Morphisms and BWT-Run Sensitivity

Gabriele Fici
Department of Mathematics and Computer Science, University of Palermo, Italy

Giuseppe Romana
Department of Mathematics and Computer Science, University of Palermo, Italy

Marinella Sciortino
Department of Mathematics and Computer Science, University of Palermo, Italy

Cristian Urbina
Department of Computer Science, University of Chile, Santiago, Chile
Centre for Biotechnology and Bioengineering (CeBiB), Santiago, Chile

Abstract

We study how the application of morphisms affects the number r of equal-letter runs in the Burrows–Wheeler Transform (BWT). This parameter has emerged as a key repetitiveness measure in compressed indexing. We focus on the notion of BWT-run sensitivity after application of morphisms. For binary alphabets, we characterize the class of injective morphisms that preserve the number of BWT-runs up to a bounded additive increase by showing that it coincides with the known class of primitivity-preserving morphisms, which are those that map primitive words to primitive words. We further prove that deciding whether a given binary morphism has bounded BWT-run sensitivity is possible in polynomial time with respect to the total length of the images of the two letters. Additionally, we explore new structural and combinatorial properties of synchronizing and recognizable morphisms. These results establish new connections between BWT-based compressibility, code theory, and symbolic dynamics.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Formal languages and automata theory; Mathematics of computing → Combinatorics on words

Keywords and phrases Burrows–Wheeler transform, BWT-runs, morphism, pure code, repetitiveness

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.49

Funding *Gabriele Fici*: Supported by MUR project PRIN 2022 APML – 20229BCXNW, funded by the European Union – Mission 4 "Education and Research" C2 – Investment 1.1.
Giuseppe Romana: Supported by the project "ACoMPA – Algorithmic and Combinatorial Methods for Pangenome Analysis" (CUP B73C24001050001) funded by the NextGeneration EU programme PNRR MUR M4 C2 Inv. 1.5 – Project ECS00000017 Tuscany Health Ecosystem (Spoke 6), CUP Master B63C22000680007 and by the INdAM – GNCS Project CUP_E53C24001950001.
Marinella Sciortino: Supported by the MUR PRIN Project "PINC, Pangenome INformatiCs: from Theory to Applications" (Grant No. 2022YRB97K), funded by Next Generation EU PNRR M4 C2, Inv. 1.1 and by the INdAM – GNCS Project CUP_E53C24001950001.
Cristian Urbina: Basal Funds FB0001 and AFB240001, ANID, Chile; FONDECYT Project 1-230755, ANID, Chile; ANID-Subdirección de Capital Humano/Doctorado Nacional/2021-21210580, ANID, Chile; NIC Chile Doctoral Scholarship, NIC, Chile.

1 Introduction

Morphisms are a powerful combinatorial mechanism for generating a collection of repetitive texts, and have been largely used in the field of combinatorics on words and formal languages [22, 31]. Formally, a morphism maps each character of an alphabet to a word over the same or another alphabet, by preserving the operation of concatenation. That is, if μ is a

morphism and u and v are words, then $\mu(uv) = \mu(u)\mu(v)$. Iterating morphisms can produce long and often highly repetitive sequences, which makes them a natural model for studying repetitiveness in words. Morphisms find applications in a wide range of contexts. Injective morphisms are widely used in information theory, data compression, and cryptography, as they define uniquely decodable codes [4]. More recently, morphisms have been employed in combination with copy-paste mechanisms to define novel compression schemes, known as NU-systems [28], further highlighting their versatility in modeling and processing repetitive data.

The Burrows–Wheeler Transform (BWT) is a reversible transformation introduced in 1994 in the field of data compression [6] and now underpins some of the most used tools in bioinformatics, such as `bwa` [21, 34] and `bowtie` [20, 19]. It permutes the characters of a text in a way that makes it more compressible, by clustering characters that precede similar contexts in the text. This property often results in long runs of identical characters, particularly in repetitive texts. The number r of such equal-letter runs, known as *BWT-runs*, has recently emerged as a measure of repetitiveness [27]. Several measures have been proposed to quantify repetitiveness in strings [26], such as the number z of phrases in the Lempel–Ziv parsing, the size g of the smallest context-free grammar generating the text, the size γ of the smallest string attractor [17, 8]. Among these, the measure r has recently attracted considerable attention due to its close connection with compressed indexing structures, such as the r-index [13], which use space proportional to r and support efficient pattern matching and retrieval in highly repetitive text collections, including genomic datasets and versioned document archives. Akagi et al. [1] explored the question of how much one character edit affects compression-based repetitiveness measures. In [14], the effect of single edit operations on the measure r has also been analyzed.

In this paper, we study how the application of a binary morphism affects the measure r, i.e., the number of BWT-runs. We focus on two notions of *BWT-run sensitivity*, which capture how much the number of BWT-runs can change when a morphism μ is applied to a word. The *additive sensitivity function* AS_μ gives, for every $n > 0$, the maximum increase in the number of BWT-runs that can occur when applying the morphism μ to any word of length n, while the *multiplicative sensitivity function* MS_μ gives, for every $n > 0$, the maximum ratio between the number of BWT-runs after and before the morphism μ is applied, over all words of length n. These notions allow us to quantify the impact of a morphism on the compressibility of the resulting text. An initial approach to the study of how morphisms affect the number of BWT-runs was given in [12], where we showed that Sturmian morphisms are the only binary injective morphisms that preserve the number of BWT-runs. Here, we tackle the problem of characterizing those binary morphisms that preserve the BWT-based compressibility of a text, in the sense that they have an additive sensitivity function bounded by a constant. We first prove that all non-injective binary morphisms have a bounded additive sensitivity function. In the injective case, which is the most interesting one, we prove that this class coincides with the known class of *primitivity-preserving morphisms*, which are those that map primitive words to primitive words. As a direct consequence, for these morphisms the multiplicative sensitivity function is also bounded. Primitivity-preserving morphisms are a well-studied class in algebraic theory of codes, and they are crucial in applications involving symbolic sequences, code synchronization, and the structural analysis of words [32, 30, 11, 25, 22, 4, 15].

In addition to establishing a novel connection between BWT-based compressed indexing, combinatorics on words, and code theory, a key contribution of our paper consists in identifying new combinatorial and structural properties of primitivity-preserving, recognizable, and

synchronizing morphisms. These properties are central to our main results but also hold independent interest in information theory and symbolic dynamics, where such morphisms play a fundamental role in coding, synchronization, and symbolic representations of dynamical systems [3, 4]. In fact, recognizability ensures that the morphic image of a word can be uniquely decomposed, up to rotations, into a sequence of morphic images of the letters of the alphabet. Synchronizing morphisms guarantee that a window of bounded length is sufficient to detect boundaries between codewords, a property that is crucial for decoding and synchronization in data streams.

We further show that all binary morphisms have bounded multiplicative sensitivity, but this result does not extend to alphabets with more than two symbols.

Furthermore, we prove that it is decidable in polynomial time whether the additive sensitivity function of a binary morphism is bounded by a constant, which makes our results practically applicable to the design of compression and indexing techniques that work directly on morphic encodings of highly repetitive text collections. Such a result builds upon fundamental results in the field of combinatorics on words, including properties of codes and solutions to word equations.

The rest of the paper is organized as follows. In Section 2, we present the preliminaries on words, morphisms, and the BWT. Section 3 introduces new combinatorial and structural properties of primitivity-preserving, recognizable, and synchronizing morphisms. Section 4 formalizes the sensitivity of r with respect to the application of morphisms and motivates our measures. Section 5 contains our main theorem characterizing the morphisms with a bounded additive sensitivity function. Section 6 discusses the multiplicative case, and Section 7 concludes with final remarks and open problems.

2 Preliminaries

Basics

Let $\Sigma = \{a_1, a_2, \ldots, a_\sigma\}$ be a finite sorted set of *letters* $a_1 < a_2 < \cdots < a_\sigma$, which we call an *alphabet*. A *finite word* $w = w[1]w[2]\cdots w[n]$ is any finite sequence of letters where $w[i] \in \Sigma$, for $i \in [1, n]$, and $n = |w|$ is the *length* of the word. The *empty word*, denoted by ε, is the unique word of length 0. The set of all finite words (resp. all non-empty words) over the alphabet Σ is denoted by Σ^* (resp. Σ^+). For a letter $a_i \in \Sigma$, $|w|_{a_i}$ denotes the number of occurrences of a_i in w. The vector $(|w|_{a_1}, \ldots, |w|_{a_\sigma})$ is called the *Parikh vector* of w.

If $u = u[1]\cdots u[n]$ and $v = v[1]\cdots v[m]$ are words, the *concatenation* of u and v is the word $uv = u[1]\cdots u[n]v[1]\cdots v[m]$. We let $\Pi_{i=1}^{k} w_i$ denote the concatenation of the words $w_1, w_2, \ldots, w_k$ in that order, and w^k the concatenation of the word w with itself k times.

