THE LEMPEL–ZIV COMPLEXITY OF FIXED POINTS OF MORPHISMS

SORIN CONSTANTINESCU[†] AND LUCIAN ILIE[†‡§]

Abstract. The Lempel–Ziv complexity is a fundamental measure of complexity for words, closely connected with the famous LZ77 compression algorithm. We investigate this complexity measure for one of the most important families of infinite words in combinatorics, namely the fixed points of morphisms. We give a complete characterization of the complexity classes which are $\Theta(1)$, $\Theta(\log n)$, and $\Theta(n^{1/k})$, $k \in \mathbb{N}$, $k \geq 2$, depending on the periodicity of the word and the growth function of the morphism. The relation with the well-known classification of Ehrenfeucht, Lee, Rozenberg, and Pansiot for factor complexity classes is also investigated. The two measures complete each other, giving an improved picture for the complexity of these infinite words.

Key words. combinatorics on words, infinite words, Lempel–Ziv complexity, fixed points, morphisms, factors

AMS subject classifications. 68R15, 68P30

1. Introduction. Before publishing their famous papers introducing the well-known compression schemes LZ77 and LZ78 in [36] and [37], resp., Lempel and Ziv introduced a complexity measure for words in [21] which attempted to detect "sufficiently random looking" sequences. In contrast with the fundamental measures of Kolmogorov [19] and Chaitin [4], Lempel and Ziv's measure is computable. The definition is purely combinatorial; its basic idea, splitting the word into minimal never seen before factors, proved to be at the core of the well-known compression algorithm LZ77, as well as subsequent variations. Another, closely related variant is to decompose the word into maximal already seen factors, as introduced by Crochemore [7] as a tool for algorithm design.

Lempel–Ziv-type complexity and factorizations have important applications in many areas, such as data compression [36, 37], string algorithms [7, 20, 25, 32], cryptography [26], molecular biology [5, 15, 16], and neural computing [1, 34, 35].

Lempel and Ziv [21] investigate various properties which are expected from a complexity measure which intends to detect randomness. They prove it to be subadditive and also that most (but not too many) sequences are complex; see [21] for details. Also, they test it against de Bruijn words, [3], as a well-established case of complex words – de Bruijn words contain as factors all words of a given length, within the minimum possible space. Therefore, they establish the first connection with the factor complexity, which is also one of our topics.

In this paper, we investigate the Lempel–Ziv complexity from the combinatorial point of view and not from an information theoretical perspective. Nevertheless, some implications of our results to data compression are obtained. We shall consider the Lempel–Ziv complexity for one of the most important classes of infinite words in combinatorics, namely the fixed points of morphisms. Many famous infinite words, such as Fibonacci or Thue-Morse, belong to this family; see, e.g., [23].

The fundamental nature of this measure allows for a complete characterization of the complexity of infinite fixed points of morphisms. The lowest complexity, constant,

[†]Department of Computer Science, University of Western Ontario, London, Ontario, N6A 5B7, CANADA

[‡]Research partially supported by NSERC.

[§]Corresponding author; e-mail: ilie@csd.uwo.ca

or $\Theta(1)$, is encountered for the simplest words, that is, ultimately periodic. For non-periodic words, the complexity depends on the growth function of the underlying morphism for the letter on which the morphism is iterated. Thus, for polynomial growth we obtain $\Theta(n^{1/k})$, $k \in \mathbb{N}$, $k \geq 2$, whereas for the exponential growth the complexity is $\Theta(\log n)$. We give examples for which each of the above complexities is reached.

An interesting by product of this research is the observation that LZ77 will succeed in compressing these infinite fixed points down to 0 bits/symbol asymptotically which is desireable of any good compression algorithm since the underlying mechanism generating these infinite words has only finite amount of information.

Our results are similar with the well-known ones of Ehrenfeucht, Lee, and Rozenberg [9], Ehrenfeuct and Rozenberg [10, 11, 12, 13, 14, 30] and Pansiot [27, 28] who provided the same characterization for the factor complexity. Comparing the two characterizations, we find out that they complete each other in an interesting way. While theirs distinguishes four complexity classes for the exponential case, ours gives an infinite hierarchy (given by the parameter k above) in the polynomial case, corresponding to their quadratic complexity.

The paper is structured as follows. After some basic definitions in the next section, we introduce the Lempel–Ziv complexity and related concepts in Section 3. Section 4 contains an important intermediate result which characterizes the complexity of powers of a morphism. Using it, our complete characterization is proved in Section 5 where examples which reach each complexity involved are shown. The comparison with the characterization of factor complexity is included in Section 6. Many problems need to be investigated about the Lempel-Ziv complexity. We propose several in the last section.

2. Basic notions. We introduce here the basic definitions and concepts we need. For further details we refer the reader to [6, 22, 23, 24].

Let Σ be an alphabet (finite non-empty set) and denote by Σ^* the free monoid generated by Σ, that is, the set of all finite words over Σ. The elements of Σ are called letters and the empty word is denoted ε. The length of a word w is denoted $|w|$ and represents the number of letters in w; e.g., $|\mathsf{abaab}| = 5$ and $|\varepsilon| = 0$.

Given the words $w, x, y, z \in \Sigma^*$ such that $w = xyz$, x is called a *prefix*, y is a *factor* and z a *suffix* of w; we use the notation $x \leq w$. If moreover $x \neq w$, then x is a *proper* prefix of w, denoted $x < w$. The prefix of length n of w is denoted $\mathsf{pref}_n(w)$.

An infinite word is a function $w : \mathbb{N} \setminus \{0\} \to \Sigma$. A finite word can be viewed as a function $w : \{1, 2, \ldots, |w|\} \to \Sigma$. In either case, the factor of w starting at position i and ending at position j, will be denoted by $w(i, j) = w_i w_{i+1} \ldots w_j$. The set of all factors of w is $F(w)$. The set of letters of Σ that actually occur in w is denoted $\Sigma(w)$. The set of infinite words over Σ is denoted Σ^ω. An infinite word w is *ultimately periodic* if $w = uvvv\ldots$, for some $u, v \in \Sigma^*$, $v \neq \varepsilon$. When we say w is non-periodic, we mean it is not ultimately periodic.

A morphism is a function $h : \Sigma^* \to \Delta^*$ such that $h(\varepsilon) = \varepsilon$ and $h(uv) = h(u)h(v)$, for all $u, v \in \Sigma^*$. Clearly, a morphism is completely defined by the images of the letters in the domain. For all our morphisms, $\Sigma = \Delta$.

A morphism $h : \Sigma^* \to \Sigma^*$ is called *non-erasing* if $h(a) \neq \varepsilon$, for all $a \in \Sigma$, *uniform* if $|h(a)| = |h(b)|$, for all $a, b \in \Sigma$, and *prolongeable* on $a \in \Sigma$ if $a < h(a)$.

If h is prolongeable on a, then $h^n(a)$ is a proper prefix of $h^{n+1}(a)$, for all $n \in \mathbb{N}$. Therefore, the sequence $(h^n(a))_{n \geq 0}$ of words defines an infinite word $h^\infty(a) \in \Sigma^\omega$ that is a fixed point of h. Formally, the i-th letter of $h^\infty(a)$ is defined as being the

i-th letter of a power $h^n(a)$ whose length is greater than i. The fact that $h^\infty(a)$ is a well-defined fixed point of h is easily verified. Also, for h and a fixed, the fixed point is unique.

