Computing the k-binomial complexity of the Tribonacci word

LIÈGE université

fnrs
LA LIBERTÉ DE CHERCHER

September 12, 2019

Marie Lejeune (FNRS grantee)
Joint work with Michel Rigo and Matthieu Rosenfeld

k-binomial complexity of Tribonacci

Let's look at the Tribonacci word

$$\mathcal{T} = 0102010010201010201001 \cdots$$

and more precisely at its factors of a given length:

k-binomial complexity of Tribonacci

Let's look at the Tribonacci word

$$\mathcal{T} = 0102010010201010201001\cdots$$

and more precisely at its factors of a given length:

$$\mathrm{Fac}_{\mathcal{T}}(1) = \{0, \quad \}.$$

k-binomial complexity of Tribonacci

Let's look at the Tribonacci word

$$\mathcal{T} = 0{\color{red}1}0201001020101020100 1 \cdots$$

and more precisely at its factors of a given length:

$$\mathsf{Fac}_{\mathcal{T}}(1) = \{0, {\color{red}1}, \ \}.$$

k-binomial complexity of Tribonacci

Let's look at the Tribonacci word

$$\mathcal{T} = 0102010010201010201001\cdots$$

and more precisely at its factors of a given length:

$$\text{Fac}_{\mathcal{T}}(1) = \{0, 1, 2\}.$$

k-binomial complexity of Tribonacci

Let's look at the Tribonacci word

$$\mathcal{T} = 0102010010201010201001\cdots$$

and more precisely at its factors of a given length:

$$\mathsf{Fac}_{\mathcal{T}}(4) = \{0102, \qquad \}.$$

k-binomial complexity of Tribonacci

Let's look at the Tribonacci word

$$\mathcal{T} = 0{\color{red}1020}1001020101020 1001 \cdots$$

and more precisely at its factors of a given length:

$$\mathsf{Fac}_{\mathcal{T}}(4) = \{0102, {\color{red}1020}, \qquad\qquad\qquad\qquad \}.$$

Let's look at the Tribonacci word

$$\mathcal{T} = 0102010010201010201001\cdots$$

and more precisely at its factors of a given length:

$$\text{Fac}_{\mathcal{T}}(4) = \{0102, 1020, 0201, \qquad \}.$$

k-binomial complexity of Tribonacci

Let's look at the Tribonacci word

$$\mathcal{T} = 010\textcolor{red}{2010}010201010201001\cdots$$

and more precisely at its factors of a given length:

$$\mathsf{Fac}_{\mathcal{T}}(4) = \{0102, 1020, 0201, \textcolor{red}{2010}, \qquad\qquad \}.$$

Let's look at the Tribonacci word

$$\mathcal{T} = 0102010010201010201001\cdots$$

and more precisely at its factors of a given length:

$$\mathrm{Fac}_{\mathcal{T}}(4) = \{0102, 1020, 0201, 2010, 0100, \quad \}.$$

k-binomial complexity of Tribonacci

Let's look at the Tribonacci word

$$\mathcal{T} = 0102010010201010201001 \cdots$$

and more precisely at its factors of a given length:

$$\mathsf{Fac}_{\mathcal{T}}(4) = \{0102, 1020, 0201, 2010, 0100, \ldots\}.$$

k-binomial complexity of Tribonacci

Let's look at the Tribonacci word

$$\mathcal{T} = 0102010010201010201001\cdots$$

and more precisely at its factors of a given length:

$$\mathrm{Fac}_{\mathcal{T}}(4) = \{0102, 1020, 0201, 2010, 0100, \ldots\}.$$

The factor complexity

$$p_{\mathcal{T}} : n \mapsto \#\mathrm{Fac}_{\mathcal{T}}(n)$$

is equal to $n \mapsto 2n + 1$.

k-binomial complexity of Tribonacci

Let's look at the Tribonacci word

$$\mathcal{T} = 0102010010201010201001\cdots$$

and more precisely at its factors of a given length:

$$\mathsf{Fac}_{\mathcal{T}}(4) = \{0102, 1020, 0201, 2010, 0100, \ldots\}.$$

The factor complexity

$$p_{\mathcal{T}} : n \mapsto \#\mathsf{Fac}_{\mathcal{T}}(n)$$

is equal to $n \mapsto 2n + 1$.
Changing this complexity into an other one,

$$\mathbf{b}_{\mathcal{T}}^{(k)} : n \mapsto \#(\mathsf{Fac}_{\mathcal{T}}(n)/\sim_k)$$

we want to show that it is still equal to $n \mapsto 2n + 1$.