For any $1 \le i \le j \le |w|$, we use the notation $w[i, j]$ to denote the word $w[i]w[i+1]\cdots w[j]$, which we call a *factor* of w. If $i > j$, then we assume $w[i, j] = \varepsilon$. We let $\mathcal{F}(w)$ denote the set of all factors of w. For any $\mathcal{L} \subseteq \Sigma^*$, we write $\mathcal{F}(\mathcal{L}) = \bigcup_{w\in\mathcal{L}} \mathcal{F}(w)$. A factor of w is *proper* if it is different from w itself. The factor $w[i, j]$ is called a *prefix* when $i = 1$, and a *suffix* when $j = n$. The *longest common prefix* between two words u and v is the longest word that is a prefix of both words. The length of this word is denoted by $lcp(u, v)$. The *longest common suffix* and the associated function *lcs* are defined symmetrically.

The *run-length encoding* of a word w, denoted $\mathsf{rle}(w)$, is the sequence of pairs (c_i, l_i) with $c_i \in \Sigma$ and $l_i > 0$, such that $w = c_1^{l_1} c_2^{l_2} \cdots c_r^{l_r}$ and $c_i \neq c_{i+1}$ for every $i \in [1, r-1]$. The length $|\mathsf{rle}(w)|$ is the number of *equal-letter runs* in w.

A *rotation*, or *conjugate*, of the word $w = w[1]w[2]\cdots w[n]$ is a word of the form $w[i+1, n]w[1, i]$, for some $0 \leq i < n$, obtained by shifting i letters cyclically. We let $\mathcal{R}(w)$ denote the multiset of all the $|w|$ rotations of w. A word in $\widetilde{\mathcal{F}}(w) := \mathcal{F}(\mathcal{R}(w))$ is called a *circular factor* of w.

A word w is *primitive* if for every word $u \in \Sigma^+$, $w = u^k$ implies $k = 1$; otherwise, w is called *non-primitive* (or *a power*). A word of length n is primitive if and only if it has exactly n distinct rotations, i.e., if $\mathcal{R}(w)$ has all-distinct elements. We let $Q(\Sigma^*)$ denote the set of all primitive words in Σ^*, and $\overline{Q(\Sigma^*)}$ the set of all non-primitive words in Σ^*. We say two non-empty words u, v *commute* if $uv = vu$. This is equivalent to saying that both uv and vu are not primitive.

Codes and morphisms

A set $X \subseteq \Sigma^+$ is a *code* if for all $m, \ell \geq 0$ and $u_i, v_j \in X$ with $i \in [1, \ell], j \in [1, m]$, the equation $u_1u_2\cdots u_\ell = v_1v_2\cdots v_m$ implies that $\ell = m$ and $u_k = v_k$, for all $k \in [1, \ell]$. Or equivalently, every word $w \in X^+$ has a unique factorization in words in X. Given a word $w \in X^+$, a word u is an *X-factor* of w if there exists a rotation w' of w (which can be w itself) that can be factored as $w' = sup$ such that $u, ps \in X^*$.

Whenever a code X consists of two words, the following property holds [16, 32].

▶ **Lemma 1.** *A set $X = \{u, v\}$, $u, v \in \Sigma^+$, is a code if and only if u and v do not commute, i.e., $uv \neq vu$.*

If u and v do not commute, then they are not powers of the same word, but in principle this does not exclude the case that either u, v, or both are non-primitive. For example, $X = \{aa, bb\}$ is a code.

Let Σ and Γ be two alphabets. A *morphism* μ is a map from Σ^* to Γ^* such that $\mu(uv) = \mu(u)\mu(v)$ for all words $u, v \in \Sigma^*$. Therefore, a morphism μ can be defined by specifying its action on the letters of Σ, and can therefore be denoted as $\mu = (\mu(a_1), \ldots, \mu(a_\sigma))$. The *size* of the morphism μ is defined as $|\mu| = \sum_{c\in\Sigma} |\mu(c)|$. When $\Sigma = \Gamma$, for all $t > 0$ and $w \in \Sigma^+$, we have $\mu^t(w) = \mu(\mu^{t-1}(w))$ and $\mu^0(w) = w$.

▶ Remark 2. Let μ be a morphism. If w and w' are conjugates, then so are $\mu(w)$ and $\mu(w')$. Moreover, since every conjugate of a power is a power, if $\mu(w)$ is a power, so is $\mu(w')$ for every conjugate w' of w.

A morphism μ is *cyclic* if there exists $z \in \Gamma^+$ such that $\mu(a) \in z^*$ for each $a \in \Sigma$. Otherwise, it is called *acyclic*.

As shown in the following proposition, there is a very strong relation between codes and injective morphisms.

▶ **Proposition 3** ([4])**.** *Let $X \subset \Gamma^*$ be a code. Then, any morphism $\mu : \Sigma^* \to \Gamma^*$ which induces a bijection of some alphabet Σ onto X is injective. Conversely, let $\mu : \Sigma^* \to \Gamma^*$ be an injective morphism. Then, $X = \mu(\Sigma)$ is a code.*

▶ Remark 4. By Lemma 1 and Proposition 3, one has that for a binary morphism $\mu : \{a, b\}^* \to \Gamma^*$, injectivity is equivalent to acyclicity, which in turn is equivalent to the condition $\mu(ab) \neq \mu(ba)$. For this reason, injective morphisms are often considered in applications, such as theory of codes, information theory, etc.

Examples of injective morphisms are the *Fibonacci morphism* $\varphi = (ab, a)$, the *Thue–Morse morphism* $\tau = (ab, ba)$, and the *period-doubling morphism* $\pi = (ab, aa)$.

From the relationship between codes and morphisms, many properties of codes are reflected in the corresponding properties of injective morphisms. Combinatorial properties of injective morphisms are explored in Section 3.

The Fibonacci morphism belongs to a wider class of morphisms called *Sturmian morphisms*, strictly related to the well-known Sturmian words [5]. Sturmian morphisms can be defined as those that can be obtained by composition from: the Fibonacci morphism φ, the morphism $E = (b, a)$, and the morphism $\tilde{\varphi} = (ba, a)$.

Let us suppose that both Σ and Γ are endowed with a total order relation that yields a lexicographic order, denoted by $<_\Gamma$ and $<_\Sigma$, respectively. A morphism $\mu : \Sigma^* \to \Gamma^*$ is *abelian order-preserving* if for every pair of distinct words $x, y \in \Sigma^*$ having the same Parikh vector, it holds that $x <_\Sigma y \iff \mu(x) <_\Gamma \mu(y)$. A morphism μ is *abelian order-reversing* if for every pair of distinct words x and y having the same Parikh vector, it holds that $x <_\Sigma y \iff \mu(x) >_\Gamma \mu(y)$. We simply write $<$ whenever Σ and Γ are clear from the context.

When $\Sigma = \{a, b\}$, the following result holds.

▶ **Lemma 5** ([12]). *Let $\mu : \{a, b\}^* \mapsto \Gamma^*$ be an acyclic morphism. Then μ is either abelian order-preserving or abelian order-reversing.*

For our purposes, the fact that binary acyclic morphisms are either abelian order-preserving or abelian order-reversing is a crucial property, since it implies that they preserve or reverse the order on the set of rotations of any given binary word.

Burrows–Wheeler Transform

The *Burrows–Wheeler transform* (BWT) of a word w, denoted by $\mathsf{bwt}(w)$, is a permutation of the letters of w obtained by sorting all the rotations of w in ascending lexicographic order and then concatenating the last letter of each rotation. The original word can be recovered if one stores the position where it appears in the list of sorted rotations. Figure 1 shows the sorted rotations of the word $w = \varphi^4(a) = abaababa$ and $bwt(w) = bbbaaaaa$.

We let $r(w)$ denote the number of equal-letter runs of $\mathsf{bwt}(w)$, i.e.,

$$r(w) = |\mathsf{rle}(\mathsf{bwt}(w))|.$$

Such a value can be considered as a measure of the repetitiveness of w. In fact, if a word w is highly repetitive, the number of equal-letter runs of its BWT tends to be small. From Figure 1, one can see that $r(abaababa) = 2$.

One can verify that for each word $v \in \mathcal{R}(w)$, $\mathsf{bwt}(v) = \mathsf{bwt}(w)$ and, consequently, $r(v) = r(w)$ and $r(\mu(v)) = r(\mu(w))$ for every morphism μ.

Let w be a non-primitive word, i.e., $w = z^p$, for some $z \in \Sigma^+$ and $p > 1$. It is well known that if $\mathsf{bwt}(z) = a_1 a_2 \cdots a_{|z|}$, then $\mathsf{bwt}(w) = a_1^p a_2^p \cdots a_{|z|}^p$ [24]. This implies that $r(w) = r(z)$.

Some results proved in [24, 29, 10] establish a strong connection between the BWT and Sturmian morphisms, as synthesized in the following theorem.