It is possible to have finite strings as fixed points of morphisms and one can also consider erasing morphisms, but the interesting case is that of non-erasing prolongeable morphisms. Therefore, when we say *fixed point* $h^\infty(a)$, we mean an infinite word obtained by iterating a non-erasing morphism h that is prolongeable on a.

3. Word histories and Lempel–Ziv complexity. Let w be a (possibly infinite) word. We now introduce a fundamental notion for the Lempel–Ziv complexity. We define the operator π that removes the final letter of a finite word w:

$$\pi(w) = w(1, |w| - 1) \ .$$

A *history* $H = (u_1, u_2, \ldots, u_n)$ of $w \neq \varepsilon$ is a factorization of w, $w = u_1 u_2 \ldots u_n$, having the property that $u_1 \in \Sigma$ and

$$\pi(u_i) \in F(\pi^2(u_1 u_2 ... u_i)) \ ,$$

for all $2 \leq i \leq n$. We assume also that all u_is are non-empty. This definition requires that any new factor u_i, excepting its last letter, appears before in the word. However, it is still possible that the whole u_i does occur before in w, or $u_i \in F(\pi(u_1 u_2 ... u_i))$. In this case u_i is called *reproductive*. Otherwise u_i is *innovative*.

Example 1. Consider the word $w = \mathsf{aaabaabbaba}$. A possible history of w is $(\mathsf{a}, \mathsf{aab}, \mathsf{aab}, \mathsf{bab}, \mathsf{a})$. The second and fourth components are innovative whereas the third and fifth are reproductive.

By definition, n is called the length of the history H and is denoted by $|H|$.

Two kinds of history are important to us. The first, directly connected to the definition of Lempel–Ziv complexity is the *exhaustive* history. A history H is exhaustive if all u_i, $2 \leq i \leq |H| - 1$, are innovative. In other words, the whole new factor u_i does not occur before in the word even if all its proper prefixes do. Clearly, the exhaustive history of a word is unique. Sometimes (e.g., in [2]) the exhaustive history is called *Lempel–Ziv factorization*.

By contrast with the exhaustive history, a *reproductive history* requires that all its factors have occurred before (they are reproductive), with the necessary exceptions of never seen before letters: a history $H = (u_1, u_2, \ldots, u_n)$ is reproductive if either

$$u_i \in F(\pi(u_1 u_2 ... u_i)) \quad \text{or} \quad u_i \notin F(\pi(u_1 u_2 ... u_i)) \text{ but then } u_i \in \Sigma.$$

The innovative factors in a reproductive history are single letters. A reproductive history need not be unique.

Example 2. For the word in Example 1, $(\mathsf{a}, \mathsf{aab}, \mathsf{aabb}, \mathsf{aba})$ is the exhaustive history, whereas $(\mathsf{a}, \mathsf{aa}, \mathsf{b}, \mathsf{aa}, \mathsf{b}, \mathsf{ba}, \mathsf{ba})$ and $(\mathsf{a}, \mathsf{aa}, \mathsf{b}, \mathsf{aab}, \mathsf{ba}, \mathsf{ba})$ are two reproductive histories.

The following result, due to [21], relates the exhaustive history with all other histories of a word.

Lemma 3. *The exhaustive history of a word is the shortest history of that word.*

By definition, the *Lempel–Ziv complexity* of a finite word w, denoted by $\mathrm{LZ}(w)$ is the length of the exhaustive history of w, that is, the number of factors in the Lempel–Ziv factorization. Therefore, by Lemma 3, for any history H, $\mathrm{LZ}(w) \leq |H|$.

The *Lempel–Ziv complexity of an infinite word* w is the function $\text{LZ}_w : \mathbb{N} \to \mathbb{N}$ defined by

$$\text{LZ}_w(n) = \text{LZ}(\mathsf{pref}_n(w))$$

as the complexity of finite prefixes of w.

REMARK 4. The Lempel–Ziv complexity of finite words can be computed in linear time by using suffix trees; see [7, 17].

4. The complexity of powers. The main result of this section is that the complexity of $h^n(a)$, as a function of n, is either linear or bounded for a non-erasing morphism h prolongeable on a. That is, either $\text{LZ}(h^n(a)) = \Theta(n)$ or $\text{LZ}(h^n(a)) = \Theta(1)$. Throughout this section, a is fixed and h is non-erasing and prolongeable on a.

Given the morphism h, we can assume, without loss of generality, that each letter of Σ occurs in $h^\infty(a)$, the fixed point of h. If that is not the case, h can be restricted to the set of those letters that do occur in w and the fixed point of the restriction will still be the same.

4.1. Maximal reproductive history. We show first that the complexity of powers is at most linear. To this end, we define the *maximal reproductive history*[1] of a finite word w, denoted $RH(w)$. For $w = w_1 w_2 \dots w_{|w|}$, $w_i \in \Sigma$, we define $RH(w) = (u_1, u_2, \dots, u_n)$ as follows:

- $u_1 = w_1$, the first letter of w,
- $u_{i+1} = \begin{cases} w_{|u_1 u_2 \dots u_i|+1}, & \text{if } w_{|u_1 u_2 \dots u_i|+1} \notin \Sigma(u_1 u_2 \dots u_i) \\ \text{longest } w \text{ with } w \in F(\pi(u_1 u_2 \dots u_i w)), & \text{if } w_{|u_1 u_2 \dots u_i|+1} \in \Sigma(u_1 u_2 \dots u_i) \end{cases}$

  for all $i \geq 2$.

With the exception of new single letters, $RH(w)$ is created by taking at each step the maximal factor that has occurred before. For the word in Example 1, the maximal reproductive history is $(\mathsf{a}, \mathsf{aa}, \mathsf{b}, \mathsf{aab}, \mathsf{ba}, \mathsf{ba})$.

From the definition it is clear that $RH(w)$ is a reproductive history. It follows from Lemma 3 that $|RH(w)| \geq \text{LZ}(w)$.

REMARK 5. The maximally reproductive history has been introduced independently by Crochemore [7] as a tool for algorithm design. It is more natural than the Lempel–Ziv factorization. Indeed, most applications we mentioned above use Crochemore's factorization. On the other hand, the two factorizations are very closely related. For historical reasons, we defined the Lempel–Ziv complexity as the number of factors in the Lempel–Ziv factorization but our assymptotical results hold as well for Crochemore's factorization. This can be seen directly, by looking at the proofs or from the following lemma which connects the lengths of the two histories.

LEMMA 6. *For any $w \in \Sigma^*$, we have*

$$\text{LZ}(w) \leq |RH(w)| \leq 2\,\text{LZ}(w) - 1.$$

Proof. The first inequality follows by Lemma 3. For the second, we show first that the maximal reproductive history is the shortest among all reproductive histories. Denote $RH(w) = (u_1, \dots, u_n)$ and consider another reproductive history, $(v_1, \dots, v_m)$.

[1] This is called *s-factorization* in [7, 25], *f-factorization* in [8], *Lempel-Ziv factorization* in [32] and *Crochemore factorization* in [2].

First, for all $1 \leq i \leq \min(n, m)$, we have $|v_1 \dots v_i| \leq |u_1 \dots u_i|$. Indeed, if this is not the case, consider the smallest i_0 for which it does not hold. In this case, $i_0 \geq 2$ and u_{i_0} appears in v_{i_0} as a factor but not at the end of v_{i_0}. Thus $|v_{i_0}| \geq 2$, so v_{i_0} is not a letter, and, by the definition of the reproductive histories, v_{i_0} must have occurred before. Therefore, u_{i_0} is not the longest prefix of $u_{i_0} \dots u_n$ which has occurred before, a contradiction. It follows immediately that $n \leq m$.

