A Separation of γ and b via Thue–Morse Words

Hideo Bannai[1,4][0000−0002−6856−5185], Mitsuru Funakoshi[2,5][0000−0002−2547−1509], Tomohiro I[3][0000−0001−9106−6192], Dominik Koeppl[1][0000−0002−8721−4444], Takuya Mieno[2,5][0000−0003−2922−9434], and Takaaki Nishimoto[4]

[1] M&D Data Science Center, Tokyo Medical and Dental University, Tokyo, Japan
{hdbn,koeppl}.dsc@tmd.ac.jp
[2] Department of Informatics, Kyushu University, Fukuoka, Japan
{mitsuru.funakoshi,takuya.mieno}@inf.kyushu-u.ac.jp
[3] Department of Artificial Intelligence, Kyushu Institute of Technology, Iizuka, Japan
tomohiro@ai.kyutech.ac.jp
[4] RIKEN Center for Advanced Intelligence Project, Tokyo, Japan
takaaki.nishimoto@riken.jp
[5] Japan Society for the Promotion of Science, Tokyo, Japan

Abstract. We prove that for $n \geq 2$, the size $b(t_n)$ of the smallest bidirectional scheme for the nth Thue–Morse word t_n is $n+2$. Since Kutsukake et al. [SPIRE 2020] show that the size $\gamma(t_n)$ of the smallest string attractor for t_n is 4 for $n \geq 4$, this shows for the first time that there is a separation between the size of the smallest string attractor γ and the size of the smallest bidirectional scheme b, i.e., there exist string families such that $\gamma = o(b)$.

1 Introduction

Repetitiveness measures for strings is an important topic in the field of string compression and indexing. Compared to traditional entropy-based measures, measures based on dictionary compression are known to better capture the repetitiveness in highly repetitive string collections [12]. Some well known examples of dictionary-compression-based measures are: the size r of the run-length Burrows–Wheeler transform [2] (RLBWT), the size z of the Lempel-Ziv 77 factorization [17], the size b of the smallest bidirectional (or macro) scheme [15].

Kempa and Prezza introduced the notion of *string attractors* [4], which gave a unifying view of dictionary-compression-based measures. A string attractor of a string is a set of positions such that any substring of the string has at least one occurrence which contains a position in the set. The size γ of the smallest string attractor of a word is a lower bound on the size of all known dictionary compression measures, but is NP-hard to compute. Kociumaka et al. [5,6] introduced another measure $\delta \leq \gamma$ that is computable in linear time, defined as the maximum over all integers k, the number of distinct substrings of length k in the string divided by k.

The landscape of the relations between these measures has been a focus of attention. For example, since z is a special case of a bidirectional scheme, $b \leq z$.

Also, $b \leq 2r$ [13] and $r = O(z \log^2 N)$ [3] hold, where N is the length of the string. Notice that a string can be represented in space (with an extra factor of $\log N$ for bits) proportional to b, r, or z. Interestingly, while δ and γ do not give a direct representation of the string, it is known that the string can be represented in $O(\delta \log \frac{N}{\delta})$ or $O(\gamma \log \frac{N}{\gamma})$ space, respectively [4,5,6]. On the other hand, Kociumaka et al. [5,6] showed that for every length N and integer $\delta \in [2, N]$, there exists a family of length-N strings having the same measure δ, that requires $\Omega(\delta \log \frac{N}{\delta} \log N)$ bits to be encoded. Analogous results for γ are not yet known [5,6,12]. The bidirectional scheme is the most powerful among the dictionary-compression-based measures. The size b of the smallest bidirectional scheme is also known to satisfy $b = O(\gamma \log \frac{N}{\gamma})$, but again, the tightness of this bound was not known [12].

Following Mantaci et al. [8,9], Kutsukake et al. [7] investigated repetitiveness measures on Thue–Morse words [14,16,11] and showed that the size of the smallest string attractor for the n-th Thue–Morse word is 4, for any $n \geq 4$. They also conjectured that the size of the smallest bidirectional scheme for the n-th Thue–Morse word (which has length $N = 2^n$) is $\Theta(\log N)$, which would imply a separation between γ and b. Possibly due to the difficulty (NP-hardness) of computing the size of the smallest bidirectional scheme of a string [15], tight bounds for b have only been discovered for a very limited family of strings, most notably standard Sturmian words [10]. This was shown from the fact that the size r of the RLBWT of every standard Sturmian word is 2, therefore implying a constant upper bound on the smallest bidirectional scheme.

In this paper, we prove Kutsukake et al.'s conjecture by showing that for any $n \geq 2$, the size $b(t_n)$ of the smallest bidirectional scheme for t_n is exactly $n + 2$. For any value of $\gamma \geq 4$, we can construct a family of strings such that $b = \Theta(\gamma \log \frac{N}{\gamma})$ and N is the length of the string. Our result shows for the first time the separation between γ and b, i.e., there are string families such that $\gamma = o(b)$.

2 Preliminaries

We consider the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. A string is an element of Σ^*. For any string $w \in \Sigma^*$, let $|w|$ denote its length, and let $w = w[0] \cdots w[|w| - 1]$. Also, for any $0 \leq i \leq j < |w|$, let $w[i..j] = w[i] \cdots w[j]$.

