

# Gapped Binomial Complexities in Sequences

Michel Rigo, Manon Stipulanti, and Markus A. Whiteland  
 Department of Mathematics, University of Liège, 4000 Liège, Belgium  
 {m.rigo, m.stipulanti, mwhiteland}@uliege.be

**Abstract**—We relate the gapped  $k$ -deck problem introduced by Golm et al. (ISIT 2022) to notions arising in the literature of combinatorics on words. We consider the complexity functions of infinite sequences that count the number of factors up to the equivalence relation of strings having equal gapped  $k$ -decks. We show that the Thue–Morse sequence, the fixed point of the substitution  $0 \mapsto 01, 1 \mapsto 10$ , has unbounded 1-gap  $k$ -binomial complexity for  $k \geq 2$ . We also show that for a Sturmian sequence and  $g \geq 1$ , all of its long enough factors are always pairwise  $g$ -gap  $k$ -binomially inequivalent for any  $k \geq 2$ .

## I. INTRODUCTION

Many situations in coding theory, bio-informatics, formal verification, and number theory are modeled by an infinite sequence  $s : \mathbb{N} \rightarrow A$  taking values in a finite alphabet  $A$ . It is also often relevant to gather information about subsequences or patterns occurring in  $s$ . In combinatorics on words, several counting functions (factor, abelian, pattern complexities) have been successfully introduced to provide insight and information on the combinatorial structure or to capture a measure of complexity of the sequence of interest [1]–[6].

In this paper, we consider a complexity function based on *gapped binomial coefficients* of strings introduced in Golm et al. [7], which are defined as follows. Let  $g \in \mathbb{N}$  and  $x = x_1 \cdots x_\ell$  and  $u$  be two strings over  $A$ . The  $g$ -gapped binomial coefficient of  $x$  and  $u$  is defined by

$$\begin{aligned} \left[ \begin{matrix} x \\ u \end{matrix} \right]_g &:= \#\{i_1 < \cdots < i_{|u|} \mid x_{i_1} \cdots x_{i_{|u|}} = u \text{ and} \\ &\quad i_{j+1} - i_j > g, 1 \leq j < |u|\}, \end{aligned}$$

where  $|u|$  denotes the length of  $u$ . So we count the number of substring occurrences of  $u$  in  $x$  with letters spaced by at least  $g$  elements. This is quite a natural concept to consider in applications such as the *just-in-time scheduling problem* where the task is to construct a string with prescribed properties such as given frequencies of the letters (which encode the different items on a production line subject to some physical constraints and ideal production rates) [8], [9]. In arithmetics, gapped binomial coefficients may also be related to some well-studied integer sequences. For a letter  $a$ ,  $\left[ \begin{matrix} a^n \\ a^k \end{matrix} \right]_1$  is the number of subsets of  $\{1, 2, \dots, n\}$  of size  $k \leq n$  and containing no consecutive integers. This sequence has entry A011973 in the celebrated Sloane’s On-Line Encyclopedia of Integer Sequences (see also its sibling A102547). For larger gaps,

we have  $\left[ \begin{matrix} a^n \\ a^k \end{matrix} \right]_g = \binom{n-g(k-1)}{k}$  where on the right-hand side is the usual binomial coefficient of integers.

The aforementioned work of Golm et al. considered the following generalized *string reconstruction problem*: when does the knowledge of  $\left[ \begin{matrix} x \\ u \end{matrix} \right]_g$  for all strings  $u$  of length at most  $k$  uniquely determine the string  $x$ ? Otherwise stated, what is the minimal length such that two distinct strings share the same gapped binomial coefficients of substrings of length at most  $k$ ? As pointed out by Golm et al. [7], such knowledge is of interest in molecular random-access storage systems for which gaps in readouts arise due to skipping effects with nanopore sequencers [10], [11]. Similar types of reconstruction problems have received much attention [7], [12]–[15].

### A. Our contributions

We bring tools and results from combinatorics on words that could be useful to information theorists. In Section II, we define  $(g, k)$ -binomially equivalent strings and then the  $(g, k)$ -binomial complexity function of infinite sequences. A better understanding of such a function may give insights on the language  $\mathcal{L}(\mathbf{x})$  of an infinite sequence  $\mathbf{x}$ , i.e., the set of finite strings (*factors*) made of contiguous letters occurring in  $\mathbf{x}$ , for which the reconstruction problem could be solved. The aim is thus to restrict the reconstruction problem to the language of an infinite sequence having a  $(g, k)$ -binomial complexity function of the same order as its factor complexity function.

In this paper, we characterize the  $(1, 2)$ -binomial equivalence of strings over a 2-letter alphabet. We observe that Lemma 2.1 allows us to connect two related notions from combinatorics on words, namely the 2-binomial and 2-abelian equivalences [16]. In Section III we briefly discuss the fact that  $(g, k)$ -binomial equivalence may be tested in polynomial time. Then we consider the ubiquitous Thue–Morse sequence [17]–[21] and Sturmian sequences. Indeed, the authors of [7] already considered a variation of the Thue–Morse sequence where gaps are produced by conveniently inserting zeroes. In Section IV we show that its  $(1, k)$ -binomial complexity function is unbounded for all  $k \geq 2$ . This result is in striking contrast with the fact that the usual (0-gap)  $k$ -binomial complexity function remains bounded [22]. Nevertheless, we conjecture that its behavior exhibits some regularities linked to base-2 expansions of integers and shows a fractal self-similar structure (see Conjecture 4.3 and Fig. 1). Sturmian sequences appear in symbolic dynamics as coding of irrational rotations or in discrete geometry as coding of straight lines [23]. We study their gapped binomial complexity functions in Section V. Balancedness of these sequences have, for instance,