Consider then the exhaustive history of w: $(t_1, \dots, t_k)$. Put, for all $2 \leq i \leq k$, $t_k = s_k a_k$, $a_k \in \Sigma$. We construct the history H obtained from the factorization $(t_1, s_2, a_2, s_3, a_3, \dots, s_k, a_k)$ by removing the empty factors, if any. We have then $|H| \leq 2k - 1 = 2\,\mathrm{LZ}(w) - 1$. By the above, $|RH(w)| \leq |H|$ which concludes the proof. □

Notice that Lemma 3 can be easily proved in a similar way.

4.2. Morphic images of histories. The next step is to iterate reproductive histories through a morphism h. We will show a way to create a reproductive history of $h(w)$, given a reproductive history of w.

Let w be a word and $H = (v_0, v_1, \dots, v_n)$ a reproductive history of w. Let $1 = i_1 < i_2 < \dots < i_{|\Sigma(w)|}$ be the indexes corresponding to the single letter factors of H that have not occurred before. We define a factorization of $h(w)$, denoted $h(H)$, by replacing all factors of w that have occurred before by their image through h and the single letters v_{i_j}, by the history $RH(h(v_{i_j}))$. We claim that this is a reproductive history of $h(w)$.

Example 7. Let us consider the Thue–Morse morphism

$$\begin{aligned} t(\mathsf{a}) &= \mathsf{ab}\ , \\ t(\mathsf{b}) &= \mathsf{ba}\ , \end{aligned}$$

and the word from Example 1, $w = \mathsf{aaabaabbaba}$. A reproductive history H (in fact, $RH(w)$) and its image through t, $t(H)$, are:

$$\begin{array}{rcl} H & = & (\mathsf{a}\ ,\ \ \mathsf{aa}\ ,\ \ \mathsf{b}\ ,\ \ \mathsf{aab}\ ,\ \ \mathsf{ba}\ ,\ \ \mathsf{ba})\ , \\ h(H) & = & (\mathsf{a},\mathsf{b},\ \ \mathsf{abab},\ \ \mathsf{b},\mathsf{a},\ \ \mathsf{ababba},\ \ \mathsf{baab},\ \ \mathsf{baab})\ . \end{array}$$

Lemma 8. *If H is a reproductive history of w, then $h(H)$ is a reproductive history of $h(w)$.*

Proof. There are two kinds of factors in $h(H)$. One originates from a factor of H that has already occurred. If a factor u has already occurred in w, then its image $h(u)$ will have also occurred in $h(w)$.

Also, each factor of the history $RH(h(v_{i_j}))$ is either a new single letter, or has already occurred in the factor $h(v_{i_j})$ of w and therefore has occurred in w.

By selecting the first occurrence of all the single letters in $h(w)$, we conclude that each factor of $h(H)$ is either a factor that has already occurred, or a letter that has not been previously seen. Equivalently, $h(H)$ is a reproductive history of $h(w)$. □

4.3. Linear upper bound. With respect to the length of $h(H)$, we note that each letter in $\Sigma(w)$, originally a standalone factor of H, is transformed into the factorization $RH(h(v_{i_j}))$ and, consequently, each letter x of w prompts a $|RH(h(x))| - 1$ increase in the length of $h(H)$:

$$|h(H)| \leq |H| + \sum_{x \in \Sigma(w)} (|RH(h(x))| - 1)\ .$$

If we assume that all letters of Σ occur in w, then the increase in length is constant, which leads us to the following result.

PROPOSITION 9. *If $h : \Sigma^* \to \Sigma^*$ is non-erasing and $a < h(a)$, $a \in \Sigma$, then* $\mathrm{LZ}(h^n(a)) = O(n)$.

Proof. We will use the above method for iteratively creating histories for $h^n(a)$ that will have a linearly increasing length.

Let n_0 be the first integer for which $h^{n_0}(a)$ contains all letters of Σ:

$$n_0 = \min\{n \in \mathbb{N} \mid \Sigma(h^n(a)) = \Sigma\}$$

and let $H_0 = RH(h^{n_0}(a))$.

Applying the above method, $h(H_0)$ is a valid history for $h^{n_0+1}(a)$ and

$$|h(H_0)| = |H_0| + \sum_{x\in\Sigma}(|RH(h(x))| - 1) \ .$$

Iterating for $n \geq n_0$, we get

$$|h^{n-n_0}(H_0)| = |H_0| + (n - n_0)\sum_{x\in\Sigma}(|RH(h(x))| - 1)$$

or

$$|h^{n-n_0}(H_0)| = A \cdot n + B$$

where $B = |H_0| - n_0 \sum\limits_{x\in\Sigma}(|RH(h(x))| - 1)$ and $A = \sum\limits_{x\in\Sigma}(|RH(h(x))| - 1)$.

Since $h^{n-n_0}(H_0)$ is a valid history for $h^n(a)$, it follows that

$$\mathrm{LZ}(h^n(a)) \leq An + B$$

or $\mathrm{LZ}(h^n(a)) = O(n)$. □

The next result gives the inferior asymptotic limit for $\mathrm{LZ}(h^n(a))$. It is obvious that $\mathrm{LZ}(h^n(a))$, as a function of n, is increasing since $h^n(a) < h^{n+1}(a)$. The remaining part of this section is dedicated to showing that the growth of the Lempel–Ziv complexity of powers is at least linear unless the fixed point word is ultimately periodic.

Throughout the rest of this section, the word u is defined by $h(a) = au$.

4.4. Some technical results. We prove next two technical lemmata to be used later in the proof of the lower bound.

LEMMA 10. *If $h^p(u)h^{p+1}(u)$ occurs at most $|h^p(u)|$ positions before its last occurrence in*

$$h^{p+2}(a) = auh(u)\ldots h^p(u)h^{p+1}(u) \ ,$$

then $h^\infty(a)$ is ultimately periodic.

Proof. Let $\alpha = h^p(u)$. Since $\alpha h(\alpha)$ occurs at most $|\alpha|$ positions from the end of $t = h^{p+2}(a)$, there exists v with $|v| \leq |\alpha|$ such that $v\alpha h(\alpha)$ is a suffix of t and also $\alpha h(\alpha)$ is a prefix of $v\alpha h(\alpha)$. Let v be the minimal word that satisfies this property – in other words, v marks the occurrence of $\alpha h(\alpha)$ that is the closest to the end of $\pi(t)$; see Fig. 1.

Both α and $\alpha h(\alpha)$ are fractional powers of v:

$$\alpha = v^n v' \text{ with } v' \leq v, n \geq 1 \tag{1}$$

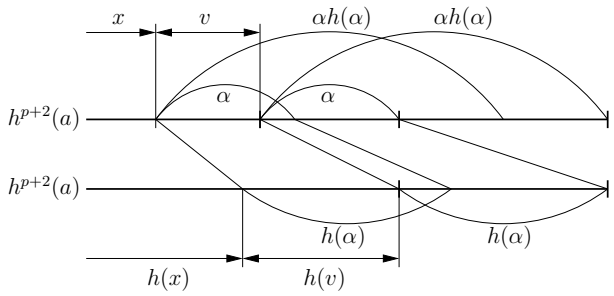

FIG. 1. *The occurrence of $\alpha h(\alpha)$ that has v as prefix is the closest to the end of $\pi(h^{p+2}(a))$.*

$$\alpha h(\alpha) = v^m v'' \text{ with } v'' \leq v, m \geq 2 \tag{2}$$

Let x be defined by $t = xv\alpha h(\alpha)$. Therefore $h(xv) = xv\alpha$. Since h is non-erasing, $|x| \leq |h(x)|$ which implies that $h(v)$ is a suffix of $v\alpha$.