k-binomial complexity of Tribonacci

1. Preliminary definitions
   - Words, factors and subwords
   - Complexity functions
   - k-binomial complexity
2. State of the art
3. Next result: the Tribonacci word
   - Definition
   - The theorem
   - Introduction to templates and their parents
   - Using templates to compute $\mathbf{b}_T^{(2)}$

k-binomial complexity of Tribonacci

1 Preliminary definitions
- Words, factors and subwords
- Complexity functions
- k-binomial complexity

2 State of the art

3 Next result: the Tribonacci word
- Definition
- The theorem
- Introduction to templates and their parents
- Using templates to compute $\mathbf{b}_T^{(2)}$

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^{m}$. A **factor** of u is a contiguous subword.

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u.

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^{m}$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Let $\binom{\mathbf{u}}{\mathbf{x}}$ denote the number of times x appears as a subword in u and $|\mathbf{u}|_{\mathbf{x}}$ the number of times it appears as a factor in u.

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition
A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Let $\binom{\mathbf{u}}{\mathbf{x}}$ denote the number of times x appears as a subword in u and $|\mathbf{u}|_{\mathbf{x}}$ the number of times it appears as a factor in u.

If $u = aababa$,

$$|u|_{ab} = ? \text{ and } \binom{u}{ab} = ?$$

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Let $\binom{\mathbf{u}}{\mathbf{x}}$ denote the number of times x appears as a subword in u and $|\mathbf{u}|_{\mathbf{x}}$ the number of times it appears as a factor in u.

If $u = aababa$,

$$|u|_{ab} = 1 \text{ and } \binom{u}{ab} = ?$$

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Let $\binom{\mathbf{u}}{\mathbf{x}}$ denote the number of times x appears as a subword in u and $|\mathbf{u}|_{\mathbf{x}}$ the number of times it appears as a factor in u.

If $u = aababa$,

$$|u|_{ab} = 2 \text{ and } \binom{u}{ab} = ?$$

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Let $\binom{\mathbf{u}}{\mathbf{x}}$ denote the number of times x appears as a subword in u and $|\mathbf{u}|_{\mathbf{x}}$ the number of times it appears as a factor in u.

If $u = aababa$,

$$|u|_{ab} = 2 \text{ and } \binom{u}{ab} = ?$$

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Let $\binom{\mathbf{u}}{\mathbf{x}}$ denote the number of times x appears as a subword in u and $|\mathbf{u}|_{\mathbf{x}}$ the number of times it appears as a factor in u.

If $u = aababa$,

$$|u|_{ab} = 2 \text{ and } \binom{u}{ab} = 1.$$

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Let $\binom{\mathbf{u}}{\mathbf{x}}$ denote the number of times x appears as a subword in u and $|\mathbf{u}|_{\mathbf{x}}$ the number of times it appears as a factor in u.

If $u = aababa$,

$$|u|_{ab} = 2 \text{ and } \binom{u}{ab} = 2.$$

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Let $\binom{\mathbf{u}}{\mathbf{x}}$ denote the number of times x appears as a subword in u and $|\mathbf{u}|_{\mathbf{x}}$ the number of times it appears as a factor in u.

If $u = aababa$,

$$|u|_{ab} = 2 \text{ and } \binom{u}{ab} = 3.$$

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Let $\binom{\mathbf{u}}{\mathbf{x}}$ denote the number of times x appears as a subword in u and $|\mathbf{u}|_{\mathbf{x}}$ the number of times it appears as a factor in u.

If $u = aababa$,

$$|u|_{ab} = 2 \text{ and } \binom{u}{ab} = 4.$$

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Let $\binom{\mathbf{u}}{\mathbf{x}}$ denote the number of times x appears as a subword in u and $|\mathbf{u}|_{\mathbf{x}}$ the number of times it appears as a factor in u.

