^j(a), \varphi^j(a_i)$. Then, for any $1 \le i \le k$, w_i' is a subword of $uw_1w_2\cdots w_{i-1}\pi(w_i')$, by the definition of the e-decomposition. It follows that any $\varphi^j(w_i')$ occurs before as well.

Now, consider any $n \ge 0$. We claim that the number of subwords in the d-decomposition of $\mathrm{pref}_n(\alpha)$ is larger than or equal to the number of subwords in the e-decomposition of $\mathrm{pref}_n(\alpha)$. (Clearly, we talk about the induced decompositions.) This is obvious for $n \le |uw|$. For larger n, we use an argument similar to the one in the proof of Theorem 10. But, the number of subwords in the d-decomposition is logarithmic in n because φ is exponential. Therefore, so is $\textsc{lz}(\mathrm{pref}_n(\alpha))$. □

Remark 13 We notice that the condition on φ being exponential in Lemma 12 is essential as shown by the following example. Take $\varphi : \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^* \to \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^*$, given by $\varphi(\mathsf{a}) = \mathsf{a}$, $\varphi(\mathsf{b}) = \mathsf{ba}$, $\varphi(\mathsf{c}) = \mathsf{cba}$. φ is prolongable on c and we have the e-decomposition of $\varphi^\infty(\mathsf{c})$

$$\varphi^\infty(\mathsf{c}) = \mathsf{c.b.a.baa.baaa.baaaa.baaaaa.baaaaaa}\cdots$$

Hence $\textsc{lz}(\mathrm{pref}_n(\varphi^\infty(\mathsf{c}))) = \Theta(\sqrt{n})$.

Lemma 12 gives a relation in the other direction between $\textsc{r}$ and $\textsc{lz}$.

Theorem 14 *There are arbitrarily long words w for which $\textsc{r}(w) = |w|$ and $\textsc{lz}(w) = \mathcal{O}(\lg |w|)$.*

Proof. Consider the morphism $m : \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^* \to \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^*$ given by $m(\mathsf{a}) = \mathsf{abc}$, $m(\mathsf{b}) = \mathsf{ac}$, $m(\mathsf{c}) = \mathsf{b}$. m is prolongable on a and denote $\mathbf{m} = m^\infty(\mathsf{a})$. It is a well known fact that $\mathbf{m}$ is square-free; see, e.g., [20]. Because $\mathbf{m}$ is square-free, we have $\textsc{r}(\mathrm{pref}_n(\mathbf{m})) = n$. Applying Lemma 12 to the word $\mathbf{m}$ gives that $\textsc{lz}(\mathrm{pref}_n(\mathbf{m})) = \mathcal{O}(\lg n)$. □

It is also worth noticing here that our repetition complexity is the same for a word and its reversed version, while the same is not necessarily true for Lempel-Ziv complexity. It is probably interesting to investigate the difference between the Lempel-Ziv complexity of a word and that of its reverse. This is, however, outside the scope of this paper.

6. Periodic infinite words and complexity of prefixes

We show in this section a strong connection between all low $\textsc{r}$-, $\textsc{lz}$-, and $\textsc{sub}$-complexities for prefixes of infinite words. This gives us also new characterizations of ultimately periodic infinite words. (The characterization (ii) resembles the famous one by Coven and Hedlund [5].)

Theorem 15 *For any infinite word α, the following assertions are equivalent:*

(i) α is ultimately periodic,

(ii) $\textsc{sub}(\mathrm{pref}_n(\alpha)) = \mathcal{O}(n)$,

(iii) $\textsc{lz}(\mathrm{pref}_n(\alpha)) = \mathcal{O}(1)$,

(iv) $\textsc{r}(\mathrm{pref}_n(\alpha)) = \lg n + \mathcal{O}(1)$.

Proof. $(i) \Leftrightarrow (ii)$. If $\alpha = uv^\omega$ is ultimately periodic, then, for any $n \ge 1$, we have that $\textsc{sub}(\mathrm{pref}_n(\alpha)) \le n|uv|$.

