On the Impact of Morphisms on BWT-Runs

Gabriele Fici
Department of Mathematics and Informatics, University of Palermo, Italy

Giuseppe Romana
Department of Mathematics and Informatics, University of Palermo, Italy

Marinella Sciortino
Department of Mathematics and Informatics, University of Palermo, Italy

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

Abstract

Morphisms are widely studied combinatorial objects that can be used for generating infinite families of words. In the context of Information theory, injective morphisms are called (variable length) codes. In Data compression, the morphisms, combined with parsing techniques, have been recently used to define new mechanisms to generate repetitive words. Here, we show that the repetitiveness induced by applying a morphism to a word can be captured by a compression scheme based on the Burrows–Wheeler Transform (BWT). In fact, we prove that, differently from other compression-based repetitiveness measures, the measure r_{bwt} (which counts the number of equal-letter runs produced by applying BWT to a word) strongly depends on the applied morphism. More in detail, we characterize the binary morphisms that preserve the value of $r_{bwt}(w)$, when applied to any binary word w containing both letters. They are precisely the Sturmian morphisms, which are well-known objects in Combinatorics on words. Moreover, we prove that it is always possible to find a binary morphism that, when applied to any binary word containing both letters, increases the number of BWT-equal letter runs by a given (even) number. In addition, we derive a method for constructing arbitrarily large families of binary words on which BWT produces a given (even) number of new equal-letter runs. Such results are obtained by using a new class of morphisms that we call Thue–Morse-like. Finally, we show that there exist binary morphisms μ for which it is possible to find words w such that the difference $r_{bwt}(\mu(w)) - r_{bwt}(w)$ is arbitrarily large.

2012 ACM Subject Classification Mathematics of computing → Combinatorics on words; Theory of computation → Data compression

Keywords and phrases Morphism, Burrows–Wheeler transform, Sturmian word, Sturmian morphism, Thue–Morse morphism, Repetitiveness measure

Digital Object Identifier 10.4230/LIPIcs.CPM.2023.10

Funding *Gabriele Fici*: Partly supported by MIUR project PRIN 2017 ADASCOML – 2017K7XPAN.
Giuseppe Romana: Partly supported by MIUR project PRIN 2017 ADASCOML – 2017K7XPAN.
Marinella Sciortino: Partly supported by the INdAM-GNCS Project (CUP E55F22000270001) and the PNRR project ITSERR (CUP B53C22001770006).
Cristian Urbina: Partly supported by ANID national doctoral scholarship – 21210580.

Acknowledgements This research was carried out during the visit of Cristian Urbina to the University of Palermo, Italy.

1 Introduction

The Burrows–Wheeler transform (BWT) is a reversible permutation of words, introduced in the Data compression field [5]. Such a transformation allows one to *boost* the effect of the run-length encoding with respect to the original word in input [27]. Due to its

myriad virtues, some of the well-known compressed text-indexes for pattern matching [11, 13] and the most used alignment tools in Bioinformatics [23, 21] are based on the BWT. The performance of the BWT is related to the repetitions of factors in the word, which is why the number of equal-letter runs of the BWT, denoted by r_{bwt}, is considered as a measure of repetitiveness [29]. Much attention has recently paid to the measure r_{bwt} both for its crucial role in designing compressed indexing data structures for highly repetitive texts [13, 16, 30] and for its combinatorial properties [25, 14].

In Combinatorics on words, morphisms are a fundamental tool for generating repetitive sequences, with multiple applications. For instance, injective morphisms, known as codes, are widely used in the fields of Information theory, Data compression, and Cryptography [2]. Recently, morphisms have been used in conjunction with copy-paste mechanisms to define novel compressors and repetitiveness measures, called NU-systems [32]. Informally speaking, a morphism is a mechanism that transforms each letter in a given input word into a corresponding image word, thus producing an output that is likely to contain longer repeated factors. The relationship between morphisms and the measure r_{bwt} has been studied in the context of a subclass of infinite words generated by morphisms, i.e., the purely morphic words [4, 12].

Here, we focus the impact of morphism application on the number of BWT equal-letter runs of finite words.

In Section 3, we prove that a binary morphism is cyclic (i.e., the images of both letters are powers of the same word) if and only if the image of every word under this morphism has the same number of BWT equal-letter runs, regardless of the input word. We also prove other results relating morphisms and words sharing the same Parikh vector (i.e., having the same number of occurrences of each letter), which can be of independent interest.

Then, in Section 4 we find a novel characterization of Sturmian morphisms [3, 28] in terms of BWT equal-letter runs: they are exactly the binary morphisms that preserve the number of BWT equal-letter runs of every binary word containing both letters of the alphabet. This characterization is interesting from a combinatorial point of view, because Sturmian morphisms are a widely studied subject [3, 28]. It also builds another bridge between Combinatorics on words and Data compression.

Further, in Section 5 we show a wide class of morphisms, which we call Thue–Morse-like morphisms, that increase the number of BWT equal-letter runs by 2 on every binary word containing both letters of the alphabet. Moreover, for each even number $2k$, we can find a wide class of binary morphisms, obtained by composing Sturmian and Thue–Morse-like morphisms, that increase the BWT equal-letter runs of every binary words by exactly $2k$. Note that this is exhaustive for the binary alphabet. In fact, unless considering powers of a single letter, every binary word has an even number of BWT equal-letter runs. In addition, we can use the aforementioned morphisms to construct arbitrarily large families of binary words having all the same number of BWT equal-letter runs, for every fixed (even) number, and converging to an infinite aperiodic word.

At the other end of the spectrum, in Section 6 we show that there are binary morphisms (in particular, the so-called period-doubling morphism) that can highly increase the number of BWT equal-letter runs of binary words. We show that the increase in the number of BWT equal-letter runs can be $\Omega(\sqrt{n})$, where n is the length of the original word. In Section 7, we show that this degree of increase cannot occur in other relevant reachable repetitiveness measures, like the size of the Lempel–Ziv parsing [9, 22], or the size g of the smallest deterministic context-free grammar generating the word [18].

We conclude in Section 8 with some final remarks, and some open questions and conjectures.

2 Preliminaries

Basic terminology

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* $w = w[1]w[2]\cdots w[|w|]$ is any finite sequence of letters where $w[i] \in \Sigma$, for $i \in [1, |w|]$, and $|w|$ is the *length* of the word. We denote by $\mathsf{alph}(w)$ the set of the letters of Σ appearing in w. The *empty word*, denoted by ε, is the unique word of length 0. The set of all finite words (resp. all finite words of positive length) over the alphabet Σ is denoted by Σ^* (resp. Σ^+). If $u = u[1]\cdots u[n]$ and $v = v[1]\cdots v[m]$ are words, the *concatenation* uv of u and v is $uv = u[1]\cdots u[n]v[1]\cdots v[m]$. 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 $w[i, j] = \varepsilon$. 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 $j = n$. We denote by $\Pi_{i=1}^{k} w_i$ the concatenation of the words $w_1, w_2, \ldots, w_k$ in that order. We denote by w^k the concatenation of the word w with itself k times. A *rotation* 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 $1 \le i \le n$, obtained by shifting i letters cyclically. We denote by $\mathcal{R}(w)$ the multiset of all the $|w|$ rotations of w. A factor of any word in $\mathcal{R}(w)$ is called a *circular factor* of w. A word is *primitive* if $w = u^k$ implies $k = 1$, or equivalently, if it cannot be written as uv for some non-empty words u and v such that $uv = vu$. A primitive word of length n has exactly n distinct rotations. If w is a *binary word* over the alphabet $\{a, b\}$, the *complement* of w, i.e., the word obtained by replacing all the a's of w by b's and all the b's by a's, is denoted by $\overline{w}$. If $w = w[1]\cdots w[n]$, the *reverse* of w is the word $w^R = w[n]\cdots w[1]$. Given a word $w \in \Sigma^*$ and $a \in \Sigma$, we denote by $|w|_a$ the number of occurrences of a in w. The *run-length encoding* of a word w, denoted by $\mathtt{rle}(w)$, is a 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 $c_i \neq c_{i+1}$. The length $|\mathtt{rle}(w)|$ is the number of *equal-letter runs* in w. The *Parikh vector* of w, denoted as $P(w)$, is the σ-tuple $(|w|_{a_1}, \ldots, |w|_{a_\sigma})$. Given two words u and v having the same length, the *Hamming distance* between u and v, denoted as $d_H(u, v)$, is the number of positions at which the corresponding letters in u and v are different. An *infinite word* $x = x[1]x[2]x[3]\cdots$ is a non-ending sequence of elements of the alphabet Σ. An infinite word x is *ultimately periodic* if there exist $u \in \Sigma^*$ and $v \in \Sigma^+$ such that $x = uvvv\cdots$; it is called *periodic* when $u = \varepsilon$; *aperiodic* if it is not ultimately periodic. If there is no ambiguity, finite words are simply called words.