If $u = aababa$,

$$|u|_{ab} = 2 \text{ and } \binom{u}{ab} = 5.$$

Factors and subwords

Let $u = u_1 u_2 \cdots u_m$ be a finite or infinite word.

Definition

A **(scattered) subword** of u is a finite subsequence of the sequence $(u_j)_{j=1}^m$. A **factor** of u is a contiguous subword.

Let $u = abacaba$. The word acb is a subword of u, but not a factor of u. The word $acab$ is a factor of u, thus also a subword of u.

Let $\binom{\mathbf{u}}{\mathbf{x}}$ denote the number of times x appears as a subword in u and $|\mathbf{u}|_{\mathbf{x}}$ the number of times it appears as a factor in u.

If $u = aababa$,

$$|u|_{ab} = 2 \text{ and } \binom{u}{ab} = 5.$$

1 Preliminary definitions
- Words, factors and subwords
- Complexity functions
- k-binomial complexity

2 State of the art

3 Next result: the Tribonacci word
- Definition
- The theorem
- Introduction to templates and their parents
- Using templates to compute $\mathbf{b}_T^{(2)}$

Factor complexity

Let $\mathbf{w}$ be an infinite word. A complexity function of $\mathbf{w}$ is an application linking every nonnegative integer n with length-n factors of $\mathbf{w}$.

Factor complexity

Let **w** be an infinite word. A complexity function of **w** is an application linking every nonnegative integer n with length-n factors of **w**.

Definition

The factor complexity of the word **w** is the function

$$p_{\mathbf{w}} : \mathbb{N} \cup \{0\} \to \mathbb{N} \cup \{0\} : n \mapsto \#\mathrm{Fac}_{\mathbf{w}}(n).$$

Factor complexity

Let **w** be an infinite word. A complexity function of **w** is an application linking every nonnegative integer n with length-n factors of **w**.

Definition

The **factor complexity** of the word **w** is the function

$$p_{\mathbf{w}} : \mathbb{N} \cup \{0\} \to \mathbb{N} \cup \{0\} : n \mapsto \#(\mathrm{Fac}_{\mathbf{w}}(n)/\sim_{=}),$$

where $u \sim_{=} v \Leftrightarrow u = v$.

Factor complexity

Let **w** be an infinite word. A complexity function of **w** is an application linking every nonnegative integer n with length-n factors of **w**.

Definition

The **factor complexity** of the word **w** is the function

$$p_{\mathbf{w}} : \mathbb{N} \cup \{0\} \to \mathbb{N} \cup \{0\} : n \mapsto \#(\mathrm{Fac}_{\mathbf{w}}(n)/\sim_=),$$

where $u \sim_= v \Leftrightarrow u = v$.

We can replace $\sim_=$ with other equivalence relations.

Other equivalence relations

Different equivalence relations from $\sim_=$ can be considered:

Other equivalence relations

Different equivalence relations from $\sim_=$ can be considered:

- abelian equivalence: $u \sim_{ab,1} v \Leftrightarrow |u|_a = |v|_a \ \forall a \in A$

Other equivalence relations

Different equivalence relations from $\sim_=$ can be considered:

- abelian equivalence: $u \sim_{ab,1} v \Leftrightarrow |u|_a = |v|_a \; \forall a \in A$

If $k \in \mathbb{N}$,

- k-abelian equivalence: $u \sim_{ab,k} v \Leftrightarrow |u|_x = |v|_x \; \forall x \in A^{\leq k}$

Other equivalence relations

Different equivalence relations from $\sim_=$ can be considered:

- abelian equivalence: $u \sim_{ab,1} v \Leftrightarrow |u|_a = |v|_a \ \forall a \in A$

If $k \in \mathbb{N}$,

- k-abelian equivalence: $u \sim_{ab,k} v \Leftrightarrow |u|_x = |v|_x \ \forall x \in A^{\leq k}$
- k-binomial equivalence: $u \sim_k v \Leftrightarrow \binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}$

Other equivalence relations

Different equivalence relations from $\sim_=$ can be considered:

- abelian equivalence: $u \sim_{ab,1} v \Leftrightarrow |u|_a = |v|_a \ \forall a \in A$

If $k \in \mathbb{N}$,

- k-abelian equivalence: $u \sim_{ab,k} v \Leftrightarrow |u|_x = |v|_x \ \forall x \in A^{\leq k}$
- k-binomial equivalence: $u \sim_k v \Leftrightarrow \binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}$

We will deal with the last one.