A *string morphism* μ is a function mapping strings to strings such that each character is replaced by a single string (deterministically), i.e., $\mu(w) = \mu(w[0]) \cdots \mu(w[|w| - 1])$ for any string w. Let $\mu^0(w) = w$, and for any integer $n \geq 1$, let $\mu^n(w) = \mu(\mu^{n-1}(w))$. Now let μ be the morphism on the binary alphabet determined by $\mu(\mathtt{a}) = \mathtt{ab}$ and $\mu(\mathtt{b}) = \mathtt{ba}$. Then the n-th *Thue–Morse word* t_n is $\mu^n(\mathtt{a})$, and its length is $|t_n| = 2^n$.

A list of strings $b_1, \ldots, b_k$ is called a *parsing* of a string S, if $S = b_1 \cdots b_k$. Each b_i $(i = 1, \ldots, k)$ is called a *phrase*. A sequence $B = ((b_1, s_1), \ldots, (b_k, s_k))$ is a *bidirectional scheme* for S, if $b_1, \ldots, b_k$, is a parsing of S and for all $i = 1, \ldots, k$, $s_i \in [0, |S| - 1] \cup \{\perp\}$, such that $s_i = \perp$ if $|b_i| = 1$, and $b_i = S[s_i..s_i + |b_i| - 1]$

otherwise. We denote the *size* k of the bidirectional scheme B by $|B|$. We call s_i the *source* of the phrase b_i.

If $|b_i| = 1$ then we stipulate that $s_i = \bot$, and call b_i a *ground* phrase. (Consequently, there are no phrases of length one that have a source being a text position.) We denote the number of ground phrases in B by $\#_g(B)$. For convenience, we denote the starting position of phrase b_i by p_i, i.e., $p_1 = 0$ and $p_i = |b_1 \cdots b_{i-1}|$ for all $i = 2, \ldots, k+1$, where p_{k+1} is defined for technical reasons.

A bidirectional scheme B for the string S defines a function $f_B : [0, |S|-1] \cup \{\bot\} \to [0, |S|-1] \cup \{\bot\}$ over positions of S, where

$$f_B(x) = \begin{cases} \bot & \text{if } x = \bot \text{ or if } x = p_i, s_i = \bot \text{ for some } i, \\ s_i + x - p_i & \text{otherwise, i.e., if } p_i \le x < p_{i+1}, s_i \neq \bot \text{ for some } i. \end{cases}$$

Let $f_B^0(x) = x$, and for any $j \ge 1$, let $f_B^j(x) = f_B(f_B^{j-1}(x))$. It is clear that if $f_B(i) \neq \bot$, then it holds that $S[i] = S[f_B(i)]$. A bidirectional scheme for S is *valid*, if there is no $i \in [0, |S|-1]$ such that the function f_B contains a cycle, that is, for every $i \in [0, |S|-1]$, there exists a $j \ge 1$ such that $f_B^j(i) = \bot$. A valid bidirectional scheme B of size k for S implies an $O(k)$-word size (compressed) representation of S, namely, the sequence $((|b_1|, s'_1), \ldots, (|b_k|, s'_k)) \subset ([1, |S|] \times ([0, |S|-1] \cup \Sigma))^k$, where $s'_i = b_i$ if $s_i = \bot$, and $s'_i = s_i$ otherwise. Note that the string S can be reconstructed from this sequence if and only if B is valid. A parsing $b_1, \ldots, b_k$ of S is *valid* if there exists a list of phrase sources $s_1, \ldots, s_k$ such that $((b_1, s_1), \ldots, (b_k, s_k))$ is a valid bidirectional scheme for S. See Figure 1 for examples of representations of valid bidirectional schemes of t_3 and t_4.

Informally, $f_B(x)$ gives the position (source) from where we want to copy the character that restores $S[x]$ when reconstructing S from the compressed representation, where $f_B(x) = \bot$ indicates that the character is stored as a ground phrase, i.e., as a literal.

It is easy to see that a valid bidirectional scheme must have at least as many ground phrases as there are different characters appearing in S (the number of ground phrases is at least $|\Sigma|$ if all characters of Σ appear in S).

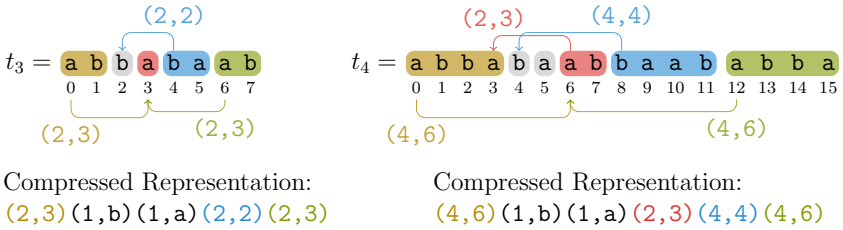


Fig. 1. Examples of valid compressed representations of t_3 and t_4.

3 Important Characteristics of Thue–Morse Words

Before proving our bounds, we first give some simple observations on Thue–Morse words that we will use later. Remember that the first index of t_n is 0, which is an even position.

