Subword Complexity of a Generalized Thue-Morse Word

John Tromp [1] and Jeffrey Shallit [2]

Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

Let $y = y(0)y(1)y(2)\cdots$ be an infinite word over a finite alphabet, and let $p_y(r)$ count the number of distinct subwords of y of length r. In this paper we determine $p_y(r)$ when $y(i) = s_2(i) \bmod k$, where $s_2(i)$ denotes the sum of the base-2 digits of i. Our method is based on determining the redundancy of a certain code for subwords of a related infinite word.

Key words: formal languages; subword complexity; Thue-Morse word

1 Introduction.

Let $y = y(0)y(1)y(2)\cdots$ be a (finite or infinite) word over an alphabet Σ, and let w be a finite word. If there exist words v, x such that $y = vwx$, then we say w is a *subword* or *factor* of y. If $|\Sigma|$ is finite, then we define $p_y(r)$, the *subword complexity* of y, to be the map which counts the number of distinct subwords of y of length r.

Computing the subword complexity for "naturally-occurring" infinite words is an interesting and challenging problem that has received much attention in the past few years; for example, see the recent survey of Allouche [1].

For example, define $s_2(n)$ to be the sum of the base-2 digits of n. Then the infinite word

$$\mathbf{t} = t(0)t(1)t(2)\cdots = 0\,1\,1\,0\,1\,0\,0\,1\;\cdots,$$

[1] Supported in part by an NSERC International Fellowship and ITRC. E-mail: `tromp@math.uwaterloo.ca`.

[2] Supported in part by a grant from NSERC Canada. Please direct all correspondence to this author. E-mail: `shallit@graceland.uwaterloo.ca`.

where $t(i) = s_2(i) \bmod 2$, is the famous *Thue-Morse word.* In 1989, Brlek [3] and de Luca & Varricchio [6] independently computed the subword complexity of $\mathbf{t}$. However, both proofs were rather complicated.

In this paper, we introduce a new technique for computing subword complexity based on determining the redundancy of a certain encoding for subwords of a related infinite word. This technique allows us to compute the subword complexity of the *generalized Thue-Morse word* t_k, defined as follows: for $k \geq 2$, and $n \geq 0$, set $t_k(n) = s_2(n) \bmod k$, and set $t_k = t_k(0)t_k(1)t_k(2)\cdots$. (N.B. This is *not* the "generalized Morse sequence on k symbols" as introduced by Martin [7], whose subword complexity was studied by Mouline [9].) Note that t_k is an infinite word over the alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$, and the Thue-Morse word $\mathbf{t}$ is just t_2.

Here is our main theorem:

Theorem 1 *Let r be an integer ≥ 0. Then*

$$p_{t_k}(r+1) = \begin{cases} k, & \text{if } r = 0; \\ k^2, & \text{if } r = 1; \\ k(kr - 2^{a-1}), & \text{if } r = 2^a + b, \text{ where } a \geq 1,\ 0 \leq b < 2^{a-1}; \\ k(kr - 2^{a-1} - b), & \text{if } r = 2^a + 2^{a-1} + b, \text{ where } a \geq 1,\ 0 \leq b < 2^{a-1}. \end{cases}$$

The proof, as we will see, is relatively simple and completely self-contained. In the special case $k = 2$, we recover the results of Brlek and de Luca & Varricchio.

Operations between a word w and an integer in this paper will be done termwise; thus, for example, $w + 1$ denotes the word formed by adding 1 to each term in w. Also, by $w \equiv x \pmod{k}$, we mean $w_i \equiv x_i \pmod{k}$ for $0 \leq i < |w| = |x|$.

Let $[w]_k$ denote the value of the string w when interpreted as a number in base k, and let ϵ denote the empty string. By $(n)_k$ we will mean the string in $\epsilon + (\Sigma_k - 0)\Sigma_k^*$ that gives the ordinary base-k representation for $n \geq 0$, and by $[w]_k$ we mean the value of the string w when interpreted as a number in base k.