All the authors contributed equally to the study. M. Stipulanti and M. Whiteland are respectively supported by the FNRS Research grant 1.B.397.20F and the FNRS Research grant 1.B.466.21F.

applications in routing arriving jobs to parallel queues [24]. Our result shows again some robustness of the factors of a Sturmian sequence: any two distinct such strings of length  $\geq 3$  are never 1-gap  $k$ -binomially equivalent for any  $k \geq 2$ . The result also holds for larger gaps and long enough factors.

## II. PRELIMINARIES

For  $k \geq 1$  set  $A^{\leq k} := \bigcup_{1 \leq j \leq k} A^j$ . For strings  $u, v \in A^*$ , we let  $|u|_v$  denote the number of *factor* occurrences of  $v$  in  $u$ . We let  $\text{prf}_\ell(u)$  (resp.,  $\text{suf}_\ell(u)$ ) denote the length- $\ell$  prefix (resp., suffix) of  $u$ . We use  $\binom{u}{v}$  in place of  $\binom{u}{v}_0$  to distinguish the 0-gapped coefficients from positive gapped coefficients.

Let  $g, k \in \mathbb{N}$  with  $g \geq 0, k \geq 1$ . Two strings  $x, y \in A^*$  are  $(g, k)$ -binomially equivalent, written  $x \sim_k^{(g)} y$ , if

$$\binom{x}{u}_g = \binom{y}{u}_g \quad \forall u \in A^{\leq k}. \quad (1)$$

The  $(0, k)$ -binomial equivalence relation is the  $k$ -binomial equivalence relation studied in [22], and we use  $\sim_k$  to denote the relation. Observe that for any  $g \in \mathbb{N}$ ,  $\sim_1^{(g)}$  is just the abelian equivalence relation  $\sim_1$ , relating two words iff they are permutations of one another. For instance,  $x = 100110$  and  $y = 011001$  are  $(1, 2)$ -binomially equivalent: they are abelian equivalent and the  $(1, 2)$ -binomial coefficients are seen as  $\binom{x}{00}_1 = 2, \binom{x}{01}_1 = 3, \binom{x}{10}_1 = 3, \text{ and } \binom{x}{11}_1 = 2$ . Also, as observed in [7], we have  $01110 \not\sim_2^{(1)} 10001$  even though the  $(1, 2)$ -binomial coefficients are equal; indeed, the strings are not abelian equivalent. Contrary to the case  $g = 0$  [22],  $\sim_k^{(g)}$  is not a congruence for  $g \geq 1, k \geq 2$ . For example,  $xy \not\sim_2^{(1)} xx$  because  $\binom{xy}{00}_1 = \binom{xy}{00}_1 + 1$ .

For a string  $x$  of length at least  $k$ , summing up gapped binomial coefficients over all substrings of length  $k$  gives

$$\sum_{u \in A^k} \binom{x}{u}_g = \binom{a^{|x|}}{a^k}_g = \binom{|x| - g(k-1)}{k}. \quad (2)$$

Hence if  $|x| < |u| + g(|v| - 1)$ , then  $\binom{x}{u}_g = 0$ . We note that  $\binom{x}{aa} = \binom{|x|_a}{2}$  for any letter  $a$ . We deduce, using (2) with  $g = 0$  and  $k = 2$ , that we have, for any  $x, y \in \{0, 1\}^*$  with  $x \sim_1 y$ ,

$$\binom{x}{aa} = \binom{y}{aa} \text{ and } \binom{x}{01} + \binom{x}{10} = \binom{y}{01} + \binom{y}{10}, \quad (3)$$

where  $a \in \{0, 1\}$ . From (1) we have  $x \sim_{k+1}^{(g)} y$  implies that  $x \sim_k^{(g)} y$ , so for a fixed  $g$  we have the chain of implications

$$x \sim_1^{(g)} y \Leftarrow x \sim_2^{(g)} y \Leftarrow \dots \Leftarrow x \sim_k^{(g)} y \Leftarrow \dots \quad (4)$$

### A. Complexity functions

For an infinite sequence  $\mathbf{x}$  over an alphabet  $A$ , we let  $\mathcal{L}(\mathbf{x})$  denote the set of its factors and  $\mathcal{L}_n(\mathbf{x})$  denote  $\mathcal{L}(\mathbf{x}) \cap A^n$ , that is, the set of length- $n$  factors of  $\mathbf{x}$ . The usual *factor complexity function*  $p_{\mathbf{x}}: \mathbb{N} \rightarrow \mathbb{N}$  counts the number  $\#\mathcal{L}_n(\mathbf{x})$  of strings of length  $n$  occurring in  $\mathbf{x}$ . It was first introduced in [1] and the reader may consult [25, §4] for a comprehensive presentation. For an equivalence relation  $\sim$ , we consider the quotient of the language  $\mathcal{L}(\mathbf{x})$  by  $\sim$  and the corresponding