Lemma 1. $\mathtt{aa}$ *and* $\mathtt{bb}$ *only occur at odd positions in* t_n*.*

Proof. The morphism μ implies that any substring of length 2 starting at an even position is either $\mu(\mathtt{a}) = \mathtt{ab}$ or $\mu(\mathtt{b}) = \mathtt{ba}$. □

Lemma 2 (Theorem 2.2.3 of [1]). t_n *has no overlapping factors, i.e., two occurrences of the same string in* t_n *never share a common position.*

Lemma 3. $\mathtt{abab}$ *and* $\mathtt{baba}$ *only occur at even positions in* t_n*.*

Proof. Suppose to the contrary that there is an occurrence of $\mathtt{abab}$ that starts at an odd position. Then, Lemma 1 implies that $\mathtt{b}$ occurs immediately left of $\mathtt{abab}$, i.e., there is an occurrence of the substring $\mathtt{babab}$, thus contradicting Lemma 2 with the substring $\mathtt{bab}$ having two overlapping occurrences. □

Let the *parity* of an integer i be $i \bmod 2 \in \{0, 1\}$.

Lemma 4. *For any substring* $w \notin \{\mathtt{aba}, \mathtt{bab}, \mathtt{ab}, \mathtt{ba}, \mathtt{a}, \mathtt{b}\}$ *of* t_n*, the parities of all occurrences of* w *in* t_n *are the same.*

Proof. Any such substring w contains at least one of $\{\mathtt{aa}, \mathtt{bb}, \mathtt{abab}, \mathtt{baba}\}$ as a substring, and thus the result follows from Lemmas 1 and 3. □

Further, we use that t_n is a prefix of t_{n+1} and $t_n[0..4] = \mathtt{abbab}$ for $n \geq 3$.

4 Upper and Lower Bounds on b

We start with the upper bound on the smallest size of a (valid) bidirectional parsing by constructing such a parsing, and subsequently show that this bound is optimal by showing a lower bound whose proof is more involved.

4.1 Upper Bound

Theorem 1 (Upper bound). *For* $n \geq 2$*, there exists a valid bidirectional scheme for* t_n *of size* $n + 2$*.*

Proof. Proof by induction. For $n = 2$ it is clear that there is a valid bidirectional scheme of size 4.

Suppose that for some $n \geq 2$, there is a valid bidirectional scheme $B_n = ((b_1, s_1), \ldots, (b_k, s_k))$ of size k for t_n. We can assume that there are at least two ground phrases $b_{i_\mathtt{a}} = t_n[p_{i_\mathtt{a}}] = \mathtt{a}$ and $b_{i_\mathtt{b}} = t_n[p_{i_\mathtt{b}}] = \mathtt{b}$. Since $t_{n+1} = \mu(t_n)$, we first consider a bidirectional scheme B' for t_{n+1} where each phrase is constructed

from phrases of B_n by applying μ, with the small exception for the two ground phrases. More precisely, the phrases of B' are $\mu(b_i)$ for $i \in [1,k] \setminus \{i_{\mathtt{a}}, i_{\mathtt{b}}\}$, and two ground phrases from each of $\mu(b_{i_{\mathtt{a}}}) = \mathtt{ab}$ and $\mu(b_{i_{\mathtt{b}}}) = \mathtt{ba}$, resulting in a parsing of size $k+2$. For each non-ground phrase $\mu(b_i)$ in B', we can either choose the source to be (i) $2p_{i_{\mathtt{a}}}$ or $2p_{i_{\mathtt{b}}}$ if its length is 2, or (ii) $2s_i$ otherwise. The latter is because $\mu(b_i) = \mu(t_n[s_i..s_i+|b_i|-1]) = \mu(t_n)[2s_i..2s_i+2|b_i|-1] = t_{n+1}[2s_i..2s_i+2|b_i|-1]$. The validity of B' follows from the validity of B_n, and $f_{B'}$ has no cycles. It is easy to see that for any position i, the parities of i and $f_{B'}(i)$ are the same (unless $f_{B'}(i) = \bot$). Thus, noticing that $t_{n+1}[3..4] = \mathtt{ab}$, (1) the source of $t_{n+1}[3] = \mathtt{a}$ at an odd position can eventually be traced to the ground phrase at position $2p_{i_{\mathtt{b}}} + 1$, and (2) the source of $t_{n+1}[4] = \mathtt{b}$ at an even position can eventually be traced to the ground phrase at position $2p_{i_{\mathtt{b}}}$.

Next, we modify B' by combining the two consecutive ground phrases $(\mathtt{a}, \bot)$ and $(\mathtt{b}, \bot)$ corresponding to $\mu(b_{i_{\mathtt{a}}})$, and replace them with a single $(\mathtt{ab}, 3)$. This results in a bidirectional scheme B'' of size $k+1$. From the above observations (1) and (2), it is clear that B'' is still valid. Thus, $B_{n+1} = B''$ is a valid bidirectional scheme for t_{n+1} of size $k+1$, thereby proving the theorem. □

The bidirectional scheme of t_4 in Figure 1 can be constructed by following the algorithmic instructions of the proof of Theorem 1.

