Templates for the k-binomial complexity of the Tribonacci word

Marie Lejeune[1,2], Michel Rigo[1], and Matthieu Rosenfeld[1]

[1] Department of Mathematics, University of Liège, Allée de la Découverte 12 (B37), B-4000 Liège, Belgium. {M.Lejeune,M.Rigo,M.Rosenfeld}@uliege.be
[2] The first author is supported by a FNRS fellowship.

Abstract. Consider k-binomial equivalence: two finite words are equivalent if they share the same subwords of length at most k with the same multiplicities. With this relation, the k-binomial complexity of an infinite word $\mathbf{x}$ maps the integer n to the number of pairwise non-equivalent factors of length n occurring in $\mathbf{x}$. In this paper based on the notion of template introduced by Currie et al., we show that, for all $k \geq 2$, the k-binomial complexity of the Tribonacci word coincides with its usual factor complexity $p(n) = 2n + 1$. A similar result was already known for Sturmian words, but the proof relies on completely different techniques that seemingly could not be applied for Tribonacci.

1 Introduction

Abelian equivalence of words has been investigated for quite a long time; e.g., in the sixties Erdös raised the question whether abelian squares can be avoided by an infinite word over an alphabet of size 4 [6,7,17]. Let Σ be a finite alphabet. We let Σ^* denote the set of all finite words over Σ. Two words u and v in Σ^* are *abelian equivalent* if one word is obtained by permuting the letters of the other word. More formally, u and v are abelian equivalent if $|u|_a = |v|_a$, for all $a \in \Sigma$, where we let $|u|_a$ denote the number of occurrences of the letter a in u.

Definition 1. *Let u be a word over an ordered alphabet $\{0, \ldots, s-1\}$. The* abelianization *or* Parikh vector *of u, denoted by $\Psi(u)$, is the column vector in $\mathbb{N}^s$*

$$(|u|_0, \ldots, |u|_{s-1})^{\intercal} .$$

With this notation, two words are abelian equivalent if and only if $\Psi(u) = \Psi(v)$. A possible generalization of abelian equivalence is the *k-binomial equivalence* based on binomial coefficient of words. An independent generalization is k-abelian equivalence where one counts factors of length at most k [10,11]. For a survey, see, for instance, [20]. We let the binomial coefficient $\binom{u}{v}$ denote the number of times v appears as a (not necessarily contiguous) subsequence of u. Let $k \geq 1$ be an integer. Two words u and v are *k-binomially equivalent*, denoted $u \sim_k v$, if $\binom{u}{x} = \binom{v}{x}$ for all words x of length at most k.

Definition 2. *Let* $\mathbf{x}$ *be an infinite word. The* k-binomial complexity function *of* $\mathbf{x}$ *is defined as*

$$b_{\mathbf{x},k} : \mathbb{N} \to \mathbb{N},\ n \mapsto \#\left(\mathrm{Fac}_n(\mathbf{x})/\sim_k\right)$$

where $\mathrm{Fac}_n(\mathbf{x})$ *is the set of factors of length* n *occurring in* $\mathbf{x}$*. For* $k = 1$*, this measure of complexity is exactly the abelian complexity.*

The celebrated theorem of Morse–Hedlund [3,4] characterizes ultimately periodic words in terms of a bounded factor complexity function. Hence, aperiodic words with the lowest factor complexity are exactly the Sturmian words characterized by $p_{\mathbf{x}}(n) = n + 1$. It is also a well-known result of Cobham that the factor complexity of any k-automatic sequence is in $\mathcal{O}(n)$. The Tribonacci word has a factor complexity $2n + 1$.

We collect the known facts about the k-binomial complexity. For all $k \geq 2$, Sturmian words have a k-binomial complexity which is the same as their factor complexity, i.e., $b_{\mathbf{x},k}(n) = n + 1$ for all n. Since $b_{\mathbf{x},k}(n) \leq b_{\mathbf{x},k+1}(n)$, the proof consists in showing that any two distinct factors of length n occurring in a given Sturmian word are never 2-binomially equivalent [18, Thm. 7]. However, the Thue–Morse word has a bounded k-binomial complexity [18, Thm. 13]. So we have a striking difference with the most usual complexity measures. Naturally, the bound on the k-binomial complexity of the Thue–Morse word depends on the parameter k because when k tends to infinity, the k-binomial equivalence gets closer to equality of factors, i.e. $b_{\mathbf{x},k}(n) = p_{\mathbf{x},k}(n)$ for all $n \leq k$, and the Thue–Morse word has a factor complexity in $\Theta(n)$. The precise results are recalled below.