2 Proof of the Main Theorem

First, we consider the subwords of

$$\mathbf{s} = s_2(0)s_2(1)s_2(2)\cdots = 0\,1\,1\,2\,1\,2\,2\,3\,\cdots,$$

an infinite word over $\mathbb{N} = \{0, 1, 2, \ldots\}$. Let

$$w = s_2(i)s_2(i+1)\cdots s_2(i+r-1)$$

be a subword of length r of $\mathbf{s}$. Then

$$w + 1 = s_2(2^j + i)s_2(2^j + i + 1)\cdots s_2(2^j + i + r - 1)$$

for all sufficiently large j. Thus we have proved:

Lemma 2 *If w is a subword of $\mathbf{s}$, then so is $w+1$. If w is a subword of t_k, then so is $(w+1) \bmod k$.*

For an integer $n \neq 0$, define $\nu_2(n)$ to be the integer exponent e such that $2^e \,||\, n$, i.e., $2^e \mid n$ but $2^{e+1} \not| n$. Then we have the following

Lemma 3 *For $n \geq 0$ we have $\nu_2(n+1) = s_2(n) - s_2(n+1) + 1$.*

PROOF. We can write n as $[x\,0\,1^j]_2$, and $n+1$ as $[x\,1\,0^j]_2$, for some $j \geq 0$ and $x \in (0+1)^*$. Hence $s_2(n) - s_2(n+1) = j - 1 = \nu_2(n+1) - 1$. $\square$

Remark 4 *This lemma is equivalent to Legendre's relation $\nu_2(n!) = n - s_2(n)$; see [5, p. 10].*

Now for $n \geq 1$ define

$$\mathbf{v} = \nu_2(1)\nu_2(2)\nu_2(3)\cdots = 0\,1\,0\,2\,0\,1\,0\,3\,\cdots,$$

and $u_k = \mathbf{v} \bmod k$. Thus, for example,

$$u_2 = 0\,1\,0\,0\,0\,1\,0\,1\cdots.$$

Then, by Lemma 3, the subwords of u_k of length r are in 1–1 correspondence with the first differences of the subwords of t_k of length $r+1$. Together with Lemma 2, this shows:

Lemma 5 *For $r \geq 0$ we have $k \cdot p_{u_k}(r) = p_{t_k}(r+1)$.*

Thus, to prove Theorem 1, it suffices to determine the subword complexity of the word u_k. We start by defining an encoding for the subwords of $\mathbf{v}$.

Let r be a fixed positive integer. We define a function $f(n,m)$ as follows: for $0 \leq n < r$ and $m \geq 0$, let

$$f(n,m) = w_0 w_1 \cdots w_{r-1},$$

where $w_n = m$ and $w_{n'} = \nu_2(n'-n)$ for $n' \neq n$. For example, for $r = 8$, we have $f(3,1) = 01010102$.

Lemma 6 *Any length r subword of $\mathbf{v}$ is equal to $f(n,m)$ for some n and m.*

PROOF. Let m be the largest term in a nonempty subword $w = w_0 w_1 \cdots w_{r-1}$ of $\mathbf{v}$. Between any two occurrences of m in $\mathbf{v}$, say $\nu_2(2^m \cdot r)$ and $\nu_2(2^m \cdot (r+2))$ (where r is odd), lies an occurrence of a larger integer, since $\nu_2(2^m \cdot (r+1)) \geq m+1$. Hence m occurs exactly once in w, say $m = w_n$, and $w_{n'} < m$ for all $n' \neq n$.

Now suppose w_n is the s'th symbol of $\mathbf{v}$, i.e. $w_n = \nu_2(s)$. Then $w_{n'} = \nu_2(s+n'-n) = i$ for some $i < m$. Now $2^m \mid s$, and $2^i \mid\mid s+n'-n$, so $2^i \mid\mid (s+n'-n)-s = n'-n$. Thus $w_{n'} = \nu_2(n'-n)$. Thus we have shown $w = f(n,m)$. $\Box$

It follows that the corresponding subword $w \bmod k$ of u_k can be encoded by the pair $(n, m \bmod k)$. We next consider what alternative codes w might have, modulo k.