4.2 Lower Bound

Theorem 2 (Lower Bound). *For $n \geq 2$, the smallest valid bidirectional scheme for t_n has size $n+2$.*

To prove Theorem 2, we would like to, in essence, do the opposite of what we did in the proof of Theorem 1, and show that we can construct a bidirectional scheme for t_{n-1} of size $k-1$, given a bidirectional scheme for t_n of size k. However, the opposite direction involves halving the size of phrases, and thus does not work straightforwardly when there are phrases of odd length. Nevertheless, we will show that this can be done in an amortized way, and show the following.

Lemma 5. *For any $n \geq 5$, if there exists a valid bidirectional scheme of size k for t_n, then, for some $1 \leq i \leq 3$, there exists a valid bidirectional scheme of size at most $k-i$ for t_{n-i}.*

Since the size of the smallest bidirectional scheme for t_2, t_3, t_4 can be confirmed to be respectively $4, 5, 6$ by computer analysis, this with Lemma 5 implies Theorem 2.

In the rest of the section, we give an algorithm that, given a bidirectional scheme B_n for t_n, constructs a bidirectional scheme B_{n-1} for t_{n-1}, and claim that applying the algorithm repeatedly i times, for some $1 \leq i \leq 3$, we obtain a bidirectional scheme B_{n-i} for t_{n-i} such that $|B_{n-i}| \leq |B_n| - i$. The algorithm consists of 3 main steps:

1. Elimination of length-1 ground phrases.

2. Elimination of odd length phrases.
3. Application of the inverse morphism μ^{-1} on all phrases of the modified parsing.

The goal of Steps 1 and 2 is to modify the phrases of B_n to construct a bidirectional scheme B'_n so that all phrases in B'_n will be of even length. When modifying the phrases, we must take care in 1) defining the source of the phrase, and 2) ensuring that no cycles are introduced in the resulting bidirectional scheme B_{n-1}. To make this clear, we temporarily relax the definition for ground phrases in B'_n during the modification, so that the ground phrases of B'_n are phrases of length 2 that start at even positions. In this way, we can be sure that any position in a length-2 phrase starting at an even position in B'_n is not involved in a cycle. In Step 3, we create a new bidirectional scheme B_{n-1} of t_{n-1} by translating all phrase lengths and sources of B'_n according to the inverse morphism μ^{-1}, i.e., we map each non-ground phrase (b'_i, s'_i) of B'_n to the phrase $(\mu^{-1}(b'_i), s'_i/2)$ in t_{n-1}. The length-2 ground phrases in B'_n become length-1 ground phrases in B_{n-1}, and thus we obtain a valid bidirectional scheme B_{n-1} for t_{n-1}, without the relaxation, and the same size as B'_n.

Eliminating Length-1 Ground Phrases The operation is done analogously and symmetrically for any length-1 ground phrase (`a` or `b`) that may occur at an even or odd position. We describe in detail the case for a ground phrase with character `a` that occurs at some odd position $2i+1$.

For a consecutive pair of positions $2i, 2i+1$, we call one a *partner* of the other. Let $i_{\mathtt{b}} = 2i$ be the partner position of the length-1 ground phrase `a`, i.e., $t_n[i_{\mathtt{b}}..i_{\mathtt{b}}+1] = \mathtt{ba}$. The idea is to (re)move the phrase boundary that separates partner positions so that the ground phrase disappears. Since we are considering the case where the ground phrase is at an odd position, we extend the phrase (b_i, s_i) containing position $i_{\mathtt{b}}$ by one character, so that it includes the length-1 ground phrase $t_n[i_{\mathtt{b}}+1] = \mathtt{a}$, thereby eliminating it. If possible, we would like to keep the source of the extended phrase the same, i.e., change (b_i, s_i) to $(b_i\mathtt{a}, s_i)$, or equivalently, change $f_{B_n}(i_{\mathtt{b}}+1) = \perp$ to $f_{B_n}(i_{\mathtt{b}}+1) = s_i + |b_i|$. Note that if the parity of $f_{B_n}(i_{\mathtt{b}})$ is equal to that of $i_{\mathtt{b}}$, this is always possible (i.e., $t_n[f_{B_n}(i_{\mathtt{b}})+1] = \mathtt{a}$ always holds). However, it may be that the position $i_{\mathtt{b}}+1$ gets involved in a cycle, due to this change. Notice that since we started from a valid (relaxed) bidirectional scheme, it is guaranteed that $i_{\mathtt{b}}$ is not involved in a cycle, i.e., $f^j_{B_n}(i_{\mathtt{b}}) \neq i_{\mathtt{b}}$ for any $j \geq 1$. Therefore, we further modify the phrase boundaries, if necessary, to ensure that the source of $t_n[i_{\mathtt{b}}+1] = \mathtt{a}$ will belong in the same phrase as the source of $t_n[i_{\mathtt{b}}] = \mathtt{b}$. This is repeated until we are sure that all these changes made to eliminate the original length-1 ground phrase `a` do not introduce any cycles in the final bidirectional scheme. In other words, we ensure, for some *sufficiently large* j', $f^j_{B_n}(i_{\mathtt{b}}+1) = f^j_{B_n}(i_{\mathtt{b}})+1$ for all $1 \leq j \leq j'$. Then, from the acyclicity of position $i_{\mathtt{b}}$, the acyclicity of position $i_{\mathtt{b}}+1$ follows.