▶ **Theorem 6.** *Let w be a word over $\{a, b\}$ that is not a power of a single letter. Then the following are equivalent:*

1. *$w = (\mu(a))^\ell$ for a Sturmian morphism μ and for some $\ell > 0$.*
2. *$r(w) = 2$.*

a	a	b	a	a	b	a	**b**
a	a	b	a	b	a	a	**b**
a	b	a	a	b	a	a	**b**
a	b	a	a	b	a	b	**a**
a	b	a	b	a	a	b	**a**
b	a	a	b	a	a	b	**a**
b	a	a	b	a	b	a	**a**
b	a	b	a	a	b	a	**a**

Figure 1 BWT-matrix of the word $\varphi^4(a) = abaababa$: for each i, the ith row corresponds to the ith rotation of $\varphi^4(a)$ in lexicographic order, and the Burrows–Wheeler Transform $\mathsf{bwt}(\varphi^4(a)) = bbbaaaaa = b^3a^5$ is highlighted in bold in the last column. So, $r(abaababa) = 2$.

3 New combinatorial properties of injective morphisms

This section focuses on some combinatorial properties and characterizations of some classes of morphisms which are well-known in the context of coding theory and symbolic dynamics. The results provided in this section may be of independent interest and will later be related to BWT-run sensitivity in the next sections.

3.1 Primitivity-Preserving Morphisms

A morphism $\mu : \Sigma^* \to \Gamma^*$ is called *primitivity-preserving* if for every $w \in Q(\Sigma^*)$, it holds that $\mu(w) \in Q(\Gamma^*)$, that is, primitive words are mapped to primitive words. Primitivity-preserving morphisms are injective, and the associated codes are known in the literature as *pure codes* [25]. Such codes have been introduced in [30] to study the relationships between locally testable languages and synchronizing properties of codes.

Given a morphism $\mu : \Sigma^* \to \Gamma^*$, we call a primitive word w a *μ–power* if $\mu(w) = z^k$, for some primitive word z and an integer $k > 1$. Intuitively, it is a word that witnesses the non-primitivity-preserving property of a morphism. By P^μ we refer to the set of all μ–power words. From the definition, hence, $P^\mu = \emptyset$ if and only if the morphism μ is primitivity-preserving.

▶ **Example 7.** Let $\pi = (ab, aa)$ be the period-doubling morphism. The word b is a π–power, since $\pi(b) = a^2$. Hence, $b \in P^\pi$, and π is not primitivity-preserving.

▶ **Example 8.** Let $\mu = (a, bab)$. The word ab is a μ–power, since $\mu(ab) = (ab)^2$. Hence, $ab \in P^\mu$, and μ is not primitivity-preserving.

In this section, we prove a new characterization of the decompositions of binary primitivity-preserving morphisms. To do so, we first recall the following lemma, characterizing the combinatorial structure of binary primitivity-preserving morphisms.

▶ **Lemma 9** ([16, Theorem 3.1])**.** *Let $\mu = (u, v)$ be an injective morphism, with u, v two distinct primitive words. Then μ is a primitivity-preserving morphism if and only if all words in $\{u^n v^m \mid n, m \geq 1\}$ are primitive.*

The following lemma describes what happens when the property of Lemma 9 is not verified. In particular, it considers the combinatorial structure of the non-primitive words generated by the morphism when applied to some primitive word distinct from a single letter. Recall that if $u^n v^m = z^k$, for some primitive word z and $k > 1$, then we can derive that $n = 1$ or $m = 1$ (see [23]).

▶ **Lemma 10** ([2, 33]). *Let $\mu = (u, v)$ be an injective morphism, and let $W = \{u^n v \mid n \geq 1\} \cup \{uv^n \mid n > 1\}$. Then, there is at most one primitive word z and one integer $k > 1$ such that $z^k \in W$, i.e. $|W \cap \overline{Q(\{a,b\}^*)}| \leq 1$. Moreover, let $Y = \mu(Q(\{a,b\}^*)^{\geq 2}) \cap \overline{Q(\{a,b\}^*)}$. Then $Y = \mathcal{R}(z^k) \cap \{u, v\}^+$.*

The next lemma provides a characterization of the structure of the set P^μ for an injective morphism μ.

▶ **Lemma 11.** *Let $\mu = (u, v)$ be an injective morphism, and let $W = \{u^n v \mid n \geq 1\} \cup \{uv^n \mid n > 1\}$. We can distinguish the two cases:*

1. *$|W \cap \overline{Q(\{a,b\}^*)}| = 0$. Then only one of the following occurs:*
   - **a.** *$P^\mu = \emptyset$;*
   - **b.** *$P^\mu = \{c\}$, for some $c \in \{a, b\}$;*
   - **c.** *$P^\mu = \{a, b\}$.*
2. *$|W \cap \overline{Q(\{a,b\}^*)}| = 1$. Then there exists a unique $w \in \{a,b\}^*$ such that $\mu(w) \in W \cap \overline{Q(\{a,b\}^*)}$, and only one of the following occurs:*
   - **a.** *$P^\mu = \mathcal{R}(w)$;*
   - **b.** *$P^\mu = \mathcal{R}(w) \cup \{c\}$, for some $c \in \{a, b\}$.*

Note that, among the cases described in Lemma 11, the Case 1a is the only one in which every primitivity-preserving morphism $\mu = (u, v)$ falls. In this case, both u and v are primitive words. If the morphism $\mu = (u, v)$ is not primitivity-preserving and $W \cap \overline{Q(\{a,b\}^*)} = \emptyset$, then we deduce from Lemma 11 that either only one between u and v is a non-primitive word (Case 1b), or both are non-primitive words (Case 1c).

A classification of the non-primitivity-preserving morphisms $\mu = (u, v)$ that fall in Cases 2a and 2b, with $W \cap \overline{Q(\{a,b\}^*)} \neq \emptyset$, and their respective μ-power words, can be derived from a result given in [15, Theorem 8].

The set of primitivity-preserving morphisms is closed under composition, as stated in the following lemma.

▶ **Lemma 12.** *Let $\mu_1 : \Sigma^* \to \Gamma^*$, $\mu_2 : \Gamma^* \to \Lambda^*$ be two morphisms. If μ_1 and μ_2 are both primitivity-preserving, then $\mu_2 \circ \mu_1$ is primitivity-preserving too.*

However, it is possible to obtain primitivity-preserving morphisms even as a composition of morphisms that do not necessarily satisfy this property. The following proposition gives a complete characterization.

▶ **Proposition 13.** *Let $\mu : \{a,b\}^* \to \{a,b\}^*$ be an injective morphism. The morphism μ is primitivity-preserving if and only if, for all $\psi, \chi : \{a,b\}^* \to \{a,b\}^*$ such that $\mu = \psi \circ \chi$, it holds that $\chi = (p, q)$ is a primitivity-preserving morphism and ψ is an injective morphism such that $P^\psi \cap \{p, q\}^+ = \emptyset$.*

▶ **Example 14.** Let $\mu = (abaa, aaab)$. One can verify that $\mu = \pi \circ \tau$, where π and τ are the period-doubling morphism and the Thue–Morse morphism, respectively. We have that π is not primitivity-preserving and $P^\pi = \{b\}$. Since τ is a primitivity-preserving morphism and $b \notin \{ab, ba\}^+$, from Proposition 13 it follows that μ is primitivity-preserving too.

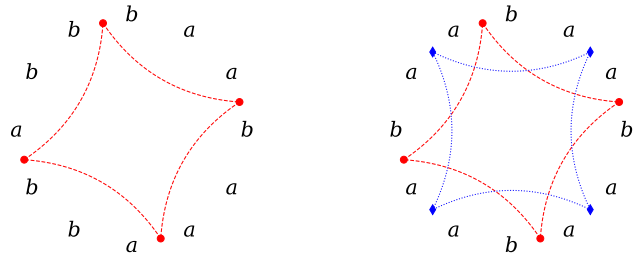
▶ **Example 15.** Let $\psi = (aba, b)$, and consider the morphism $\mu = (abab, baba) = \psi \circ \tau$, where τ is the Thue–Morse morphism, which is primitivity-preserving. In this case, ψ is not primitivity-preserving (since $\psi(ab) = (ab)^2$), nor is μ.

3.2 Recognizable Morphisms

In this subsection, we focus on some structural and combinatorial properties of morphisms that generate words admitting a unique factorization in circular factors, similarly to the notion of circular code [4].

Let $\mathcal{L} \subseteq \Sigma^*$. An injective morphism $\mu : \Sigma^* \to \Gamma^*$ is *recognizable* on $\mu(\mathcal{L})$ if for every non-empty word $w \in \mu(\mathcal{L})$ and every word $w' \in \mathcal{R}(w)$, there exist, and are unique, $p \in \Gamma^+$, $q \in \Gamma^*$, $z \in \Sigma^*$, and $c \in \Sigma$, such that $w' = q\mu(z)p$ and $pq = \mu(c)$. If $\mathcal{L} = \Sigma^*$, we simply say that μ is recognizable.