Conversely, assume $\textsc{sub}(\mathrm{pref}_n(\alpha)) \leq cn$, for any $n \geq 1$. Consider some $n \geq 2$ and let v be the shortest suffix of $\mathrm{pref}_n(\alpha)$ which is not a subword of $\mathrm{pref}_{n-1}(\alpha)$; denote $\alpha = uva\alpha'$, $u, v \in A^*$, $a \in A$. Any proper suffix of v appears as a subword in $\mathrm{pref}_{n-1}(\alpha)$ while any suffix of $\mathrm{pref}_n(\alpha)$ which has v as suffix does not. Thus, denoting $\mathrm{diff}_n(\alpha) = \textsc{sub}(\mathrm{pref}_n(\alpha)) - \textsc{sub}(\mathrm{pref}_{n-1}(\alpha))$, we have $\mathrm{diff}_n(\alpha) = |u| + 1$.

Also, va is not a subword of $\mathrm{pref}_n(\alpha)$. Thus, $\mathrm{diff}_{n+1}(\alpha) \geq |u| + 1$ and hence the sequence $(\mathrm{diff}_n(\alpha))_{n\geq 2}$ is increasing. We claim that $\mathrm{diff}_n(\alpha) \leq c$, for all $n \geq 1$. Indeed, if this is not true for some n, then $\textsc{sub}(\mathrm{pref}_{n(c+1)+1}(\alpha)) \geq (n(c+1)+1)c$, a contradiction. Therefore, there is $c' \leq c$, such that $\mathrm{diff}_n(\alpha) = c'$, for any n. Now, any long enough prefix of α will have a period shorter than c' which implies that α is ultimately periodic.

(i) ⇔ *(iii)*. (iii) is equivalent with the fact that the e-decomposition of the whole α has finitely many elements, the last of which must be an infinite word. This is equivalent with α being ultimately periodic.

(i) ⇔ *(iv)*. If $\alpha = uv^\omega$ is ultimately periodic, then, for any $n \geq 1$,

$$\textsc{r}(\mathrm{pref}_n(\alpha)) \leq \lg n + \textsc{r}(u) + \textsc{r}(v) + \max_{v' \leq_{\mathrm{pref}} v} \textsc{r}(v').$$

Consider an arbitrary long enough word w such that $\textsc{r}(w) \leq \lg(|w|) + c$ and those repetitions in the sequence (1) corresponding to w which are not included in any other repetitions from (1). We may assume, according to (i) and (ii) in Lemma 8, that these repetitions appear at the beginning of the sequence; assume they are the first m_0. Assume also the number of letters which do not belong to any of the repetitions in (1) for w is o. Then, using the relation (2), Lemma 2, and the fact that

$$\sum_{j=1}^{m_0} p_j e_j = |w| - o,$$

we have

$$\begin{aligned}
\textsc{r}(w) &= o + \sum_{j=1}^{m_0} \big(\textsc{r}(w[i_j, i_j + p_j - 1]) + \lceil \lg(e_j + 1) \rceil\big) \\
&\geq o + m_0 + \lg\big(\prod_{j=1}^{m_0} (p_j e_j)\big) \\
&= o + m_0 + \lg(|w| - o) + \lg\big(\prod_{j=1}^{m_0} (p_j e_j)\big) - \lg\big(\sum_{j=1}^{m_0} (p_j e_j)\big).
\end{aligned}$$

Now, $\textsc{r}(w) \leq \lg(|w|) + c$ implies that $m_0 \leq c$, $o \leq 2c$, and

$$\prod_{j=1}^{m_0} (p_j e_j) \leq 10^c \sum_{j=1}^{m_0} (p_j e_j).$$

Therefore, if $p_r e_r$ is largest among $p_j e_j$, $1 \leq j \leq m_0$, then all others $p_j e_j$ are bounded by $2 \cdot 10^c$.