There are six cases where the process terminates, as shown in Figure 2 (Case 3 is further divided into two sub-cases). As noted above, as long as the parity of $f^j_{B_n}(i_{\mathtt{b}})$ is the same as that of $f^{j-1}_{B_n}(i_{\mathtt{b}})$, the character of $f^j_{B_n}(i_{\mathtt{b}})$'s partner

is always a, and we can ensure that $f_{B_n}^{j-1}(i_\mathtt{b})$ and $f_{B_n}^{j-1}(i_\mathtt{b}+1)$ are in the same phrase by only (possibly) setting $f_{B_n}^{j}(i_\mathtt{b}+1) = f_{B_n}^{j}(i_\mathtt{b})+1$. Thus, we consider the cases where $j' \geq 1$ is the smallest integer such that the parities of $f_{B_n}^{j'-1}(i_\mathtt{b})$ and $f_{B_n}^{j'}(i_\mathtt{b})$ differ, in which case, Lemma 4 implies that $f_{B_n}^{j'-1}(i_\mathtt{b})$ is contained in a phrase in $\{\mathtt{aba}, \mathtt{bab}, \mathtt{ab}, \mathtt{ba}, \mathtt{b}\}$. Each of the six cases corresponds to a distinct occurrence of b in the strings of this set. We show that in each case, we can modify the phrases so that both $f_{B_n}^{j'-1}(i_\mathtt{b})$ and $f_{B_n}^{j'-1}(i_\mathtt{b}+1)$ are in the same length-2 phrase, i.e., a relaxed ground phrase, and be sure that $i_\mathtt{b}+1$ will not be involved in a cycle in the final bidirectional scheme. The details of each case are described in Figure 2.

Although Cases 1, 2, 4 introduce a new length-1 ground phrase, the number of phrase boundaries that separate partner positions always decreases at the starting point, and never increases. Therefore the whole process terminates at some point, at which point, all length-1 ground phrases have been eliminated.

Eliminating Odd Length Phrases In this step, we eliminate all remaining phrases with odd lengths. Since there are no more length-1 ground phrases, we first focus on removing phrases aba and bab of length 3. Below, we describe the operation for removing a phrase aba that starts at an odd position. The other cases are analogous or symmetric.

Starting with an occurrence of phrase aba that starts at an odd position $i_\mathtt{b}+1$, we know that this phrase is preceded by b. We move the phrase boundary that separates partner positions, so that the length-3 phrase shrinks to a length-2 phrase starting at an even position, i.e., a relaxed ground phrase, in this case, by expanding the phrase to its left. Since we have changed the source of the a at position $i_\mathtt{b}+1$, we ensure that for some sufficiently large j', $f_{B_n}^{j}(i_\mathtt{b}+1) = f_{B_n}^{j}(i_\mathtt{b})+1$ for all $1 \leq j \leq j'$, as we did for the elimination of length-1 ground phrases, so that $i_\mathtt{b}+1$ is not involved in a cycle.

There are five cases where the process terminates, as shown in Figure 3. As noted previously, as long as the parity of $f_{B_n}^{j}(i_\mathtt{b})$ is the same as that of $f_{B_n}^{j-1}(i_\mathtt{b})$, then the character of $f_{B_n}^{j}(i_\mathtt{b})$'s partner is always a, and we can ensure that $f_{B_n}^{j-1}(i_\mathtt{b})$ and $f_{B_n}^{j-1}(i_\mathtt{b}+1)$ are in the same phrase by only (possibly) setting $f_{B_n}^{j}(i_\mathtt{b}+1) = f_{B_n}^{j}(i_\mathtt{b})+1$. Thus, we consider the cases where $j' \geq 1$ is the smallest integer such that the parities of $f_{B_n}^{j'-1}(i_\mathtt{b})$ and $f_{B_n}^{j'}(i_\mathtt{b})$ differ, in which case, Lemma 4 and the previous step implies that $f_{B_n}^{j'-1}(i_\mathtt{b})$ is contained in a phrase in $\{\mathtt{aba}, \mathtt{bab}, \mathtt{ab}, \mathtt{ba}\}$. Each of the five cases corresponds to a distinct occurrence of b in strings of this set. The details of each case are described in Figure 3.

After eliminating all phrases aba and bab of length 3, all remaining phrases are either of length 2 or do not belong to the set $\{\mathtt{aba}, \mathtt{bab}, \mathtt{ab}, \mathtt{ba}, \mathtt{a}, \mathtt{b}\}$. Therefore, we can move all phrase boundaries that separate partner positions to the right (or all of them to the left) and update the sources accordingly without introducing cycles, since length-2 phrases starting at odd positions become re-