In other words, every image under a recognizable morphism μ has a unique circular factorization in words of $\mu(\mathcal{L})$. Equivalently, a recognizable morphism on $\mathcal{L}$ can be regarded as an injective map on the necklaces over $\mathcal{L}$, i.e., for all $x, y \in \mathcal{L}$, it holds that $\mathcal{R}(\mu(x)) = \mathcal{R}(\mu(y))$ if and only if $\mathcal{R}(x) = \mathcal{R}(y)$.



(a) $w = baaabbabbbaa$. **(b)** $w = baabaabaabaa$.

Figure 2 On the left, the unique circular factorization of $w = baaabbabbbaa$ into $\mu_1(a) = baa$ and $\mu_1(b) = abb$. On the right, two distinct circular factorizations of $w = baabaabaabaa$ into $\mu_2(a) = baa$ and $\mu_2(b) = aba$, respectively in blue and red.

▶ **Example 16.** Let us consider the injective morphism $\mu_1 = (baa, abb)$. Such a morphism is recognizable since every word in $\mu_1(\{a, b\}^*)$ has a unique circular factorization into the words $\mu_1(a)$ and $\mu_1(b)$, as shown in Figure 2a.

The recognizability of a morphism is well studied in the context of bi-infinite words and symbolic dynamics [3]. Here, it is adapted to necklaces, or circular words, which can be seen as periodic bi-infinite words. Note that most of the results known in the literature on bi-infinite words focus on the aperiodic case. Therefore, the results provided in this section can also be interpreted as contributions toward the less-explored setting of periodic bi-infinite words.

The following lemma establishes the close relationship between recognizable morphisms on $\mu(\Sigma^*)$ and a property related to the so-called very pure codes, which are properly included in the class of pure codes [30].

▶ **Lemma 17** ([4, Proposition 7.1.1])**.** *An injective morphism $\mu : \Sigma^* \to \Gamma^*$ is recognizable if and only if for every $u, v \in \Gamma^*$, if $uv, vu \in \mu(\Sigma^*)$ then $u, v \in \mu(\Sigma^*)$.*

In the case of binary injective morphisms, the next lemma provides a useful characterization of recognizable morphisms.

▶ **Lemma 18.** *A primitivity-preserving morphism $\mu = (u, v)$ is recognizable if and only if u and v are not conjugates.*

▶ **Example 19.** Consider the injective morphism $\mu_2 = (baa, aba)$. Since $\mu_2(a)$ and $\mu_2(b)$ are conjugates, by Lemma 18, μ_2 is not recognizable on $\mu_2(\{a,b\}^*)$, as shown in Figure 2b, where two distinct circular factorizations of the word *baabaabaabaa* are indicated. Indeed, $\mu_2(aaaa)$ and $\mu_2(bbbb)$ are equal up to rotations. Figure 3a shows two distinct circular factorizations of $(ab)^6$ into $\tau(a)$ and $\tau(b)$. Hence, τ is also not recognizable. One can note that $\mu_2 = \tilde{\varphi} \circ \tau$, where $\tilde{\varphi} = (ba, a)$, confirming the characterization established in the following theorem.

The following result shows an important structural property of the primitivity-preserving morphisms that are not recognizable. In particular, we prove that they can always be obtained by composing another morphism with the Thue–Morse morphism.

▶ **Theorem 20.** *Let $\mu : \{a,b\}^* \to \Gamma^*$ be a primitivity-preserving morphism. Then exactly one of the following cases occurs:*

1. *μ is recognizable;*
2. *$\mu = \psi \circ \tau$ for some injective morphism ψ, where τ is the Thue–Morse morphism.*

Proof. We show that, under the hypothesis of the theorem, it holds that $\mu = (uv, vu)$ if and only if $\mu = \psi \circ \tau$ for some injective morphism ψ, and by Lemma 18, the thesis directly follows. For the first direction, one can define $\psi = (u, v)$, and therefore $\mu = (\psi(\tau(a)), \psi(\tau(b))) = (\psi(ab), \psi(ba)) = (uv, vu)$. For the other direction, if $\mu = \psi \circ \tau$, then $\mu = (\psi(\tau(a)), \psi(\tau(b))) = (\psi(ab), \psi(ba)) = (\psi(a)\psi(b), \psi(b)\psi(a))$, and the thesis follows. ◀

3.3 Synchronizing Morphisms

An important notion in the context of injective morphisms is that of synchronization pair, which intuitively marks a position within a factor of a morphic image where the boundary between two codewords can be uniquely identified. Synchronization provides a way to "align" a segment of the morphic image of a circular word with the images of the letters of the alphabet.

Let $\mu : \Sigma^* \to \Gamma^*$, $\mathcal{L} \subseteq \Sigma^*$, and $u \in \widetilde{\mathcal{F}}(\mu(\mathcal{L}))$. We say that (u_1, u_2) is a *synchronization pair* of u on $\mu(\mathcal{L})$ if $u = u_1u_2$ and, for all $v_1, v_2 \in \Gamma^*$ and $f \in \widetilde{\mathcal{F}}(\mathcal{L})$, $v_1uv_2 = \mu(f)$ implies $v_1u_1 = \mu(f_1)$ and $u_2v_2 = \mu(f_2)$, for some $f_1, f_2 \in \widetilde{\mathcal{F}}(\mathcal{L})$ such that $f_1f_2 = f$.

Observe that a morphism μ is recognizable on $\mu(\mathcal{L}) \subseteq \mu(\Sigma^*)$ if and only if every word $w \in \mu(\mathcal{L})$ admits at least one synchronization pair, since from it one can uniquely recover the preimage $w' = \mu^{-1}(w)$, up to rotations.

The following notion of synchronization with delay gives a quantitative measure of the width of a window sliding along the morphic image of a circular word that guarantees the detection of a synchronization point. This is an adaptation to our setting of a definition commonly used in the context of HD0L-systems [7, 18].

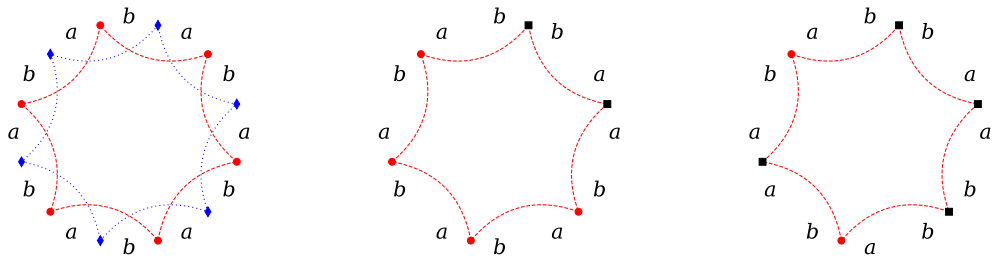
We say that a morphism $\mu : \Sigma^* \to \Gamma^*$ is *synchronizing with delay $k > 0$* for $w \in \mu(\Sigma^*)$ if every circular factor of w of length at least k admits a synchronization pair. Given $\mathcal{L} \subseteq \Sigma^*$, we say that μ is *synchronizing with delay $K > 0$* for $\mu(\mathcal{L})$ if it is synchronizing with a finite delay for every $w \in \mathcal{L}$ and

$$\sup \left\{ \min_{x \in \mathcal{L}} \{k \mid \mu \text{ is synchronizing with delay } k \text{ for } \mu(x)\} \right\} \leq K.$$

It has been proved [30, Theorem 5.1] that a morphism is recognizable if and only if it is synchronizing with finite delay for $\mu(\Sigma^*)$. The following example shows that the recognizability of a morphism on a proper subset of $\mu(\Sigma^*)$ does not necessarily imply being synchronizing with finite delay for that subset.

▶ **Example 21.** Let τ be the Thue–Morse morphism. Figure 3a shows that the Thue–Morse morphism τ is not recognizable. In fact, $\tau(aaaaaa)$ and $\tau(bbbbbb)$ are equal up to rotations and produce distinct circular factorizations. However, τ is recognizable on $\tau(\mathcal{L})$, where $\mathcal{L} = \{a^n b, b^n a \mid n > 0\}$, as shown in Figure 3b. Observe that even though τ is not recognizable, τ is recognizable on $\tau(\mathcal{L})$, since $(\tau(a), \tau(b))$ and $(\tau(b), \tau(a))$ are synchronization pairs that occur in every word of $\tau(\mathcal{L})$. In the figure, the unique circular factorization of $\tau(a^5 b) = (ab)^5 ba$ is depicted; the two black squares identify the synchronization pairs (b, b) and (a, a). Moreover, τ is synchronizing with delay 11 on $(ab)^5 ba$; more in general it is synchronizing with delay $2n+1$ for $(ab)^n ba$ and for $(ba)^n ab$. However, τ is not synchronizing with finite delay for $\tau(\mathcal{L})$, since the supremum of all minimum finite delays for all the words in $\mathcal{L}$ is unbounded.