Theorem 3. *[18] Let* $k \geq 1$*. There exists* $C_k > 0$ *such that the* k*-binomial complexity of the Thue–Morse word* $\mathbf{t}$ *satisfies* $b_{\mathbf{t},k}(n) \leq C_k$ *for all* $n \geq 0$*.*

Theorem 4. *[12] We let* $\mathbf{t}$ *denote the Thue–Morse word over a* 2*-letter alphabet. Let* k *be a positive integer. For all* $n \leq 2^k - 1$*, we have* $b_{\mathbf{t},k}(n) = p_{\mathbf{t}}(n)$*. For all* $n \geq 2^k$*, we have*

$$b_{\mathbf{t},k}(n) = \begin{cases} 3 \cdot 2^k - 3, & \text{if } n \equiv 0 \pmod{2^k}; \\ 3 \cdot 2^k - 4, & \text{otherwise.} \end{cases}$$

When investigating this new k-binomial complexity measure, it is naturally interesting to consider various well-known words or families of words. A natural choice is therefore to try computing the k-binomial complexity of the Tribonacci word $\mathcal{T} = 010201001020101\cdots$, fixed point of the morphism $\tau : 0 \mapsto 01,\ 1 \mapsto 02,\ 2 \mapsto 0$. From computer experiments, the second author made the conjecture in 2014 that, for all $k \geq 2$, its k-binomial complexity is the same as the usual factor complexity $2n+1$ [19]. As in the Sturmian case, it is enough to show that given any two distinct factors of length n occurring in the Tribonacci word, these two factors are not 2-binomially equivalent. Surprisingly, classical combinatorial techniques seemed to be unsuccessful. We make an extensive use of the concepts of *templates* and their ancestors similar to what can be found in [1,2,8] where avoidance of abelian repetitions is considered. Closely related, let us also mention

[5] where a morphic word avoiding three consecutive factors of the same size and same sum is given. Recently F. Liétard proposed algorithmic proofs for morphic words avoiding additive powers [14].

This paper is organized as follows. In Section 2, we recall the notion of Parikh matrix and extend it to our k-binomial context. In particular, this extended matrix is built from the classical one using the Kronecker product: binomial coefficients can be nicely represented in terms of this product. In Section 3 we define and adapt the notions of templates and ancestors to our purpose. To solve our problem, we need to show the finiteness of some set of realizable ancestors. To that end, in Section 4, we first get several bounds related to Parikh vectors of factors of the Tribonacci word. Consequently, we deduce bounds on the realizable ancestors. We put together the results of these last two sections to establish the main theorem in Section 5. Similarly to [1,2,8,14,16], our proof is a computer-assisted one.

2 Basics

Let $\Sigma = \{0, \ldots, s-1\}$ be an ordered alphabet of size s. As mentioned in the introduction, it is enough to consider 2-binomial equivalence but everything in this section generalizes well to k-binomial equivalence.

Definition 5. *Let w be a finite word over Σ. We will make an extensive use of its* extended Parikh vector *denoted by $\Phi(w)$ and defined as follows. We set*

$$\Phi(w) := \left(|w|_0, \ldots, |w|_{s-1}, \binom{w}{00}, \binom{w}{01}, \ldots, \binom{w}{(s-1)(s-1)} \right)^{\mathsf{T}}.$$

It is a column vector of size $s(s+1)$ and we assume that the s^2 subwords of length 2 are lexicographically ordered.

Take the word $u = 10010201010$ which is a factor of length 11 occurring in the Tribonacci word. Its extended Parikh vector is given by

$$\Phi(u) = \big(6, 4, 1, 15, 11, 3, 13, 6, 2, 3, 2, 0\big)^{\mathsf{T}}.$$

With this notation, $\Phi(u) = \Phi(v)$ if and only if $u \sim_2 v$.

For a vector $\mathbf{d} \in \mathbb{Z}^n$, $n \geq s$, we let $\mathbf{d}|_{0,\ldots,s-1}$ denote the vector in $\mathbb{Z}^s$ made of the first s coordinates of $\mathbf{d}$. In particular over an alphabet of size s, $\Phi(w)|_{0,\ldots,s-1} = \Psi(w)$.

We let $A \otimes B$ denote the usual Kronecker product of two matrices $A \in \mathbb{Z}^{m \times n}$ and $B \in \mathbb{Z}^{p \times q}$. It is a block-matrix in $\mathbb{Z}^{mp \times nq}$ defined by

$$A \otimes B := \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}.$$

Let m be an integer. We let $P_m \in \mathbb{Z}^{(m(m+1))\times m^2}$ denote the matrix such that for all, i, j,

$$[P_m]_{i,j} = \begin{cases} 1, & \text{if } i = j + m; \\ 0, & \text{otherwise.} \end{cases}$$

This matrix adds m zeros at the beginning of a column vector of size m^2. We start with two straightforward lemmas and the introduction of an extended Parikh matrix.

Lemma 6. *Let u and v be two words over an alphabet of size s. We have*

$$\Phi(uv) = \Phi(u) + \Phi(v) + P_s\big(\Psi(u) \otimes \Psi(v)\big).$$

Proof. The first two terms in the statement take into account the separate contributions of u and v to the different coefficients. Nevertheless, subwords of length 2 can also be obtained by taking their first letter in u and their second one in v. This is exactly the contribution of the third term. Observe that $\Psi(u) \otimes \Psi(v)$ is a column vector of size s^2. Applying P_s will add s zeros on top because the contribution of individual letters has already been taken into account in the first two terms. □

The classical *Parikh matrix* M'_h associated with a morphism h is a useful tool in combinatorics on words (not to be confused with the notion of Parikh matrix of a word introduced in 2000 by Mateescu et al.). Over an ordered s-letter alphabet, it is defined from its columns as a $s \times s$ matrix

$$M'_h = \big(\Psi(h(0)) \cdots \Psi(h(s-1))\big)$$

and it readily satisfies

$$\Psi(h(u)) = M'_h \Psi(u), \quad \forall u \in \Sigma^* .$$

For the Tribonacci morphism, it is given by

$$M'_\tau = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} . \tag{1}$$

Definition 7. *Mimicking the Parikh matrix and its use, one can define an extended Parikh matrix M_h associated with a morphism h defined over an ordered s-letter alphabet. It is a $s(s+1) \times s(s+1)$ matrix satisfying*

$$\Phi(h(u)) = M_h \Phi(u), \quad \forall u \in \Sigma^* . \tag{2}$$

The existence of the extended Parikh matrix satisfying (2) is ensured by the next result.