By applying h to (1), we get that $h(\alpha) = h(v)^n h(v')$. This indicates that $h(v)h(\alpha)$ has period $|h(v)|$. However, $h(v)h(\alpha)$ is a suffix of $v\alpha h(\alpha)$ which has period $|v|$. By Fine and Wilf's theorem (see [6, 23]) $h(v)h(\alpha)$ has the period $d = \gcd(|v|, |h(v)|)$.

If $d < |v|$ then $h(v)$ has period d. Since α has period $|v|$, all factors of α of length $|v|$ are circular shifts of v. Consequently, the circular shift of v occurring at the end of $v\alpha$ is completely covered by $h(v)$ and, therefore, that particular circular shift of v has period d. However, the length of v is a multiple of d, so v is a non-trivial power of one of its proper prefixes of length d. In this case, we could find another occurrence of $\alpha h(\alpha)$ closer to the end t which contradicts the choice of v.

Therefore $d = |v|$ or $|v|$ divides $|h(v)|$. Furthermore $h(v)$ is a factor of some power of v since it is a factor of $v\alpha$, a fractional power of v. Let $r \in \mathbb{N}$ be defined by $r|v| = |h(v)|$. It follows that $h(v)$ is a circular shift of v^r. Inductively, if $h^s(v)$ is a circular shift of v^{r^s}, then $h^{s+1}(v)$ is a circular shift of $h(v)^{r^s}$ which is a circular shift of $(v^r)^{r^s} = v^{r^{s+1}}$. This implies that $|h^s(v)| = r^s|v|$ for all $s \geq 0$.

Because $h(v)$ is a suffix of $v\alpha$, it follows that $h^{s+1}(v)$ is a suffix of $h^s(v)h^s(\alpha)$. Inductively, if $h^s(v)$ has period $|v|$, then it is a power of some word of length $|v|$. Since $h^s(v)h^s(\alpha)$ is a fractional power of $h^s(v)$ by (1), it must also have period $|v|$ which implies that $h^{s+1}(v)$ has period v.

We have established that all $h^s(v)$ have period $|v|$ and their lengths are all multiples of $|v|$. We can now apply h^s to (1) and obtain

$$h^s(\alpha) = h^s(v)^n h^s(v') \text{ with } v' \leq v, n \geq 1.$$

Since $h^s(v)$ has period $|v|$ and its length is a multiple of that period, $h^s(\alpha)$ must also have the period $|v|$.

By a similar argument, using (2), $h^s(\alpha)h^{s+1}(\alpha)$ has period $|v|$. As this holds for all $s \geq 0$ and $|h^s(\alpha)| \geq |v|$, it follows that $\alpha h(\alpha) \dots h^s(\alpha) \dots$ has period $|v|$. ∎

LEMMA 11. *If $h^q(u)h^{q+1}(u)h^{q+2}(u)$ occurs before its last occurrence in*

$$h^{q+3}(a) = auh(u) \dots h^q(u)h^{q+1}(u)h^{q+2}(u)$$

and $|h^q(u)| < |h^{q+1}(u)|$, then $h^\infty(a)$ is ultimately periodic.

Proof. Let $\alpha = h^q(u)$ and $t = h^{q+3}(a)$. If $\alpha h(\alpha)h^2(\alpha)$ occurs at most $|\alpha|$ positions from its last occurrence in t as a suffix, then, by Lemma 10, $h^\infty(a)$ is ultimately periodic.

Otherwise, there exist words x and y such that

$$t = x\alpha y\alpha h(\alpha)h^2(\alpha) = h(x)h(\alpha)h(y)h(\alpha)h^2(\alpha)$$

and $h(\alpha)h^2(\alpha)$ is a prefix of $y\alpha h(\alpha)h^2(\alpha)$; see Fig. 2. By taking the lengths of the two factorizations of t, we have

$$|x| + |\alpha| + |y| + |\alpha| = |h(x)| + |h(\alpha)| + |h(y)|$$

or

$$|h(x)| - |x| = |\alpha| - (|h(\alpha)| - |\alpha|) - (|h(y)| - |y|) \ .$$

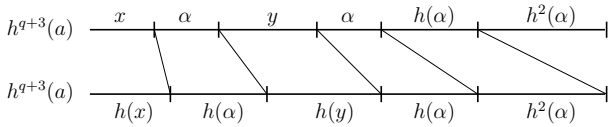

FIG. 2. *Here $h(\alpha)h^2(\alpha)$ is a prefix of $y\alpha h(\alpha)h^2(\alpha)$ and so $h^2(\alpha)$ is a prefix of $h(y)h(\alpha)h^2(\alpha)$.*

Since h is non-erasing, $|h(x)| - |x| < |\alpha|$. However, $h(\alpha)h^2(\alpha)$ is a prefix of $y\alpha h(\alpha)h^2(\alpha)$, so $h^2(\alpha)$ is a prefix of $h(y)h(\alpha)h^2(\alpha)$. This leads to $h(\alpha)h^2(\alpha)$ occurring at position $|h(x)|-|x|$ in the first occurrence of $\alpha h(\alpha)h^2(\alpha)$. Consequently, $h(\alpha)h^2(\alpha)$ occurs in t at distance less than $|h(\alpha)|$ symbols before its last occurrence in t which makes the Lemma 10 applicable for $p = q+1$ since $h(\alpha)h^2(\alpha) = h^p(u)h^{p+1}(u)$ occurs at most $|h(\alpha)| = |h^p(u)|$ positions before its last occurrence in $t = h^{q+3}(a) = h^{p+2}(a)$. □

4.5. Growth functions. In order to be able to use Lemma 11, we need to find values of q for which $|h^q(u)| < |h^{q+1}(u)|$. It is clear that $|h^q(u)| \le |h^{q+1}(u)|$ and, if there exists a letter z in $h^q(u)$ satisfying $|h(z)| \ge 2$, the inequality is strict. We shall prove that such powers must exist or else the fixed point is ultimately periodic. We need more definitions and results.

The *growth function* of the letter $x \in \Sigma$ in h is the function $h_x : \mathbb{N} \to \mathbb{N}$ defined by

$$h_x(n) = |h^n(x)| \ .$$

The following result from [31, 33] is very useful.

LEMMA 12. *There exist an integer $e_a \ge 0$ and an algebraic real number $\rho_a \ge 1$ such that*

$$h_a(n) = \Theta(n^{e_a}\rho_a^n) \ .$$

The pair (e_a, ρ_a) is called the *growth index* of a in h. We say that h_a (and a as well) is called *bounded*, *polynomial*, and *exponential* if a's growth index w.r.t. h is $(0,1)$, $(>0,=1)$, $(\ge 0,>1)$, resp.

EXAMPLE 13. All letters of a uniform morphism with images of length k share the same growth index: $(0,k)$. For instance, the growth index of a for the Thue-Morse morphism of Example 7 is $(0,2)$.