Assume now $\textsc{r}(\mathrm{pref}_n(\alpha)) \leq \lg n + c$, for any $n \geq 1$. Applying the above to $w = \mathrm{pref}_n(\alpha)$, what we proved is that, in any long enough prefix of α, there is a repetition not included in others which may be arbitrarily long, while everything else is bounded by $c' = 2c + 2(c-1)10^c$; that is, we have $\mathrm{pref}_n(\alpha) = uw^e v$ with $|uv| \le c'$. If we take also $\mathrm{pref}_{n+c'+1}(\alpha) = xy^r z$, $|xz| \le c'$, then $w^e \ne y^r$. Put $w = (\rho(w))^f$, $y = (\rho(y))^s$. If one of ef and rs is at least 3 and n is large enough, then Fine and Wilf's lemma, see [15, 16], will imply that both w^e and y^r are powers of a word shorter than c'. On the other hand, for any large enough n, we can find such a situation because of the following result due to Chrochemore and Rytter [8]: if $u^2 <_{\mathrm{pref}} v^2 <_{\mathrm{pref}} w^2$ ($<_{\mathrm{pref}}$ is the proper prefix relation), u is primitive, and $v \notin u^*$, then $|u| + |v| \le |w|$. □

Remark 16 We notice that in Theorem 15 we have at (ii) and (iii) the order of the lower bound for the respective complexity from Lemmas 5 and 7 while at (iv) we have a stronger condition: the lower bound in Lemma 2 plus a constant instead of $\mathcal{O}(\lg n)$. In fact, $\mathcal{O}(\lg n)$ is not good as shown by the following example. Consider the word $w_k = ((\cdots(\mathsf{bab})^{10})\mathsf{ba}^2\mathsf{b})^{10^2}\mathsf{ba}^3\mathsf{b})^{10^3}\cdots\mathsf{ba}^k\mathsf{b})^{10^k}$. We have $w_k \le_{\mathrm{pref}} w_{k+1}$, so we can construct the infinite word $\mathbf{w} = \lim_{k\to\infty} w_k$. It can be shown that $\mathrm{R}(\mathrm{pref}_n(\mathbf{w})) = \mathcal{O}(\lg n)$. But $\mathbf{w}$ is not ultimately periodic. Therefore, we have from the R-complexity a slightly finer characterization of ultimately periodic words.

7. De Bruijn words and subword complexity

We next investigate the case of words with high subword complexity. We consider *de Bruijn* words $\mathbf{b}_k$ (see [2]) which have very high SUB-complexity. For $k \ge 1$, a de Bruijn word $\mathbf{b}_k \in A^*$ has the properties $|\mathbf{b}_k| = \mathrm{card}(A)^k + k - 1$ and $\mathrm{SUB}(\mathbf{b}_k) \cap A^k = A^k$; that is, $\mathbf{b}_k$ has as subwords all words of length k and any two subwords of length k of $\mathbf{b}_k$ starting from different positions are different. (There are many such words but our result holds for all of them.)

If $\mathrm{card}(A) = l$, then the number of all subwords of $\mathbf{b}_k$ is $\mathrm{SUB}(\mathbf{b}_k) = \frac{l^k-1}{l-1} + \frac{l^k(l^k+1)}{2}$. So, not only that $\mathrm{SUB}(\mathbf{b}_k)$ is of the order of maximal subword complexity in Lemma 5, but also the difference between the upper bound in Lemma 5 and $\mathrm{SUB}(\mathbf{b}_k)$ is of strictly lower order: $\mathcal{O}(|\mathbf{b}_k| \lg |\mathbf{b}_k|)$. We show that de Bruijn words have also high repetition complexity.

Theorem 17 $\mathrm{R}(\mathbf{b}_k) = \Omega\Big(\frac{|\mathbf{b}_k| \lg\lg |\mathbf{b}_k|}{\lg |\mathbf{b}_k|}\Big)$.

Proof. Since no two subwords of length k in $\mathbf{b}_k$ are the same, it follows that any repetition (i, p, e) in $\mathbf{b}_k$ verifies $|pe| < 2k$. Using Lemma 8 for $\mathbf{b}_k$ and the bounds in Lemma 2, we have, for some $1 \le m_0 \le m$,

$$\mathrm{R}(\mathbf{b}_k) \ge \sum_{i=1}^{m_0} (\lceil \lg(p_i e_i + 1)\rceil + 1) \ge \lg((2k)^{n/2k}) = \Omega\Big(\frac{n \lg\lg n}{\lg n}\Big).$$

□

8. Computing the repetition complexity

We show in this section that the repetition complexity can be computed in time $\mathcal{O}(n^3(\log n)^2)$. Due to the very intricate nature of repetitions, this problem is by no means trivial. (A good example of how complex the repetitions in a word can be are the Fibonacci words.) In fact, it can be seen as a restricted case of the optimal data compression which is NP-complete; see [21, 10].