complexity function maps  $n \in \mathbb{N}$  to  $\#(\mathcal{L}_n(\mathbf{x})/\sim)$ . For example, we let  $b_{\mathbf{x}}^{(g,k)}$  denote the  $(g, k)$ -binomial complexity function with  $b_{\mathbf{x}}^{(g,k)}(n) = \#(\mathcal{L}_n(\mathbf{x})/\sim_k^{(g)})$ . For shorthand, we let  $b_{\mathbf{x}}^{(k)} := b_{\mathbf{x}}^{(0,k)}$ . From (4), it follows that, for a fixed gap  $g$  and for all  $n \in \mathbb{N}$ ,

$$b_{\mathbf{x}}^{(g,1)}(n) \leq b_{\mathbf{x}}^{(g,2)}(n) \leq \dots \leq b_{\mathbf{x}}^{(g,k)}(n) \leq \dots \leq p_{\mathbf{x}}(n). \quad (5)$$

For a brief example, consider the Thue–Morse sequence  $\mathbf{t} = 0110100110010110\dots$ , a fixed point of the morphism  $\varphi: 0 \mapsto 01, 1 \mapsto 10$ . One can check that there are 16 different length-6 factors, hence  $p_{\mathbf{t}}(6) = 16$ . Among these, the only  $\sim_2^{(1)}$ -equivalent pairs are  $010011 \sim_2^{(1)} 001101, 100110 \sim_2^{(1)} 011001$ , and  $110010 \sim_2^{(1)} 101100$ ; hence  $b_{\mathbf{t}}^{(1,2)}(6) = 13$ .

### B. Another equivalence relation and a first characterization

We define a family of equivalence relations that appear in our considerations, introduced in [26] and further studied in [16]. Let  $k \geq 1$  be an integer. Two strings  $u, v$  are  $k$ -abelian equivalent, written  $u \equiv_k v$ , if  $|u|_w = |v|_w$  for each string  $w$  of length at most  $k$ . Observe that  $\equiv_k$  implies  $\sim_1$  for all  $k \geq 1$ .

Note that if  $h$  is an integer with  $g < h$ , then  $\binom{u}{v}_g \geq \binom{u}{v}_h$ . More precisely, for any string  $u$  and any letters  $a, b \in A$ ,

$$\binom{u}{ab}_1 = \binom{u}{ab} - |u|_{ab} \text{ and } \binom{u}{ab}_{g+1} = \binom{u}{ab}_g - \sum_{x \in A^g} |u|_{axb}. \quad (6)$$

Hence by induction, we have

$$\binom{u}{ab}_{g+1} = \binom{u}{ab} - \sum_{x \in A^{\leq g}} |u|_{axb}. \quad (7)$$

As a consequence of the above, if  $u \sim_2 v$  and  $u \equiv_{g+1} v$ , then  $u \sim_2^{(g)} v$ . The following lemma characterizes the  $(1, 2)$ -binomial equivalence over binary strings. For any letter  $a \in \{0, 1\}$ , we let  $\bar{a}$  denote its *complement* letter, i.e.,  $\bar{a} = 1 - a$ .

*Lemma 2.1:* For  $u, v \in \{0, 1\}^*$ , we have  $u \sim_2^{(1)} v$  iff

- (i)  $u \sim_2 v$  and  $u \equiv_2 v$ ; or
- (ii)  $u \sim_1 v$  and there exists  $a \in \{0, 1\}$  such that  $u = a\bar{u}'\bar{a}$ ,  $v = \bar{a}v'a$ , and  $\binom{u}{a\bar{a}} - \binom{v}{a\bar{a}} = |u|_{a\bar{a}} - |v|_{a\bar{a}} = 1$ .

*Proof:* We first show that (i) or (ii) implies  $u \sim_2^{(1)} v$ . If  $u$  and  $v$  satisfy (i), then  $u \sim_2^{(1)} v$  follows from (6). Assume then that  $u$  and  $v$  satisfy (ii). Since  $u \sim_1 v$ , it suffices to show that the four differences  $\binom{u}{ab}_1 - \binom{v}{ab}_1, a, b \in \{0, 1\}$ , vanish to find  $u \sim_2^{(1)} v$ . First notice that by (6) and the assumption,

$$\binom{u}{a\bar{a}}_1 - \binom{v}{a\bar{a}}_1 = \binom{u}{a\bar{a}} - \binom{v}{a\bar{a}} - |u|_{a\bar{a}} + |v|_{a\bar{a}} = 1 - 1 = 0.$$

Using (3) we get  $\binom{u}{a\bar{a}}_1 - \binom{v}{a\bar{a}}_1 = |v|_{aa} - |u|_{aa}$  from (6). We claim that these quantities vanish. To this end, note that  $|x|_{a\bar{a}} + |x|_{aa} = |x|_a - |\text{suf}_1(x)|_a$  for any binary string  $x$ . This fact and the assumptions in (ii) lead to the desired calculation

$$|u|_{aa} - |v|_{aa} = |u|_{aa} - |v|_{aa} + |u|_{a\bar{a}} - |v|_{a\bar{a}} - 1 = 0. \quad (8)$$

Consider next  $\binom{u}{\bar{a}a}_1$ ; now  $\binom{u}{\bar{a}a} - \binom{v}{\bar{a}a} = \binom{v}{\bar{a}a} - \binom{u}{\bar{a}a} = -1$  from (3) which, together with (6), gives

$$\binom{u}{\bar{a}a}_1 - \binom{v}{\bar{a}a}_1 = |v|_{\bar{a}a} - |u|_{\bar{a}a} - 1. \quad (9)$$