Lemma 8. *Let M'_h be the Parikh matrix associated with some morphism h. The extended Parikh matrix of h has the following form:*

$$M_h = \begin{pmatrix} M'_h & 0 & \cdots & 0 \\ \star & & & \\ \star & & M'_h \otimes M'_h & \\ \star & & & \end{pmatrix}.$$

In particular, $\det(M_h) = \det(M'_h)^{2s+1}$. *Moreover, if the alphabet is of size s, then $M_h P_s = P_s(M'_h \otimes M'_h)$. If M'_h is non-singular, then M_h is non-singular and M_h^{-1} is a block-triangular matrix of the same form as M_h with diagonal blocks $M_h'^{-1}$ and $M_h'^{-1} \otimes M_h'^{-1}$.*

Proof. Since the first s components of $\Phi(u)$ give the usual Parikh vector, M'_h is the upper-left corner of M_h. For the last s^2 components of $\Phi(u)$ dealing with binomial coefficients of subwords of length 2, it is shown in [12] that, for all $a, b \in \Sigma$,

$$\binom{h(u)}{ab} = \sum_{c \in \Sigma} \binom{h(c)}{ab} |u|_c + \sum_{x_1 x_2 \in \Sigma^2} \binom{h(x_1)}{a} \binom{h(x_2)}{b} \binom{u}{x_1 x_2}.$$

In this expression, the first sum corresponds to the $s \times s$ submatrices marked as $\star$ and the second sum exactly corresponds to the Kronecker product $M'_h \otimes M'_h$. Indeed, if we index M'_h on Σ and $M'_h \otimes M'_h$ on Σ^2, we have

$$\binom{h(x_1)}{a} \binom{h(x_2)}{b} = [M'_h]_{a,x_1} \, [M'_h]_{b,x_2} = [M'_h \otimes M'_h]_{ab,x_1x_2}. \qquad \square$$

This extended Parikh matrix was also used in [15] (for avoidance problems).

3 Templates and ancestors

For this section, let $h : \Sigma^* \to \Sigma^*$ be any primitive (prolongable) morphism. Let M'_h be its Parikh matrix and M_h be its extended Parikh matrix. We let $s := \#\Sigma$. Recall that a prefix (resp., a suffix) of a word w is *proper* if it is different from w (and thus, possibly empty).

Definition 9. *The* language *of h, denoted by $\mathcal{L}(h)$, is the set of factors of any of its non-empty fixed points (if h is primitive, they all have the same language). The set* $\mathrm{Pref}(h)$ *(resp.,* $\mathrm{Suff}(h)$*) is the set of proper prefixes (resp., proper suffixes) of the words in $\{h(a) \mid a \in \Sigma\}$, e.g., for the Tribonacci morphism,* $\mathrm{Pref}(\tau) = \{\varepsilon, 0\}$ *and* $\mathrm{Suff}(\tau) = \{\varepsilon, 1, 2\}$. *Such a notation can be extended to h^n. If $u \in \mathcal{L}(h)$, there exist a shortest $p_u \in \mathrm{Pref}(h)$, a shortest $s_u \in \mathrm{Suff}(h)$ and $u' \in \mathcal{L}(h)$ such that $h(u') = p_u u s_u$.*

In the following definition, the index **b** (resp., **e**) stands for beginning (resp., end).

Definition 10. *A* template *is a* 5*-tuple of the form* $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ *where* $a_1, a_2 \in \Sigma$, $\mathbf{d} \in \mathbb{Z}^{s(s+1)}$ *and* $\mathbf{D_b}, \mathbf{D_e} \in \mathbb{Z}^s$. *A pair of words* (u, v) *is a* realization *of (or* realizes*) the template* t *if:*

- $\Phi(u) - \Phi(v) = \mathbf{d} + P_s\big(\mathbf{D_b} \otimes \Psi(u) + \Psi(u) \otimes \mathbf{D_e}\big)$,
- *there exist* u' *and* v' *such that* $u = u'a_1$ *and* $v = v'a_2$.

A template t *is* realizable by h *if there is a pair of words in* $\mathcal{L}(h)$ *that realize* t.

Given two factors u and v, the template of the form $[\Phi(u) - \Phi(v), \mathbf{0}, \mathbf{0}, a_1, a_2]$ is obviously realizable by h, where a_1 (resp., a_2) is the last letter of u (resp., v).

Due to the presence of P_s in the above definition, note that if a template is realizable by a pair (u, v), then the corresponding vector $\mathbf{d}$ is such that $\mathbf{d}|_{0,\dots,s-1} = \Psi(u) - \Psi(v)$.