Morphisms

Let Σ and Γ be two alphabets. A *morphism* is a map μ 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 be denoted as $\mu \equiv (\mu(a_1), \ldots, \mu(a_\sigma))$. When $\Sigma = \Gamma = \{a, b\}$, μ is called a *binary morphism*. A morphism μ is called *prolongable* on a letter $a \in \Sigma$ if $\mu(a) = au$ for some $u \in \Sigma^+$. If for all $a \in \Sigma$ it holds that $\mu(a) \neq \varepsilon$, then the morphism μ is called *non-erasing*. From now on, we will consider non-erasing morphisms, unless stated explicitly otherwise. If there exists k such that $|\mu(a)| = k$ for every $a \in \Sigma$, then the morphism is called *k-uniform*. A 1-uniform morphism is called a *coding*. Given a morphism μ prolongable on some letter $a \in \Sigma$, the family of words $\{a, \mu(a), \ldots, \mu^i(a), \ldots\}$ are prefixes of a unique infinite word $\mu^\infty(a) = \lim_{i\to\infty} \mu^i(a)$, that is a fixed point of μ. Such an infinite word is then called *purely morphic*. An infinite word is *morphic* if it is obtained by applying a coding to a purely morphic word. 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*. Note that the fixed point of a cyclic morphism is periodic. In the case of a binary morphism, it is known that μ is cyclic if and only if $\mu(ab) = \mu(ba)$.

Sturmian words and Sturmian morphisms

Let $\Sigma = \{a, b\}$. A word $w \in \Sigma^*$ is called *balanced* if the difference of the number of a's (or, equivalently, b's) in every two factors of the same length of w is at most 1. An infinite word x is balanced if every finite factor of x is balanced. A finite word w is *circularly balanced* if each word in $\mathcal{R}(w)$ is balanced.

An infinite word over $\Sigma = \{a, b\}$ is a *Sturmian word* if it has exactly $n+1$ distinct factors of length n for every $n \geq 0$. The theory of Sturmian words is very well studied (see [24] for a reference). For example, the following characterization is well known.

▶ **Theorem 1.** *An infinite word over $\Sigma = \{a, b\}$ is Sturmian if and only if it is balanced and aperiodic.*

A class of Sturmian words, called *characteristic Sturmian words*, can be constructed by using finite words, called *standard Sturmian words*, defined recursively as follows. Given an infinite sequence of integers $(d_0, d_1, d_2, \ldots)$, with $d_0 \geq 0, d_i > 0$ for all $i > 0$, called *directive sequence*, the associated standard Sturmian words are defined by $s_0 = b$, $s_1 = a$, and $s_{i+1} = s_i^{d_{i-1}} s_{i-1}$, for $i \geq 1$. A characteristic Sturmian word is the limit of an infinite sequence of standard Sturmian words, i.e., $s = \lim_{i\to\infty} s_i$. Note that standard Sturmian words are finite words also appearing as extremal case for several algorithms and data structures [19, 7, 26, 37].

A *Sturmian morphism* is a morphism that maps infinite Sturmian words to infinite Sturmian words. Some combinatorial characterizations of Sturmian morphisms have been proved in [3]. In particular, a binary morphism μ is Sturmian if and only if it is acyclic and *balanced* (i.e., it maps balanced words to balanced words). Berstel and Séébold [3] also proved the following characterization:

▶ **Theorem 2.** *An acyclic morphism μ is Sturmian if and only if it is locally Sturmian, that is, there exists a Sturmian word s such that $\mu(s)$ is Sturmian.*

Let us denote the following morphisms:

$$E : \begin{cases} a \mapsto b \\ b \mapsto a \end{cases} \qquad \varphi : \begin{cases} a \mapsto ab \\ b \mapsto a \end{cases} \qquad \tilde{\varphi} : \begin{cases} a \mapsto ba \\ b \mapsto a \end{cases}$$

The morphism φ is called the *Fibonacci morphism*, since its fixed point is the Fibonacci word $abaababaabaababaab\cdots$. The monoid $\{E, \varphi, \tilde{\varphi}\}^*$ generated by E, φ, and $\tilde{\varphi}$, by using the composition operator $\circ$, is known as the *Sturm monoid*. The following theorem, proved in [28], shows the combinatorial structure of Sturmian morphisms.

▶ **Theorem 3.** *A morphism is Sturmian if and only if it belongs to $\{E, \varphi, \tilde{\varphi}\}^*$.*

Burrows–Wheeler transform

The *Burrows–Wheeler transform* (BWT) of a word w, denoted by $\mathsf{bwt}(w)$, is a permutation of w obtained by sorting all its rotations in lexicographical order and then concatenating the last symbol of each rotation. The original word can be recovered if one stores the position where it appears in the list of sorted rotations. If a word is highly repetitive, the number of equal-letter runs of the BWT tends to be small. In fact, Kempa and Kociumaka have shown that r_{bwt} is never too far from the size of the Lempel-Ziv parsing, a widely used repetitiveness measure [16]. Hence applying run-length encoding to the BWT is very effective. Because of this, the value $r_{bwt}(w) = |\mathtt{rle}(\mathsf{bwt}(w))|$ that counts the number of *BWT-runs* of w, i.e., equal-letter runs of $\mathsf{bwt}(w)$, is used as a measure for capturing the repetitiveness of the word

w. To understand the particularities of the BWT of a word w, sometimes it is useful to think about the *BWT-matrix* of the sorted rotations of w. It is not difficult to see that, when w is a word such that $\mathsf{alph}(w) = \{a, b\}$, then $r_{bwt}(w)$ is an even number.

The Burrows–Wheeler transform is strictly related to the notions of balance, Sturmian word and morphism, as shown in the following proposition.

▶ **Proposition 4.** *Let w be a word such that $\mathsf{alph}(w) = \{a, b\}$. Then the following are equivalent:*

1. *w is circularly balanced;*
2. *$w \in \mathcal{R}(s^\ell)$, for some standard Sturmian word s and for some $\ell > 0$;*
3. *$r_{bwt}(w) = 2$;*
4. *$w = (\mu(a))^\ell$ for a Sturmian morphism μ and for some $\ell > 0$.*

Proof. The equivalence of 1, 2 and 3 is in [26, 35]. The equivalence with 4 is in [8] (see also Proposition 10 in [34]). ◀

3 Morphisms and sorted rotations of words

We start by introducing some definitions regarding the rotations of morphic images of words.

▶ **Definition 5.** *Let $\mu : \Sigma^* \mapsto \Gamma^*$ be a morphism. Then, we define the multisets*

$$\mathcal{I}_\mu(w) = \{\mu(w') \mid w' \in \mathcal{R}(w)\}$$
$$\mathcal{S}_\mu(w) = \{v\mu(w')u \mid u, v \in \Gamma^+, uv = \mu(a) \text{ for some } a \in \Sigma, \text{ and } aw' \in \mathcal{R}(w)\}.$$

The multiset $\mathcal{I}_\mu(w)$ corresponds to the rotations of $\mu(w)$ obtained by applying μ to the rotations of w. The multiset $\mathcal{S}_\mu(w)$ corresponds to all the remaining rotations of $\mu(w)$. We refer to the multiset $\mathcal{I}_\mu(w)$ as the *I-rotations* of $\mu(w)$, and to the multiset $\mathcal{S}_\mu(w)$ as the *S-rotations* of $\mu(w)$. These two multisets could have elements that end up being equal, as we show in the following example.

▶ **Example 6.** Let $\mu \equiv (a, bab)$, which is an acyclic binary morphism. Then, ab is primitive but $\mu(ab) = abab$ is not. Moreover, $\mathcal{I}_\mu(w) = \{abab, baba\} = \mathcal{S}_\mu(w)$.

We now prove some combinatorial properties of words having the same Parikh vector. By using such properties, we prove that, in the case of the binary alphabet, the lexicographic order among the rotations of a given word is either preserved or reversed, after a morphism is applied. This is a key point to show that the number of BWT-runs cannot decrease after the application of a binary morphism. This is no longer true for larger alphabets.

The following lemma shows that distinct words having the same Parikh vector must have Hamming distance of at least 2.