We give next another example which, although simple from algorithmic point of view, shows that a word can have exponentially many optimal reductions; which again makes the problem hard. Consider the Morse-Hedlund infinite word $\mathbf{m}$, [20], defined as $\mathbf{m} = m^{\infty}(\mathsf{a})$, where $m : \{\mathsf{a},\mathsf{b},\mathsf{c}\}^* \to \{\mathsf{a},\mathsf{b},\mathsf{c}\}^*$, $m(\mathsf{a}) = \mathsf{abc}$, $m(\mathsf{b}) = \mathsf{ac}$, $m(\mathsf{c}) = \mathsf{b}$. For the morphism φ given by $\varphi(\mathsf{a}) = \mathsf{ababa}$, $\varphi(\mathsf{b}) = \mathsf{a'b'a'b'a'}$, $\varphi(\mathsf{c}) = \mathsf{a''b''a''b''a''}$, we have that $\varphi(\mathrm{pref}_n(\mathbf{m}))$ has length $5n$ and 2^n optimal reductions.

We present next our method to compute $\mathrm{R}(w)$. The following observation is the basis of our algorithm. For any non-empty word w, we have

$$\mathrm{R}(w) = \min\Big(\min_{w=uv}\big(\mathrm{R}(u)+\mathrm{R}(v)\big), \min_{k|\mathrm{ord}(w)}\Big(\mathrm{R}(\rho(w)^{\frac{\mathrm{ord}(w)}{k}}) + |\mathrm{dec}(k)|\Big)\Big). \qquad (4)$$

Based on (4), we use then dynamic programming to compute the repetition complexities of all subwords of w.

Theorem 18 *The repetition complexity of w, for $|w| = n$, can be computed in time $\mathcal{O}(n^3(\log n)^2)$.*

Proof. We prove first the equality (4). The complexity $\mathrm{R}(w)$ can be derived in two ways, depending on whether the first repetition (i_1, p_1, e_1) in a sequence (1) yielding an optimal reduction covers the whole w or not; that is, $w[i_1, i_1+p_1-1]^{e_1} = w$.

When this repetition does not cover the whole w, we have either $1 < i_1$ or $i_1 + p_1e_1 - 1 < |w|$. In the former case take $u = w[1, i_1 - 1]$, $v = w[i_1, |w|]$, in the latter $u = w[1, i_1+p_1e_1-1]$, $v = w[i_1+p_1e_1, |w|]$. In either case $\mathrm{R}(w) = \mathrm{R}(u)+\mathrm{R}(v)$.

When this repetition consists of the whole w, then e_1 must divide the order of w and $w[1, p_1] = \rho(w)^{\frac{\mathrm{ord}(w)}{e_1}}$. (Notice that $i_1 = 1$ and $\mathrm{ord}(w) \geq 2$.) This is a well-known fact in combinatorics on words; see [15, 16]. Briefly, the argument of the proof is that $w[1, p_1]^{e_1} = \rho(w)^{\mathrm{ord}(w)}$ for $e_1 \geq 2$, $\mathrm{ord}(w) \geq 2$, implies, by Fine and Wilf's lemma, that $w[1, p_1]$ and $\rho(w)$ are powers of the same word z. Since $\rho(w)$ is primitive, it cannot be a nontrivial power of a word and so $z = \rho(w)$.

The equality (4) is used in Algorithm 19 to compute $\mathrm{R}(w)$.