## V. COMPLEXITIES OF STURMIAN SEQUENCES

Sturmian sequences are those infinite sequences  $\mathbf{s}$ , for which the factor complexity function  $p_{\mathbf{s}}(n) = n + 1$  for all  $n \geq 0$ . See [31, §2] for a comprehensive introduction of their theory. They are also characterized as those aperiodic binary sequences  $\mathbf{s}$  that are *balanced*, i.e., for all  $u, v \in \mathcal{L}(\mathbf{s})$ ,  $\|u\|_1 - \|v\|_1 \leq 1$  whenever  $|u| = |v|$ . The archetypical Sturmian sequence is the *Fibonacci word*  $\mathbf{f} = 0100101\dots$ , i.e., the fixed point of the morphism  $0 \mapsto 01, 1 \mapsto 0$ . For a Sturmian sequence  $\mathbf{s}$ , we have  $b_{\mathbf{s}}^{(k)} = p_{\mathbf{s}}$  for all  $k \geq 2$  [22, Thm. 7], while  $b_{\mathbf{s}}^{(1)}(n) = 2$  for all  $n \geq 1$ . In fact, Sturmian sequences are characterized by the property that, for some  $k \geq 2$ ,  $b_{\mathbf{s}}^{(k)} = p_{\mathbf{s}}$  [30]. The main result we show is the following.

*Theorem 5.1:* For any Sturmian sequence  $\mathbf{s}$  and any  $k \geq 2$ ,  $b_{\mathbf{s}}^{(1,k)}(2) = 2$  and  $b_{\mathbf{s}}^{(1,k)}(n) = n + 1$  otherwise.

From (5),  $b_{\mathbf{s}}^{(1,2)}(n) \leq b_{\mathbf{s}}^{(1,k)}(n) \leq p_{\mathbf{s}}(n) = n + 1$ , so it is enough to prove the claim for  $k = 2$ .

### A. Preparatory results on Sturmian sequences

Let  $\mathbf{s}$  be a Sturmian sequence that contains  $00$ ; thus any occurrence of  $1$  is isolated. There then exists  $k \in \mathbb{N}$  such that each  $1$  is followed by  $0^k 1$  or  $0^{k+1} 1$ . Letting  $\mu: 0 \mapsto 0^k 1, 1 \mapsto 0^{k+1} 1$ , there is a sequence  $\mathbf{s}'$  such that  $\mu(\mathbf{s}') = 0^j \mathbf{s}$  for some  $j \leq k+1$ . A remarkable result tells us that  $\mathbf{s}'$  is also a Sturmian sequence (see, e.g., [23, §2]). It follows that any factor  $x$  of  $\mathbf{s}$  can be written in the form  $0^r 10^{k+\epsilon_0} 1 \dots 0^{k+\epsilon_{m-1}} 10^s$ , where  $r, s \leq k+1$ , and  $\epsilon = \epsilon_0 \dots \epsilon_{m-1}$  is a factor of  $\mathbf{s}'$ . See [22, Cor. 9] for details. For such a binary string  $\epsilon$ , we define for each  $0 \leq \ell < m$  the quantities

$$S_{\epsilon}(\ell) := \sum_{i=0}^{\ell} (m-i)\epsilon_i \quad \text{and} \quad S_{\epsilon} := S_{\epsilon}(m-1). \quad (10)$$

Then for a string  $x = 0^r 10^{k+\epsilon_0} 1 \dots 0^{k+\epsilon_{m-1}} 10^s$ , we find

$$\binom{x}{01} = r(m+1) + S_{\epsilon} + k \frac{m(m+1)}{2}. \quad (11)$$

(See [22, Rem. 4] for details.)

For two binary strings  $x$  and  $y$  of equal length, we let  $\Delta_{x,y}(i) := |\text{prf}_i(x)|_1 - |\text{prf}_i(y)|_1$  for each  $1 \leq i \leq |x|$ . For the sake of conciseness, when the strings  $x$  and  $y$  are clear from the context, we write  $\Delta(i)$  instead.

*Lemma 5.2:* If  $x$  and  $y$  are length- $m$  factors of a Sturmian sequence, then  $|\Delta_{x,y}(i)| \leq 1$  for each  $i = 1, \dots, m$ . Furthermore, all non-zero  $\Delta_{x,y}(i)$  have the same sign.

*Proof:* If  $x \neq y$ , there is a minimal  $i \leq m$  such that  $x_i \neq y_i$ . Without loss of generality, we may assume that  $\Delta(i) = 1$ . Then  $\Delta(j) \geq 0$  for all  $j \geq i$ , as otherwise

$$|y[i+1, j]|_1 - |x[i+1, j]|_1 \geq 2,$$

contradicting the balancedness of the Sturmian sequence. By symmetric arguments,  $\Delta(j) \geq 0$  for all  $j \leq i$  as well. ■

The next lemma is key in our forthcoming arguments.

*Lemma 5.3:* Let  $\epsilon$  and  $\epsilon'$  be distinct length- $m$  factors of a Sturmian sequence. Then we have  $0 < |S_{\epsilon} - S_{\epsilon'}| \leq m$ . If moreover  $\epsilon \sim_1 \epsilon'$ , then  $0 < |S_{\epsilon} - S_{\epsilon'}| < m$ .