Let us consider the morphism h defined by:

$$h(\mathsf{a}) = \mathsf{ab}\ ,$$
$$h(\mathsf{b}) = \mathsf{bc}\ ,$$
$$h(\mathsf{c}) = \mathsf{c}\ .$$

The growth index of a is $(2,1)$, the growth index of b is $(1,1)$ and, finally, the growth index of c is $(0,1)$.

4.6. The associated graph. We introduce the following graph which is very useful for some proofs. Given a morphism $h : \Sigma^* \to \Sigma^*$, we denote the sets of bounded, polynomial, and exponential letters by Σ_B, Σ_P, and Σ_E, resp. The *graph associated with* h is the directed graph

$$G^h = (\Sigma, \{(a,b) \mid b \in F(h(a))\})\ .$$

Thus, the vertices of G^h are the letters of the alphabet and there is an edge from a to b if b appears in the image of a.

Consider its subgraphs G^h_X, induced by the sets Σ_X, $X \in \{B, P, E\}$, of vertices, resp. A few observations about the graphs we just defined are in order:

1. Any letter a belonging to two distinct cycles of G^h is exponential, as some power $h^r(a)$ would contain at least two as.
2. Let us fix the order $B < P < E$. Then for any X and any $a \in \Sigma_X$, the image $h(a)$ of a must contain at least one letter from Σ_X and cannot contain any letter from Σ_Y, for any $Y > X$.
3. The above observation implies that, as soon as Σ_X is non-empty, there is a cycle (which might be a loop) in G^h_X and from each vertex in G^h_X there is a path leading to a vertex in a cycle (everything in G^h_X).

Example 14. Consider the morphism h:

$$h(\mathsf{a}) = \mathsf{acb}\ ,$$
$$h(\mathsf{b}) = \mathsf{bca}\ ,$$
$$h(\mathsf{c}) = \mathsf{c}\ .$$

The graph G^h is shown in Fig. 3. This is also the graph of a different morphism: $\mathsf{a} \mapsto \mathsf{abc}$, $\mathsf{b} \mapsto \mathsf{bac}$, $\mathsf{c} \mapsto \mathsf{c}$ which indicates that different morphisms can produce isomorphic graphs.

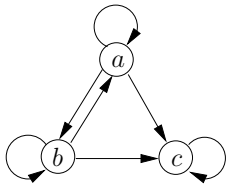

Fig. 3. *The graph G^h for Example 14.*

4.7. Linear lower bound for non-periodic words. We need only one more lemma before proving the main result of this section.

Lemma 15. *If $h(a) = au$, $u \in \Sigma^*$, $u \neq \varepsilon$, then there exist $m, p \in \mathbb{N}$ such that $|h^{m+jp}(a)| < |h^{m+jp+1}(a)|$, for all $j \geq 0$, or else $h^\infty(a)$ is ultimately periodic.*

Proof. Since h is prolongeable on a, it means a is not bounded. Assume $a \in \Sigma_P$; the case $a \in \Sigma_E$ is similar. Denote also $u = u_1 u_2 \dots u_{|u|}$, $u_i \in \Sigma$. We have in G^h the edges (a, a) and (a, u_i), for all $1 \le i \le |u|$.

If all u_is are in Σ_B, then $|h^n(u)|$ is bounded as $|h^n(u)| = \sum_{i=1}^{|u|} h_{u_i}(n)$. Hence, we can find n and r such that $h^n(u) = h^{n+r}(u)$, implying that $h^\infty(a) = auh(u)h^2(u)\dots$ is ultimately periodic.

Assume $u_i \in \Sigma_P$, for some i. By the above properties of G^h, we can find in G_P^h a path from u_i to a vertex which belongs to a cycle which is also in G_P^h. There must be a vertex, say z, in that cycle, whose outdegree is at least two, otherwise, all vertices in the cycle would be bounded. If we denote the length of the path from u_i to z by m and the length of the cycle by p, then $|h^{m+jp}| < |h^{m+jp+1}|$, for all $j \ge 0$, as claimed. □

Using Lemmata 11 and 15, we obtain, for all $j \ge 0$, that either

$$h^{m+jp}(u)h^{m+jp+1}(u)h^{m+jp+2}(u) \tag{3}$$

has never occurred before or w is ultimately periodic.

If we assume $w = h^\infty(a)$ to be non-periodic, then all factors of the form (3) can never occur before their last occurrence. This shows that there must exist a factor in the exhaustive history of w that ends within each distinct factor of the above mentioned form. It follows that $\textsc{lz}(h^n(a)) \ge \frac{1}{k}(n - n_0) + \textsc{lz}(h^{n_0+1}(a))$ or $\textsc{lz}(h^n(a)) = \Omega(n)$.

Combining this result with Proposition 9 we obtain that $\textsc{lz}(h^n(a))$ is either constant or linear. On the other hand, the fact that ultimate periodicity is equivalent to a bounded Lempel–Ziv complexity has been mentioned in [18]. Therefore, we proved the main result of this section.

Proposition 16. *For a non-erasing morphism h that admits the fixed point $h^\infty(a)$, $\textsc{lz}(h^n(a))$ is either $\Theta(1)$ if $h^\infty(a)$ is ultimately periodic or $\Theta(n)$ otherwise.*

5. Growth functions and infinite word complexity. Let w be an infinite word generated by iterating a non-erasing morphism h, $w = h^\infty(a)$. The prefix of a given length m of w will fall between two consecutive powers of h:

$$h^{n(m)}(a) \le \mathsf{pref}_m(w) < h^{n(m)+1}(a) \tag{4}$$

for a $n(m) \in \mathbb{N}$. If $\textsc{lz}(h^n(a))$ is bounded then $\textsc{lz}_w(n)$ is bounded. This establishes our first case for the complexity of $\textsc{lz}_w(\cdot)$, $\Theta(1)$.

When $\textsc{lz}(h^n(a))$ is not bounded, it has to be linear, by Proposition 16. Then a is not bounded and hence, by Lemma 12, we distinguish two cases:

1. $\rho_a = 1$ (h_a is polynomial). Then $|h^n(a)| = \Theta(n^{e_a})$ or $n(m) = \Theta(m^{1/e_a})$. Since, by (4), $\textsc{lz}(h^{n(m)}(a)) \le \textsc{lz}(\mathsf{pref}_m(w)) \le \textsc{lz}(h^{n(m)+1}(a))$ and $\textsc{lz}(h^n(a)) = \Theta(n)$, it follows that $\textsc{lz}_w(m) = \Theta(m^{1/e_a})$.
2. $\rho_a > 1$ (h_a is exponential). There exist ρ_1 and ρ_2 positive numbers such that $\rho_1^n \le |h^n(a)| \le \rho_2^n$ which means that $n(m) = \Theta(\log m)$. By the same argument, $\textsc{lz}_w(m) = \Theta(\log m)$

Notice however that h_a growing does not imply $\textsc{lz}_w(\cdot)$ unbounded. For instance, if $h(\mathsf{a}) = \mathsf{ab}$, $h(\mathsf{b}) = \mathsf{b}$, then h_a is polynomial but $w = h^\infty(\mathsf{a}) = \mathsf{abbb}\dots$ has bounded $\textsc{lz}_w(\cdot)$. For the exponential case we can take $h(\mathsf{a}) = \mathsf{aa}$ whose fixed point also has bounded Lempel–Ziv complexity.

