On repetitiveness measures of Thue-Morse words

Kanaru Kutsukake[1], Takuya Matsumoto[1],
Yuto Nakashima[1][0000-0001-6269-9353],
Shunsuke Inenaga[1,2][0000-0002-1833-010X], Hideo Bannai[3][0000-0002-6856-5185],
and Masayuki Takeda[1][0000-0002-6138-1607]

[1] Department of Informatics, Kyushu University, Fukuoka, Japan
[2] PRESTO, Japan Science and Technology Agency, Kawaguchi, Japan
[3] M&D Data Science Center, Tokyo Medical and Dental University, Tokyo, Japan
{kutsukake.kanaru,matsumoto.takuya,yuto.nakashima,
inenaga,takeda}@inf.kyushu-u.ac.jp
hdbn.dsc@tmd.ac.jp

Abstract. We show that the size $\gamma(t_n)$ of the smallest string attractor of the nth Thue-Morse word t_n is 4 for any $n \geq 4$, disproving the conjecture by Mantaci et al. [ICTCS 2019] that it is n. We also show that $\delta(t_n) = \frac{10}{3+2^{4-n}}$ for $n \geq 3$, where $\delta(w)$ is the maximum over all $k = 1, \ldots, |w|$, the number of distinct substrings of length k in w divided by k, which is a measure of repetitiveness recently studied by Kociumaka et al. [LATIN 2020]. Furthermore, we show that the number $z(t_n)$ of factors in the self-referencing Lempel-Ziv factorization of t_n is exactly $2n$.

Keywords: String attractors · Thue-Morse words

1 Introduction

Measures which indicate the repetitiveness in a string is a hot and important topic in the field of string compression. For example, given string w, the size $g(w)$ of the smallest grammar that derives solely w [5], the number $z(w)$ of factors in the Lempel-Ziv factorization [12], the number $r(w)$ of runs in the Burrows-Wheeler transform [4] (RLBWT), and the size $b(w)$ of the smallest bidirectional scheme (or macro schemes) [18]. Recently, Kempa and Prezza proposed the notion of *string attractors* [10], and showed that the size $\gamma(w)$ of the smallest string attractor of w is a lower bound on the size of the compressed representation for these dictionary compression schemes. While $z(w)$ and $r(w)$ are known to be computable in linear time, it is NP-hard to compute $g(w), b(w), \gamma(w)$ [7,18,10].

To further understand these measures, Mantaci et al. [13] studied the size of the smallest string attractor in several well known family of strings. In particular, they showed a size-2 string attractor for standard Sturmian words which is the smallest possible. They further showed a string attractor of size n for the nth Thue-Morse word t_n, and conjectured it to be the smallest.

In this paper, we continue this line of work, and investigate the exact values of various repetitive measures of the nth Thue-Morse word t_n. More specifically, we show that the size $\gamma(t_n)$ of the smallest string attractor of t_n is 4 for $n \geq 4$, disproving Mantaci et al.'s conjecture. Furthermore, we give the exact value $\delta(t_n) = \frac{10}{3+2^{4-n}}$ for $n \geq 3$, of the repetitiveness measure recently studied by Kociumaka et al. [11], and the size $z(t_n) = 2n$ of the self-referencing LZ77 factorization.

We note that for any standard Sturmian word s, $z(s) = \Theta(\log |s|)$ [1], while the size $r(s)$ of the RLBWT is always constant [14]. On the other hand, $z(t_n)$ and $r(t_n)$ are both $\Theta(n)$, i.e., logarithmic in the length $|t_n|$ (the former due to [1] as well as this work, and the latter due to [3]). This shows that Thue-Morse words are an example where the size of smallest string attractor is *not* a tight lower bound on the size of the smallest of the known efficiently computable dictionary compressed representation, namely, $\min\{z(w), r(w)\}$. We also conjecture that $b(t_n) = \Theta(n)$, which would seem to imply that the size of the smallest string attractor is not a tight lower bound for *all* currently known dictionary compression schemes.