*Proof:* The first part of the statement is shown within the proof of [22, Lem. 10]. We give a sketch of the proof here, as a careful analysis allows also to conclude the second claim.

Assume without loss of generality that  $S_{\epsilon} - S_{\epsilon'} \geq 0$ . Hence the minimal index at which the strings differ is positive (i.e.,  $\epsilon$  has a  $1$  while  $\epsilon'$  has a  $0$ ). It can be shown that, if  $i_0, \dots, i_t$ , with  $t \geq 0$ , are the indices in  $\{0, \dots, m-1\}$  at which the strings  $\epsilon = \epsilon_0 \dots \epsilon_{m-1}$  and  $\epsilon' = \epsilon'_0 \dots \epsilon'_{m-1}$  disagree, then

$$S_{\epsilon} - S_{\epsilon'} = \sum_{r=0}^{\lceil t/2 \rceil - 1} (i_{2r+1} - i_{2r}) + \begin{cases} m - i_t & \text{if } t \text{ is even} \\ 0 & \text{if } t \text{ is odd,} \end{cases} \quad (12)$$

where we understand the first sum as empty if  $t = 0$ . Using the bounds  $1 \leq i_{2r+1} - i_{2r} < i_{2r+2} - i_{2r}$ , one can show that  $\lceil t/2 \rceil < S_{\epsilon} - S_{\epsilon'} \leq m - i_0 - \lceil t/2 \rceil$ , and the first claim follows immediately. For the second claim, we notice that  $\epsilon \sim_1 \epsilon'$  implies that  $t$  must be odd. Hence the above bounds show that  $S_{\epsilon} - S_{\epsilon'} < m$ . ■

### B. Proof of the (1, 2)-binomial complexity

*Proof of Theorem 5.1:* Without loss of generality, we may assume that  $00$  appears in the Sturmian sequence  $\mathbf{s}$  but  $11$  does not. The claim holds for  $n \leq 2$  by straightforward inspection (and observing that  $01 \sim_2^{(1)} 10$ ). Let then  $n \geq 3$  and let  $u, v$  be distinct length- $n$  factors of  $\mathbf{s}$ . Assume, towards a contradiction, that  $u \sim_2^{(1)} v$ . By [22, Thm. 7],  $u \not\sim_2 v$ . Therefore, Lemma 2.1 implies that  $u \sim_1 v$  and, without loss of generality,  $u = 0u'1$  and  $v = 1v'0$ . Notice that  $u$  must contain at least two  $1$ 's; otherwise  $\begin{bmatrix} u \\ 01 \end{bmatrix}_1 = 0 \neq \begin{bmatrix} v \\ 01 \end{bmatrix}_1$ . Write

$$u = 0^r 10^{k+\epsilon_0} 1 \dots 0^{k+\epsilon_{m-1}} 1; \quad v = 10^{k+\epsilon'_0} 1 \dots 0^{k+\epsilon'_{m-1}} 10^s$$

with  $r, s > 0$ ,  $\epsilon = \epsilon_0 \dots \epsilon_{m-1}$ , and  $\epsilon' = \epsilon'_0 \dots \epsilon'_{m-1}$ . By Lemma 2.1(ii) on the one hand and (11) on the other, we find

$$1 = \binom{u}{01} - \binom{v}{01} = r(m+1) + S_{\epsilon} - S_{\epsilon'}. \quad (13)$$

By Lemma 5.3, we have  $0 < |S_{\epsilon} - S_{\epsilon'}| \leq m$ , so we conclude that  $r = 1$  and  $S_{\epsilon} - S_{\epsilon'} = -m$ . It can be similarly shown that also  $s = 1$ .

Now we show  $\epsilon \sim_1 \epsilon'$ . Indeed, (6) together with  $u \sim_2^{(1)} v$  and  $u \sim_1 v$  yield  $0 = \binom{u}{00} - \binom{v}{00} = |u|_{00} - |v|_{00}$ . Recalling  $r = s = 1$ , by straightforward counting we get

$$|u|_{00} = m(k-1) + \sum_{i=0}^{m-1} \epsilon_i \quad \text{and} \quad |v|_{00} = m(k-1) + \sum_{i=0}^{m-1} \epsilon'_i.$$

Since these quantities are equal, we conclude that  $\sum_{i=0}^{m-1} \epsilon_i = \sum_{i=0}^{m-1} \epsilon'_i$ , i.e.,  $\epsilon \sim_1 \epsilon'$  as desired.