1 Preliminary definitions
- Words, factors and subwords
- Complexity functions
- k-binomial complexity

2 State of the art

3 Next result: the Tribonacci word
- Definition
- The theorem
- Introduction to templates and their parents
- Using templates to compute $\mathbf{b}_T^{(2)}$

k-binomial equivalence

Definition (Reminder)

Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

k-binomial equivalence

Definition (Reminder)

Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent.
Indeed,

k-binomial equivalence

Definition (Reminder)

Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 1 = \binom{v}{a}$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \; \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent.
Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 1 = \binom{v}{b}$$

k-binomial equivalence

Definition (Reminder)

Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 2 = \binom{v}{b}$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 3 = \binom{v}{b}$$

k-binomial equivalence

Definition (Reminder)

Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}$$

k-binomial equivalence

Definition (Reminder)

Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \; \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa}$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent.
Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 1 = \binom{v}{ab}$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bb\textcolor{red}{a}ab\textcolor{red}{b}$ and $v = b\textcolor{red}{a}b\textcolor{red}{b}ab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = \textcolor{red}{2} = \binom{v}{ab}$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 3 = \binom{v}{ab}$$

k-binomial equivalence

Definition (Reminder)

Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}, \binom{u}{ba} = 1 = \binom{v}{ba}.$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent.
Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}, \binom{u}{ba} = 2 = \binom{v}{ba}.$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent.
Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}, \binom{u}{ba} = 3 = \binom{v}{ba}.$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}, \binom{u}{ba} = 4 = \binom{v}{ba}.$$

k-binomial equivalence

Definition (Reminder)
Let u and v be two finite words. They are **k-binomially equivalent** if

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

The words $u = bbaabb$ and $v = babbab$ are 2-binomially equivalent. Indeed,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}, \binom{u}{ba} = 4 = \binom{v}{ba}.$$

Some properties

1. For all words u, v and for every nonnegative integer k,

$$u \sim_{k+1} v \Rightarrow u \sim_k v.$$

Some properties

1. For all words u, v and for every nonnegative integer k,

$$u \sim_{k+1} v \Rightarrow u \sim_k v.$$

2. For all words u, v,

$$u \sim_1 v \Leftrightarrow u \sim_{ab,1} v.$$

Some properties

1. For all words u, v and for every nonnegative integer k,

$$u \sim_{k+1} v \Rightarrow u \sim_k v.$$

2. For all words u, v,

$$u \sim_1 v \Leftrightarrow u \sim_{ab,1} v.$$

Indeed, the words u and v are 1-abelian equivalent if

$$\binom{u}{a} = |u|_a = |v|_a = \binom{v}{a} \; \forall a \in A.$$

k-binomial complexity

Definition

If $\mathbf{w}$ is an infinite word, we can define the function

$$\mathbf{b}_{\mathbf{w}}^{(k)} : \mathbb{N} \to \mathbb{N} : n \mapsto \#(\mathrm{Fac}_{\mathbf{w}}(n)/\sim_k),$$

which is called the k-binomial complexity of $\mathbf{w}$.

k-binomial complexity

Definition

If $\mathbf{w}$ is an infinite word, we can define the function

$$\mathbf{b}_{\mathbf{w}}^{(k)} : \mathbb{N} \to \mathbb{N} : n \mapsto \#(\mathsf{Fac}_{\mathbf{w}}(n)/\sim_k),$$

which is called the **k-binomial complexity** of $\mathbf{w}$.

We have an order relation between the different complexity functions:

$$\rho_{\mathbf{w}}^{ab}(n) \leq \mathbf{b}_{\mathbf{w}}^{(k)}(n) \leq \mathbf{b}_{\mathbf{w}}^{(k+1)}(n) \leq p_{\mathbf{w}}(n) \quad \forall n \in \mathbb{N}, k \in \mathbb{N}^+$$

where $\rho_{\mathbf{w}}^{ab}$ is the abelian complexity function of the word $\mathbf{w}$.

k-binomial complexity of Tribonacci

1. Preliminary definitions
   - Words, factors and subwords
   - Complexity functions
   - k-binomial complexity
2. State of the art
3. Next result: the Tribonacci word
   - Definition
   - The theorem
   - Introduction to templates and their parents
   - Using templates to compute $\mathbf{b}_T^{(2)}$

A famous word...