Let $\ell(w)$ denote the size of the Lyndon factorization [6] of w. It is known that for any w, $\ell(w) = O(g(w))$ [8] and $\ell(w) = O(z(w))$ [20], although it can be much smaller. Interestingly, it is also known that $\ell(t_n) = \Theta(n)$ (Theorem 3.1, Remark 3.8 of [9]). Thus, if $b(t_n) = \Theta(n)$, then $\ell(t_n)$ would be an asymptotically tight lower bound for the smallest size of known dictionary compression schemes for t_n, while $\gamma(t_n)$ is not.

Table 1 summarizes what we know so far.

Table 1. Repetitiveness measures for the n-th Thue-Morse word t_n.

measure	description	value	reference
$z(t_n)$	Size of Lempel-Ziv factorization with self-reference	$2n$	[1], this work
$r(t_n)$	Number of same-character runs in BWT	$2n-2$	[3]
$\ell(t_n)$	Size of Lyndon factorization	$\left\lfloor \frac{3n-2}{2} \right\rfloor$	[9]
$b(t_n)$	Size of smallest bidirectional scheme	open	N/A
$\gamma(t_n)$	Size of smallest string attractor	4 $(n \geq 4)$	this work
$\delta(t_n)$	maximum of subword complexity divided by subword length	$\frac{10}{3+2^{4-n}}$ $(n \geq 3)$	this work

2 Preliminaries

Let Σ denote a set of symbols called the alphabet. An element of Σ^* is called a string. For any $k \geq 0$, let Σ^k denote the set of strings of length exactly k. For any string w, the length of w is denoted by $|w|$. For any $1 \leq i \leq |w|$, let $w[i]$ denote the ith symbol of w, and for any $1 \leq i \leq j \leq |w|$, let $w[i..j] = w[i]w[i+1] \cdots w[j]$.

If $w = xyz$ for strings $x, y, z \in \Sigma^*$, then x, y, z are respectively called a prefix, substring, suffix of w. We denote by $\mathit{Substr}(w)$, the set of substrings of w.

In this paper, we will only consider the binary alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. For any string $w \in \Sigma^*$, let $\overline{w}$ denote the string obtained from w by changing all occurrences of a (resp. b) to b (resp. a).

Definition 1 (Thue-Morse Words [16,19,15]). *The n-th Thue-Morse word t_n is a string over a binary alphabet $\{\mathtt{a}, \mathtt{b}\}$ defined recursively as follows: $t_0 = \mathtt{a}$, and for any $n > 0$, $t_n = t_{n-1}\overline{t_{n-1}}$.*

It is a simple observation that $|t_n| = 2^n$ for any $n \geq 0$.

Below, we define the repetitiveness measures used in this paper:

String attractors [10] For any string w, a set Γ of positions in w is a string attractor of w, if, for any substring x of w, there is an occurrence of x in w that contains a position in Γ. For any string w, we will denote the size of a smallest string attractor of w as $\gamma(w)$.

δ [17,11]
For any string w,

$$\delta(w) = \max_{k=1,..,|w|} \left(|\Sigma^k \cap \mathit{Substr}(w)|/k \right).$$

LZ factorization [12] For any string w, the LZ factorization of w is the sequence $f_1, \ldots, f_z$ of non-empty strings such that $w = f_1 \cdots f_z$, and for any $1 \leq i \leq z$, f_i is the longest prefix of $f_i \cdots f_z$ which has at least two occurrences in $f_1 \cdots f_i$, or, $|f_i| = 1$ otherwise. We denote the size of the LZ factorization of string w as $z(w)$.

It is known that $\delta(w) \leq \gamma(w) \leq z(w), r(w)$ for any w [7,10].

3 Repetitive Measures of Thue-Morse Words

3.1 $\gamma(t_n)$

Mantaci et al. [13] showed the following explicit string attractor of size n for the n-th Thue-Morse word.