Remark 11. There exist an infinite number of realizable templates. Actually, for any choice of words u, v and vectors $\mathbf{D_b}, \mathbf{D_e}$ in $\mathbb{Z}^s$, there exists a convenient $\mathbf{d} \in \mathbb{Z}^{s(s+1)}$.

Lemma 12. *Let* h *be a primitive morphism. Let* $T := \{[\mathbf{0}, \mathbf{0}, \mathbf{0}, a_1, a_2] : a_1 \neq a_2\}$. *The factorial complexity and the* 2*-binomial complexity of any fixed point of* h *are equal if and only if all templates from* T *are non-realizable by* h.

Proof. The factorial complexity is not the same as the 2-binomial complexity if and only if there exists a pair of factors (u, v) such that $u \neq v$ and $\Phi(u) = \Phi(v)$.

The two words of any realization of an element in T are 2-binomially equivalent and are different since they do not have the same last letter. Thus, if there is a realization of an element of T then the factorial complexity and the 2-binomial complexity are not equal.

Now, for the other direction, suppose that the two complexity functions are not equal: we have a pair of words (u, v) such that $u \neq v$ and $\Phi(u) = \Phi(v)$. Since $u \neq v$ and $|u| = |v|$, there exist $u', v', s \in \Sigma^*$ and $a, b \in \Sigma$ with $a \neq b$ such that $u = u'as$ and $v = v'bs$ (observe that s is the longest common suffix of u and v). Then $\Phi(u'a) = \Phi(v'b)$ so the pair $(u'a, v'b)$ realizes $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$, which belongs to T. □

The idea in the next definition is that any long factor of a fixed point of a morphism must be the image of a shorter factor, up to (short) prefix and suffix. So the relation corresponds to the various relationships among the binomial coefficients that must hold if this is to be the case. For more details, the reader is invited to read the proof of [13, Lemma 15].

Definition 13. *Let* $t' = [\mathbf{d}', \mathbf{D}'_\mathbf{b}, \mathbf{D}'_\mathbf{e}, a'_1, a'_2]$ *and* $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ *be two templates and* h *be a morphism. We say that* t' is a parent by h of t *if there exist* $p_u, p_v \in \mathrm{Pref}(h)$ *and* $s_u, s_v \in \mathrm{Suff}(h)$ *such that:*

- $\mathbf{d}'$ *is given by*

$$M_h\mathbf{d}' = \mathbf{d} + \Phi(p_u s_u) - \Phi(p_v s_v) + P_s\big(\Psi(p_v) \otimes \mathbf{d}|_{0,\dots,s-1} + \mathbf{d}|_{0,\dots,s-1} \otimes \Psi(s_v)\big)$$
$$- P_s\big((\mathbf{D_b} + \Psi(p_u) - \Psi(p_v)) \otimes \Psi(p_u s_u) + \Psi(p_u s_u) \otimes (\mathbf{D_e} + \Psi(s_u) - \Psi(s_v))\big);$$

- *the value of* $\mathbf{D}'_{\mathbf{b}}$ *is given by* $M'_h\mathbf{D}'_{\mathbf{b}} = \mathbf{D}_{\mathbf{b}} + \Psi(p_u) - \Psi(p_v)$*;*
- *the value of* $\mathbf{D}'_{\mathbf{e}}$ *is given by* $M'_h\mathbf{D}'_{\mathbf{e}} = \mathbf{D}_{\mathbf{e}} + \Psi(s_u) - \Psi(s_v)$*;*
- $a_1 s_u$ *is a suffix of* $h(a'_1)$*;*
- $a_2 s_u$ *is a suffix of* $h(a'_2)$*.*

We let $\mathrm{Par}_h(t)$ denote the set of parents by h of t.

Remark 14. Observe that for any given template t, $\mathrm{Par}_h(t)$ is finite and easy to compute as long as M_h and M'_h are non-singular. Indeed, the sets $\mathrm{Pref}(h)$ and $\mathrm{Suff}(h)$ are finite. For the Tribonacci word, $\#\,\mathrm{Pref}(\tau) = 2$, $\#\,\mathrm{Suff}(\tau) = 3$ and thus $\#\,\mathrm{Par}_\tau(t) \leq 36$. At this stage, it is not required for a parent to be realizable.

More interestingly there is a link between preimages of the realization by h of a template and realization by h of the parents of the template. We make that link explicit in the following Lemma.

Lemma 15. *Let* h *be a morphism. Assume that* $det(M'_h) = \pm 1$*. Let* t *be a template,* $u, v, v', u' \in \mathcal{L}(h)$*,* $p_u, p_v \in \mathrm{Pref}(h)$ *and* $s_u, s_v \in \mathrm{Suff}(h)$ *such that:*

- $h(u') = p_u u s_u$ *and* $h(v') = p_v v s_v$*;*
- s_u *is a proper suffix of the image of the last letter of* u'*;*
- s_v *is a proper suffix of the image of the last letter of* v'*;*
- (u, v) *realizes* t*.*

Then there exists a parent t' *of* t *such that* (u', v') *realizes* t'*.*

This motivates the following definitions.