▶ **Lemma 7.** *Let $w_1, w_2 \in \Sigma^*$ be such that $w_1 \neq w_2$ and $P(w_1) = P(w_2)$. Then, $d_H(w_1, w_2) \geq 2$.*

Proof. By definition of d_H, we have that $d_H(w_1, w_2) = 0$ if and only if $w_1 = w_2$. So, let us suppose by contradiction that $d_H(w_1, w_2) = 1$. Then, there exist two finite words $u, v \in \Sigma^*$ and two distinct indices $i < j \in [1, \sigma]$ such that $w_1 = ua_iv$ and $w_2 = ua_jv$. It follows that the Parikh vectors of w_1 and w_2 are respectively

$$P(w_1) = (|u|_{a_1} + |v|_{a_1}, \ldots, |u|_{a_i} + |v|_{a_i} + 1, \ldots, |u|_{a_j} + |v|_{a_j}, \ldots, |u|_{a_\sigma} + |u|_{a_\sigma})$$

and

$$P(w_2) = (|u|_{a_1} + |v|_{a_1}, \ldots, |u|_{a_i} + |v|_{a_i}, \ldots, |u|_{a_j} + |v|_{a_j} + 1, \ldots, |u|_{a_\sigma} + |u|_{a_\sigma}).$$

Thus, we obtain that the $P(w_1) \neq P(w_2)$, a contradiction. ◀

Since all the words in the same conjugacy class share the same Parikh vector, we can derive the following

▶ **Corollary 8.** *Let $w \in \Sigma^*$ be a word. Then, for every word $w' \in \mathcal{R}(w)$ such that $w' \neq w$, one has $d_H(w, w') \geq 2$.*

Here, we introduce and study new properties of some classes of morphisms, which are related to the number of BWT-runs.

▶ **Definition 9.** *A morphism μ is* abelian order-preserving *if for every pair of distinct words x and y having the same Parikh vector, it holds that $x < y \iff \mu(x) < \mu(y)$.*

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 $x < y \iff \mu(x) > \mu(y)$.*

In general, a morphism can be neither abelian order-preserving nor abelian order-reversing:

▶ **Example 10.** A cyclic morphism is trivially not abelian order-preserving nor abelian order-reversing. The acyclic morphism $\mu \equiv (b, a, c)$ is also neither of them. This can be verified on the rotations of the word *abc*.

However, all acyclic morphisms with a binary domain are either abelian order-preserving or abelian order-reversing, as we show in the following lemma.

▶ **Lemma 11.** *Let $\mu : \{a, b\}^* \mapsto \Sigma^*$ be an acyclic morphism. Then, μ is either abelian order-preserving or abelian order-reversing.*

Proof. Let $\mu \equiv (\alpha, \beta)$ be an acyclic morphism (i.e., $\alpha\beta \neq \beta\alpha$). For the proof, we assume that $|\alpha| \leq |\beta|$, and the other case is treated symmetrically. Factorize μ as $(\alpha, \beta) = (\alpha, \alpha^k v)$, where $k \geq 0$ is as big as possible. This factorization is unique, and α is not a prefix of v, otherwise, k is not as big as possible. Also, $v \neq \varepsilon$ and $v \neq \alpha$ because the morphism μ is acyclic. Let $x = uaz_1$ and $y = ubz_2$ be two distinct binary words with the same Parikh vector. Note that a b has to appear in z_1, since otherwise x has fewer b's than y. Let $z_1 = a^t b z_1'$ for some $t \geq 0$ and $z_1' \in \{a, b\}^*$. We can write $x = uaa^t b z_1'$. Then, $\mu(x) = \mu(u)\alpha^k \alpha\alpha^t v \mu(z_1')$ and $\mu(y) = \mu(u)\alpha^k v \mu(z_2)$. We proceed by case analysis.

If v is not a prefix of α, then the order between $\mu(x)$ and $\mu(y)$ depends only on the order between α and v. The reason is that $\mu(x)$ and $\mu(y)$ share a common prefix $\mu(u)\alpha^k$, followed by α and v respectively, which differ at some position from left to right. Hence, if $\alpha < v$, we obtain $x < y \iff \mu(x) < \mu(y)$; if $v < \alpha$, then we obtain $x < y \iff \mu(x) > \mu(y)$.

If v is a proper prefix of α and $k > 0$, rewrite $\mu(y) = \mu(u)\alpha^k v\alpha z_2'$. We can do this because y has to have at least one letter after ub and both images α and β start with α (in the case of β because $k > 0$). We note that the common prefix $\mu(u)\alpha^k$ is followed by αv in $\mu(x)$ (αv is a prefix of $\alpha\alpha$), and by $v\alpha$ in the case of $\mu(y)$. The order between $\mu(x)$ and $\mu(y)$ is then completely determined by the order between αv and $v\alpha$. This happens because αv and $v\alpha$ are words of the same length which must be distinct, as implied by the inequality $\alpha\beta = \alpha\alpha^k v \neq \beta\alpha = \alpha^k v\alpha$. Hence, if $\alpha v < v\alpha$, we obtain $x < y \iff \mu(x) < \mu(y)$; if $v\alpha < \alpha v$, then we obtain $x < y \iff \mu(x) > \mu(y)$.

No other case is possible. By construction, α is not a prefix of v. Also, $\alpha \neq v$, so if v is a prefix of α, it has to be a proper prefix. If this is the case, as $|\alpha| \leq |\alpha^k v|$ and $|v| < |\alpha|$, k has to be at least 1. ◀

Using Lemma 11 we can easily derive the following corollary.

▶ **Corollary 12.** *Let w be a binary word and let μ be an acyclic morphism. Then, for all pairs of rotations u, v of w, either $u < v \iff \mu(u) < \mu(v)$ (when μ is abelian order-preserving), or $u < v \iff \mu(u) > \mu(v)$ (when μ is abelian order-reversing).*

We introduce new measures to study how the action of a morphism affects the BWT-runs.

▶ **Definition 13.** *Let μ be a morphism and w a word. We define*

$$\Delta^+_\mu(w) = r_{bwt}(\mu(w)) - r_{bwt}(w)$$

and

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

Acyclic binary morphisms cannot decrease the number of BWT-runs of any word.

▶ **Theorem 14.** *Let $\mu : \{a, b\}^* \mapsto \Sigma^*$ be an acyclic morphism. Then $\Delta^+_\mu(w) \geq 0$ for every $w \in \{a, b\}^*$.*

Proof. Let $\mu \equiv (\alpha, \beta)$. Since $r_{bwt}(w) = r_{bwt}(w^m)$ for every $w \in \Sigma^*$ and $m > 1$, let us assume that w is primitive. For the proof, we assume that $|\alpha| \geq |\beta|$, and the other case is treated symmetrically. First, let us consider the case where β is not a suffix of α. Let moreover $x \in \Sigma^*$ be the longest common suffix between α and β. It follows that there exist $\alpha', \beta' \in \Sigma^+$ such that $\alpha = \alpha' x$ and $\beta = \beta' x$, and that the last symbol of α' is different from the last of β' (otherwise x would be longer). Let $\mathcal{R}_x(\mu(w))$ denote the multiset of rotations of $\mu(w)$ with x as a prefix. Note that if $x = \varepsilon$, then $\mathcal{R}_x(\mu(w)) = \mathcal{I}_\mu(w)$. Since x appears in both α and β, it follows that $|\mathcal{R}_x(\mu(w))| \geq |w|$. Specifically, for each $i \in [1, |w|]$, there exists $t_i \in \mathcal{R}_x(\mu(w))$ such that $t_i = x\mu(w[i+1, |w|] \cdot w[1, i-1])v$, where v is either α' or β', depending on whether $w[i]$ is a or b respectively. The lexicographical order of these $|w|$ rotations of $\mu(w)$ with the same prefix correspond to the lexicographical order of the rotations in $\mathcal{I}_\mu(w)$, since by Corollary 8 the words $\bigcup_{i=1}^{|w|}\{\mu(w[i+1, |w|] \cdot w[1, i-1])\}$ must differ in at least one position. By Corollary 12 this is either in the same or in the reverse order with respect to the sorting of the rotations of w. Thus, there exists an injective coding $\lambda : \{a, b\}^* \mapsto \Sigma'^* \subseteq \Sigma$ such that either $\lambda(\mathsf{bwt}(w))$ or $\lambda(\mathsf{bwt}(w)^R)$ is a subsequence of $\mathsf{bwt}(\mu(w))$, and therefore $r_{bwt}(\mu(w)) \geq r_{bwt}(w)$.

Let us now consider the case where β is suffix of α. Then, there exists a primitive word $u \in \Sigma^+$ and two integers $p \geq q \geq 1$ such that $\beta = u^q$, and $\alpha = \alpha' u^p$, with $\alpha' \in \Sigma^+$ that does not have u as suffix. Note that $\alpha' \neq \varepsilon$, otherwise we would have $\alpha\beta = u^p u^q = u^q u^p = \beta\alpha$, i.e. μ would not be acyclic. Let x be the longest common suffix between α' and u. If $x \neq \alpha'$, from analogous arguments to the case where β is not a suffix of α, we have at least $r_{bwt}(w)$ equal-letter runs in $\mathcal{R}_{xu^p}(\mu(w))$. Otherwise, if $x = \alpha'$, let us consider the word $y \in \Sigma^+$ such that $u = yx$. We can then consider the longest common suffix x' between xy and yx, which must be a proper suffix (otherwise u would not be primitive), and apply the same reasoning over the set $\mathcal{R}_{x'xu^p}(\mu(w))$ and the thesis follows. ◀

The following example shows that Theorem 14 does not hold in the case of larger alphabets.