The classical **Thue–Morse word**, defined as the fixed point of the morphism

$$\varphi : \{0,1\}^* \to \{0,1\}^* : \left\{ \begin{array}{lcl} 0 & \mapsto & 01; \\ 1 & \mapsto & 10, \end{array} \right.$$

has a bounded *k*-binomial complexity.

A famous word...

The classical **Thue–Morse word**, defined as the fixed point of the morphism

$$\varphi : \{0,1\}^* \to \{0,1\}^* : \left\{ \begin{array}{lcl} 0 & \mapsto & 01; \\ 1 & \mapsto & 10, \end{array} \right.$$

has a bounded k-binomial complexity. The exact value is known.

Theorem (M. L., J. Leroy, M. Rigo, 2018)

Let k be a positive integer. For every $n \leq 2^k - 1$, we have

$$\mathbf{b}_{\mathbf{t}}^{(k)}(n) = p_{\mathbf{t}}(n),$$

while for every $n \geq 2^k$,

$$\mathbf{b}_{\mathbf{t}}^{(k)}(n) = \left\{ \begin{array}{ll} 3 \cdot 2^k - 3, & \text{if } n \equiv 0 \pmod{2^k}; \\ 3 \cdot 2^k - 4, & \text{otherwise.} \end{array} \right.$$

Another family

A **Sturmian word** is an infinite word having, as factor complexity, $p(n) = n + 1$ for all $n \in \mathbb{N}$.

Another family

A **Sturmian word** is an infinite word having, as factor complexity, $p(n) = n+1$ for all $n \in \mathbb{N}$.

Theorem (M. Rigo, P. Salimov, 2015)
Let $\mathbf{w}$ be a Sturmian word. We have

$$\mathbf{b}_{\mathbf{w}}^{(k)}(n) = p_{\mathbf{w}}(n) = n+1,$$

for all $n \in \mathbb{N}$ and for all $k \geq 2$.

Another family

A **Sturmian word** is an infinite word having, as factor complexity, $p(n) = n+1$ for all $n \in \mathbb{N}$.

Theorem (M. Rigo, P. Salimov, 2015)
Let $\mathbf{w}$ be a Sturmian word. We have

$$\mathbf{b}_{\mathbf{w}}^{(k)}(n) = p_{\mathbf{w}}(n) = n+1,$$

for all $n \in \mathbb{N}$ and for all $k \geq 2$.

Since $\mathbf{b}_{\mathbf{w}}^{(k)}(n) \leq \mathbf{b}_{\mathbf{w}}^{(k+1)}(n) \leq p_{\mathbf{w}}(n)$, it suffices to show that

$$\mathbf{b}_{\mathbf{w}}^{(2)}(n) = p_{\mathbf{w}}(n).$$

k-binomial complexity of Tribonacci

1 Preliminary definitions
- Words, factors and subwords
- Complexity functions
- k-binomial complexity

2 State of the art

3 Next result: the Tribonacci word
- Definition
- The theorem
- Introduction to templates and their parents
- Using templates to compute $\mathbf{b}_T^{(2)}$

k-binomial complexity of Tribonacci

1 Preliminary definitions
- Words, factors and subwords
- Complexity functions
- k-binomial complexity

2 State of the art

3 Next result: the Tribonacci word
- Definition
- The theorem
- Introduction to templates and their parents
- Using templates to compute $\mathbf{b}_T^{(2)}$

The Tribonacci word

From now, let $A = \{0, 1, 2\}$. Let us define the Tribonacci word.

The Tribonacci word

From now, let $A = \{0, 1, 2\}$. Let us define the Tribonacci word.

Definition

The **Tribonacci word** $\mathcal{T}$ is the fixed point of the morphism

$$\tau : \{0,1,2\}^* \to \{0,1,2\}^* : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$$

From now, let $A = \{0, 1, 2\}$. Let us define the Tribonacci word.