The second part of Lemma 5.3 implies  $S_{\epsilon} - S_{\epsilon'} > -m$ , however, contrary to the assertion that  $S_{\epsilon} - S_{\epsilon'} = -m$ . This contradiction suffices for the proof of the theorem. ■

### C. On larger gaps in Sturmian sequences

Let us consider larger gaps in Sturmian sequences. Take  $g \geq 2$ . For factors of length at most  $g+1$ , the  $(g, 2)$ -binomial equivalence is determined by abelian equivalence, and hence the  $(g, 2)$ -complexity of a Sturmian sequence is constant 2

up to length  $g + 1$ . At length  $g + 2$ , we see that the  $(g, 2)$ -binomial equivalence is determined by abelian equivalence together with the first and last letters (which determine exactly one occurrence of a  $(g, 2)$ -binomial coefficient). This implies that at length  $g + 2 \geq 4$ , the  $(1, 2)$ -complexity coincides with the 2-abelian complexity, which in turn is known to be 4 in a Sturmian sequence (cf. [16]). For longer factors, the situation becomes slightly more complicated. We can show that there exist distinct length- $2g$  factors  $u, v$  of a Sturmian sequence  $s$  for which  $u \sim_2^{(g)} v$ . Indeed, there exist arbitrarily long factors  $z$  such that  $z01z$  and  $z10z$  appear in  $s$  (cf. [16]). Letting  $|z| \geq g - 1$  and setting  $x = \text{prf}_{g-1}(z)$ ,  $y = \text{suf}_{g-1}(z)$ , the factors  $y01x$  and  $y10x$  are  $(1, 2)$ -binomially equivalent: the central letters 01 (or 10) do not appear as part of a  $g$ -gapped substring, and the strings are otherwise equal.

We have the following result pertaining to long factors.

**Theorem 5.4:** Let  $g, k \geq 2$ . For a Sturmian sequence  $s$  which does not contain 11,  $b_s^{(g,k)}(n) = p_s(n) = n + 1$  for all  $n$  such that any length- $n$  factor of  $s$  contains more than  $g$  occurrences of 1.

Again it is sufficient to prove the claim for  $k = 2$  for any  $g$ . It is a well-known fact that for a length- $n$  factor of a Sturmian sequence of slope  $\alpha$  such that  $\lfloor \alpha n \rfloor > g$ , its interpretation as a mechanical sequence intersects the horizontal lines of the unit grid more than  $g$  times and thus contains more than  $g$  1's. Hence, the above theorem can be reformulated as: *For a Sturmian sequence  $s_\alpha$  of slope  $\alpha < 1/2$ ,  $b_{s_\alpha}^{(g,k)}(n) = p_s(n)$  for all  $n$  such that  $\lfloor \alpha n \rfloor > g$ .*

*Proof sketch of Theorem 5.4:* Towards a contradiction, assume that there exist two distinct such factors  $u, v$  of  $s$  such that  $u \sim_2^{(g)} v$ . Hence, for all  $a, b \in \{0, 1\}$  we have  $\binom{u}{ab} - \binom{v}{ab} = \sum_{x \in A <_g} |u|_{axb} - |v|_{axb}$  by (7). When  $a = b$  we have  $\binom{u}{ab} - \binom{v}{ab} = 0$  by (3). Therefore, we can add  $0 = \sum_{x \in A <_g} |u|_{1x1} - |v|_{1x1}$  to  $\binom{u}{01} - \binom{v}{01}$  without changing its value (which is positive without loss of generality, and not vanishing, as otherwise  $u = v$  by [22, Thm. 7]). Hence we find that

$$\begin{aligned} \binom{u}{01} - \binom{v}{01} &= \sum_{x \in A <_g} |u|_{0x1} - |v|_{0x1} + \sum_{x \in A <_g} |u|_{1x1} - |v|_{1x1} \\ &= \sum_{1 \leq |y| \leq g} |u|_{y1} - |v|_{y1} = \sum_{i=1}^g \Delta_{v,u}(i) \leq g, \end{aligned}$$

where in the last equality we use the facts that  $u \sim_1 v$  and  $\sum_{|y|=i} |v|_{y1} = |v|_1 - |\text{prf}_i(v)|_1$ . The bound  $g$  follows from  $u$  and  $v$  being factors of  $s$ , so we may use Lemma 5.2. By a similar argument, we also find  $\binom{u}{01} - \binom{v}{01} = \sum_{i=1}^g \Delta_{\tilde{u},\tilde{v}}(i)$  (where  $\tilde{x}$  is the reversal of the string  $x$ ).

We thus have three formulas to count  $\binom{u}{01} - \binom{v}{01}$ ; the above two and one obtained by applying (11) to  $u$  and  $v$  (with suitable factorizations):  $\binom{u}{01} - \binom{v}{01} = (r - r')(m + 1) + S_\epsilon - S_{\epsilon'}$ , where  $m + 1 > g$ . By carefully examining  $\sum_{i=1}^g \Delta_{v,u}(i)$ , we can express its value as a sum similar to (12). To have equality between  $\sum_{i=1}^g \Delta_{v,u}(i)$  and the aforementioned sum, we conclude that  $i_t$ , the last position where  $\epsilon$  and  $\epsilon'$  differ, must appear within the first  $m/2$  letters of  $\epsilon$  and  $\epsilon'$  due to the

assumption on  $g < m + 1$ . This means that we cannot see a difference in the last  $g$  letters of  $u$  and  $v$ , which renders  $\sum_{i=1}^g \Delta_{\tilde{u},\tilde{v}}(i) = 0$  contrary to it being positive.  $\blacksquare$

## VI. CONCLUSIONS

As stated in the introduction, the concept of  $(g, k)$ -binomial equivalence has some potential applications in information theory while, as we have shown in this article, it also allows to make links between several notions arising in combinatorics on words. From a theoretical point of view, gapped binomial coefficients and gapped binomial complexity functions open the way to study new concepts such as Pascal-like triangles, applications in formal language theory (generalizing *piecewise testable* languages or *Simon congruence* [32] by introducing a gapped support), links to Parikh matrices, etc. Another interesting direction is to efficiently generate the strings of an equivalence class. Inspired by [33], [34], we search for string manipulations preserving the  $(1, 2)$ -binomial equivalence. To this end, it is straightforward to check the following assertions.