▶ **Example 15.** Consider the acyclic morphism $\mu \equiv (b, a, c)$. Then, $\mathsf{bwt}(bcba) = bcab$ and $\mathsf{bwt}(\mu(bcba)) = \mathsf{bwt}(acab) = cbaa$.

An immediate consequence of Theorem 14 is the following.

▶ **Corollary 16.** *Let $\mu : \{a,b\}^* \mapsto \Sigma^*$ be an acyclic morphism. Then, $\Delta^{\times}_{\mu}(w) \geq 1$, for every $w \in \{a,b\}^*$.*

The following theorem provides a characterization of cyclic morphisms in terms of the number of BWT-runs.

▶ **Theorem 17.** *A morphism $\mu : \{a,b\}^* \mapsto \Sigma^*$ is cyclic if and only if there exists $k > 0$ such that $r_{bwt}(\mu(w)) = k$ for all $w \in \{a,b\}^*$.*

Proof. If $\mu \equiv (\alpha, \beta)$ is cyclic then there exists a primitive word $u \in \Sigma^*$ such that $\alpha = u^p$ and $\beta = u^q$, for some $p, q \geq 0$. Therefore, for each word $w \in \{a,b\}^*$, we have $r_{bwt}(\mu(w)) = r_{bwt}(u^{p\cdot|w|_a+q\cdot|w|_b}) = r_{bwt}(u)$. The other implication is a consequence of Theorem 14. In fact, by contraposition for each $k > 0$ we can find a word w such that $r_{bwt}(w) > k$ (for instance, the i-th Thue–Morse finite word such that $i > \frac{k}{2}$ [4]), which leads to $r_{bwt}(\mu(w)) \geq r_{bwt}(w) > k$ as well. ◀

4 Binary morphisms preserving r_{bwt}

This section is devoted to characterizing binary morphisms such that the number of BWT equal-letter runs is preserved after the action of the morphism on any binary word. First, we show with an example that this property is not trivial.

▶ **Example 18.** Let $\theta \equiv (ab, aa)$ be the period-doubling morphism. It can be verified that $\Delta^{+}_{\theta}(ab) = 0$, $\Delta^{+}_{\theta}(aab) = 2$, and $\Delta^{+}_{\theta}(aaabbaabab) = 4$.

Next, we show that every Sturmian morphism fixes the number of BWT-runs. From the definition of E, φ, and $\tilde{\varphi}$, and by Lemma 11, we derive the following.

▶ **Lemma 19.** *Let $w \in \{a,b\}^*$ be a binary word. Then, for all pairs of rotations u and v of w, and for each $\chi \in \{E, \varphi, \tilde{\varphi}\}$, it holds that $u < v$ if and only if $\chi(u) > \chi(v)$.*

We prove that the number of BWT-runs is preserved by the morphisms that are the generators of the Sturmian morphisms. Note that from the following lemma a method can be derived to construct $\mathsf{bwt}(\mu(w))$ starting from $\mathsf{bwt}(w)$, for every Sturmian morphism μ and every binary word w.

▶ **Lemma 20.** *Let $w \in \{a,b\}^*$ be a binary word with $|\mathsf{alph}(w)| = 2$. Then, for all $\chi \in \{E, \varphi, \tilde{\varphi}\}$, one has $r_{bwt}(w) = r_{bwt}(\chi(w))$. More in detail, one has $\mathsf{bwt}(E(w)) = \overline{\mathsf{bwt}(w)^R}$ and $\mathsf{bwt}(\varphi(w)) = \mathsf{bwt}(\tilde{\varphi}(w)) = \overline{\mathsf{bwt}(w)^R} \cdot a^{|w|_a}$.*

Proof. Since for each word w and each integer $k > 0$ we have $r_{bwt}(w) = r_{bwt}(w^k)$, let us assume that w is a primitive word. From Lemma 19, the case $\chi = E$ is trivial: in fact, from it follows that $\mathsf{bwt}(E(w)) = \overline{\mathsf{bwt}(w)^R}$, and therefore $r_{bwt}(w) = r_{bwt}(E(w))$.

For the case $\chi = \varphi$ one can observe that every b that occurs in $\varphi(w)$ is obtained from $\varphi(a)$, and therefore it is always preceded by an a. Thus, the rotations of $\varphi(w)$ left to cover are all those starting with an a, which therefore must also start with either $\varphi(a)$ or $\varphi(b)$. By Lemma 19, and by observing that $\varphi(a)$ ends with a b and $\varphi(b)$ ends with an a, we have that $\mathsf{bwt}(\varphi(w)) = \overline{\mathsf{bwt}(w)^R} \cdot a^{|w|_a}$. Thus, we need to check if the run of a's at the end merges with the last symbol of $\overline{\mathsf{bwt}(w)^R}$. This is equivalent to checking that the first symbol of $\mathsf{bwt}(w)$ is a b, and by contradiction if the first rotation in lexicographical order is ua for some $u \in \{a,b\}^{n-1}$, then au is a conjugate of w and $au < ua$ for each binary word w, a contradiction.

For the case $\chi = \tilde{\varphi}$, one can see for any binary word $w = w_1 w_2 \cdots w_n$ we have that $\varphi(w) = \varphi(w_1 w_2 \cdots w_n) = a v_1 a v_2 \cdots a v_n$, where for each $i \in [1, n]$ we have $v_i = b$ if $w_i = a$, or $v_i = \varepsilon$ if $w_i = b$. On the other hand, for the same word w we have $\tilde{\varphi}(w) = \tilde{\varphi}(w_1 w_2 \cdots w_n) = v_1 a v_2 \cdots a v_n a$, where analogously to the previous case $v_i = b$ if $w_i = a$, or $v_i = \varepsilon$ if $w_i = b$. One can notice that $\varphi(w)$ and $\tilde{\varphi}(w)$ are conjugate, and the thesis follows. ◀

A graphical interpretation of Lemma 20 is shown in Figure 1.

$M(w)$		$M(\varphi(w))$		$M(\tilde{\varphi}(w))$		
aabba	*b*	**a.a.ab.a.ab.a**	**b.**	*a.*	*a.a.ba.a.ba.*	*b*
abaab	*b*	**a.ab.a.ab.ab.**	**a.**	**a.**	**a.ba.a.ba.b**	**a.**
abbab	*a*	**a.ab.ab.a.a.a**	**b.**	*a.*	*a.ba.ba.a.a.*	*b*
baabb	*a*	**ab.a.a.ab.a.a**	**b.**	*a.*	*ba.a.a.ba.a.*	*b*
babaa	*b*	**ab.a.ab.ab.a.**	**a.**	**a.**	**ba.a.ba.ba.**	**a.**
bbaba	*a*	**ab.ab.a.a.ab.**	**a.**	**a.**	**ba.ba.a.a.b**	**a.**
		b.a.a.ab.a.ab.	*a*	**b**	**a.a.a.ba.a.b**	**a.**
		b.a.ab.ab.a.a.	*a*	**b**	**a.a.ba.ba.a.**	**a.**
		b.ab.a.a.ab.a.	*a*	**b**	**a.ba.a.a.ba.**	**a.**

Figure 1 From left to right, the BWT-matrix for the words $w = abbaba$, $\varphi(w)$, and $\tilde{\varphi}(w)$ respectively. For $M(\varphi(w))$ and $M(\tilde{\varphi}(w))$, we separate with dots the images of symbols from w. The rotations in bold of $M(\varphi(w))$ and $M(\tilde{\varphi}(w))$ correspond to the words in $\mathcal{I}_{\varphi}(w)$ and $\mathcal{I}_{\tilde{\varphi}}(w)$ respectively. The block of rotations in gray at the end of both $M(\varphi(w))$ and $M(\tilde{\varphi}(w))$ are in correspondence of the equal-letter run of a's of length $|w|_a$, which occurs for every $w \in \{a, b\}^*$. One can see that $\mathsf{bwt}(\varphi(w)) = \mathsf{bwt}(\tilde{\varphi}(w)) = \overline{\mathsf{bwt}(w)^R} \cdot a^{|w|_a}$.

The following theorem shows a new characterization of Sturmian morphisms.