Let $\mathcal{L}' = \tau(\{a, b\}^*) = \{ab, ba\}^*$. Unlike the previous cases, the synchronization pairs (a, a) and (b, b) occur in every word of $\widetilde{\mathcal{F}}(\tau(\mathcal{L}'))$ of length at least 5, hence τ is synchronizing with delay 5 for $\tau(\mathcal{L}')$. In fact, as shown in Figure 3c, every factor of $\tau(abbaab) = abbababaabbba$ having length at least 5 contains a black square that identifies a synchronization pair.



(a) $w = abababababab$. **(b)** $w = ababababbba$. **(c)** $w = abbabaababba$.

Figure 3 Circular factorizations into $\tau(a) = ab$ and $\tau(b) = ba$ are depicted, where τ is the Thue–Morse morphism. On the left, two distinct circular factorizations of $(ab)^6$ in blue and red, respectively; in the center, the unique circular factorizations of $w = ababababbba$; on the right, the unique circular factorizations of $w = abbabaababba$. Each black square identifies a synchronization pair.

We now give a new combinatorial characterization of synchronizing morphisms with finite delay on $\mu(\mathcal{L})$, for any $\mathcal{L} \subseteq \{a, b\}^*$. This characterization is based on the powers of single letters occurring in $\mathcal{L}$ in the case of non-recognizable primitivity-preserving morphisms, while it is based on the μ-power words in the case of non-primitivity-preserving morphisms.

▶ **Lemma 22.** *Let $\mu : \{a, b\}^* \to \Gamma^*$ be a non-recognizable primitivity-preserving morphism and let $\widetilde{\mathcal{F}}_a = \widetilde{\mathcal{F}}(\mathcal{L}) \cap \{a\}^*$ and $\widetilde{\mathcal{F}}_b = \widetilde{\mathcal{F}}(\mathcal{L}) \cap \{b\}^*$, where $\mathcal{L} \subseteq \{a, b\}^*$. Then μ is synchronizing with finite delay on $\mu(\mathcal{L})$ if and only if at least one of the sets $\widetilde{\mathcal{F}}_a$ or $\widetilde{\mathcal{F}}_b$ is finite.*

By using analogous techniques, one can prove that for any non-primitivity-preserving morphism, there exists a $k > 0$ such that each k-length factor w in $\widetilde{\mathcal{F}}(\mu(\Sigma^*))$ has a synchronization pair, unless $w \in \widetilde{\mathcal{F}}(\mu(z^*))$ for some $z \in P^\mu$.

▶ **Lemma 23.** *Let $\mu : \{a, b\}^* \to \Gamma^*$ be a non-primitivity-preserving morphism. Then, there exists an integer $k > 0$ such that every factor $w \in \Gamma^k \cap (\widetilde{\mathcal{F}}(\mu(\Sigma^*)) \setminus \widetilde{\mathcal{F}}(\{\mu(z^*) \mid z \in P^\mu\}))$ has a synchronization pair.*

From the previous lemma, the following result can be derived:

▶ **Theorem 24.** *Let $\mu : \{a, b\}^* \to \Gamma^*$ be a non-primitivity-preserving morphism and let $\mathcal{L} \subseteq \{a, b\}^*$. Then μ is synchronizing with finite delay on $\mu(\mathcal{L})$ if and only if the set $\widetilde{\mathcal{F}}(\mathcal{L}) \cap \{w^* \mid w \in P^\mu\}$ is finite.*

Table 1 The first column contains the list of all words of length 5, up to rotations. This is not restrictive, since rotations of the same word have the same value of r. The columns $r(w)$ and $r(\pi(w))$ are used to compute $AS_\pi(5)$.

w	$\mathsf{bwt}(w)$	$r(w)$	$\pi(w)$	$\mathsf{bwt}(\pi(w))$	$r(\pi(w))$
$aaaab$	$baaaa$	2	$abababababaa$	$babbbaaaaa$	4
$aaabb$	$baaba$	4	$abababaaaa$	$baaabbaaaa$	4
$aabab$	$bbaaa$	2	$ababaaababaa$	$bbaabaaaaa$	4
$aabbb$	$babba$	4	$ababaaaaaa$	$baaaaabaaa$	4
$ababb$	$bbbaa$	2	$abaaabaaaa$	$babaaaaaaa$	4
$abbbb$	$bbbba$	2	$abaaaaaaaa$	$baaaaaaaaa$	2

4 Sensitivity of the Measure r to the Application of Morphisms

Let μ be a morphism and w a word. In [12] we defined:

$$\Delta^+_\mu(w) = r(\mu(w)) - r(w)$$

and

$$\Delta^\times_\mu(w) = \frac{r(\mu(w))}{r(w)}.$$

Notice that $\Delta^+_\mu(w)$ may be negative for some word w. For example, let μ be the morphism over a 3-letter alphabet $\{a, b, c\}$ defined as $\mu = (b, a, c)$ and let $w = bcba$. One has that $r(w) = |\mathsf{rle}(\mathsf{bwt}(bcba))| = |\mathsf{rle}(bcab)| = 4$ and $r(\mu(w)) = r(acab) = |\mathsf{rle}(\mathsf{bwt}(acab))| = |\mathsf{rle}(cbaa)| = 3$. However, when μ is defined over a binary alphabet, one can prove that $\Delta^+_\mu(w)$ is always non-negative [12, Theorem 14].

▶ **Definition 25.** *The* BWT additive sensitivity function *and* BWT multiplicative sensitivity function *for a morphism μ are, respectively, the functions*

$$AS_\mu(n) = \max_{w\in\Sigma^n}(\Delta^+_\mu(w)) \quad \textit{and} \quad MS_\mu(n) = \max_{w\in\Sigma^n}(\Delta^\times_\mu(w))$$

Note that the additive sensitivity function is always a non-negative function, as for every n, $\Delta^+_\mu(a^n) \geq 0$ for any letter a.

▶ **Example 26.** Let us consider the period-doubling morphism π. Let us compute the value of the BWT additive sensitivity function for π when $n = 5$. From Table 1, it is possible to conclude that $AS_\pi(5) = MS_\pi(5) = 2$.

▶ Remark 27. Observe that for any pair of morphisms $\mu_1 : \Sigma^* \to \Gamma^*$ and $\mu_2 : \Gamma^* \to \Lambda^*$, and any word $w \in \Sigma^*$, it holds that $\Delta^+_{\mu_1\circ\mu_2}(w) = \Delta^+_{\mu_2}(w) + \Delta^+_{\mu_1}(\mu_2(w))$.

The following lemma shows that cyclic morphisms produce words with a fixed number of BWT-runs, whatever the words on which they are applied.

▶ **Lemma 28.** *Let $\mu : \{a, b\}^* \to \Gamma^*$ be a cyclic morphism. Then, there exist two constants $k^+_\mu, k^\times_\mu$, which depend on μ, such that $AS_\mu(n) = k^+_\mu$ and $MS_\mu(n) = k^\times_\mu$, for all $n \geq 2$.*

Proof. Recall that a binary morphism is cyclic if and only if there exist two integers $t_1, t_2 > 0$ and a non-empty word $z \in \Gamma^+$ such that $\mu(a) = z^{t_1}$ and $\mu(b) = z^{t_2}$. Hence, for each word $w \in \Sigma^+$ it holds that $r(\mu(w)) = r(z^{|w|_a t_1 + |w|_b t_2}) = r(z)$. Let us fix the claimed constants

$k_\mu = r(z) - 2$ and $k'_\mu = r(z)/2$. For all $n \geq 2$, let us consider the word $s_n = a^{n-1}b$. By Theorem 6, it follows that $r(s_n) = 2$. The proof follows by observing that since $r(\mu(w))$ is constant, the values of Δ^+_μ and $\Delta^\times_\mu$ are maximal when $r(w)$ assumes the smallest value, that in the case of binary words is 2, i.e., $AS_\mu(n) = \max_{w\in\Sigma^n}(r(\mu(w) - r(w)) = r(z) - r(s_n) = k^+_\mu$ and $MS_\mu(n) = \max_{w\in\Sigma^n}(r(\mu(w)/r(w)) = r(z)/r(s_n) = k^\times_\mu$. ◀

▶ **Example 29.** Let us consider the cyclic morphism $\mu = (ababbba, (ababbba)^2)$. It is possible to verify that for every $w \in \{a, b\}^+$, one has $\mu(w) = (ababbba)^p$, for some integer $p > 0$ depending on w. This means that $r(\mu(w)) = r(ababbba) = 6$ for every $w \in \{a, b\}^+$. For every length n, we can consider the word $a^{n-1}b$. We have $r(a^{n-1}b) = 2$, which is the lowest value that r can take on a binary word. Then, $AS_\mu(n) = 6 - 2 = 4$ and $MS_\mu(n) = 6/2 = 3$, for $n \geq 2$.