Theorem 1 (Theorem 8 of [13]). *A string attractor of the n-th Thue Morse word, with $n \geq 3$ is*

$$\left\{2^{n-1} + 1\right\} \cup \{3 \cdot 2^{i-2} \mid i = 2, \ldots, n\}.$$

To prove our new upperbound of 4 for the smallest string attractor of t_n for $n \geq 4$, we first show the following lemma.

Lemma 1. *Let*

$$N_n = \{t_{n-1}\overline{t_{n-1}}\} \cup \left(\bigcup_{k=0}^{n-2} \{t_k\overline{t_k}, \overline{t_k}t_k\} \right).$$

Then, for any substring w and $n \geq 2$, there exists $s \in N(n)$ such that the occurrence of w in s contains the center of s (i.e., position $|s|/2$).

Proof. Consider the recursively defined perfect binary tree with t_n as the root, with t_{n-1} and $\overline{t_{n-1}}$ respectively as its left and right children (See Fig. 1). The leaves consist of either t_0 or $\overline{t_0}$, each corresponding to a position of t_n. If $|w| = 1$, then, we can choose $t_1 = t_0\overline{t_0} = \mathtt{ab}$ for $\mathtt{a}$ and $t_2 = t_1\overline{t_1} = \mathtt{abba}$ for $\mathtt{b}$. For any substring $w = t_n[i..j]$ of length at least 2, consider the lowest common ancestor of leaves corresponding to $t_n[i]$ and $t_n[j]$. Each node of the tree is $t_n = t_{n-1}\overline{t_{n-1}}$ if it is the root, or otherwise, either $t_{k+1} = t_k\overline{t_k}$ or $\overline{t_{k+1}} = \overline{t_k}t_k$ for some $0 \le k \le n-2$. Since w is a substring that starts in the left child and ends in the right child of the lowest common ancestor, the occurrence of w must contain the center, and the lemma holds. □

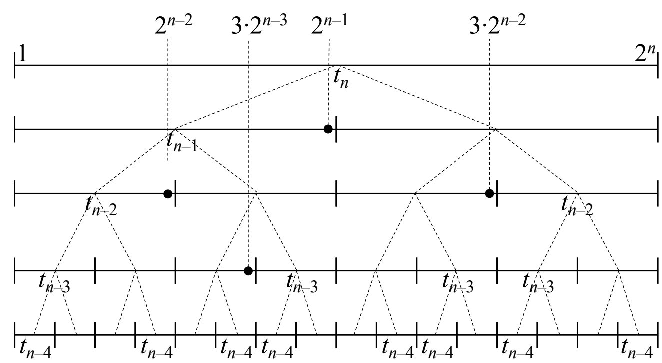

Fig. 1. A representation of t_n as a perfect binary tree (shown to depth 4) introduced in the proof of Lemma 1. For each level where segments are labeled with t_k, non-labeled segments represent $\overline{t_k}$. The black circles depict the four positions in K_n defined in Theorem 2, at the node at which the center of the parent coincides with the position.

Theorem 2. *For any $n \ge 4$, the set*

$$K_n = \left\{2^{n-2}, 3 \cdot 2^{n-3}, 2^{n-1}, 3 \cdot 2^{n-2}\right\}$$

is a string attractor of t_n.