▶ **Theorem 21.** *Let μ be a binary morphism. Then, the following are equivalent:*
1. $\Delta^+_\mu(w) = 0$ *for every word w with* $|\mathsf{alph}(w)| = 2$*;*
2. *μ is a Sturmian morphism.*

Proof. By Theorem 3 and Lemma 20, all Sturmian morphisms preserve the number of BWT-runs. Conversely, suppose that μ preserves the number of BWT-runs. By Theorem 17, such a morphism must be acyclic. Let $s = \lim s_i$ be a characteristic Sturmian word. For every i, the word $\mu(s_i)$ has 2 runs in its BWT, hence it is circularly balanced (Proposition 4). Let us consider the word $\mu(s) = \lim \mu(s_i)$. It is balanced and aperiodic, since it is obtained by applying an acyclic morphism to a Sturmian word [6]. Then, $\mu(s)$ is Sturmian by using Theorem 1, whence μ is a Sturmian morphism by applying Theorem 2. ◀

5 Binary morphisms increasing r_{bwt} by a constant

The next step after characterizing Sturmian morphisms as those fixing BWT equal-letter runs on binary words, is to find other binary morphisms that increase the number of BWT-runs always by the same fixed constant. Remind that if such a constant exists, it has to be an even integer because the BWT of any binary word starts with b and ends with a.

We show that for every $k > 0$, we can find a morphism increasing the BWT-runs of any binary word by exactly $2k$. We do so by showing a family of binary morphisms that increase the BWT-runs always by 2, which then we can compose as we want. This family is formed by binary morphisms that are similar to the famous Thue–Morse morphism $\tau \equiv (ab, ba)$. The

structure of the BWT of Thue–Morse words has been studied before and it is well understood [4, 10]. We generalize such results by showing how to derive $\mathsf{bwt}(\mu(w))$ from $\mathsf{bwt}(w)$ for every Thue–Morse-like morphism μ and every binary word w.

▶ **Definition 22.** *A binary morphism is* Thue–Morse-like *if it has the form* $\tau_{p,q} \equiv (ab^p, ba^q)$ *for some* $p, q > 0$.

We prove the following proposition, which is crucial to obtain the main result of this section. Figure 2 highlights the key aspects of the proof.

▶ **Proposition 23.** *For every binary word* w *such that* $\mathsf{alph}(w) = \{a, b\}$*, the I-rotations of* $\tau_{p,q}(w)$ *are contiguous in the BWT-matrix of* $\tau_{p,q}(w)$*, and their last letters spell* $\overline{\mathsf{bwt}(w)}$*.*

Proof. Let w be a binary word of length n such that $\mathsf{alph}(w) = \{a, b\}$. Observe that $\tau_{p,q} \equiv (ab^p, ba^q)$ is abelian order-preserving, so the I-rotations of $\tau_{p,q}(w)$ maintain their relative order. Because $\tau_{p,q}(a)$ ends with b and $\tau_{p,q}(b)$ ends with a, if we consider only the I-rotations of $\tau_{p,q}(w)$ and take the last letter of each, we obtain $\overline{\mathsf{bwt}(w)}$, which starts with a and ends with b. It remains to show that all the I-rotations of $\tau_{p,q}(w)$ are contiguous in its BWT-matrix.

If $p > 1$, each S-rotation starting with a, has to start either with $a^i b$ for some $2 \leq i \leq q+1$, or with aba^q, and both of these prefixes are smaller than ab^p. If $p = 1$, an S-rotation starting with a is smaller than the word $a(ba^q)^{n-1}ba^{q-1}$, which is smaller than a rotation having $(ab)^i ba$ as a prefix for some $0 < i < n$. The I-rotations that start with a have prefixes of such type. In both cases, we obtain that the S-rotations starting with a are smaller than the I-rotations starting with a. A symmetric argument shows that S-rotations starting with b are greater than the I-rotations starting with b. Thus, the I-rotations are contiguous and the thesis holds. ◀

Now we are ready to show that Thue–Morse-like morphisms increase the number of BWT-runs of binary words always by 2.

▶ **Lemma 24.** *For every binary word* w *such that* $\mathsf{alph}(w) = \{a, b\}$*, it holds that*

$$\mathsf{bwt}(\tau_{p,q}(w)) = b^{|w|_b} a^{(q-1)|w|_b} \cdot \overline{\mathsf{bwt}(w)} \cdot b^{(p-1)|w|_a} a^{|w|_a},$$

and that $r_{bwt}(\tau_{p,q}(w)) = r_{bwt}(w) + 2$.

Proof. We show that the block of $\mathsf{bwt}(\tau_{p,q}(w))$ that corresponds to the S-rotations starting with the letter a is equal to $b^{|w|_b} a^{(q-1)|w|_b}$. If $q = 1$, all the S-rotations starting with a end with the letter b. If $q > 1$, the only S-rotations that start with a and end with b have as a prefix either $a^{q+1}b$ or $a^q ba^q$. The smallest S-rotation starting with a and ending with a starts with $a^q b^p ab$ or $a^q b^p ba$. Hence, S-rotations starting with a and ending with b appear before those ending with a.

It follows that the block of $\mathsf{bwt}(\tau_{p,q}(w))$ defined by the S-rotations starting with a spells $b^{|w|_b} a^{(q-1)|w|_b}$, because of their order, and because each of these rotations is in correspondence with some specific a inside $\tau_{p,q}(b)$ for some specific b of w. Only one of these a's per image produces a rotation ending with b, and the other $q-1$ a's yield rotations ending with a.

Showing that the block of $\mathsf{bwt}(\tau_{p,q}(w))$ corresponding to the S-rotations starting with the letter b equals $b^{(p-1)|w|_a} a^{|w|_a}$ is handled symmetrically.

By using Proposition 23, we obtain

$$\mathsf{bwt}(\tau_{p,q}(w)) = b^{|w|_b} a^{(q-1)|w|_b} \cdot \overline{\mathsf{bwt}(w)} \cdot b^{(p-1)|w|_a} a^{|w|_a}.$$

As $\overline{\mathsf{bwt}(w)}$ starts with a and ends with b, we have that $r_{bwt}(\tau_{p,q}(w)) = r_{bwt}(w) + 2$, and the thesis holds. ◀

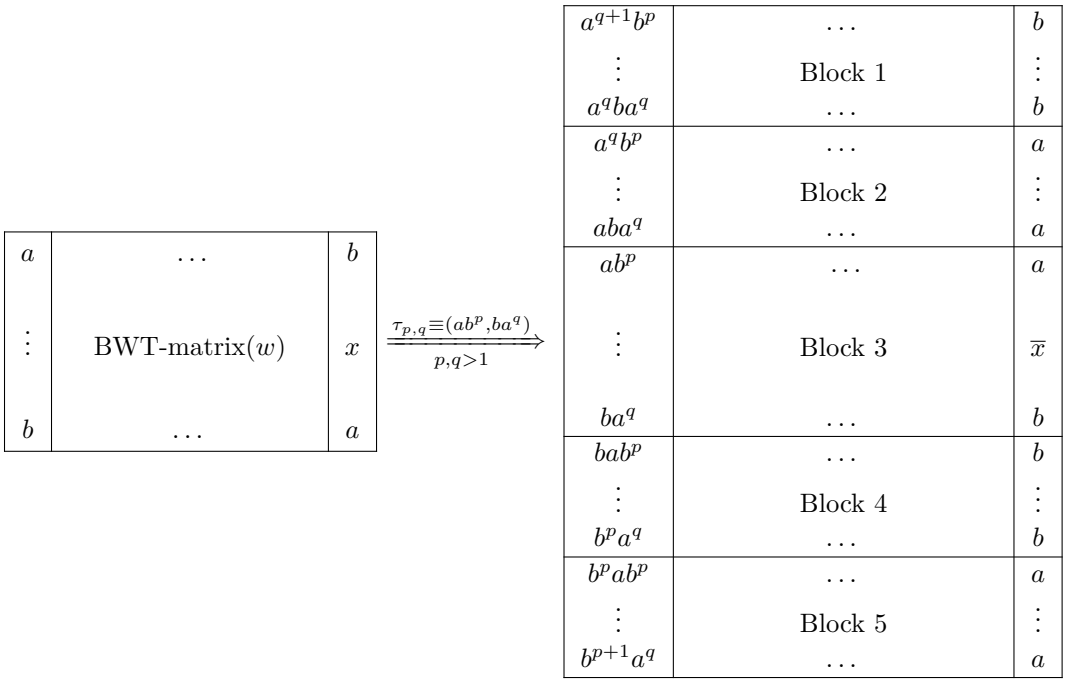

Figure 2 Scheme showing the action of a Thue–Morse-like morphism $\tau_{p,q} \equiv (ab^p, ba^q)$ with $p, q > 1$ on a binary word w with $\mathsf{alph}(w) = \{a, b\}$. At the left is the BWT-matrix of w. At the right is the BWT-matrix of $\tau_{p,q}(w)$. The cases where $p = 1$ or $q = 1$ are similar with Block 2 or Block 4 omitted.

As a consequence of Theorem 21 and Lemma 24, we obtain the following corollary.