Lemma 7 *Let $0 \leq n, n' < r$, $0 \leq m, m' < k$, and $(n,m) \neq (n',m')$. Then $f(n,m) \equiv f(n',m') \pmod{k}$ iff the following four conditions hold:*

(i) $d := |n'-n| = 2^i$ *for some integer* $i \geq 0$*;*
(ii) $m \equiv m' \equiv i \pmod{k}$*;*
(iii) if $n < n'$ *then* $n - d < 0$ *and* $n' + d \geq r$*;*
(iv) if $n' < n$ *then* $n' - d < 0$ *and* $n + d \geq r$*.*

PROOF. Let $f(n,m) = w_0 w_1 \cdots w_{r-1}$ and $f(n',m') = x_0 x_1 \cdots x_{r-1}$. All congruences in the proof are $\pmod{k}$.

First, suppose the stated conditions hold. Then by condition (2) we have

$$w_n = m \equiv i = \nu_2(n-n') = \nu_2(n'-n) = x_n;$$

hence $w_n \equiv x_n$. Similarly, $w_{n'} \equiv x_{n'}$. To complete the proof of this direction, it suffices to show that $w_{n''} = x_{n''}$ for any $n'' \in \{0, 1, \ldots, r-1\} \setminus \{n, n'\}$. If not, then without loss of generality, assume that $\nu_2(n''-n) = w_{n''} < x_{n''} = \nu_2(n''-n')$. Then, as in the proof of Lemma 6, we have $w_{n''} = \nu_2((n''-n) - (n''-n')) = \nu_2(n'-n)$. But by condition (1), $i = \nu_2(n'-n)$, so $d = 2^i \mid n''-n$, and $2d = 2^{i+1} \mid n''-n'$. Hence $|n''-n| \geq d$ and $|n''-n'| \geq 2d$. This contradicts conditions (3) and (4).

Now we prove the other direction. Let $w = f(n,m)$ and $x = f(n',m')$, and suppose $w \equiv x$. Then if $n = n'$, we have $m = w_n \equiv w_{n'} = m'$; so $m = m'$, a contradiction. Hence $n \neq n'$; without loss of generality we may assume $n < n'$ and set $d = n' - n$.

We now show d is a power of 2. For assume not; then $s_2(d) \geq 2$, and we can write the binary expansion of d as follows: $(d)_2 = 1\,x\,1\,0^a$, where $x \in (0+1)^*$ and $a \geq 0$. Then $d = 2^a + y \cdot 2^{a+1}$, where $y = [1x]_2 \geq 1$. Now define $t = n + 2^{a+1}$. Clearly $n < t < n'$. Then $w_t = \nu_2(t-n) = a+1$, while

$$x_t = \nu_2(n'-t) = \nu_2(n+d-t) = \nu_2(2^a + (y-1)2^{a+1}) = a,$$

so $w_t \not\equiv x_t$. This contradiction shows d is indeed a power of 2, say $d = 2^i$ for some $i \geq 0$. Thus condition (1) is proved.

To prove (2), we observe that $m = w_n \equiv x_n = \nu_2(n-n') = i$, and similarly $m' = x_{n'} \equiv w_{n'} = \nu_2(n'-n) = i$.

To prove condition (3), first suppose that $n - d \geq 0$. Then by definition of f, $w_{n-d} = \nu_2((n-d)-n) = \nu_2(-d) = i$, while $x_{n-d} = \nu_2((n-d)-n') = \nu_2(-2d) = i+1$, so $w_{n-d} \not\equiv x_{n-d}$.

Condition (4) handles the case $n > n'$, and follows similarly. This completes the proof.

We now prove a lemma about subwords of u_k having multiple codes.