Proof. Let w be an arbitrary substring of t_n. From Lemma 1, it suffices to show that any element in N_n has an occurrence in t_n whose center coincides with a position in K_n. For $t_{n-1}\overline{t_{n-1}}$, $t_{n-2}\overline{t_{n-2}}$, $\overline{t_{n-2}}t_{n-2}$, and $\overline{t_{n-3}}t_{n-3}$, it is clear from Fig. 1 that their centers respectively coincide with the four elements of K_n. Furthermore, there is an occurrence of $t_{n-3}\overline{t_{n-3}}$ whose center coincides with that of $t_{n-1}\overline{t_{n-1}}$, and thus with an element of K_n. More generally, for any $2 \le k \le n-2$, each occurrence of $t_k\overline{t_k}$ implies an occurrence of $t_{k-2}\overline{t_{k-2}}$ whose centers coincide. This is because

$$\begin{aligned} t_k\overline{t_k} &= t_{k-1}\overline{t_{k-1}t_{k-1}}t_{k-1} \\ &= t_{k-1}\overline{t_{k-2}}t_{k-2}\overline{t_{k-2}}t_{k-2}t_{k-1}. \end{aligned}$$

The same argument holds for $\overline{t_{k-2}}t_{k-2}$ by considering $\overline{t_k}t_k$. The theorem follows from a simple induction. $\square$

Theorem 3. *$\gamma(t_n) = 4$ for any $n \geq 4$.*

Proof. Theorem 2 implies $\gamma(t_n) \leq 4$. From Theorem 4 shown in the next subsection, we have $\delta(t_n) > 3$ for $n \geq 6$. Since $\gamma(t_n)$ is an integer which cannot be smaller than $\delta(t_n)$, it follows that $\gamma(t_n) \geq 4$ for $n \geq 6$. For $n = 4, 5$, it can be shown by exhaustive search that there is no string attractor of size 3. $\square$

3.2 $\delta(t_n)$

Brlek [2] investigated the number of distinct substrings of length m in t_n, and gave an exact formula. Below is a summary of his result which will be a key to computing $\delta(t_n)$.

Lemma 2 (Proposition 4.2, Corollary 4.2.1, Proposition 4.4 of [2]). *The number $P_n(m)$ of distinct substrings of length $m \geq 3$ in t_n ($n \geq 3$) is:*

$$P_n(m) = \begin{cases} 2^n - m + 1 & 2^{n-2} + 1 \leq m \leq 2^n \\ 6 \cdot 2^{q-1} + 4p & 3 \leq m \leq 2^{n-2}, 0 < p \leq 2^{q-1} \\ 8 \cdot 2^{q-1} + 2p & 3 \leq m \leq 2^{n-2}, 2^{q-1} < p \leq 2^q \end{cases}$$

where p, q are values uniquely determined by $m = 2^q + p + 1$ and $0 < p \leq 2^q$.

Theorem 4.

$$\delta(t_n) = \begin{cases} 1 & n = 0 \\ 2 & n = 1, 2 \\ \frac{10}{3+2^{4-n}} & n \geq 3 \end{cases}$$

Proof. We only consider $n \geq 3$ below. The number of distinct substrings of length 1 and 2 in t_n, are respectively 2 and 4. For $2^{n-2} + 1 \leq m \leq 2^n$,

$$\max_{2^{n-2}+1 \leq m \leq 2^n} \frac{P_n(m)}{m} = \max_{2^{n-2}+1 \leq m \leq 2^n} \left\{ \frac{2^n + 1}{m} - 1 \right\} = \frac{2^n + 1}{2^{n-2} + 1} - 1 = \frac{3}{1 + 2^{2-n}}.$$

For $3 \leq m \leq 2^{n-2}$ and fixed q, it is easy to verify that $P_n(m)/m$ is increasing when $0 < p \leq 2^{q-1}$, and non-increasing when $2^{q-1} < p \leq 2^q$, because

$$\left(\frac{6 \cdot 2^{q-1} + 4p}{2^q + p + 1} \right)' = \frac{4(2^q + p + 1) - (6 \cdot 2^{q-1} + 4p)}{(2^q + p + 1)^2} = \frac{2^q + 4}{(2^q + p + 1)^2} > 0$$

and

$$\left(\frac{8 \cdot 2^{q-1} + 2p}{2^q + p + 1} \right)' = \frac{2(2^q + p + 1) - (8 \cdot 2^{q-1} + 2p)}{(2^q + p + 1)^2} = \frac{(2 - 4 \cdot 2^{q-1})}{(2^q + p + 1)^2} \leq 0.$$