Also, in the first case above, we cannot have $e_a = 1$ as this implies bounded Lempel–Ziv complexity, contradicting the assumption on $\textsc{lz}(h^n(a))$. Indeed, $e_a = 1$

implies $|h^n(a)| = \Theta(n)$ and so $|h^{n+1}(a)| - |h^n(a)|$ is bounded. Assuming $h(a) = au$, $u \neq \varepsilon$, we have $h^n(a) = auh(u)h^2(u)\cdots h^{n-1}(u)$. Consequently $|h^n(u)|$ is bounded hence we can find $h^n(u) = h^{n+p}(u)$ which implies $w = h^\infty(a)$ is ultimately periodic.

We have just proved the main result of the paper:

THEOREM 1. *For a fixed point infinite word $w = h^\infty(a)$ of a non-erasing morphism h, we have:*

1. *The Lempel–Ziv complexity of w is $\Theta(1)$ if and only if w is ultimately periodic.*
2. *If w is not ultimately periodic then the Lempel–Ziv complexity of w is $\Theta(\log n)$ or $\Theta(n^{1/k})$, $k \in \mathbb{N}$, $k \geq 2$, depending on whether h_a is exponential or polynomial, resp.*

Notice that the logarithmic Lempel–Ziv complexity in the exponential case was already proved in a different context by Ilie et al. [18, Lemma 12].

REMARK 17. Notice that the Lempel–Ziv complexity of fixed points is lower than the maximal Lempel–Ziv complexity, in the sense that there is no fixed point whose Lempel–Ziv complexity is of the order $\Theta(\frac{n}{\log n})$, which is the order of the maximum Lempel–Ziv complexity for finite words of length n, as proved by Lempel and Ziv [21].

Furthermore, since the LZ77-compressed size of a word w is $\Theta(\text{LZ}(w)\log|w|)$, it follows that the LZ77 compression algorithm will succeed in compressing the fixed points down to 0 bits/symbol asymptotically which is desireable of any good compression algorithm since the underlying mechanism generating these infinite words has only finite amount of information. Therefore, this is a positive conclusion regarding the usage of this algorithm to find random sequences, stating that the algorithm won't misclasify the infinite words considered in this paper.

REMARK 18. For a morphism h prolongeable on a, it is decidable to which of the classes in Theorem 1 its Lempel–Ziv complexity function belongs. First of all, a test for ultimate periodicity can be found in [29]. Assuming that the fixed point is not ultimately periodic, h_a is exponential if and only if there exists some letter b, accessible from a, deriving in a number of steps in a word containing two occurrences of b (see [31]). As noted above, this is equivalent with b belonging to two different cycles in the associated graph. This can be easily tested for each letter. An algorithm that decides whether or not h_a is exponential only needs to check if any of the letters belonging to two different cycles are reachable from a.

5.1. Examples. We give next examples showing that all the above complexities are indeed possible.

EXAMPLE 19. The highest Lempel–Ziv complexity is realized for $k = 2$, that means, $O(\sqrt{n})$, for the three letter morphism h_3 given by

$$\begin{aligned} h_3(\mathsf{a}) &= \mathsf{ab}\ , \\ h_3(\mathsf{b}) &= \mathsf{bc}\ , \\ h_3(\mathsf{c}) &= \mathsf{c}\ , \end{aligned}$$

for which $h_3^n(\mathsf{a}) = \mathsf{abc}^0\mathsf{bc}^1\ldots\mathsf{bc}^{n-1}$. Clearly, the growth function of a, $(h_3)_\mathsf{a}$, is quadratic whereas the complexity of powers is exactly linear which gives a final Lempel–Ziv complexity of $\sqrt{n}$; this can be checked directly by constructing the exhaustive history of $h_3^\infty(\mathsf{a})$:

$$(\mathsf{a}, \mathsf{b}, \mathsf{bc}, \mathsf{bc}^2, \mathsf{bc}^3, \ldots)\ .$$

This example can be easily extended to k letters. Let

$$h_k : \{\mathsf{a}_1, \mathsf{a}_2, \ldots, \mathsf{a}_k\}^* \to \{\mathsf{a}_1, \mathsf{a}_2, \ldots, \mathsf{a}_k\}^*$$

be defined by

$$\begin{aligned} h_k(\mathsf{a}_1) &= \mathsf{a}_1\mathsf{a}_2 \ , \\ h_k(\mathsf{a}_2) &= \mathsf{a}_2\mathsf{a}_3 \ , \\ &\vdots \\ h_k(\mathsf{a}_{k-1}) &= \mathsf{a}_{k-1}\mathsf{a}_k \ , \\ h_k(\mathsf{a}_k) &= \mathsf{a}_k \ . \end{aligned}$$

We have that $(h_k)_{\mathsf{a}_1}$ is a polynomial of degree $k-1$ (see [31, Theorem 3.5]). We can also see that directly, as follows. Note that h_k restricted to $\{\mathsf{a}_2, \mathsf{a}_3, \ldots, \mathsf{a}_k\}^*$ is actually h_{k-1} modulo the renaming $\mathsf{a}_2 = \mathsf{a}_1, \mathsf{a}_3 = \mathsf{a}_2, \ldots, \mathsf{a}_k = \mathsf{a}_{k-1}$. Since

$$|(h_k)_{\mathsf{a}_1}(n)| = |\mathsf{a}_1\mathsf{a}_2 h(\mathsf{a}_2)\ldots h_k^{n-1}(\mathsf{a}_2)| = 1 + \sum_{i=0}^{n-1} |(h_{k-1})_{\mathsf{a}_1}(n)| \ ,$$

we conclude inductively that, if $(h_{k-1})_{\mathsf{a}_1}(n) = \Theta(n^{k-2})$, then $(h_k)_{\mathsf{a}_1}(n) = \Theta(n^{k-1})$. The base case follows from the previous example for $k = 3$.

Consequently, the Lempel–Ziv complexity of the fixed point $h_k^\infty(\mathsf{a}_1)$ is $\Theta(\sqrt[k-1]{n})$. These examples illustrate the polynomial case.

Example 20. With respect to the exponential case, any uniform morphism with images of length k has a growth function of exactly k^n. Since the complexity of powers is linear for non-periodic words, the Lempel–Ziv complexity of the fixed point will be $\Theta(\log n)$.

Such an example is the famous Thue–Morse morphism, see Example 7, which fits the requirements for $k = 2$. Both fixed points $t^\infty(\mathsf{a})$ and $t^\infty(\mathsf{b})$ are non-periodic and the growth functions associated with both letters are exactly 2^n. Their Lempel–Ziv complexity is $\Theta(\log n)$.

Example 21. Another famous example is given by the Fibonacci morphism

$$\begin{aligned} f(\mathsf{a}) &= \mathsf{ab} \ , \\ f(\mathsf{b}) &= \mathsf{a} \ , \end{aligned}$$

for which we can precisely compute the value of $\textsc{lz}(f^n(\mathsf{a})) = n + 1$. The powers of the Fibonacci morphism grow exponentially, at the rate $\left(\frac{1-\sqrt{5}}{2}\right)^n + \left(\frac{1+\sqrt{5}}{2}\right)^n$ and therefore the Lempel–Ziv complexity of the infinite word is again $\Theta(\log n)$.