Algorithm 19 (computing the repetition complexity)

```
 Input: $w$ with $|w| = n$
 Output: $\mathrm{R}(w)$
1. compute $p(i,j) = |\rho(w[i,j])|$ and $\mathrm{ord}(i,j) = \mathrm{ord}(w[i,j])$, for all $1 \leq i \leq j \leq n$
2. compute (steps 3..12) $r(i,j) = \mathrm{R}(w[i,j])$, for all $1 \leq i \leq j \leq n$
3.   for $i$ from $n$ downto 1 do
4.       $r(i,i) = 1$; $r(i,i+1) = 1$
5.       for $j$ from $i+2$ to $n$ do
6.           $r(i,j) = j - i + 1$
7.           for $k$ from $i$ to $j-1$ do
8.               $r(i,j) = \min\big(r(i,j), r(i,k) + r(k+1,j)\big)$
9.           if $p(i,j) \neq j - i + 1$ then
10.              for $k$ from 1 to $\mathrm{ord}(i,j)$ do
11.                  if $(k \mid \mathrm{ord}(i,j))$ then
12.                      $r(i,j) = \min\big(r(i,j), r(i, i + k * p(i,j) - 1) + |\mathrm{dec}(\frac{\mathrm{ord}(i,j)}{k})|\big)$
13. output $\mathrm{R}(w) = r(1,n)$
```

Here is a description of the first step of the algorithm. It computes the shortest periods of all subwords $w[i, j]$, $1 \leq i \leq |w|$, $i \leq j \leq |w|$. For a fixed i, this can be done in time $\mathcal{O}(n - i)$ using a linear pattern matching algorithm. This is done as follows. A *border* of w is a subword of w which is both prefix and suffix of w. There is a one-to-one correspondence between borders and periods: p is a period of w iff w has a border of length $|w| - p$. By matching w shifted j positions against w itself, the length l of the longest common prefix (of w and $w[j+1, |w|]$) gives the longest border of $\mathrm{pref}_{j+l}(w)$, which, in turn, gives its shortest period. Therefore, we have the shortest periods of all prefixes of w in time $\mathcal{O}(|w|)$. Knowing the shortest period p of w, we can compute both the order and the length of the primitive root in time $\mathcal{O}(\log^2 |w|)$: if the shortest period divides the length of w, then the $\rho(w) = \mathrm{pref}_p(w)$, $\mathrm{ord}(w) = \frac{|w|}{p}$, otherwise, $\rho(w) = w$, $\mathrm{ord}(w) = 1$. For all $w[i, j]$, this can be done in time $\mathcal{O}(|w|^2 \log^2 |w|)$.

Then, using (4), we compute the repetition complexities for all subwords of w. The complexity of this step is seen in the Algorithm 19 to be $\mathcal{O}(n^3(\log n)^2)$. The algorithm is presented in pseudocode. Its correctness comes from (4). □

9. Conclusions and further research

We investigate the repetitions in words from the point of view of complexity of words. Our work is related to the study of repetitions in words in general, see, e.g., [13], but our goals are different. We want to measure the complexity of words using their repetitions. We introduce the notion of repetition complexity of a word and discuss its appropriateness by comparison with other potential candidates.

We give results which relate our complexity to well-known complexity measures like subword or Lempel-Ziv complexity. These turn out to give interesting results about infinite words. We mention here several problems which deserve further investigation.

The algorithm we gave for computing the repetition complexity is, of course, not very fast (compared to usual algorithms dealing with repetitions in words, e.g., [6, 1, 17, 13]), but it seems difficult to give very fast algorithms. Notice that we used dynamic programming and, based on this idea, we cannot find algorithms with sub-quadratic time. Completely new ideas and properties of words are needed for fast algorithms.

Another problem is that the algorithm is not of much use if we try to compute (or only approximate) the repetition complexity of some families of words, say all prefixes of the Fibonacci infinite word. Some different tools for lower bounds are needed.

We showed in Theorem 17 that de Bruijn words have high repetition complexity. We believe they have in fact linear complexity, that is, $\mathrm{R}(\mathbf{b}_k) = \Theta(|\mathbf{b}_k|)$.

A related complexity which we did not discuss here can be naturally defined using rational repetitions. For instance, consider the words $\mathsf{abcdabc}$ and $\mathsf{abcdefg}$. Both have R-complexity 7 as none contains any integer repetitions, although the former contains clearly more repetition than the latter. Using rational powers, we may write $\mathsf{abcdabc} = (\mathsf{abcd})^{7/4}$ which takes only 6 units of space.