Therefore, for a fixed q, the maximum value of $\frac{P_n(m)}{m}$ is obtained when $p = 2^{q-1}$, i.e., $\frac{6 \cdot 2^{q-1} + 4 \cdot 2^{q-1}}{2^q + 2^{q-1} + 1} = \frac{10 \cdot 2^{q-1}}{3 \cdot 2^{q-1} + 1} = \frac{10}{3 + 2^{1-q}}$. Since this is increasing in q, we have that $\max_{3 \leq m \leq 2^{n-2}} \frac{P_n(m)}{m}$ is obtained by choosing the largest possible $q = n-3$ (where $p = 2^{q-1} = 2^{n-4}$, and thus $m = 2^{n-3} + 2^{n-4} + 1 = 3 \cdot 2^{n-4} + 1 \leq 2^{n-2}$), which gives us the final result $\delta(t_n) = \max\{\frac{2}{1}, \frac{4}{2}, \frac{10}{3+2^{4-n}}, \frac{3}{1+2^{2-n}}\} = \frac{10}{3+2^{4-n}}$. $\square$

3.3 LZ77

We consider the size $z(t_n)$ of the LZ factorization. Although Berstel and Savelli [1] have given a complete characterization of the LZ factorization for the infinite Thue-Morse word, we show an alternate proof in terms of the n-th Thue-Morse word. Below is an important lemma, again by Brlek, we will use.

Lemma 3 (Corollary 4.1.1 of [2]). *The word t_n has one and only one occurrence of every factor w such that $|w| \geq 2^{n-2} + 1$.*

Theorem 5. *For any $n \geq 1$, $z(t_n) = 2n$.*

Proof. Clearly, $z(t_1) = 2$. Since $t_k = t_{k-1}\overline{t_{k-1}} = t_{k-2}\overline{t_{k-2}t_{k-2}}t_{k-2}$, it is easy to see that $z(t_k) \leq z(t_{k-1}) + 2$, because $\overline{t_{k-2}}$ and t_{k-2} respectively have earlier occurrences in t_k. Thus, $z(t_n) \leq 2n$. On the otherhand, Lemma 3 implies that the substring $t_k[2^{k-1}..3 \cdot 2^{k-2}]$ of length $2^{k-2} + 1$ cannot be a single LZ factor, implying that position $2^{k-1}(= |t_{k-1}|)$ and position $3 \cdot 2^{k-2}(> |t_{k-1}|)$ belong to different factors. Similarly, the substring $t[3 \cdot 2^{k-2}..2^k]$ of length $2^{k-2} + 1$ cannot cannot be a single LZ factor, implying that position $3 \cdot 2^{k-2}$ and position 2^k belong to different factors. Thus, $z(t_{k+1}) \geq z(t_k) + 2$, implying $z(t_n) \geq 2n$. □

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP18K18002 (YN), JP17H01697 (SI), JP16H02783, JP20H04141 (HB), JP18H04098 (MT), and JST PRESTO Grant Number JPMJPR1922 (SI).