6. Comparison with factor complexity. We dedicate the final section to a comparison between the Lempel–Ziv complexity and the factor complexity for infinite words generated by morphisms. The factor complexity is a natural function defined as the number of factors of a certain length occurring in an infinite word. For a word $w \in \Sigma^\omega$, this is

$$f_w(n) = \mathrm{card}(\{u \in \Sigma^* \mid u \in F(w), |u| = n\}) \ .$$

The investigation of factor complexity for the fixed points of morphisms has been initiated by Ehrenfeucht, Lee, and Rozenberg in [9] (they actually considered the closely related D0L-systems) and continued by Ehrenfeucht and Rozenberg in a series of papers, see [10, 11, 12, 13, 14, 30]. The classification was completed by Pansiot [27, 28] who found also the missing complexity class $\Theta(n \log\log n)$.

The following definitions appear, with different names, in [6]. The morphism h is called[2]

- *non-growing* if there exists a bounded letter in Σ,
- *u-exponential* if $\rho_a = \rho_b > 1$, $e_a = e_b = 1$, for all $a, b \in \Sigma$,
- *p-exponential* if $\rho_a = \rho_b > 1$, for all a, b and $e_a > 1$, for some a, and
- *e-exponential* if $\rho_a > 1$, for all a and $\rho_a > \rho_b$, for some a, b.

Here is Pansiot's characterization:

THEOREM 2 (Ehrenfeucht, Lee, Rozenberg, Pansiot). *Let $w = h^\infty(a)$ be an infinite non-periodic word of factor complexity $f_w(\cdot)$.*

1. *If h is growing, then $f_w(n)$ is either $\Theta(n)$, $\Theta(n \log\log n)$ or $\Theta(n \log n)$, depending on whether h is u-, p- or e-exponential, resp.*
2. *If h is not-growing, then either*
   (a) *w has arbitrarily large factors over the set of bounded letters and then $f_w(n) = \Theta(n^2)$ or*
   (b) *w has finitely many factors over the set of bounded letters and then $f_w(n)$ can be any of $\Theta(n)$, $\Theta(n \log\log n)$ or $\Theta(n \log n)$.*

In order to establish a correspondence with our hierarchy, we note that, in the first case of Theorem 2, the function h_a is exponential, which implies a logarithmic Lempel–Ziv complexity. However, a logarithmic Lempel–Ziv complexity does not necessarily imply one of the n, $n \log\log n$ or $n \log n$ cases for the factor complexity as it is illustrated by the following example.

EXAMPLE 22. Consider the morphism h given by

$$\begin{aligned} h(\mathsf{a}) &= \mathsf{abc}\ , \\ h(\mathsf{b}) &= \mathsf{bac}\ , \\ h(\mathsf{c}) &= \mathsf{c}\ . \end{aligned}$$

Since h_a grows exponentially, $\mathrm{LZ}(h^\infty(\mathsf{a}))$ is, by Theorem 1, logarithmic. However, there exist arbitrarily large factors of $h^\infty(\mathsf{a})$ of the form c^n (c is bounded) which implies a $\Theta(n^2)$ factor complexity.

On the other hand, a radical-type Lempel–Ziv complexity does imply a quadratic factor complexity. To prove this, we need again the associated graph.

LEMMA 23. *Assume $h : \Sigma^* \to \Sigma^*$ is a non-erasing morphism prolongeable on $a \in \Sigma$. If h_a is polynomial, then there exist arbitrarily large factors over Σ_B in $h^\infty(a)$.*

Proof. Consider the associated graph introduced above. First, since h_a is polynomial, G_E^h must be empty.

By the properties of G^h, there exists at least one cycle in G_P^h, say C. If there is a vertex of C which has other outgoing edges (different from the one in C) in G_P^h, then any path starting with such an edge cannot go back to C (this would make the letters of C exponential). Therefore, further cycles can be constructed. As Σ_P is finite, there must be a cycle C' in G_P^h which has no outgoing edges in G_P^h except for those in the cycle. On the other hand, at least one vertex(letter) of C', say b, has an outgoing edge to a vertex in G_B^h. We have then $h(b) = ubv$, $uv \in \Sigma_B^*$, $uv \neq \varepsilon$. The letter b will create in $h^\infty(a)$ arbitrarily long factors from Σ_B^*, as claimed. □

Therefore, Theorems 1 and 2, Example 22 and Lemma 23 imply that the correspondence between Lempel–Ziv and factor complexities for fixed points of morphisms

[2]What we call *u-*, *p-*, and *e-exponential* are *quasi-uniform*, *polynomially diverging*, and *exponentially diverging*, resp., in [6, 27, 28]. We changed the terminology so that it does not conflict with the corresponding notions for h_a.

	Lempel–Ziv complexity	Factor complexity
$h^\infty(a)$ is ultimately periodic	$\Theta(1)$	$\Theta(1)$
$h^\infty(a)$ is not ultimately periodic and h_a is polynomial	$\Theta(n^{\frac{1}{2}})$	$\Theta(n^2)$
	$\Theta(n^{\frac{1}{3}})$	
	$\vdots$	
	$\Theta(n^{\frac{1}{k}})$	
	$\vdots$	
$h^\infty(a)$ is not ultimately periodic and h_a is exponential	$\Theta(\log n)$	$\Theta(n^2)$
		$\Theta(n \log n)$
		$\Theta(n \log \log n)$
		$\Theta(n)$

TABLE 1
Lempel–Ziv vs. factor complexity

is shown in Table 1, where all intersections are indeed possible.

We see that both measures of complexity recognize ultimately periodic words as having bounded complexity, the lowest class of complexity.

In the nontrivial case of non-periodic fixed points, the Lempel–Ziv complexity groups together all words $h^\infty(a)$ with h_a exponential, whereas the factor complexity distinguishes four different complexities. On the other hand, the factor complexity does not make any distinction among words with h_a polynomial, whereas Lempel–Ziv gives an infinite hierarchy.

7. Further research. Most combinatorial aspects of the Lempel–Ziv complexity need to be investigated. We mention a few problems below:

1. Characterize the fixed points of morphisms in each Lempel–Ziv complexity class (especially $\Theta(n^{\frac{1}{k}})$).
2. What is the connection between k in $\Theta(n^{1/k})$ and $\mathrm{card}(\Sigma)$?
3. Investigate the relations, in general, between Lempel–Ziv complexity and other complexity measures, especially the factor complexity.
4. How is Lempel–Ziv complexity affected by operations on words? For concatenation, it is subadditive, that is, $\mathrm{LZ}(uv) \leq \mathrm{LZ}(u) + \mathrm{LZ}(v)$, as proved by Lempel and Ziv [21]. Also, it is easy to see that it is monotonic for prefixes, that is, $\mathrm{LZ}(u) \leq \mathrm{LZ}(uv)$. But the same is not true for suffixes. Here is a counterexample: $\mathrm{LZ}(\mathsf{a.ab.aaba}) = 3$, $\mathrm{LZ}(\mathsf{a.b.aa.ba}) = 4$. Also, the behaviour with respect to the reversal operation (already asked in [18]) should be investigated, that is, the relation between the Lempel–Ziv complexity of w and that of w^R, the reversal of w.
5. Another complexity measure can be defined naturally from the factorization used in the LZ78 compression algorithm, which is: $w = u_1.u_2.\cdots.u_n$, such that, for all $i \geq 2$, u_i is the shortest prefix of $u_i u_{i+1} \ldots u_n$ that does not belong to the set $\{u_1, u_2, \ldots, u_{i-1}\}$. That means u_i may have appeared as a factor of $\pi(u_1 u_2 \ldots u_i)$ but not as a member of the factorization so far.