Definition 16. *A template* t' *is an* ancestor by h *(resp.,* realizable ancestor*) of a template* t *if there exists a sequence of* $n \geq 1$ *templates (resp., realizable templates)* $t = t_1, t_2, \ldots, t_n = t'$ *such that for all* $i \in \{1, \ldots, n-1\}$*,* t_{i+1} *is a parent by* h *of* t_i*. For a template* t*, we denote by* $\mathrm{RAnc}_h(t)$ *the set of all the realizable ancestors by* h *of* t*. We may omit "by* h*" when the morphism is clear from the context.*

Let $|h| = \max_{a \in \Sigma} |h(a)|$, usually called the *width* of h. To prove that two different factors of Tribonacci are never 2-binomially equivalent, we will make use of Lemma 12. The next result will help us to show that no template of the set T defined in Lemma 12 is realizable by τ.

Proposition 17. *Let* L *be a positive integer. Let* h *be a primitive morphism and* t_0 *be a template. If there exists a pair of words in* $\mathcal{L}(h)$ *that is a realization of* t_0*, then*

- *either* t_0 *has a realization* $(u, v) \in \mathcal{L}(h) \times \mathcal{L}(h)$ *such that* $\min(|u|, |v|) \leq L$ *or,*
- *there exists a realization* $(u, v) \in \mathcal{L}(h) \times \mathcal{L}(h)$ *of a template* t *of* $\mathrm{RAnc}_h(t_0)$ *with* $L \leq \min(|u|, |v|) \leq |h|L$ *.*

Proof. Let (u, v) be a pair of factors of $\mathcal{L}(h)$ realizing t_0. If $\min(|u|, |v|) \leq L$, there is nothing left to prove. Assume therefore that $\min(|u|, |v|) > L$.

Since v is a factor of $\mathcal{L}(h)$, there are sequences of words $v = v_1, v_2, \ldots, v_n \in \Sigma^*$, $p_1, \ldots, p_{n-1} \in \mathrm{Pref}(h)$ and $s_1, \ldots, s_{n-1} \in \mathrm{Suff}(h)$ such that, for all $i < n$, $h(v_{i+1}) = p_i v_i s_i$ and $L \leq |v_n| \leq |h|L$. Moreover we may force that, for all $i < n$, s_i is a proper suffix of the image of the last letter of v_{i+1}.

Similarly, since u is a factor of $\mathcal{L}(h)$, there are sequences of words $u = u_1$, $u_2, \ldots, u_\ell \in \Sigma^*$ and $p'_1, \ldots, p'_{\ell-1} \in \mathrm{Pref}(h)$ and $s'_1, \ldots, s'_{\ell-1} \in \mathrm{Suff}(h)$ such that, for all $i < \ell$, $h(u_{i+1}) = p'_i u_i s'_i$ and $L \leq |u_\ell| \leq L|h|$.

Let $m = \min(n, \ell)$. We can simply apply Lemma 15 inductively m times. We obtain a template t' which is an ancestor of t_0 and is realized by (u_m, v_m). Since $m = \min(n, \ell)$, $L \leq \min(|u_m|, |v_m|) \leq |h|L$. This concludes the proof. □

4 Bounding realizable templates for the Tribonacci word

Recall that τ denote the Tribonacci morphism and that $\mathcal{T}$ is the Tribonacci word. The matrix M'_τ was given in (1). Since it is primitive, we may use Perron's theorem. Densities of letters $0, 1, 2$ exist and are denoted respectively by α_0, α_1 and α_2. Let $\theta \approx 1.839$ be the Perron eigenvalue of τ. Recall that $\boldsymbol{\alpha} = \begin{pmatrix}\alpha_0 & \alpha_1 & \alpha_2\end{pmatrix}^\intercal$ is an eigenvector of τ associated with θ. Let

$$\Delta = \{(\delta_0, \delta_1) : -1.5 \leq \delta_0, \delta_1, \delta_0 + \delta_1 \leq 1.5\}.$$

4.1 Bounds on extended Parikh vectors

We can obtain two different kinds of bounds on extended Parikh vectors of factors of the Tribonacci word. First we essentially take care of the large eigenvalues.

Proposition 18. *Let $\mathbf{r}$ be a left eigenvector of M_τ having λ as associated eigenvalue. If $|\lambda| < \theta$, then there exists a constant $C(\mathbf{r})$ such that, for all factors w of $\mathcal{T}$,*

$$\frac{|\mathbf{r} \cdot \Phi(w)|}{|w|} \leq C(\mathbf{r}).$$

One can see [13] for the details. If we fix $n, \ell \in \mathbb{N}$, the bound given in the proof is the following one:

$$\max_{\substack{u \in \mathcal{L}(\tau) \\ |u| \leq \ell}} \left\{ \frac{|\mathbf{r} \cdot \Phi(u)|}{|u|}, \frac{|\lambda|^n}{\iota(\ell, n)\theta^n} \max_{\substack{u \in \mathcal{L}(\tau) \\ |u| \leq \ell}} \frac{|\mathbf{r} \cdot \Phi(u)|}{|u|} + c_3(\mathbf{r}) \frac{\iota(\ell, n)\theta^n}{\iota(\ell, n)\theta^n - |\lambda|^n} \right\},$$

where

$$\iota(\ell, n) = \frac{\ell}{\ell + \theta^n \left(2 + \frac{1.5}{\theta - 1}\right)}$$

and

$$c_3(\mathbf{r}) = \max_{\substack{p\in\mathrm{Pref}(\tau^n)\\ s\in\mathrm{Suff}(\tau^n)}} \Bigg\{ |\mathbf{r}\cdot P_3\left(\Psi(p)\otimes\boldsymbol{\alpha}+\boldsymbol{\alpha}\otimes\Psi(s)\right)| + \frac{1}{\ell}\max_{\boldsymbol{\delta}\in[-1.5,1.5]^3} |\mathbf{r}\cdot\left(\Phi(ps)+P_3\left(\Psi(p)\otimes\boldsymbol{\delta}+\boldsymbol{\delta}\otimes\Psi(s)\right)\right)| \Bigg\}.$$