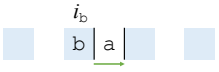
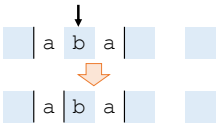
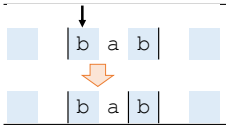
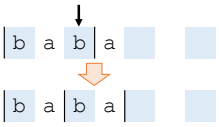
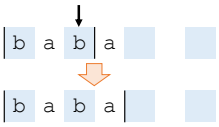
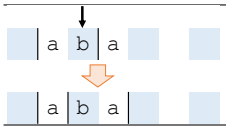
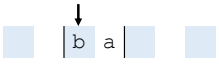
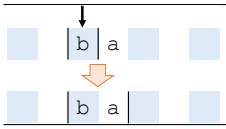
	Starting point. We eliminate the length-1 ground phrase $t_n[i_{\mathtt{b}}+1] = \mathtt{a}$ at an odd position $i_{\mathtt{b}}+1$, by including it in the same phrase as is partner $t_n[i_{\mathtt{b}}] = \mathtt{b}$, in this case, on its left. We can do this by modifying the source of a to point to the position next to the source of b, i.e., setting $f_{B_n}(i_{\mathtt{b}}+1) = f_{B_n}(i_{\mathtt{b}})+1$. This is done recursively at the source positions, until we reach one of the following cases, where the source of b no longer points to a position of the same parity.
	Case 1: We introduce a new length-1 ground phrase, and modify the boundaries. We are done since both $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})] = \mathtt{b}$ and $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})+1] = \mathtt{a}$ are in a length-2 phrase starting at an even position, i.e., a relaxed ground phrase. We are sure that $i_{\mathtt{b}}+1$ is not involved in a cycle. We recursively apply the procedure to the new length-1 ground phrase a at an odd position.
	Case 2: Same as Case 1, with the exception that the new length-1 ground phrase is b at an even position.
	Case 3-1: This case is when there are no consecutive phrases of ba and ba in the current bidirectional scheme. We introduce a new length-2 phrase, and modify the boundaries. We are done since both $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})] = \mathtt{b}$ and $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})+1] = \mathtt{a}$ are in a length-2 phrase starting at an even position, i.e., a relaxed ground phrase.
	Case 3-2: This case is when there already are consecutive phrases of ba and ba in the current bidirectional scheme. We create the phrase baba and make its source be the consecutive phrases of ba and ba (possibly constructed in Case 3-1). No cycles are introduced since the new source are relaxed ground phrases.
	Case 4: Same as Case 1.
	Case 5: There is nothing to do. We are done since both $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})] = \mathtt{b}$ and $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})+1] = \mathtt{a}$ are in a length-2 phrase starting at an even position, i.e., a relaxed ground phrase.
	Case 6: We expand the phrase to include its partner. We are done, since both $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})] = \mathtt{b}$ and $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})+1] = \mathtt{a}$ are in a length-2 phrase starting at an even position, i.e., a relaxed ground phrase.

Fig. 2. Terminal cases for eliminating a length-1 ground phrase $t_n[i_{\mathtt{b}}+1] = \mathtt{a}$ at an odd position $i_{\mathtt{b}}+1$ (see Section 4.2). The shaded squares are even positions. The vertical bars denote phrase boundaries. The black arrow points to the position $f_{B_n}^{j'-1}(i_{\mathtt{b}})$, where $j' \geq 1$ is the smallest integer such that the parities of $f_{B_n}^{j'-1}(i_{\mathtt{b}})$ and $f_{B_n}^{j'}(i_{\mathtt{b}})$ differ. The first line and second line of each case (except Case 5) respectively show the phrase boundaries before and after the modification.

laxed ground phrases, and the occurrences of each of the other phrases have the same parity due to Lemma 4. Thus, we now have a valid bidirectional scheme B'_n where all phrases are of even length, and length-2 phrases are considered to be relaxed ground phrases.

Analysis of the Number of Phrases It is easy to see that Steps 2 and 3 do not increase the number of phrases. Also, Step 2 does not decrease the number of length-2 phrases that start at even positions, i.e., relaxed ground phrases, created in Step 1, which will become ground phrases in B_{n-1}. Thus, we focus on the analysis of Step 1.

Examining each case of Fig. 2, we can see that while at the start we eliminate a length-1 ground phrase and decrease the number of phrases, Cases 1, 2, 3-1, and 4 introduce a new phrase, thus do not change the total number of phrases. Also, notice that in Case 6, two ground phrases are eliminated, while the total number of phrases decreases only by one, since the second length-1 ground phrase is expanded. Case 3-1 can occur in total at most twice, once for consecutive phrases of `ba` and once for consecutive phrases of `ab`. Thus, we obtain the following inequality:

$$|B_{n-1}| \leq |B_n| - \lceil (\#_g(B_n) - 2)/2 \rceil. \tag{1}$$

If $|B_{n-1}| \leq |B_n| - 1$, then we can choose $i = 1$ for Lemma 5 and are done. Otherwise, $|B_{n-1}| = |B_n|$. This implies that $\#_g(B_n) = 2$, and also that Case 3-1 was applied twice. Thus, there exists at least 2 phrases of `ab` and `ba` each, which are converted by μ^{-1} to ground phrases in B_{n-1}, implying $\#_g(B_{n-1}) \geq 4$. Then, applying Equation (1) for $n-2$, we have

$$\begin{aligned} |B_{n-2}| &\leq |B_{n-1}| - \lceil (\#_g(B_{n-1}) - 2)/2 \rceil \\ &\leq |B_{n-1}| - 1 = |B_n| - 1. \end{aligned}$$