In particular, this factorization is a history. Denoting the new complexity by $\mathrm{LZ}_{78}(w)$ we have by Lemma 3 that $\mathrm{LZ}(w) \leq \mathrm{LZ}_{78}(w)$. Investigating this complexity measure is certainly of interest. The precise relation between the two complexity measures is not obvious and it may be that different techniques are required for investigating LZ_{78}.

Acknowledgement. The authors would like to thank the anonymous referees for very careful reading of the paper and for useful comments which helped improving the clarity of the presentation. Also, the second part of Remark 17 has been suggested by one of the referees.

REFERENCES

[1] J. M. Amigó, J. Szczepański, E. Wajnryb, and M. V. Sanchez-Vives, *Estimating the entropy rate of spike trains via Lempel-Ziv complexity*, Neural Computation 16(4) (2004) 717 – 736.
[2] J. Berstel and A. Savelli, *Crochemore factorization of Sturmian and other infinite words*, Proc. of MFCS'06, Lecture Notes in Comput. Sci. 4162, Springer, Berlin, 2006, 157 – 166.
[3] N.G. de Bruijn, *A combinatorial problem*, Nederl. Akad. Wetensch. Proc. 49 (1946) 758 – 764.
[4] G. Chaitin, *On the length of programs for computing finite binary sequences*, J. Assoc. Comput. Mach. 13 (1966) 547 – 569.
[5] X. Chen, S. Kwong and M. Li, *A compression algorithm for DNA sequences*, IEEE Engineering in Medicine and Biology Magazine 20(4) (2001) 61 – 66.
[6] C. Choffrut and J. Karhumäki, *Combinatorics on words*, in: G. Rozenberg, A. Salomaa, eds., Handbook of Formal Languages, Vol. I, Springer-Verlag, Berlin, Heidelberg, 1997, 329 – 438.
[7] M. Crochemore, *Recherche linéaire d'un carré dans un mot*, Comptes Rendus Acad. Sci. Paris Sér.I Math 296 (1983) 781 – 784.
[8] M. Crochemore and W. Rytter, *Text algorithms*, Oxford University Press, New York, 1994.
[9] A. Ehrenfeucht, K.P. Lee and G. Rozenberg, *Subword complexities of various classes of deterministic developmental languages without interaction*, Theoret. Comput. Sci. 1 (1975) 59 – 75.
[10] A. Ehrenfeucht and G. Rozenberg, *On the subword complexities of square-free D0L-languages*, Theoret. Comput. Sci. 16 (1981) 25 – 32.
[11] A. Ehrenfeucht and G. Rozenberg, *On the subword complexities of D0L-languages with a constant distribution*, Theoret. Comput. Sci. 13 (1981) 108 – 113.
[12] A. Ehrenfeucht and G. Rozenberg, *On the subword complexities of homomorphic images of languages*, RAIRO Informatique Théorique 16 (1982) 303 – 316.
[13] A. Ehrenfeucht and G. Rozenberg, *On the subword complexities of locally catenative D0L-languages*, Information Processing Letters 16 (1982) 7 – 9.
[14] A. Ehrenfeucht and G. Rozenberg, *On the subword complexities of m-free D0L-languages*, Information Processing Letters 17 (1983) 121 – 124..
[15] M. Farach, M.O. Noordewier, S.A. Savari, L.A. Shepp, A.D. Wyner, J. Ziv, *On the entropy of DNA: algorithms and measurements based on memory and rapid convergence*, Proc. of SODA'95, 1995, 48 – 57.
[16] V.D. Gusev, V.A. Kulichkov, O.M. Chupakhina, *The Lempel-Ziv complexity and local structure analysis of genomes*, Biosystems 30(1-3) (1993) 183 – 200.
[17] D. Gusfield, *Algorithms on Strings, Trees, and Sequences. Computer Science and Computational Biology*, Cambridge University Press, Cambridge, 1997.
[18] L. Ilie, S. Yu and K. Zhang, *Word complexity and repetitions in words*, Internat. J. Found. Comput. Sci. 15(1) (2004) 41 – 55.
[19] A.N. Kolmogorov, *Three approaches to the quantitative definition of information*, Probl. Inform. Transmission 1 (1965) 1 – 7.
[20] R. Kolpakov and G. Kucherov, *Finding maximal repetitions in a word in linear time*, Proc. of the 40th Annual Symposium on Foundations of Computer Science, IEEE Computer Soc., Los Alamitos, CA, 1999, 596 – 604.
[21] A. Lempel and J. Ziv, *On the complexity of finite sequences*, IEEE Trans. Inform. Theory 92(1) (1976) 75 – 81.
[22] M. Lothaire, *Combinatorics on Words*, Addison-Wesley, Reading, MA, 1983, (reprinted with corrections, Cambridge Univ. Press, Cambridge, 1997).
[23] M. Lothaire, *Algebraic Combinatorics on Words*, Cambridge Univ. Press, 2002.

[24] M. Lothaire, *Applied Combinatorics on Words*, Cambridge Univ. Press, 2005.
[25] M.G. Main, *Detecting leftmost maximal periodicities*, Discrete Appl. Math. 25(1-2) (1989) 145 – 153.
[26] S. Mund, *Ziv-Lempel complexity for periodic sequences and its cryptographic application*, Advances in Cryptology – EUROCRYPT '91, Lecture Notes in Comput. Sci. 547, Springer-Verlag, 1991, 114 – 126.
[27] J.-J. Pansiot, *Bornes inférieures sur la complexité des facteurs des mots infinis engendrés par morphismes itérés*, Proc. of STACS'84, Lecture Notes in Comput. Sci. 166, Springer, Berlin, 1984, 230 – 240.
[28] J.-J. Pansiot, *Complexité des facteurs des mots infinis engendrés par morphismes itérés*, Proc. of ICALP'84, Lecture Notes in Comput. Sci. 172, Springer, Berlin, 1984, 380 – 389.
[29] J.-J. Pansiot, *Decidability of Periodicity for Infinite Words*, RAIRO Theoretical Informatics and Applications, 20, 1986, 43 – 46.
[30] G. Rozenberg, *On subwords of formal languages*, Proc. of Fundamentals of Computation Theory, Lecture Notes in Comput. Sci. 117, Springer, Berlin-New York, 1981, 328 – 333.
[31] G. Rozenberg and A. Salomaa, *The Mathematical Theory of L Systems*, Academic Press, 1980.
[32] W. Rytter, *Application of Lempel-Ziv factorization to the approximation of grammar-based compression*, Theoret. Comput. Sci. 302(1-3) (2003) 211 – 222.
[33] A. Salomaa and M. Soittola, *Automata-Theoretic Aspects of Formal Power Series*, Springer, New York, 1978.
[34] J. Szczepański, M. Amigó, E. Wajnryb, and M.V. Sanchez-Vives, *Application of Lempel-Ziv complexity to the analysis of neural discharges*, Network: Computation in Neural Systems 14(2) (2003) 335 – 350.
[35] J. Szczepański, J. M. Amigó, E. Wajnryb, and M. V. Sanchez-Vives, *Characterizing spike trains with Lempel-Ziv complexity*, Neurocomputing 58-60 (2004) 79 – 84.
[36] J. Ziv and A. Lempel, *A universal algorithm for sequential data compression*, IEEE Trans. Inform. Theory 23(3) (1977) 337 – 343.
[37] J. Ziv and A. Lempel, *Compression of individual sequences via variable-rate coding*, IEEE Trans. Inform. Theory 24(5) (1978) 530 – 536.