The following characterization of Sturmian morphisms in terms of the BWT additive sensitivity function was proved in [12].

▶ **Proposition 30** ([12, Theorem 21])**.** *Let μ be a binary injective morphism. Then $AS_\mu(n) = 0$ for every $n \geq 2$ if and only if μ is a Sturmian morphism.*

In the same paper, we showed that the Thue–Morse morphism τ increases by 2 the BWT-runs of every binary word, while in the case of the period-doubling morphism π, for each $n \geq 2$ we can find an n-length word w for which $\Delta^+_\pi(w) = \Theta(\sqrt{n})$. We summarize these results in the following proposition.

▶ **Proposition 31** ([12, Lemma 24 and Proposition 31])**.** *Let τ and π be the Thue–Morse and the period-doubling morphisms, respectively. The following properties hold:*

1. *$AS_\tau(n) = 2$, for all $n \geq 2$;*
2. *$AS_\pi(n) = \Omega(\sqrt{n})$.*

Note that τ is not the only morphism for which the additive sensitivity function is 2. In [12] it is proved that this property also holds for the *Thue–Morse-like* morphisms $\tau_{p,q} = (ab^p, ba^q)$, for some $p, q > 0$, and any composition of these morphisms with any Sturmian morphism.

▶ **Example 32.** Let us consider the morphism $\mu = (abbaab, ababba)$. By using Remark 27, it is possible to verify that $\mu = \tau \circ \varphi \circ \tau$, where τ and φ are, respectively, the Thue–Morse and the Fibonacci morphism. By using Propositions 30 and 31, item 1, it follows that $AS_\mu(n) = 4$ for all $n \geq 2$.

5 Characterization of Binary BWT-Run Preserving Morphisms

In this section, we are interested in morphisms having a bounded BWT additive sensitivity function.

▶ **Definition 33.** *Let $k \geq 0$ be an integer. A morphism $\mu : \Sigma^* \to \Gamma^*$ is called* k-BWT-run preserving *if for all $n \geq |\Sigma|$, $AS_\mu(n) \leq k$. We simply say* BWT-run preserving *if such a k exists.*

By Lemma 28, every cyclic morphism is BWT-run preserving since it has a bounded BWT additive sensitivity function. As a main result of this section, we characterize the acyclic (and therefore, by Remark 4, injective) binary morphisms which are BWT-run preserving. In particular, we prove that they coincide with the primitivity-preserving morphisms.

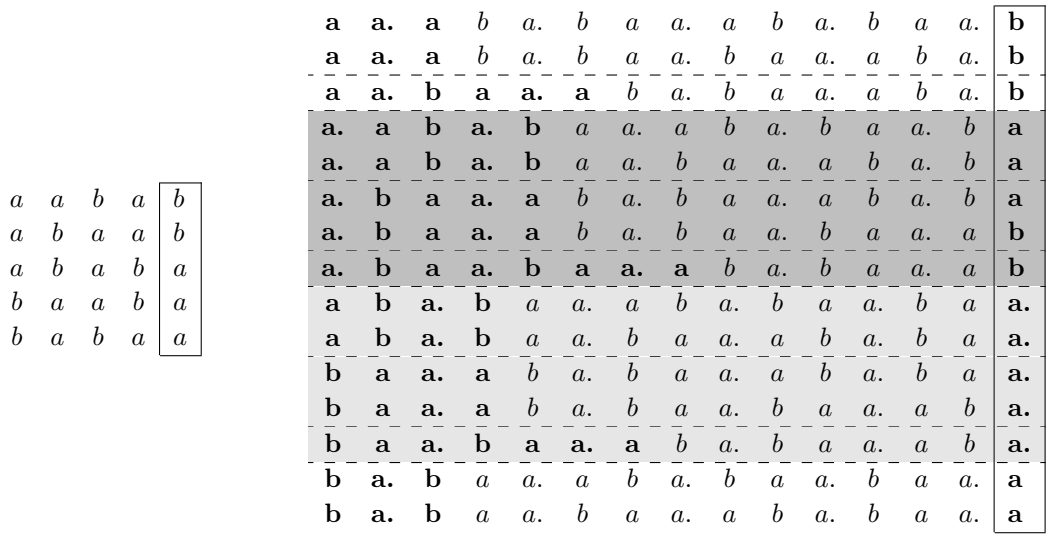


Figure 4 Comparison of the BWT–matrices for the word $w = aabab$ (on the left) and its image after application of the morphism $\mu = (baa, aba)$ (on the right). The dashed lines partition the rotations according to the shortest prefixes with at least one synchronization pair (highlighted in bold). The rotations in light gray correspond to the words in $\mu(\mathcal{R}(w))$. The rotations in dark gray correspond to the rotations where $\mathsf{bwt}(w)$ is spelled in reverse order.

We first give a lemma, in which we prove that the finite-delay synchronization of a morphism on the images of a language results in a bounded increase in the number of BWT-runs.

▶ **Lemma 34.** *Let $\mathcal{L} \subseteq \Sigma^*$, where $\Sigma = \{a, b\}$, and let $\mu : \Sigma^* \to \Gamma^*$ be synchronizing with delay $k > 0$ on $\mu(\mathcal{L})$. Then, there exists $k' > 0$ such that $\Delta^+_\mu(u) \leq k'$ for all $u \in \mathcal{L}$.*

The following lemma proves one direction of the main result.

▶ **Lemma 35.** *Let $\mu : \{a, b\}^* \to \Gamma^*$ be an injective morphism. If μ is primitivity-preserving, then μ is BWT-run preserving.*

Proof. If μ is primitivity-preserving, then by Theorem 20, either (i) μ is recognizable or (ii) there exist an integer $t > 0$ and a morphism $\psi : \{a, b\}^* \to \Gamma^*$ such that $\mu = \psi \circ \tau^t$ and $\psi \neq \psi' \circ \tau$ for all $\psi' : \{a, b\}^* \to \Gamma^*$. If we fall in case (i), the thesis follows from the equivalence between recognizable morphisms and synchronizing morphisms with bounded delay [30, Theorem 5.1] and Lemma 34.

If instead we fall in case (ii), then by Proposition 31, it follows that τ increases the BWT-runs by (at most) 2. Hence, the thesis is equivalent to showing that there exists $k \geq 0$ such that $r(\psi(u)) \leq r(u) + k$, for every $u \in \tau^t(\{a, b\}^*)$: by using Remark 27, this would prove that the BWT additive sensitive function is bounded by a constant, that is $AS_\mu(n) = \max_{w \in \Sigma^n}(\Delta^+_\mu(w)) = \max_{w \in \Sigma^n}(\Delta^+_\psi(\tau^t(w)) + \Delta^+_{\tau^t}(w)) \leq k + 2t$ for all $n > 0$. By Proposition 13, we can distinguish between two subcases: (ii.a) ψ is recognizable and (ii.b) ψ is not primitivity-preserving and $\tau^t(a) \notin P^\psi$. If (ii.a), the proof follows analogously to (i). If (ii.b), then by Lemma 10 all the ψ-power words in the set P^ψ consist of either a single letter or rotations of a unique word in the set $\{a^n b \mid n \geq 1\} \cup \{ab^n \mid n > 1\}$. Observe that $\widetilde{\mathcal{F}}(\tau^t(\Sigma^*)) \cap \{(a^n b)^m, (ab^n)^m \mid n \geq 3, m \geq 1\} = \emptyset$ and $\widetilde{\mathcal{F}}(\tau^t(\Sigma^*)) \cap \{(a^2 b)^m, (ab^2)^m \mid m \geq$

$1\} = \{a^2b, ab^2\}$, for all $t \geq 1$, while $\widetilde{\mathcal{F}}(\tau^t(\Sigma^*)) \cap \{(ab)^m \mid m \geq 1\} = \{ab, (ab)^2\}$ for $t \geq 2$ and $\widetilde{\mathcal{F}}(\tau(\Sigma^*)) \cap \{(ab)^m \mid m \geq 1\} = \{(ab)^m \mid m \geq 1\}$. Observe that only the latter does not have a finite size; however, for $t = 1$, by hypothesis, we have $ab \notin P^\psi$. Hence, $\widetilde{\mathcal{F}}(\tau^t(\Sigma^*))$ contains only a finite number of powers of elements from P^ψ, and the proof follows by Theorem 24. ◀

Now we prove the opposite direction. We consider a class of morphisms that we use to decompose a generic morphism. For any $p > 1$, let $\rho_p : \{a, b\}^* \to \{a, b\}^*$ denote the injective morphism (a, b^p). Observe that if $p > 1$, then ρ_p is not primitivity-preserving.

In the following proposition, we prove that such morphisms have an unbounded additive sensitivity function.