If $|B_{n-2}| \leq |B_n| - 2$, then we can choose $i = 2$ for Lemma 5. Otherwise, $|B_{n-2}| = |B_n| - 1$. This implies that $\#_g(B_{n-1}) = 4$ and that Case 3-1 was applied twice, and Case 6 was applied once. Therefore, we get $\#_g(B_{n-2}) \geq 5$. Finally, applying Equation (1) for $n-3$, we have

$$\begin{aligned} |B_{n-3}| &\leq |B_{n-2}| - \lceil (\#_g(B_{n-2}) - 2)/2 \rceil \\ &\leq |B_{n-2}| - 2 \\ &= |B_n| - 3. \end{aligned}$$

This proves Lemma 5, and thus Theorem 2.

5 Conclusion

We have shown that for any $n \geq 2$, the size $b(t_n)$ of the smallest bidirectional scheme for the n-th Thue–Morse word t_n is exactly $n + 2$. From the result that

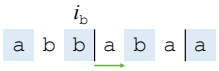
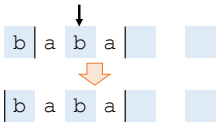
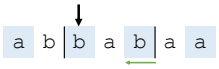
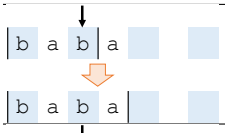
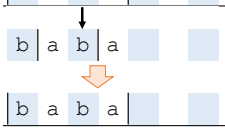
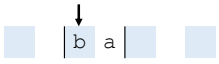
	Starting point. We wish to eliminate the length-3 phrase `aba` starting at an odd position $i_{\mathtt{b}}+1$. We move the boundary so that the length-3 phrase shrinks to a length-2 phrase that starts at an even position. In this case, we extend the phrase on its left side to include $t_n[i_{\mathtt{b}}+1]=\mathtt{a}$. We can do this by modifying the source of `a` to point to the position next to the source of `b`, i.e., setting $f(i_{\mathtt{b}}+1)=f(i_{\mathtt{b}})+1$. This is done recursively at the source positions, until we reach one of the following cases, where the source of `b` no longer points to a position of the same parity.
	Case 1: Noticing that the phrase is a substring of `baba`, we expand the phrase and make the source point to $i_{\mathtt{b}}$. This is possible because $t_n[i_{\mathtt{b}}..i_{\mathtt{b}}+3]=\mathtt{baba}$. Also, since each of the `ba`'s can finally be traced to the relaxed ground phrase created at the starting point, i.e., `ba` at position $i_{\mathtt{b}}+2$, we are done.
	Case 2: We reach a different length-3 phrase `bab` that we wish to eliminate, that ends at an even position. We recursively apply the procedure to eliminate the phrase `bab`. In this case, that will shrink this length-3 phrase to a length-2 phrase by expanding the phrase to its right. Then, we are done with the elimination of the original length-3 phrase `aba`, since $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})]=\mathtt{b}$ and $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})+1]=\mathtt{a}$ are in a length-2 phrase starting at an even position, i.e., a relaxed ground phrase.
	Case 3: Same as Case 1.
	Case 4: Same as Case 1.
	Case 5: There is nothing to do. We are done, since both $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})]=\mathtt{b}$ and $t_n[f_{B_n}^{j'-1}(i_{\mathtt{b}})+1]=\mathtt{a}$ are in a length-2 phrase starting at an even position, i.e., a relaxed ground phrase.

Fig. 3. Terminal cases for eliminating a length-3 phrase `aba` that starts at an odd position $i_{\mathtt{b}}+1$ (see Section 4.2). The shaded squares are even positions. The vertical bars denote phrase boundaries. The black arrow points to the position $f_{B_n}^{j'-1}(i_{\mathtt{b}})$, where $j'\geq 1$ is the smallest integer such that the parities of $f_{B_n}^{j'-1}(i_{\mathtt{b}})$ and $f_{B_n}^{j'}(i_{\mathtt{b}})$ differ. The first and second lines in Cases 1, 3, 4 show the phrase boundaries before and after the modification. The characters outside the phrase considered for each case can be inferred from being a partner of a phrase, and also from Lemma 2.

the smallest string attractor of t_n is 4 for any $n\geq 4$ [7] and that $|t_n|=2^n$, we have shown that Thue–Morse words are an example of a family of strings

$\{S_n\}_{n\geq 1}$ in which each string S_n has $b(S_n) = \Theta(\gamma(S_n) \log \frac{|S_n|}{\gamma(S_n)})$ as the size of its smallest bidirectional parsing, where $\gamma(S_n)$ is the size of its smallest string attractor, and $|S_n| = 2^n$ is its length. Note that we can generalize this to hold for any $\gamma \geq 4$: Given a $\gamma \geq 4$, concatenate $k = \lfloor \gamma/4 \rfloor$ copies of t_n, each using distinct letters from a different binary alphabet. Finally, we add (γ mod 4) more distinct characters to make the smallest string attractor of the resulting string exactly γ. We thus can obtain a string of length $N = k \cdot 2^n + O(1)$ with $b = \Theta(kn) = \Theta(\gamma \log \frac{N}{\gamma})$. Whether this can be achieved for any γ by a family of binary strings is not yet known.