**Lemma 6.1:** For binary strings  $u, v, x, y, z \in \{0, 1\}^*$  and  $a, b \in \{0, 1\}$ , we have

- 1)  $xa10bya01bz \sim_2^{(1)} xa01bya10bz$ ;
- 2)  $xa10a01ay \sim_2^{(1)} xa01a10ay$ ;
- 3)  $xa10ay\bar{a}01\bar{a}z \sim_2^{(1)} xa01ay\bar{a}10\bar{a}z$ .

**Proposition 6.2:** Let  $u, v, x, y \in \{0, 1\}^*$  and  $a \in \{0, 1\}$ . If  $1u1 \sim_2 0v0$ ,  $|1u1|_{00} = |0v0|_{00} + 1$ , and  $|1u1|_{01} = |0v0|_{01}$ , then  $xa1u1\bar{a}y \sim_2^{(1)} xa0v0\bar{a}y$ . In particular,  $xa1001\bar{a}y \sim_2^{(1)} xa0110\bar{a}y$ .

*Proof:* Since  $1u1 \sim_2 0v0$ , the letters  $a$  and  $\bar{a}$  playing the role of a buffer, it is enough to show that  $w = a1u1\bar{a} \sim_2^{(1)} a0v0\bar{a} = z$ . Using (6), either  $a$  or  $\bar{a}$  contributes to  $|u|_0$  substrings 00 occurring in  $w$ , so

$$\left[ \begin{matrix} w \\ 00 \end{matrix} \right]_1 = \binom{w}{00} - |w|_{00} = |u|_0 + \binom{1u1}{00} - |1u1|_{00}$$

and similarly,

$$\left[ \begin{matrix} z \\ 00 \end{matrix} \right]_1 = \binom{z}{00} - |z|_{00} = 2 + |v|_0 + \binom{0v0}{00} - |0v0|_{00} - 1.$$

The two coefficients are equal since  $1u1 \sim_1 0v0$  and thus  $|u|_0 = 2 + |v|_0$ . We may show in the same way that  $\left[ \begin{matrix} w \\ 01 \end{matrix} \right]_1 = \left[ \begin{matrix} z \\ 01 \end{matrix} \right]_1$ . Since  $1u1$  and  $0v0$  start and end with equal letters,  $|1u1|_{10} = |1u1|_{01} = |0v0|_{01} = |0v0|_{10}$  and the same conclusion holds for  $\left[ \begin{matrix} w \\ 10 \end{matrix} \right]_1 = \left[ \begin{matrix} z \\ 10 \end{matrix} \right]_1$ . From (2), there is no need to compute the fourth coefficient for 11.  $\blacksquare$

For instance, the following strings comprise a full  $\sim_2^{(1)}$ -equivalence class and we have underlined the substrings that are manipulated thanks to the above results to highlight equivalence from one string to the next:

$$\begin{aligned} &0000\underline{1110101}, 0001\underline{0110011}, 000\underline{11001011}, \underline{00101000111}, \\ &01000\underline{100111}, \underline{01000011011}, 00100011101. \end{aligned}$$

## ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their valuable suggestions which greatly helped to clarify the exposition of the results and to improve the quality of the text.