Moreover, integers n and ℓ have to verify

$$\frac{|\lambda|^n}{\iota(\ell,n)\theta^n} < 1\,.$$

The bound given by Proposition 18 will only be useful for the eigenvalues λ such that $|\lambda| \geq 1$. When $|\lambda| < 1$, the following result is stronger.

Proposition 19. *Let $\mathbf{r}$ be an eigenvector of M_τ and λ be the associated eigenvalue. If $|\lambda| < 1$, then there exists a constant $C(\mathbf{r})$ such that for all factors w of $\mathcal{T}$,*

$$|\mathbf{r}\cdot\Phi(w)| \leq C(\mathbf{r}).$$

For every $n \in \mathbb{N}$, the constant

$$C(\mathbf{r}) = \frac{1}{1-|\lambda|^n} \max_{\substack{p\in\mathrm{Pref}(\tau^n)\\ s\in\mathrm{Suff}(\tau^n)}} \max_{\boldsymbol{\delta}\in[-1.5,1.5]^3} |\mathbf{r}\cdot\left(\Phi(ps)+P_3\left(\Psi(p)\otimes\boldsymbol{\delta}+\boldsymbol{\delta}\otimes\Psi(s)\right)\right)|$$

is convenient. See [13] for details.

4.2 Bounds on templates

This subsection contains several lemmas giving necessary conditions on templates to be realizable by τ.

Lemma 20. *Let λ be an eigenvalue of M_τ such that $|\lambda| < 1$. For every left eigenvector $\mathbf{r}$ of M_τ associated with λ and for every realizable template $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$,*

$$\min_{(\delta_0,\delta_1)\in\Delta} \left| \mathbf{r}\cdot\left(\mathbf{d}+P_3\left(\mathbf{D_b}\otimes\begin{pmatrix}\delta_0\\ \delta_1\\ -\delta_0-\delta_1\end{pmatrix}+\begin{pmatrix}\delta_0\\ \delta_1\\ -\delta_0-\delta_1\end{pmatrix}\otimes\mathbf{D_e}\right)\right)\right| \leq 2C(\mathbf{r})$$

where $C(\mathbf{r})$ is the constant from Proposition 19.

This bound is not so easy to use because of the complicated minimum. It can be computed using tools from optimization. However, we can simply use this bound as follows.

For the sake of notation, let

$$f(\delta_0,\delta_1) = \mathbf{r}\cdot\left(\mathbf{d}+P_3\left(\mathbf{D_b}\otimes\begin{pmatrix}\delta_0\\ \delta_1\\ -\delta_0-\delta_1\end{pmatrix}+\begin{pmatrix}\delta_0\\ \delta_1\\ -\delta_0-\delta_1\end{pmatrix}\otimes\mathbf{D_e}\right)\right).$$

Then

$$\min_{(\delta_0,\delta_1)\in\Delta} |f(\delta_0,\delta_1)| \geq \sqrt{\min_{(\delta_0,\delta_1)\in\Delta} \mathrm{Re}\left(f(\delta_0,\delta_1)\right)^2 + \min_{(\delta_0,\delta_1)\in\Delta} \mathrm{Im}\left(f(\delta_0,\delta_1)\right)^2}.$$

Let I_{Re} and I_{Im} be intervals such that

$$I_{Re} = \left[\min_{(\delta_0,\delta_1)\in\Delta} \mathrm{Re}\left(f(\delta_0,\delta_1)\right), \max_{(\delta_0,\delta_1)\in\Delta} \mathrm{Re}\left(f(\delta_0,\delta_1)\right)\right]$$

and

$$I_{Im} = \left[\min_{(\delta_0,\delta_1)\in\Delta} \mathrm{Im}\left(f(\delta_0,\delta_1)\right), \max_{(\delta_0,\delta_1)\in\Delta} \mathrm{Im}\left(f(\delta_0,\delta_1)\right)\right].$$

Then

$$\min_{(\delta_0,\delta_1)\in\Delta} |f(\delta_0,\delta_1)| \geq \sqrt{\min_{y\in I_{Re}} y^2 + \min_{y\in I_{Im}} y^2}.$$

Thus any template for which this last quantity is greater than $2C(\mathbf{r})$ is not realizable.

Observe that each of the four interval bounds is reached for a vertex of the polytope, that is $\begin{pmatrix}\delta_0\\ \delta_1\end{pmatrix} \in \left\{\begin{pmatrix}1.5\\ -1.5\end{pmatrix}, \begin{pmatrix}1.5\\ 0\end{pmatrix}, \begin{pmatrix}0\\ 1.5\end{pmatrix}, \begin{pmatrix}-1.5\\ 1.5\end{pmatrix}, \begin{pmatrix}-1.5\\ 0\end{pmatrix}, \begin{pmatrix}0\\ -1.5\end{pmatrix}\right\}$. This is due to the fact that f is linear (and thus convex) over the convex set Δ.