▶ **Proposition 36.** *Let $\rho_p = (a, b^p)$, for some $p > 1$. Then, $AS_{\rho_p}(n) = \Omega(\sqrt{n})$.*

In the following proposition, we consider a larger class of morphisms with an unbounded additive sensitivity function.

▶ **Proposition 37.** *Given an injective morphism $\mu : \{a, b\}^* \to \Gamma^*$, let $u, v \in Q(\Gamma^*)$ and $p, q \geq 1$ such that $\mu = (u^p, v^q)$. Then,*

$$\mu = \eta \circ \rho_q \circ E \circ \rho_p \circ E$$

where $\eta = (u, v)$. Moreover, if $pq > 1$, then $AS_\mu(n) = \Omega(\sqrt{n})$.

The following lemma shows that if a morphism has bounded additive sensitivity, then it is primitivity-preserving.

▶ **Lemma 38.** *Let $\mu : \{a, b\}^* \to \Gamma^*$ be a non-primitivity-preserving injective morphism. For each $k > 0$, there exists a word w such that $\Delta^+_\mu(w) > k$.*

Proof. If $\mu(a)$ or $\mu(b)$ are not primitive, then the thesis follows by Proposition 37, so let us assume that $\mu(a), \mu(b) \in Q(\Gamma^*)$.

Recall that a Lyndon word is a primitive word that is lexicographically smaller than all its proper conjugates. Since μ is injective, not primitivity-preserving, and both images are primitive words, by Lemma 11 there exists some Lyndon word $x \in P^\mu$ such that $|x| > 1$, $\mu(x) = z^t$ for some $t > 1$, and z is primitive. Let $\psi = (a, x)$ and $\eta = (\mu(a), z^t)$. Observe that: i) $\mu(\psi(a)) = \mu(a)$; and ii) $\mu(\psi(b)) = \mu(x) = z^t$. Hence, $\eta = \mu \circ \psi$. Then, by Proposition 37, there exists a word $w \in \{a, b\}^n$ such that $\Delta^+_\eta(w) = \Theta(\sqrt{n})$. Since the concatenation uv of two Lyndon words u and v, with $u < v$, is a Lyndon word (see [9]), then, for every $m \geq 1$, $a^m x$ and $a x^m$ are Lyndon words, hence, by Lemma 9, the morphism $\psi = (a, x)$ is primitivity-preserving, and by Lemma 35 ψ is BWT-run preserving. Finally, by using Remark 27, one has $\Delta^+_\eta(w) = \Delta^+_\mu(\psi(w)) + \Delta^+_\psi(w) = \Delta^+_\mu(\psi(w)) + O(1) = \Theta(\sqrt{n})$, and the thesis follows. ◀

▶ **Example 39.** Let $\mu = (ba, ababaa)$. Let $x = aab$ and $z = babaa$, where x is Lyndon and z is primitive. It holds that $\mu(x) = \mu(aab) = ba \cdot ba \cdot ababaa = (babaa)^2 = z^2$. We define the morphisms $\psi = (a, aab)$ and $\eta = (ba, (babaa)^2)$, as described in the proof of Lemma 38. Indeed, $\eta = \mu \circ \psi$, as $\mu(\psi(a)) = \mu(a) = ba = \eta(a)$ and $\mu(\psi(b)) = \mu(aab) = (babaa)^2 = \eta(b)$. The morphism η can be written as $\eta = (ba, babaa) \circ \rho_2$, and by Proposition 37 there exists $w \in \{a, b\}^*$ such that $r(\eta(w)) - r(w) = \Theta(\sqrt{n})$. On the other hand, ψ is primitivity-preserving, so it must be the case that $r(\mu(\psi(w))) - r(\psi(w)) = \Theta(\sqrt{n})$.

From Lemmas 35 and 38, the main result of the paper can be derived.

▶ **Theorem 40.** *Let $\mu : \{a,b\}^* \to \Gamma^*$ be an injective morphism. Then μ is BWT-run preserving if and only if it is primitivity-preserving.*

Finally, we can show that there exists a finite test case, as stated in the following theorem.

▶ **Theorem 41.** *Let $\mu : \{a,b\}^* \to \Gamma^*$ be an injective morphism. It is decidable in polynomial time in the size of μ whether μ is BWT-run preserving.*

Proof. By Lemma 9, to decide whether a given morphism $\mu = (u, v)$, for some $u, v \in \Gamma^+$, is primitivity-preserving, we have to check the primitiveness of all the possible non-trivial solutions of the equation $u^\ell v^m = z^n$. Let $t_{\max} = \max\{|u|, |v|\}$ and $t_{\min} = \min\{|u|, |v|\}$. In [15], it has been proved that there are at most $O(t_{\max}/t_{\min})$ words to check, each of these having length $\Theta(|u| + |v|)$. Since the primitiveness can be checked in linear time in the size of the words, the total time complexity is $O(t^2_{\max}/t_{\min})$. ◀

6 Morphisms with bounded multiplicative sensitivity

Even though in the case of binary morphisms the additive sensitivity is not always bounded by a constant, it is natural to wonder whether the multiplicative sensitivity is. As shown in the following example, this is not the case when the alphabet size is greater than 2.

▶ **Example 42.** Let $f^{\$}_k = \varphi^k(a)\$$ be the k-th Fibonacci word with a letter \$ such that $\$ < a < b$ appended. Define μ as $\mu(\$) = \$$, $\mu(a) = ab$, and $\mu(b) = a$. Then $\mu(f^{\$}_{2k}) = f^{\$}_{2k+1}$. It is known that $r(f^{\$}_{2k+1})/r(f^{\$}_{2k}) = \Omega(\log n)$ [14]. Hence, $MS_\mu(n) = \Omega(\log n)$.

Observe that if $p > 1$, then the morphism $\rho_p = (a, b^p)$ is not primitivity-preserving. We first show that $MS_{\rho_{p>1}}(n)$ is bounded.

▶ **Lemma 43.** *Let $w \in \{a,b\}^*$ be a word that contains at least two a's and one b. Let t be the length of the longest* circular run *of b's in w (i.e., the longest run of b's in any string in $\mathcal{R}(w)$). It holds that*

$$r(\rho_p(w)) \le r(w) + 2\left|\widetilde{\mathcal{F}}(w) \cap \bigcup\{ab^i a \mid i \in [1,t]\}\right|,$$

where $\rho_p = (a, b^p)$. Moreover, it holds $\Delta^+_{\rho_p}(w) \le 2r(w)$ and $\Delta^\times_{\rho_p}(w) \le 3$.

Proof. Let t be the length of the longest circular run of b's in w. Since $\rho_p = (a, b^p)$ is order-preserving, and because the last characters of $\rho_p(a)$ and $\rho_p(b)$ are a and b, respectively, the sequence obtained by taking the last character of each image of the lexicographically sorted rotations of w is exactly $\mathsf{bwt}(w)$. Moreover, the string formed by concatenating the last characters of the range of rotations starting with a in the BWT matrix of $\rho_p(w)$ is exactly $\mathsf{bwt}(w)[1, |w|_a]$. Similarly, the string formed by concatenating the last characters of the (disjoint) ranges of rotations starting with $b^{ip}a$ for $i \in [1,t]$ is exactly $\mathsf{bwt}(w)[|w|_a + 1, |w|]$. Strictly in between the ranges of rotations starting with $b^{(i-1)p}a$ and $b^{ip}a$ for some $i \in [1,t]$, there is a range of rotations starting with $b^{(i-1)p+s}a$ for each $s \in [1, p-1]$, all ending with the character b. In the worst case, each of these blocks of rotations can only increase the number of runs by 2. Hence, the additive increase is at most 2 times the number of circular factors of the form ab^ia in w. This proves the first claim of the proposition.

For the second claim, observe that a change of character occurs in correspondence with each block of rotations starting with b^ia, for each i such that $ab^ia \in \widetilde{\mathcal{F}}(w)$. Hence, the second claim follows because $|\widetilde{\mathcal{F}}(w) \cap \bigcup\{ab^ia \mid i \in [1,t]\}| \le r(w)$. ◀

We now give a sketch of the main result of this section.

▶ **Theorem 44.** *For every morphism $\mu : \{a, b\}^* \to \Gamma^*$, there exists an integer k_μ such that $MS_\mu(n) \leq k_\mu$.*

Proof (sketch). We assume μ is injective, as otherwise the result follows from Lemma 28. By Proposition 37, μ can be decomposed as $\mu = \eta \circ \rho_q \circ E \circ \rho_p \circ E$ with $\eta = (u, v)$ and $u, v \in Q(\Sigma^*)$. By Lemma 43, both $MS_{\rho_p}(n) \leq 3$ and $MS_{\rho_q}(n) \leq 3$, hence $MS_\mu(n)$ is bounded if and only if $MS_\eta(n)$ is bounded. If η is primitivity-preserving, then by Lemma 35 we are done. Hence, we are left to show the case when η is not primitivity-preserving and both images are primitive. We give a sketch for this case.