## REFERENCES

- [1] A. Ehrenfeucht, K. P. Lee, and G. Rozenberg, "Subword complexities of various classes of deterministic developmental languages without interactions," *Theoret. Comput. Sci.*, vol. 1, no. 1, pp. 59–75, 1975.
- [2] M. Rigo, "Relations on words," *Indag. Math. (N.S.)*, vol. 28, no. 1, pp. 183–204, 2017.
- [3] T. Kamae and L. Zamboni, "Sequence entropy and the maximal pattern complexity of infinite words," *Ergodic Theory Dynam. Systems*, vol. 22, no. 4, pp. 1191–1199, 2002.
- [4] J.-P. Allouche and J. Shallit, *Automatic sequences: Theory, applications, generalizations*. Cambridge University Press, Cambridge, 2003.
- [5] L. Schaeffer and J. Shallit, "String attractors for automatic sequences," 2021, (preprint). [Online]. Available: <https://arxiv.org/abs/2012.06840>
- [6] A. Restivo, G. Romana, and M. Sciortino, "String attractors and infinite words," in *LATIN 2022: Theoretical informatics*, ser. Lecture Notes in Comput. Sci. Springer, Cham, 2022, vol. 13568, pp. 426–442.
- [7] R. Golm, M. Nahvi, R. Gabrys, and O. Milenkovic, "The gapped k-deck problem," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 49–54.
- [8] N. Brauner and Y. Crama, "The maximum deviation just-in-time scheduling problem," *Discrete Appl. Math.*, vol. 134, no. 1-3, pp. 25–50, 2004.
- [9] N. Brauner and V. Jost, "Small deviations, JIT sequencing and symmetric case of Fraenkel's conjecture," *Discrete Math.*, vol. 308, no. 11, pp. 2319–2324, 2008.
- [10] S. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free dna-based data storage," *Sci. Rep.*, vol. 7, no. 1, pp. 1–6, 2017.
- [11] L. C. Meiser, P. L. Antkowiak, J. Koch, W. D. Chen, A. X. Kohll, W. J. Stark, R. Heckel, and R. N. Grass, "Reading and writing digital data in dna," *Nat. Protoc.*, vol. 15, no. 1, pp. 86–101, 2020.
- [12] L. I. Kalašnik, "The reconstruction of a word from fragments," in *Numerical mathematics and computer technology, No. IV (Russian)*. Akad. Nauk Ukrain. SSR Fiz.-Tehn.-Inst. Nizkih Temperatur, Kharkov, 1973, pp. 56–57, 137.
- [13] B. Manvel, A. D. Meyerowitz, A. J. Schwenk, K. W. Smith, and P. K. Stockmeyer, "Reconstruction of sequences," *Discrete Math.*, vol. 94, no. 3, pp. 209–219, 1991.
- [14] P. Fleischmann, M. Lejeune, F. Manea, D. Nowotka, and M. Rigo, "Reconstructing words from right-bounded-block words," *Int. J. Found. Comput. Sci.*, vol. 32, no. 6, pp. 619–640, 2021.
- [15] G. Richomme and M. Rosenfeld, "Reconstructing words using queries on subwords or factors," 2023. [Online]. Available: <https://arxiv.org/abs/2301.01571>
- [16] J. Karhumäki, A. Saarela, and L. Q. Zamboni, "On a generalization of abelian equivalence and complexity of infinite words," *J. Comb. Theory, Ser. A*, vol. 120, no. 8, pp. 2189–2206, 2013.
- [17] J.-P. Allouche and J. Shallit, "The ubiquitous Prouhet–Thue–Morse sequence," in *Sequences and their Applications*, C. Ding, T. Hellesteth, and H. Niederreiter, Eds. London: Springer London, 1999, pp. 1–16.
- [18] J.-P. Allouche, "Thue, combinatorics on words, and conjectures inspired by the Thue–Morse sequence," *J. Théor. Nombres Bordeaux*, vol. 27, no. 2, pp. 375–388, 2015.
- [19] M. Lejeune, J. Leroy, and M. Rigo, "Computing the  $k$ -binomial complexity of the Thue–Morse word," *J. Comb. Theory, Ser. A*, vol. 176, p. 44, 2020.
- [20] A. Parreau, M. Rigo, E. Rowland, and É. Vandomme, "A new approach to the 2-regularity of the  $\ell$ -abelian complexity of 2-automatic sequences," *Electron. J. Comb.*, vol. 22, no. 1, 2015.
- [21] F. Greinecker, "On the 2-abelian complexity of the Thue–Morse word," *Theoretical Computer Science*, vol. 593, pp. 88–105, 2015.
- [22] M. Rigo and P. Salimov, "Another generalization of abelian equivalence: binomial complexity of infinite words," *Theor. Comput. Sci.*, vol. 601, pp. 47–57, 2015.
- [23] M. Lothaire, *Combinatorics on Words*. Cambridge Mathematical Library. Cambridge University Press, 1997.
- [24] A. Hordijk and D. A. van der Laan, "Bounds for deterministic periodic routing sequences," in *Integer programming and combinatorial optimization (Utrecht, 2001)*, ser. Lecture Notes in Comput. Sci. Springer, Berlin, 2001, vol. 2081, pp. 236–250.
- [25] V. Berthé and M. Rigo, Eds., *Combinatorics, automata and number theory*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 2010, vol. 135.
- [26] J. Karhumäki, "Generalized Parikh mappings and homomorphisms," *Inform. and Control*, vol. 47, no. 3, pp. 155–165, 1980. [Online]. Available: [https://doi.org/10.1016/S0019-9958\(80\)90493-3](https://doi.org/10.1016/S0019-9958(80)90493-3)
- [27] D. D. Freydenberger, P. Gawrychowski, J. Karhumäki, F. Manea, and w. Rytter, "Testing  $k$ -binomial equivalence," in *Multidisciplinary Creativity: homage to Gheorghe Paun on his 65th birthday*. Bucharest, Ed. Spandugino, 2015, pp. 239–248.
- [28] W.-G. Tzeng, "A polynomial-time algorithm for the equivalence of probabilistic automata," *SIAM J. Comput.*, vol. 21, no. 2, pp. 216–227, 1992.
- [29] E. Rowland, <https://ericrowland.github.io/packages.html>.
- [30] M. Rigo, M. Stipulanti, and M. A. Whiteland, "Characterizations of families of morphisms and words via binomial complexities," 2022. [Online]. Available: <https://arxiv.org/abs/2201.04603>
- [31] M. Lothaire, *Algebraic combinatorics on words*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 2002, vol. 90. [Online]. Available: <https://doi.org/10.1017/CBO9781107326019>
- [32] I. Simon, "Piecewise testable events," in *Automata Theory and Formal Languages*, H. Brakhage, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1975, pp. 214–222.
- [33] S. Fossé and G. Richomme, "Some characterizations of Parikh matrix equivalent binary words," *Inform. Process. Lett.*, vol. 92, no. 2, pp. 77–82, 2004.
- [34] J. Karhumäki, S. Puzynina, M. Rao, and M. A. Whiteland, "On cardinalities of  $k$ -abelian equivalence classes," *Theoretical Computer Science*, vol. 658, pp. 190–204, 2017.