References

1. Berstel, J., Savelli, A.: Crochemore factorization of Sturmian and other infinite words. In: Proc. 31st International Symposium on Mathematical Foundations of Computer Science (MFCS 2006). Lecture Notes in Computer Science, vol. 4162, pp. 157–166. Springer (2006), https://doi.org/10.1007/11821069_14
2. Brlek, S.: Enumeration of factors in the Thue-Morse word. Discrete Applied Mathematics **24**(1), 83 – 96 (1989), https://doi.org/10.1016/0166-218X(92)90274-E
3. Brlek, S., Frosini, A., Mancini, I., Pergola, E., Rinaldi, S.: Burrows-Wheeler transform of words defined by morphisms. In: Proc. 30th International Workshop on Combinatorial Algorithms (IWOCA 2019). Lecture Notes in Computer Science, vol. 11638, pp. 393–404. Springer (2019), https://doi.org/10.1007/978-3-030-25005-8_32
4. Burrows, M., Wheeler, D.J.: A block-sorting lossless data compression algorithm. SRC Research Report **124** (1994)
5. Charikar, M., Lehman, E., Liu, D., Panigrahy, R., Prabhakaran, M., Sahai, A., Shelat, A.: The smallest grammar problem. IEEE Trans. Inf. Theory **51**(7), 2554–2576 (2005), https://doi.org/10.1109/TIT.2005.850116
6. Chen, K.T., Fox, R.H., Lyndon, R.C.: Free differential calculus, IV. the quotient groups of the lower central series. Annals of Mathematics **68**(1), 81–95 (1958), http://www.jstor.org/stable/1970044
7. Christiansen, A.R., Ettienne, M.B., Kociumaka, T., Navarro, G., Prezza, N.: Optimal-time dictionary-compressed indexes (2019), http://arxiv.org/abs/1811.12779v6
8. I, T., Nakashima, Y., Inenaga, S., Bannai, H., Takeda, M.: Faster Lyndon factorization algorithms for SLP and LZ78 compressed text. In: Proc. 20th International Symposium on String Processing and Information Retrieval (SPIRE 2013). Lecture Notes in Computer Science, vol. 8214, pp. 174–185. Springer (2013), https://doi.org/10.1007/978-3-319-02432-5_21
9. Ido, A., Melançon, G.: Lyndon factorization of the Thue-Morse word and its relatives. Discret. Math. Theor. Comput. Sci. **1**(1), 43–52 (1997), http://dmtcs.episciences.org/233
10. Kempa, D., Prezza, N.: At the roots of dictionary compression: string attractors. In: Diakonikolas, I., Kempe, D., Henzinger, M. (eds.) Proc. 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018). pp. 827–840. ACM (2018), https://doi.org/10.1145/3188745.3188814
11. Kociumaka, T., Navarro, G., Prezza, N.: Towards a definitive measure of repetitiveness. In: Proc. 14th Latin American Symposium on Theoretical Informatics (LATIN) (2020), https://arxiv.org/abs/1910.02151, to appear
12. Lempel, A., Ziv, J.: On the complexity of finite sequences. IEEE Trans. Inf. Theory **22**(1), 75–81 (1976), https://doi.org/10.1109/TIT.1976.1055501
13. Mantaci, S., Restivo, A., Romana, G., Rosone, G., Sciortino, M.: String attractors and combinatorics on words. In: Proc. 20th Italian Conference on Theoretical Computer Science (ICTCS 2019). pp. 57–71 (2019), http://ceur-ws.org/Vol-2504/paper8.pdf
14. 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
15. Morse, M.: Recurrent geodesics on a surface of negative curvature. Trans. Am. Math. Soc. **22**, 84100 (1921)

16. Prouhet, E.: Mémoire sur quelques relations entre les puissances des nombres. C. R. Acad. Sci. Paris Sér. **133**, 225 (1851)
17. Raskhodnikova, S., Ron, D., Rubinfeld, R., Smith, A.D.: Sublinear algorithms for approximating string compressibility. Algorithmica **65**(3), 685–709 (2013), https://doi.org/10.1007/s00453-012-9618-6
18. 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
19. Thue, A.: Über unendliche zeichenreihen. Norske vid. Selsk. Skr. Mat. Nat. Kl. **7**, 1–22 (1906)
20. Urabe, Y., Nakashima, Y., Inenaga, S., Bannai, H., Takeda, M.: On the size of overlapping Lempel-Ziv and Lyndon factorizations. In: 30th Annual Symposium on Combinatorial Pattern Matching (CPM 2019). pp. 29:1–29:11 (2019), https://doi.org/10.4230/LIPIcs.CPM.2019.29