This allows us to remove many templates from the set of templates, but this is not enough to obtain a finite set, so we need to somehow use the bounds on the other eigenvectors as well.

Lemma 21. *Let L be a positive integer. Let λ be an eigenvalue of M_τ such that $|\lambda| < \theta$. Then, for all eigenvectors $\mathbf{r}$ of M_τ associated with λ, there exists a constant $C(\mathbf{r})$ such that for any template $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ realized by a pair of factors of the Tribonacci word (u, v) with $|u| \geq L$, we have*

$$|\mathbf{r}\cdot P_3\left(\mathbf{D_b}\otimes\boldsymbol{\alpha}+\boldsymbol{\alpha}\otimes\mathbf{D_e}\right)| \leq \frac{2L-\sum_{i=1}^3 \mathbf{d}_i}{L} C(\mathbf{r}) + \max_{(\delta_0,\delta_1)\in\Delta} \frac{|\mathbf{r}\cdot\left(\mathbf{d}+P_3\left(\mathbf{D_b}\otimes\boldsymbol{\delta}+\boldsymbol{\delta}\otimes\mathbf{D_e}\right)\right)|}{L}.$$

The quantity of the l.h.s. and the first term on the r.h.s. are straightforward to compute. For the last term, it is not difficult to show that the maximum is in fact necessarily reached on a vertex of the polytope, that is

$$\max_{(\delta_0,\delta_1)\in\Delta} \frac{|\mathbf{r}\cdot\left(\mathbf{d}+P_3\left(\mathbf{D_b}\otimes\boldsymbol{\delta}+\boldsymbol{\delta}\otimes\mathbf{D_e}\right)\right)|}{L} \leq \max_{\left(\begin{smallmatrix}\delta_0\\ \delta_1\end{smallmatrix}\right) \in \left\{\left(\begin{smallmatrix}1.5\\ 0\end{smallmatrix}\right), \left(\begin{smallmatrix}1.5\\ -1.5\end{smallmatrix}\right), \left(\begin{smallmatrix}0\\ 1.5\end{smallmatrix}\right), \left(\begin{smallmatrix}-1.5\\ 0\end{smallmatrix}\right), \left(\begin{smallmatrix}-1.5\\ 1.5\end{smallmatrix}\right), \left(\begin{smallmatrix}0\\ -1.5\end{smallmatrix}\right)\right\}} \frac{|\mathbf{r}\cdot\left(\mathbf{d}+P_3\left(\mathbf{D_b}\otimes\boldsymbol{\delta}+\boldsymbol{\delta}\otimes\mathbf{D_e}\right)\right)|}{L}.$$

5 Proof of the main result

With all these lemmas, we are ready to show our main result.

Theorem 22. *Two factors of the Tribonacci word are 2-binomially equivalent if and only if they are equal.*

Proof. Let $T = \{[\mathbf{0}, \mathbf{0}, \mathbf{0}, a_1, a_2] : a_1 \neq a_2\}$. Let us show that no template from T is realizable. Let $L = 15$. We can easily check with a computer that no pair of factors of $\mathcal{T}$ with $\min(|u|, |v|) \leq L$ realizes a template t from the set T. Indeed, since for all $t \in T$, $\mathbf{d} = \mathbf{0}$, $\mathbf{D_b} = \mathbf{0}$ and $\mathbf{D_e} = \mathbf{0}$, we know that a pair of words (u, v) realizes t if and only if $\Phi(u) - \Phi(v) = 0$. It just suffices to check that for all $n \leq L$, $b_{\mathcal{T},2}(n) = p_{\mathcal{T}}(n)$.

Now, from Proposition 17, if $t \in T$ is realized then one of its ancestors is realized by a pair (u, v) with $L \leq \min(|u|, |v|) \leq 2L$.

Lemmas 20 and 21 give us two sets of inequalities that any template realized by a pair (u, v) of factors of Tribonacci with $|u| \geq L$ must respect. Let X be the set of templates that respect the bounds. Let $A_0 = T$ and, for all i, let $A_{i+1} = \{\mathrm{Par}_\tau(t) \cap X : t \in A_i\}$. Then clearly $\mathrm{RAnc}_\tau(t) \subseteq \bigcup_{i \in \mathbb{N}} A_i$. Each A_i can be easily computed and it can be checked by a computer program that the set $\bigcup_{i \in \mathbb{N}} A_i$ is finite.