▶ **Corollary 25.** *Given a non-negative even integer $2t$, there exists a binary morphism μ such that $\Delta^+_\mu(w) = 2t$ and $\Delta^\times_\mu(w) \leq t + 1$, for every word w with $|\mathsf{alph}(w)| = 2$.*

Proof. We can construct the morphism $\mu \in (\{E, \varphi, \tilde{\varphi}\} \cup \{(ab^p, ba^q) \mid p, q > 0\})^*$ such that μ is obtained by composing, in any order, exactly t morphisms taken in the set $\{(ab^p, ba^q) \mid p, q > 0\}$ and an arbitrary number of Sturmian morphisms. By Theorem 21 and Lemma 24, it holds that $\Delta^+_\mu(w) = 2t$. The value of the function $\Delta^\times_\mu(w) = (r_{bwt}(w)+2t)/r_{bwt}(w) = 1+2t/r_{bwt}(w)$ is maximized when $r_{bwt}(w) = 2$. This maximum is $\Delta^\times_\mu(w) = t + 1$. ◀

We conclude this section by showing a simple algorithm that allows us to construct an arbitrarily large family of words $w_1, w_2, \ldots$ with exactly $2t$ BWT-runs each. In Algorithm 1, a morphism μ such that $\Delta^+_\mu(w) = 2(t-1)$ for every binary word is required. Note that Corollary 25 assures that such a morphism exists.

Moreover, each word w_i is a prefix of the next word w_{i+1}, so that the infinite word $w = \lim_{i\to\infty} w_i$ is well defined, and it is aperiodic. This is given because it holds for the (implicit) standard Sturmian words s_i for $i \in [1, k]$ being used, which are circularly balanced (i.e., $r_{bwt}(w) = 2$ on them, as reported in Proposition 4), and their limit is a characteristic Sturmian word, which is aperiodic.

Algorithm 1 Algorithm for constructing words with $2t$ BWT-runs.

Require: A morphism μ with $\Delta^+_\mu(w) = 2(t-1)$. A sequence of positive integers $d_1, \ldots, d_k$.
Ensure: A sequence of words $w_1, w_2, \ldots, w_k$ where $r_{bwt}(w_i) = 2t$ for any $1 \leq i \leq k$.

$w_{-1} \longleftarrow \mu(b)$
$w_0 \longleftarrow \mu(a)$
for $i \in [1, k]$ **do**
  $w_i \longleftarrow w_{i-1}^{d_i} w_{i-2}$
end for
return $w_1, \ldots, w_k$

6 Morphisms with an unbounded increase on r_{bwt}

There exist morphisms that do not behave as well as Sturmian and Thue–Morse-like morphisms with respect to r_{bwt}. If we consider an alphabet of size greater than 2, we can always find a morphism μ such that the values $\Delta^+_\mu(w)$ and $\Delta^\times_\mu(w)$ are arbitrarily large.

▶ **Lemma 26.** *Let $\Sigma = \{c_1, \ldots, c_k, a, b\}$ with $k \geq 1$. Let $\varphi \equiv (ab, a)$ be the Fibonacci morphism. Then, $r_{bwt}(w) = k + 3$ if w belongs to $\{\varphi^{2i}(a)c_1c_2\cdots c_k \mid i \geq 1\}$.*

Proof. We prove the result by induction on $k \geq 1$. Observe that the words $\varphi^{2i}(a)$ for $i \geq 1$ are Fibonacci words ending with the letter a. It is known that in these words, if we append the letter c_1 smaller than a at the end, then the number of runs becomes 4 [31, Theorem 11]. For the inductive step, suppose that $r_{bwt}(\varphi^{2i}(a)c_1 \ldots c_{k-1}) = k + 2$. When appending c_k at the end, the rotations that do not start with c_k keep their relative order, and the rotation that originally ended with c_{k-1} now ends with c_k. Hence, they define the same number of runs as before. The rotation starting with c_k can be found after the rotation starting with c_{k-1}, which does not end with b, and before the first rotation starting with a, which ends with b. Hence, the number of runs increases by 1. Thus, $r_{bwt}(\varphi^{2i}(a)c_1 \ldots c_k) = k + 3$. ◀

▶ **Lemma 27.** *Let $\Sigma = \{c_1, \ldots, c_k, a, b\}$ with $k \geq 1$. Let $\varphi \equiv (ab, a)$ be the Fibonacci morphism, and $\mu \equiv (c_1, c_2, \ldots, c_k, ab, a)$ be a morphism on the alphabet Σ. Then, $r_{bwt}(w) = \Omega(\log n)$ for every $w \in \{\mu(\varphi^{2i}(a)c_1c_2\cdots c_k) \mid i \geq 1\}$.*

Proof. The morphism μ maps a Fibonacci word ending with a having $c_1 \ldots c_k$ appended at the end, to the next Fibonacci word, which ends with b, having $c_1 \ldots c_k$ appended at the end. For $k = 1$, it is known that the number of runs in this family is $\Omega(\log n)$ [15]. In a similar way to Lemma 26, it is possible to prove by induction that appending c_k at the end of $\mu(\varphi^{2i}(a)c_1c_2c_{k-1})$ adds 2 runs when $k = 2$ and exactly 1 new run when $k > 2$. ◀

▶ **Proposition 28.** *For each alphabet Σ with size greater than 2 there exist a morphism μ, satisfying that for every k, there is a word $w \in \Sigma^*$ such that $\Delta^+_\mu(w) \geq k$ and $\Delta^\times_\mu(w) \geq k$.*

Proof. This is immediate from Lemma 26 together with Lemma 27. ◀

Finding examples like the previous ones for binary morphisms is trickier, but at least in the case of Δ^+_μ, it is possible. An example of a binary morphism for which the value $\Delta^+_\mu(w)$ can be arbitrarily large is the *period-doubling morphism* denoted by θ and defined by the rules $\theta(a) = ab$ and $\theta(b) = aa$.

▶ **Lemma 29.** *Let θ be the period-doubling morphism. For any positive integer k there exist a word w such that $\Delta^+_\theta(w) > k$.*

Proof. W.l.o.g assume that $k > 2$. For $i \in [2, k]$ define the words

$s_i = ab^i a \cdot u_i$ and $e_i = ab^i a \cdot u_i^R$, where $u_i = a^{2k-i}ba^{i-2}$.

We say that s_i is a *starting factor*, and e_i is an *ending factor*. Observe that s_i (resp. u_i) is always smaller than e_i (resp. u_i^R). Moreover, it holds that if $i < j$, then $u_i < u_j < u_j^R < u_i^R$. We define the word $w_k = (\Pi_{i=2}^{k} s_i e_i)a^k$ and show that $\Delta_\theta^+(w_k) = 2k$. Figure 3 shows the structure of both BWTs and highlights the increase in the number of runs.

a	$\dots$	$\dots$	b
$\vdots$	$\vdots$	$\vdots$	x
a	$b^k a u_k^R$	$\dots$	a
ba	$a^{2k}b$	$\dots$	a
$\vdots$	$\vdots$	$\vdots$	y
ba	bb	$\dots$	a
b^2a	u_2	$\dots$	a
b^2a	u_3	$\dots$	ab
$\vdots$	$\vdots$	$\vdots$	$\vdots$
b^2a	u_k	$\dots$	ab^{k-2}
b^2a	u_k^R	$\dots$	ab^{k-2}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
b^2a	u_3^R	$\dots$	ab
b^2a	u_2^R	$\dots$	a
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$b^k a$	u_k	$\dots$	a
$b^k a$	u_k^R	$\dots$	a

$\xrightarrow[\text{reverse order}]{\theta \equiv (ab, aa)}$

b	$\dots$	$\dots$	a
ab	$\dots$	$\dots$	aa
$\vdots$	$\vdots$	$\vdots$	$\overline{x}$
ab	$\theta(b^k a u_k^R)$	$\dots$	ab
$(aa)b$	$\dots$	$\dots$	a
$(aa)ab$	$\theta(a^{2k}b)$	$\dots$	ab
$\vdots$	$\vdots$	$\vdots$	$\overline{y}$
$(aa)ab$	$\theta(bb)$	$\dots$	ab
$(aa)^2b$	$\dots$	$\dots$	a
$(aa)^2ab$	$\theta(u_2)$	$\dots$	ab
$(aa)^2ab$	$\theta(u_3)$	$\dots$	aa
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(aa)^2ab$	$\theta(u_k)$	$\dots$	aa
$(aa)^2ab$	$\theta(u_k^R)$	$\dots$	aa
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(aa)^2ab$	$\theta(u_3^R)$	$\dots$	aa
$(aa)^2ab$	$\theta(u_2^R)$	$\dots$	ab
$(aa)^3b$	$\dots$	$\dots$	a
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(aa)^k b$	$\dots$	$\dots$	a
$(aa)^k ab$	$\theta(u_k)$	$\dots$	ab
$(aa)^k ab$	$\theta(u_k^R)$	$\dots$	ab

Figure 3 To the left is the BWT-matrix of w_k. To the right is the BWT-matrix of $\theta(w_k)$, here displayed in reverse order. Each gray row represents a block of rotations from $\mathcal{S}_\theta(w_k)$ starting with the same prefix, highlighted in the first column. Each one of these block except the first one yields 2 extra runs on $\mathsf{bwt}(\theta(w_k))$. The words x and y correspond to the concatenation of the last letters of blocks of the BWT-matrix of w_k whose form is unknown, but do not play a role in the increase on $r_{bwt}(\theta(w_k))$.