Finally, a problem which we have not approached concerns the connection between our complexity and randomness. It should be investigated how random are the words with high repetition complexity, in particular the square-free words.

References

1. A. Apostolico and F. Preparata, "Optimal off-line detection of repetitions in a string," *Theoret. Comput. Sci.* **22** (1983) 297 – 315.
2. N.G. de Bruijn, "A combinatorial problem," *Proc. Kon. Ned. Akad. Wetensch.* **49** (1946) 758–764.
3. G.J. Chaitin, "Information-theoretic limitations of formal systems," *J. Assoc. Comput. Mach.* **21** (1974) 403 – 424.
4. C. Choffrut and J. Karhumäki, "Combinatorics of Words," in *Handbook of Formal Languages, Vol. I*, eds. G. Rozenberg and A. Salomaa (Springer-Verlag, Berlin, 1997) pp. 329 – 438.
5. E.M. Coven and G. Hedlund, "Sequences with minimal block growth," *Math. Sytems Theory* **7** (1973) 138 – 153.
6. M. Crochemore, "An optimal algorithm for computing the repetitions in a word," *Inform. Proc. Lett.* **12** (5) (1981) 244 – 250.
7. M. Crochemore and W. Rytter, *Text Algorithms* (Oxford Univ. Press, 1994).
8. M. Crochemore and W. Rytter, "Squares, cubes, and time-space efficient string matching," *Algorithmica* **13** (1995) 405 – 425.
9. F. Dejean, "Sur un théorème de Thue," *J. Combin. Theory, Ser. A* **13** (1972) 90–99.
10. M.R. Garey and D.S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-completeness* (W.H. Freeman and Co., San Francisco, 1979).
11. G. Hansel, D. Perrin, and I. Simon, "Compression and entropy," *Proc. of STACS'92*, Lecture Notes in Comput. Sci. 577, Springer-Verlag, 1992, 515 – 528.
12. A.N. Kolmogorov, "Three approaches to the quantitative definition of information," *Probl. Inform. Transmission* **1** (1965) 1 – 7.
13. R. Kolpakov and G. Kucherov, "Finding maximal repetitions in a word in linear time," *Proc. of FOCS'99*, 596 – 604.
14. A. Lempel and J. Ziv, "On the complexity of finite sequences," *IEEE Trans. Information Theory* **22**(1) (1976) 75–81.
15. M. Lothaire, *Combinatorics on Words* (Addison-Wesley, Reading, MA, 1983).
16. M. Lothaire, *Algebraic Combinatorics on Words* (Cambridge Univ. Press, 2002).
17. M. Main and R. Lorentz, "An $\mathcal{O}(n \lg n)$ algorithm for finding all repetitions in a string," *J. Algorithms* **5** (1984) 422 – 432.
18. M. Main, "Detecting leftmost maximal periodicities," *Discrete Appl. Math.* **25** (1989) 145 – 153.
19. P. Martin-Löf, "The definition of random sequences," *Inform. and Control* **9** (1966) 602 – 619.
20. M. Morse and G. Hedlund, "Unending chess, symbolic dynamics and a problem in semigroups," *Duke Math. J.* **11** (1944) 1 – 7.
21. J.A. Storer and T.G. Szymanski, "The macro model for data compression," *Proc. of 10th STOC*, 1978, 30 – 39.
22. A. Thue, "Uber unendliche Zeichenreihen," *Norske Vid. Selsk. Skr. Mat.-Nat. Kl. (Kristiania)* **7** (1906) 1 – 22.
23. A. Thue, "Uber die gegenseitige Lage gleicher Teile gewisser Zeichenreihen," *Norske Vid. Selsk. Skr. Mat.-Nat. Kl. (Kristiania)* **5** (1912) 1 – 67.

24. J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Trans. Information Theory* **23** (3) (1977) 337 – 343.

25. J. Ziv and A. Lempel, “Compression of individual sequences via variable length encoding,” *IEEE Trans. Information Theory* **24** (5) (1978) 530 – 536.