We can finally check with a computer that there is no pair (u, v) of factors of $\mathcal{T}$ with $L \leq \min(|u|, |v|) \leq 2L$ that realizes any element of $\bigcup_{i \in \mathbb{N}} A_i$. Thus no template of T is realizable. By Lemma 12, we can conclude that the 2-binomial complexity of the Tribonacci word is equal to its factorial complexity. □

Accompanying this paper is an implementation in Mathematica of all the computations described in this theorem and in the previous lemmas and propositions. We also have a C++ implementation that is much faster, but uses machine floating point arithmetic whose accuracy cannot be guaranteed (in this case, however, we obtain exactly the same set of templates). Diagonalizing the matrix of Tribonacci gives 4 eigenvectors to which Lemma 20 can be applied. Since there are two pairs of conjugate complex vectors, it is useless to keep more than one of each pair. However, by taking a linear combination of these two, we get another eigenvector to which we can apply Lemma 20 (in practice we only do that once, but we could take as many vectors as we want from this 2-dimensional space). For this conjugation reason, we also only keep 4 of the 6 eigenvectors that correspond to an eigenvalue of norm less than 1. For each of these 7 eigenvectors, we choose[3] $\ell = 600$ and the best $1 \leq n \leq 6$ when applying Lemma 20 or Lemma 21. The rest is done as described in the article. We obtain a set of 241544 templates.

[3] Remember that we work on τ^n and that increasing n and ℓ tend to give us better bounds but increases the computation time.

6 Conclusion

We used an algorithm to show that the 2-binomial complexity of the Tribonacci word is equal to its factorial complexity. It seems that our method can be turned into an algorithm that can decide under some mild conditions whether the factorial complexity of a given morphic word is equal to its k-binomial complexity. In fact, by keeping track of the first letter of each word in templates, the "if" in Proposition 17 can be replaced by an "if and only if" (some technicalities could allow us to apply it even if the matrix is singular). Moreover, with arguments similar to the ideas from [16], one could show that we also have bounds on the eigenvectors that correspond to larger eigenvalues and that the number of templates that respect these bounds is always finite (one might need no eigenvalue has norm 1).

Observe that the notion of template was first introduced in the context of avoidability of abelian powers [8] and, as one could expect, it seems that our technique also gives a decision algorithm for the avoidability of k-binomial powers in morphic words (and even avoidability of patterns in the k-binomial sense).

References

1. A. Aberkane, J. Currie, A cyclic binary morphism avoiding abelian fourth powers, *Theoret. Comput. Sci.* **410** (2009), 44–52.
2. A. Aberkane, J. Currie, N. Rampersad, The number of ternary words avoiding abelian cubes grows exponentially, *J. Integer Seq.* **7** (2004), Article 04.2.7.
3. J.-P. Allouche, J. Shallit, *Automatic sequences. Theory, applications, generalizations*, Cambridge University Press (2003).
4. V. Berthé, M. Rigo (Eds.), *Combinatorics, Automata and Number Theory*, Encyclopedia Math. Appl., **135**, Cambridge Univ. Press (2010).
5. J. Cassaigne, J. Currie, L. Schaeffer, J. Shallit, Avoiding three consecutive blocks of the same size and same sum, *J. ACM* **61** (2014), no. 2, Art. 10.
6. J. Cassaigne, G. Richomme, K. Saari, L.Q. Zamboni, Avoiding Abelian powers in binary words with bounded Abelian complexity, *Internat. J. Found. Comput. Sci.* **22** (2011), 905–920.
7. J. Currie, N. Rampersad, Recurrent words with constant Abelian complexity, *Adv. Appl. Math.* **47** (2011), 116–124.
8. J. Currie, N. Rampersad, Fixed points avoiding Abelian k-powers, *J. Combin. Theory Ser. A* **119** (2012), 942–948.
9. A. Graham, *Kronecker products and matrix calculus: with applications*, Ellis Horwood Series in Mathematics and its Applications, Halsted Press [John Wiley & Sons, Inc., New York, (1981).
10. J. Karhumäki, A. Saarela, L. Q. Zamboni, On a generalization of Abelian equivalence and complexity of infinite words, *J. Combin. Theory Ser. A* **120** (2013), 2189–2206.
11. J. Karhumäki, A. Saarela, L. Q. Zamboni, Variations of the Morse-Hedlund theorem for k-abelian equivalence, *Lect. Notes in Comput. Sci.* **8633** (2014), 203–214.
12. M. Lejeune, J. Leroy, M. Rigo, Computing the k-binomial complexity of the Thue–Morse word, `arXiv:1812.07330`, 34 pages.

13. M. Lejeune, M. Rigo, M. Rosenfeld, Templates for the k-binomial complexity of the Tribonacci word (long version), `http://hdl.handle.net/2268/234215`, 23 pages.
14. F. Liétard, avoiding additive powers, talk at *Mons TCS days*, Bordeaux 10–14 september 2018.
15. M. Rao, M. Rigo, P. Salimov, Avoiding 2-binomial squares and cubes, *Theoret. Comput. Sci.* **572** (2015), 83–91.
16. M. Rao and M. Rosenfeld, Avoiding two consecutive blocks of same size and same sum over $\mathbb{Z}^2$, *SIAM Journal on Disc. Math.* **32:4** (2018), 2381–2397
17. G. Richomme, K. Saari, L.Q. Zamboni, Balance and Abelian complexity of the Tribonacci word, *Adv. Appl. Math.* **45** (2010), 212–231.
18. M. Rigo, P. Salimov, Another generalization of abelian equivalence: binomial complexity of infinite words, *Theoret. Comput. Sci.* **601** (2015), 47–57.
19. M. Rigo, invited talk, Streams II, Lorentz Center, Leiden, January 2014.
20. M. Rigo, Relations on words, *Indag. Math. (N.S.)* **28** (2017), 183–204.