Consider the rotations of w_k starting with $b^i a$ with $1 < i \leq k$. The left shift of the unique rotation starting with the i-th starting factor, and the left shift of the unique rotation starting with the i-th ending factor, are the smallest and greater, respectively. Both of them end with the letter a. The remaining rotations starting with $b^i a$ (if any) have to end with b because in them the prefix $b^i a$ corresponds to a suffix of a longer run of b's followed by an a.

In the case of the rotations of w_k starting with ba, the one starting in the last b of e_k, has $ba^k a^k$ as a prefix, so it is the smallest of them. Also, this rotation is preceded by the factor $ab^k aa^{k-2}$, which ends in a. The greatest rotation starting with ba is the one starting with the b preceding e_2, which is followed by abb and preceded also by an a. In the case of the rotations of w_k starting with a, the smallest of them ends with the letter b as in any binary word. The greatest is the rotation starting with $ab^k a u_k^R$ which is preceded by the letter a.

With the general structure of the BWT of w_k in mind, now we analyse the BWT of $\theta(w_k)$. The morphism θ is order-reversing and all the I-rotations of $\theta(w_k)$ start with the letter a. S-rotations of $\theta(w_k)$ starting with the letter b are always preceded by an a, and it is easy to see that this run of a's merges with the last a in the greatest I-rotation. The S-rotations starting with an a have an even number of a's before the first b appears, and also end with the letter a. This implies that they appear grouped after all the I-rotations of the form $(aa)^iab$ for some $1 \leq i \leq k$, and before all the I-rotations starting with $(aa)^{i-1}ab$. As the smallest and greatest rotations of each of these blocks of I-rotations end with b (because of the action of θ), it follows that the group of S-rotations starting with $(aa)^ib$ increases the number of runs of the BWT of $\theta(w_k)$ by 2 with respect to the BWT of w_k. This happens for $1 \leq i \leq k$, so the overall increase in r_{bwt} after applying the morphism θ is exactly $2k$. ◀

From the lemma above we can deduce that there are binary morphisms that can greatly increase the number of BWT-runs of some words. We define the sensitivity of BWT-runs to morphism application in a similar way to how Akagi et al. define the sensitivity of repetitiveness measures to edit operations [1].

▶ **Definition 30.** *The* BWT additive sensitivity *and* BWT multiplicative sensitivity *for a morphism μ are respectively, the functions*

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

▶ **Proposition 31.** *Let θ be the period-doubling morphism. It holds that $AS_\theta(n) = \Omega(\sqrt{n})$.*

Proof. The length of the words w_k in Lemma 29 is $n = \Theta(k^2)$. We showed that $\Delta^+_\theta(w_k) = 2k = \Theta(\sqrt{n})$ on these words. For values of n in between $|w_k|$ and $|w_{k+1}|$, it is easy to see that for the word $w_k a^j$ for $0 < j < |w_{k+1}| - |w_k|$, it still holds that $\Delta^+_\theta(w_k a^j) = 2k$, as none of the key aspects of the proof of Lemma 29 changes. Thus, the claim is true. ◀

7 On the impact of morphisms on other repetitiveness measures

Morphisms behave very differently when other repetitiveness measures are considered. For a general survey on repetitiveness measures see [29]. For instance, any morphism μ increases the size of the Lempel-Ziv parsing [22] of any word by at most an additive constant depending only on μ. This holds for any alphabet size, as shown by Constantinescu and Ilie [9, Lemma 8].

▶ **Lemma 32.** *Let $\mu : \Sigma^* \to \Gamma^*$ be any morphism. For every word w, it holds that $z(\mu(w)) \leq z(w) + k$ where k is a constant depending only on μ.*

The result of Constantinescu and Ilie can easily be extended to the LZ parsing without overlaps [22], the optimal (not the greedy) LZ-end parsing [20], and bidirectional macro-schemes [38]. We can show a similar result for the measure $g(w)$ defined as the size of the smallest deterministic context-free grammar generating only w [18]. This can be further generalized to the size of the smallest run-length context-free grammar [33], and also to the size of the smallest collage system [17].

▶ **Lemma 33.** *Let $\mu : \Sigma^* \to \Gamma^*$ be any (possible erasing) morphism. For every word w, it holds that $g(\mu(w)) \leq g(w) + k$ where k is a constant depending only on μ.*

Proof. Given a deterministic context-free grammar G of size $|G|$ generating w, we construct a grammar generating $\mu(w)$. For each occurrence of a terminal symbol a in any rule of the grammar, replace it with a new non-terminal A_a. For each terminal symbol add the rule

$A_a \to \mu(a)$. The size of the resulting grammar is $g' \leq |G| + k$ where $k = \sum_{a\in\Sigma} |\mu(a)|$. Let G be the smallest grammar generating w, and then the thesis holds. If the resulting grammar has erasing rules, we can delete them, and replace the occurrences of those erasing variables in other variables by ε. We repeat this recursively. The size of the resulting grammar can only decrease, so the thesis still holds. ◀

If for some fixed measure and morphism, this morphism increases the value of the measure always by at most a fixed constant, then we can derive an easy upper bound for the family of words obtained by iterating that morphism.

▶ **Proposition 34.** *Let ρ be a repetitiveness measure and μ be a morphism. Suppose that for every word w it holds that $\rho(\mu(w)) \leq \rho(w) + k$ for a constant k depending only on ρ and μ. Then, $\rho = O(i)$ in the family $\{\mu^i(w) \mid i \geq 0\}$.*

Proof. Let $k' = \rho(\mu(w))$. We show by induction that $\rho(\mu^i(w)) \leq ki + k'$ for any $i \geq 1$. For $i = 1$, clearly $\rho(\mu(w)) \leq k + k'$. Let $i > 1$ and suppose the claim is true for $i - 1$. Then, $\rho(\mu^i(w)) \leq \rho(\mu^{i-1}(w)) + k \leq (k(i-1) + k') + k \leq ki + k'$. ◀

The families on the proposition above are known as *D0L-sequences* [36]. As a direct consequence of Lemma 33 and Proposition 34, it holds that all repetitiveness measures upper-bounded by g are $O(i)$ on the family of words belonging to a fixed D0L-sequence. In fact, the result we obtain is even more general because we can apply any morphism to words obtained from a D0L-sequence increasing the size of the grammar only by a fixed constant.

▶ **Proposition 35.** *For every (possibly erasing) morphisms μ and λ, and every word w, it holds that $g = O(i)$ in the family $\{\lambda \circ \mu^i(w) \mid i \geq 0\}$.*

Proof. By Lemma 33 and Proposition 34, it holds that $g(\mu^i(w)) = O(i)$ for every (possibly erasing) morphism μ. By Lemma 33, one has $g(\lambda \circ \mu^i(w)) \leq g(\mu^i(w)) + k$ for every (possibly erasing) morphism λ, and a constant k depending on λ. Thus, $g(\lambda \circ \mu^i(w)) = O(i)$. ◀

It is unknown if an analogous result is true for r_{bwt}. In fact, even for the restricted case of purely morphic words, this is known to hold only for the binary case [12].

8 Conclusions and further work

In this work, we have studied the impact of morphism application on the number of BWT equal-letter runs of finite words.

Firstly, we characterized Sturmian morphisms as the binary morphisms preserving the number of BWT-runs for all words w such that $|\mathsf{alph}(w)| = 2$. Besides being interesting on its own, when this characterization is in conjunction with the rest of our results, it allows us to construct binary words with any possible number of BWT-runs, and morphisms with known behavior. This can have practical applications, for instance, in experimentation. In fact, we showed an infinite family of binary morphisms called Thue–Morse-like morphisms, which increase the number of BWT-runs of binary words by 2. As a consequence, we have extended the results of Brlek et al. [4] on the number of BWT-runs of words generated by iterating the composition of the Fibonacci morphism with the Thue–Morse morphism to any composition of Sturmian morphisms and Thue–Morse-like morphisms. Also, we are able to construct infinite sequences of words of increasing length, having all exactly $2k$ BWT-runs, and converging to an aperiodic infinite word at their limit. While the result on Sturmian morphisms is a complete characterization, it is unknown if the compositions of Thue–Morse-like and Sturmian morphisms are the only binary morphisms increasing the number of BWT-runs exactly by 2. The following question is left open.