Let $\mu = (u, v)$ be a non-primitivity-preserving injective morphism with $u, v \in Q(\Sigma^*)$. By Lemma 10, there exists a primitive word x with $|x| > 1$, such that $P^\mu = \mathcal{R}(x)$ and $\mu(x) = z^t$ with $z \in Q(\Sigma^*)$ and $t > 1$.

As a consequence of Lemma 23, there exists an integer $k > 0$, which depends only on μ, such that every rotation with a k-length prefix $y \notin \Gamma^k \cap \widetilde{\mathcal{F}}(\{z\}^*)$ contains a synchronization pair. Hence, we can partition these rotations according to their length-k prefix, and the characters preceding these rotations can be determined.

The remaining rotations starting with a power of some rotation of z are handled in a similar (though more complicated) fashion with respect to how rotations starting with a power of b were handled in Lemma 43. This yields an upper-bound for MS_μ depending on the value $|z|$ instead of 3. ◀

7 Conclusions and future work

In this paper, we have provided a complete characterization of binary injective morphisms that preserve the number of BWT-runs up to a bounded additive increase. We have shown that this class coincides with the class of binary primitivity-preserving morphisms.

Primitivity-preserving morphisms could be considered a general effective tool for studying and evaluating repetitiveness measures, since such measures remain invariant, up to small constants, when applied to powers of a word. This suggests that such morphisms could be seen as a unifying framework for the analysis and comparison of different repetitiveness measures.

It would be interesting to explore the design of compression and indexing techniques based on BWT-runs that operate directly on morphic encodings of highly repetitive text collections. This could have applications, for example, in the domain of privacy-preserving algorithms. Although our current approach allows for polynomial-time decision procedures for testing whether a given binary morphism is BWT-run preserving or, equivalently, primitivity-preserving, more efficient algorithms could yield significant improvements in terms of scalability and practical performance.

Furthermore, BWT-run sensitivity could support a new classification of morphisms, providing new insights for their structural behavior and the impact on repetitiveness measures.

Finally, we plan to investigate how to extend our results to morphisms over larger alphabets.

References

1 Tooru Akagi, Mitsuru Funakoshi, and Shunsuke Inenaga. Sensitivity of string compressors and repetitiveness measures. *Information and Computation*, 291:104999, 2023. doi:10.1016/J.IC.2022.104999.

2 Evelyne Barbin-Le Rest and Michel Le Rest. Sur la combinatoire des codes à deux mots. *Theoretical Computer Science*, 41:61–80, 1985. doi:10.1016/0304-3975(85)90060-X.

3 Marie-Pierre Béal, Dominique Perrin, and Antonio Restivo. Unambiguously coded shifts. *European Journal of Combinatorics*, 119:103812, 2024. doi:10.1016/J.EJC.2023.103812.

4 Jean Berstel, Dominique Perrin, and Christophe Reutenauer. *Codes and Automata*, volume 129 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 2010.

5 Jean Berstel and Patrice Séébold. A Characterization of Sturmian Morphisms. In *MFCS*, volume 711 of *Lecture Notes in Computer Science*, pages 281–290. Springer, 1993. doi:10.1007/3-540-57182-5_20.

6 Michael Burrows and David Wheeler. A block sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, 1994.

7 Julien Cassaigne. An algorithm to test if a given circular HD0L-language avoids a pattern. In *IFIP Congress (1)*, volume A-51 of *IFIP Transactions*, pages 459–464. North-Holland, 1994.

8 Julien Cassaigne, France Gheeraert, Antonio Restivo, Giuseppe Romana, Marinella Sciortino, and Manon Stipulanti. New string attractor-based complexities for infinite words. *Journal of Combinatorial Theory, Series A*, 208:105936, 2024. doi:10.1016/J.JCTA.2024.105936.

9 Kuo Tsai Chen, Ralph H. Fox, and Roger C. Lyndon. Free differential calculus, IV. The quotient groups of the lower central series. *Annals of Mathematics*, 68(1):81–95, 1958.

10 Wai-Fong Chuan. Sturmian morphisms and alpha-words. *Theoretical Computer Science*, 225(1-2):129–148, 1999. doi:10.1016/S0304-3975(97)00239-9.

11 Pál Dömösi, Sándor Horváth, Masami Ito, László Kászonyi, and Masashi Katsura. Formal languages consisting of primitive words. In Zoltán Ésik, editor, *Fundamentals of Computation Theory*, pages 194–203, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg. doi:10.1007/3-540-57163-9_15.

12 Gabriele Fici, Giuseppe Romana, Marinella Sciortino, and Cristian Urbina. On the Impact of Morphisms on BWT-Runs. In *CPM*, volume 259 of *LIPIcs*, pages 10:1–10:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.CPM.2023.10.

13 Travis Gagie, Gonzalo Navarro, and Nicola Prezza. Fully Functional Suffix Trees and Optimal Text Searching in BWT-Runs Bounded Space. *Journal of the ACM*, 67(1):2:1–2:54, 2020. doi:10.1145/3375890.

14 Sara Giuliani, Shunsuke Inenaga, Zsuzsanna Lipták, Giuseppe Romana, Marinella Sciortino, and Cristian Urbina. Bit catastrophes for the Burrows-Wheeler transform. *Theory of Computing Systems*, 69(2):19, 2025. doi:10.1007/S00224-024-10212-9.

15 Štěpán Holub, Martin Raska, and Stepán Starosta. Binary codes that do not preserve primitivity. *Journal of Automated Reasoning*, 67(3):25, 2023. doi:10.1007/S10817-023-09674-2.

16 Cheng-Chi Huang. A note on pure codes. *Acta Informatica*, 47(5-6):347–357, 2010. doi:10.1007/S00236-010-0122-7.

17 Dominik Kempa and Nicola Prezza. At the roots of dictionary compression: string attractors. In *STOC*, pages 827–840. ACM, 2018. doi:10.1145/3188745.3188814.

18 Karel Klouda and Stepán Starosta. Characterization of circular D0L-systems. *Theor. Comput. Sci.*, 790:131–137, 2019. doi:10.1016/J.TCS.2019.04.021.

19 Ben Langmead and Steven L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–359, 2012.

20 Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25, 2009.

21 Heng Li and Richard Durbin. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics*, 26(5):589–595, 2010. doi:10.1093/BIOINFORMATICS/BTP698.

22 M. Lothaire. *Combinatorics on Words*. Cambridge University Press, 1997.

23 Roger C. Lyndon and Marcel-Paul Schützenberger. The equation $a^m = b^n c^p$ in a free group. *Michigan Mathematical Journal*, 9(4):289–298, 1962.

24 Sabrina Mantaci, Antonio Restivo, and Marinella Sciortino. Burrows–Wheeler transform and Sturmian words. *Information Processing Letters*, 86(5):241–246, 2003. doi:10.1016/S0020-0190(02)00512-4.

25 Victor Mitrana. Primitive morphisms. *Information Processing Letters*, 64(6):277–281, 1997. doi:10.1016/S0020-0190(97)00178-6.

26 Gonzalo Navarro. Indexing highly repetitive string collections, part I: Repetitiveness measures. *ACM Computing Surveys*, 54(2):article 29, 2021.

27 Gonzalo Navarro. The compression power of the BWT: technical perspective. *Communications of the ACM*, 65(6):90, 2022. doi:10.1145/3531443.

28 Gonzalo Navarro and Cristian Urbina. Repetitiveness measures based on string morphisms. *Theoretical Computer Science*, 1043:115259, 2025. doi:10.1016/J.TCS.2025.115259.

29 Geneviève Paquin. On a generalization of Christoffel words: epichristoffel words. *Theoretical Computer Science*, 410(38-40):3782–3791, 2009. doi:10.1016/J.TCS.2009.05.014.

30 Antonio Restivo. On a Question of McNaughton and Papert. *Information and Control*, 25(1):93–101, 1974. doi:10.1016/S0019-9958(74)90821-3.

31 Michel Rigo. *Formal Languages, Automata and Numeration Systems 1: Introduction to Combinatorics on Words*. Wiley, 2014.

32 Huei-Jan Shyr and Gabriel Thierrin. Codes, languages and MOL schemes. *RAIRO Theoretical Informatics and Applications*, 11(4):293–301, 1977. doi:10.1051/ITA/1977110402931.

33 Huei-Jan Shyr and Shyr-Shen Yu. Non-primitive words in the language p^+q^+. *Soochow Journal of Mathematics*, 20:535–546, 1994.

34 Md. Vasimuddin, Sanchit Misra, Heng Li, and Srinivas Aluru. Efficient architecture-aware acceleration of BWA-MEM for multicore systems. In *2019 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2019, Rio de Janeiro, Brazil, May 20-24, 2019*, pages 314–324, Los Alamitos, CA, 2019. IEEE Computer Society. doi:10.1109/IPDPS.2019.00041.