Our result shows for the first time the separation between γ and b, i.e., there are string families such that $\gamma = o(b)$. Although it is still open whether $O(\gamma \log N)$ bits is enough to represent any string of length N, it seems not possible by dictionary compression, i.e., copy/pasting within the string.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP20H04141 (HB), JP20J21147 (MF), JP19K20213 (TI), JP21K17701 (DK), JP20J11983 (TM).

References

1. Berstel, J., Reutenauer, C.: Square-free words and idempotent semigroups. In: Lothaire, M. (ed.) Combinatorics on Words, p. 18–38. Cambridge Mathematical Library, Cambridge University Press, 2 edn. (1997). https://doi.org/10.1017/CBO9780511566097.005
2. Burrows, M., Wheeler, D.J.: A block-sorting lossless data compression algorithm. Tech. rep. (1994)
3. Kempa, D., Kociumaka, T.: Resolution of the burrows-wheeler transform conjecture. In: 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020. pp. 1002–1013. IEEE (2020). https://doi.org/10.1109/FOCS46700.2020.00097, https://doi.org/10.1109/FOCS46700.2020.00097
4. Kempa, D., Prezza, N.: At the roots of dictionary compression: string attractors. In: Diakonikolas, I., Kempe, D., Henzinger, M. (eds.) Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018. pp. 827–840. ACM (2018). https://doi.org/10.1145/3188745.3188814, https://doi.org/10.1145/3188745.3188814
5. Kociumaka, T., Navarro, G., Prezza, N.: Towards a definitive measure of repetitiveness. In: Kohayakawa, Y., Miyazawa, F.K. (eds.) LATIN 2020: Theoretical Informatics - 14th Latin American Symposium, São Paulo, Brazil, January 5-8, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12118, pp. 207–219. Springer (2020). https://doi.org/10.1007/978-3-030-61792-9_17, https://doi.org/10.1007/978-3-030-61792-9_17
6. Kociumaka, T., Navarro, G., Prezza, N.: Towards a definitive compressibility measure for repetitive sequences (2021)

7. Kutsukake, K., Matsumoto, T., Nakashima, Y., Inenaga, S., Bannai, H., Takeda, M.: On repetitiveness measures of thue-morse words. In: Boucher, C., Thankachan, S.V. (eds.) String Processing and Information Retrieval - 27th International Symposium, SPIRE 2020, Orlando, FL, USA, October 13-15, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12303, pp. 213–220. Springer (2020). https://doi.org/10.1007/978-3-030-59212-7_15, https://doi.org/10.1007/978-3-030-59212-7_15
8. Mantaci, S., Restivo, A., Romana, G., Rosone, G., Sciortino, M.: String attractors and combinatorics on words. In: Cherubini, A., Sabadini, N., Tini, S. (eds.) Proceedings of the 20th Italian Conference on Theoretical Computer Science, ICTCS 2019, Como, Italy, September 9-11, 2019. CEUR Workshop Proceedings, vol. 2504, pp. 57–71. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2504/paper8.pdf
9. Mantaci, S., Restivo, A., Romana, G., Rosone, G., Sciortino, M.: A combinatorial view on string attractors. Theor. Comput. Sci. **850**, 236–248 (2021). https://doi.org/10.1016/j.tcs.2020.11.006, https://doi.org/10.1016/j.tcs.2020.11.006
10. Mantaci, S., Restivo, A., Sciortino, M.: Burrows-Wheeler transform and Sturmian words. Inf. Process. Lett. **86**(5), 241–246 (2003), https://doi.org/10.1016/S0020-0190(02)00512-4
11. Morse, M.: Recurrent geodesics on a surface of negative curvature. Trans. Am. Math. Soc. **22**, 84–100 (1921)
12. Navarro, G.: Indexing highly repetitive string collections, part i: Repetitiveness measures. ACM Comput. Surv. **54**(2) (Mar 2021). https://doi.org/10.1145/3434399, https://doi.org/10.1145/3434399
13. Navarro, G., Ochoa, C., Prezza, N.: On the approximation ratio of ordered parsings. IEEE Trans. Inf. Theory **67**(2), 1008–1026 (2021). https://doi.org/10.1109/TIT.2020.3042746, https://doi.org/10.1109/TIT.2020.3042746
14. Prouhet, E.: Mémoire sur quelques relations entre les puissances des nombres. C. R. Acad. Sci. Paris Sér. **133**, 225 (1851)
15. Storer, J.A., Szymanski, T.G.: Data compression via textual substitution. J. ACM **29**(4), 928–951 (1982), https://doi.org/10.1145/322344.322346
16. Thue, A.: Über unendliche zeichenreihen. Norske vid. Selsk. Skr. Mat. Nat. Kl. **7**, 1–22 (1906)
17. Ziv, J., Lempel, A.: A universal algorithm for sequential data compression. IEEE Trans. Inf. Theory **23**(3), 337–343 (1977). https://doi.org/10.1109/TIT.1977.1055714, https://doi.org/10.1109/TIT.1977.1055714