Lemma 8 *Each subword w of u_k of length r corresponds to at most two distinct encodings $f(n,m)$. If $r = 2^a + b$ with $a \geq 1$ and $0 \leq b < 2^{a-1}$, then exactly 2^{a-1} words have two codes. If $r = 2^a + 2^{a-1} + b$ with $a \geq 1$ and $0 \leq b < 2^{a-1}$, then exactly $2^{a-1} + b$ words have two codes.*

PROOF. If the four conditions of Lemma 7 hold for a given n, then an easy case analysis based on conditions (3) and (4) shows that n' is unique. Thus each word w has one or two codes.

By Lemma 7, the number of subwords of u_k of length r having two codes is exactly the number of pairs (n,n') with $0 \leq n < n' < r$, for which $d = n' - n = 2^i$ for some i, $n - d < 0$, and $n' + d \geq r$.

When $r = 2^a + b$ with $a \geq 1$ and $0 \leq b < 2^{a-1}$, there are b such pairs $(n, n+2^a)$ for $0 \leq n < b$, and $2^{a-1} - b$ pairs $(n, n+2^{a-1})$ for $b \leq n < 2^{a-1}$. This gives a total of 2^{a-1} pairs.

When $r = 2^a + 2^{a-1} + b$ with $a \geq 1$ and $0 \leq b < 2^{a-1}$, there are $2^{a-1} + b$ such pairs $(n, n + 2^a)$ for $0 \leq n < 2^{a-1} + b$. $\square$

We have therefore proved the following theorem:

Theorem 9 *If $r = 2^a + b$ with $a \geq 1$ and $0 \leq b < 2^{a-1}$, then $p_{u_k}(r) = kr - 2^{a-1}$. If $r = 2^a + 2^{a-1} + b$ with $a \geq 1$ and $0 \leq b < 2^{a-1}$, then $p_{u_k}(r) = kr - (2^{a-1} + b)$.*

Theorem 1 now follows by combining Theorem 9 and Lemma 5. $\square$

3 Concluding Remarks.

It follows from our result that the sequence $(p_{t_k}(r))_{r \geq 0}$ is 2-regular in the sense of Allouche and Shallit [2]. Furthermore, it is easy to see that $p_{t_k}(r + 1) - p_{t_k}(r) \leq k^2$, so that the sequence $(p_{t_k}(r + 1) - p_{t_k}(r))_{r \geq 0}$ is 2-automatic (or 2-recognizable) in the sense of Cobham [4].

For more general results along these lines, see [8].

Acknowledgement

We thank Drew Vandeth and the referees for suggesting improvements to this paper.

References

[1] J.-P. Allouche. Sur la complexité des suites infinies. *Bull. Belgian Math. Soc.* **1** (1994), 133–143.

[2] J.-P. Allouche and J. O. Shallit. The ring of k–regular sequences. *Theoret. Comput. Sci.* **98** (1992), 163–187.

[3] S. Brlek. Enumeration of factors in the Thue-Morse word. *Disc. Appl. Math.* **24** (1989), 83–96.

[4] A. Cobham. Uniform tag sequences. *Math. Systems Theory* **6** (1972), 164–192.

[5] A.-M. Legendre. *Théorie des Nombres.* Firmin Didot Frères, Paris, 1830.

[6] A. de Luca and S. Varricchio. Some combinatorial properties of the Thue-Morse sequence and a problem in semigroups. *Theoret. Comput. Sci.* **63** (1989), 333–348.

[7] J. C. Martin. Generalized Morse sequences on n symbols. *Proc. Amer. Math. Soc.* **54** (1976), 379–383.

[8] B. Mossé. Notions de reconnaissabilité pour les substitions et complexité des suites automatiques. Technical Report 93-21, Laboratoire de Mathématiques Discrètes, Université de Marseille, 1993.

[9] J. Mouline. *Contribution à l'étude de la complexité des suites substitutives.* PhD thesis, Université de Provence, France, 1990.