▶ **Question 36.** *What is a sufficient and necessary condition for a binary morphism μ, to have $\Delta^+_\mu(w) \leq 2k$ (where $k > 0$) for every binary word w?*

We showed that when the alphabet size of the domain is $\sigma > 2$, the values $\Delta^+_\mu(w)$ and $\Delta^\times_\mu(w)$ can be arbitrarily large for some morphisms. In the case of the binary alphabet, we went further and showed that there exists morphisms where $AS_\mu(w) = \Omega(\sqrt{n})$. We plan to extend such a result by studying morphisms where all the images of letters are primitive words. On the other hand, it is unknown if the value $\Delta^\times_\mu(w)$ can be unbounded for some morphism μ with binary domain. We conjecture that this is not the case.

▶ **Conjecture 37.** *For every morphism $\mu : \{a, b\}^* \mapsto \Sigma^*$, we can find a constant k such that $\Delta^\times_\mu(w) \leq k$, for every word $w \in \{a, b\}^*$.*

If Conjecture 37 were true, the following conjecture on images of standard Sturmian words would also be true.

▶ **Conjecture 38.** *For every morphism $\mu : \{a, b\}^* \mapsto \Sigma^*$ and every sequence of standard Sturmian words $(s_i)_{i\in\mathbb{N}}$, it holds that $r_{bwt}(\mu(s_i)) = \Theta(1)$.*

Finally, we showed that the impact of morphism application on BWT-runs, is quite different from the impact of morphisms on other repetitiveness measures based on popular compression schemes, like context-free grammars and LZ factorizations. In these measures, the additive increase after morphism application is bounded by a constant depending only on the morphism and the measure. This raises the following question, which is true in the case of smallest grammars and (some) variants of the LZ parsing, but unknown in the case of BWT equal-letter runs.

▶ **Question 39.** *Does it hold that $r_{bwt}(w) = O(i)$ when $w \in \{\lambda \circ \mu^i(a) \mid i \geq 0\}$, for every pair of morphisms $\mu : \{a, b\}^* \mapsto \{a, b\}^*$ and $\lambda : \{a, b\}^* \to \Sigma^*$?*

We are working on proving or refuting these questions and conjectures. In the future, we plan to study how to extend the results on morphism fixing BWT-runs, and morphisms increasing BWT-runs by a fixed natural number, to alphabets of size greater than 2.

References

1 Tooru Akagi, Mitsuru Funakoshi, and Shunsuke Inenaga. Sensitivity of string compressors and repetitiveness measures. *Inf. Comput.*, 291:104999, 2023.

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

3 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.

4 Srecko Brlek, Andrea Frosini, Ilaria Mancini, Elisa Pergola, and Simone Rinaldi. Burrows-wheeler transform of words defined by morphisms. In *IWOCA*, volume 11638 of *Lecture Notes in Computer Science*, pages 393–404. Springer, 2019.

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

6 Julien Cassaigne. Sequences with grouped factors. In *DLT*, pages 211–222. Aristotle University of Thessaloniki, 1997.

7 Giusi Castiglione, Antonio Restivo, and Marinella Sciortino. Circular Sturmian words and Hopcroft's algorithm. *Theor. Comput. Sci.*, 410(43):4372–4381, 2009.

8 Wai-Fong Chuan. Sturmian morphisms and alpha-words. *Theor. Comput. Sci.*, 225(1-2):129–148, 1999.

9 Sorin Constantinescu and Lucian Ilie. The lempel–ziv complexity of fixed points of morphisms. *SIAM J. Discret. Math.*, 21(2):466–481, 2007.

10 Maxime Crochemore, Thierry Lecroq, and Wojciech Rytter. *125 Problems in Text Algorithms: with Solutions*. Cambridge University Press, 2021.

11 Paolo Ferragina and Giovanni Manzini. Opportunistic Data Structures with Applications. In *FOCS*, pages 390–398. IEEE Computer Society, 2000.

12 Andrea Frosini, Ilaria Mancini, Simone Rinaldi, Giuseppe Romana, and Marinella Sciortino. Logarithmic equal-letter runs for BWT of purely morphic words. In *DLT*, volume 13257 of *Lect. Notes Comput. Sc.*, pages 139–151. Springer, 2022.

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

14 Sara Giuliani, Shunsuke Inenaga, Zsuzsanna Lipták, Nicola Prezza, Marinella Sciortino, and Anna Toffanello. Novel results on the number of runs of the burrows-wheeler-transform. In *SOFSEM*, volume 12607 of *Lecture Notes in Computer Science*, pages 249–262. Springer, 2021.

15 Sara Giuliani, Shunsuke Inenaga, Zsuzsanna Lipták, Giuseppe Romana, Marinella Sciortino, and Cristian Urbina. Bit catastrophes for the Burrows-Wheeler Transform. In *DLT*, volume 13911 of *Lect. Notes Comput. Sc.* Springer, 2023.

16 Dominik Kempa and Tomasz Kociumaka. Resolution of the Burrows-Wheeler Transform Conjecture. *Commun. ACM*, 65(6):91–98, 2022.

17 Takuya Kida, Tetsuya Matsumoto, Yusuke Shibata, Masayuki Takeda, Ayumi Shinohara, and Setsuo Arikawa. Collage system: a unifying framework for compressed pattern matching. *Theor. Comput. Sci.*, 298(1):253–272, 2003.

18 John C. Kieffer and En-Hui Yang. Grammar-based codes: A new class of universal lossless source codes. *IEEE Trans. Inf. Theory*, 46(3):737–754, 2000.

19 Donald E. Knuth, James H. Morris Jr., and Vaughan R. Pratt. Fast pattern matching in strings. *SIAM J. Comput.*, 6(2):323–350, 1977.

20 Sebastian Kreft and Gonzalo Navarro. LZ77-Like Compression with Fast Random Access. In *DCC*, pages 239–248. IEEE, 2010.

21 Ben Langmead and Steven Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature methods*, 9:357–359, March 2012.

22 Abraham Lempel and Jacob Ziv. On the complexity of finite sequences. *IEEE Trans. Inf. Theory*, 22(1):75–81, 1976.

23 Heng Li and Richard Durbin. Fast and accurate long-read alignment with Burrows–Wheeler transform. *Bioinform.*, 26(5):589–595, 2010.

24 M. Lothaire. *Algebraic Combinatorics on Words*. Encyclopedia of Mathematics and its Applications. Cambridge Univ. Press, New York, NY, USA, 2002.

25 Sabrina Mantaci, Antonio Restivo, Giovanna Rosone, Marinella Sciortino, and Luca Versari. Measuring the clustering effect of BWT via RLE. *Theoret. Comput. Sci.*, 698:79–87, 2017.

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

27 Giovanni Manzini. An analysis of the Burrows–Wheeler transform. *J. ACM*, 48(3):407–430, 2001.

28 Filippo Mignosi and Patrice Séébold. Morphismes sturmiens et règles de Rauzy. *Journal de théorie des nombres de Bordeaux*, 5(2):221–233, 1993.

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

30 Gonzalo Navarro. The compression power of the BWT: technical perspective. *Commun. ACM*, 65(6):90, 2022.

31 Gonzalo Navarro, Carlos Ochoa, and Nicola Prezza. On the approximation ratio of ordered parsings. *IEEE Trans. Inf. Theory*, 67(2):1008–1026, 2021.

32 Gonzalo Navarro and Cristian Urbina. On stricter reachable repetitiveness measures. In *SPIRE*, volume 12944 of *Lect. Notes Comput. Sci.*, pages 193–206. Springer, 2021.

33 Takaaki Nishimoto, Tomohiro I, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Fully Dynamic Data Structure for LCE Queries in Compressed Space. In *MFCS*, volume 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 72:1–72:15, 2016.

34 Geneviève Paquin. On a generalization of Christoffel words: epichristoffel words. *Theor. Comput. Sci.*, 410(38-40):3782–3791, 2009. doi:10.1016/j.tcs.2009.05.014.

35 Antonio Restivo and Giovanna Rosone. Balancing and clustering of words in the Burrows-Wheeler transform. *Theor. Comput. Sci.*, 412(27):3019–3032, 2011. doi:10.1016/j.tcs.2010.11.040.

36 Grzegorz Rozenberg and Arto Salomaa. *The mathematical theory of L systems*. Academic press, 1980.

37 Marinella Sciortino and Luca Q. Zamboni. Suffix automata and standard sturmian words. In *DLT*, volume 4588 of *Lect. Notes Comput. Sc.*, pages 382–398. Springer, 2007.

38 James A. Storer and Thomas G. Szymanski. Data compression via textual substitution. *J. ACM, !month = oct*, 29(4):928–951, 1982.