Definition

The **Tribonacci word** $\mathcal{T}$ is the fixed point of the morphism

$$\tau : \{0,1,2\}^* \to \{0,1,2\}^* : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$$

We have

$$\mathcal{T} = \underline{0}1 \cdots$$

From now, let $A = \{0, 1, 2\}$. Let us define the Tribonacci word.

Definition

The **Tribonacci word** $\mathcal{T}$ is the fixed point of the morphism

$$\tau : \{0,1,2\}^* \to \{0,1,2\}^* : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$$

We have

$$\mathcal{T} = 0\underline{1}02 \cdots$$

The Tribonacci word

From now, let $A = \{0, 1, 2\}$. Let us define the Tribonacci word.

Definition
The **Tribonacci word** $\mathcal{T}$ is the fixed point of the morphism

$$\tau : \{0,1,2\}^* \to \{0,1,2\}^* : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$$

We have

$$\mathcal{T} = 01\underline{0}2\textcolor{red}{01}\cdots$$

From now, let $A = \{0, 1, 2\}$. Let us define the Tribonacci word.

Definition

The **Tribonacci word** $\mathcal{T}$ is the fixed point of the morphism

$$\tau : \{0,1,2\}^* \to \{0,1,2\}^* : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$$

We have

$$\mathcal{T} = 010\underline{2}010 \cdots$$

The Tribonacci word

From now, let $A = \{0, 1, 2\}$. Let us define the Tribonacci word.

Definition

The **Tribonacci word** $\mathcal{T}$ is the fixed point of the morphism

$$\tau : \{0,1,2\}^* \to \{0,1,2\}^* : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$$

We have

$$\mathcal{T} = 0102\underline{0}1001\cdots$$

The Tribonacci word

From now, let $A = \{0, 1, 2\}$. Let us define the Tribonacci word.

Definition

The **Tribonacci word** $\mathcal{T}$ is the fixed point of the morphism

$$\tau : \{0,1,2\}^* \to \{0,1,2\}^* : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$$

We have

$$\mathcal{T} = 01020\underline{1}00102 \cdots$$

The Tribonacci word

From now, let $A = \{0, 1, 2\}$. Let us define the Tribonacci word.

Definition

The **Tribonacci word** $\mathcal{T}$ is the fixed point of the morphism

$$\tau : \{0,1,2\}^* \to \{0,1,2\}^* : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$$

We have

$$\mathcal{T} = 0102010\underline{0}10201\cdots$$

The Tribonacci word

From now, let $A = \{0, 1, 2\}$. Let us define the Tribonacci word.

Definition

The **Tribonacci word** $\mathcal{T}$ is the fixed point of the morphism

$$\tau : \{0,1,2\}^* \to \{0,1,2\}^* : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$$

We have

$$\mathcal{T} = 0102010010201 \cdots$$

1 Preliminary definitions
- Words, factors and subwords
- Complexity functions
- k-binomial complexity

2 State of the art

3 Next result: the Tribonacci word
- Definition
- The theorem
- Introduction to templates and their parents
- Using templates to compute $\mathbf{b}_T^{(2)}$

Its k-binomial complexity

The next result was first conjectured by Michel Rigo, and then proved.

Theorem (M. L., M. Rigo, M. Rosenfeld, 2019)
For all $k \geq 2$, the k-binomial complexity of the Tribonacci word equals its factor complexity.

Its k-binomial complexity

The next result was first conjectured by Michel Rigo, and then proved.

Theorem (M. L., M. Rigo, M. Rosenfeld, 2019)
For all $k \geq 2$, the k-binomial complexity of the Tribonacci word equals its factor complexity.

To prove this result, it suffices to show that, for all $n \in \mathbb{N}$,

$$\begin{cases} u, v \in \mathrm{Fac}_{\mathcal{T}}(n) \\ u \neq v \end{cases} \Rightarrow \quad u \not\sim_2 v.$$

Its k-binomial complexity

The **Parikh vector** of a word u is classicaly defined as

$$\Psi(u) := \left(\binom{u}{0} \quad \binom{u}{1} \quad \binom{u}{2}\right)^{\intercal} \in \mathbb{N}^3 \,.$$

Its k-binomial complexity

The **Parikh vector** of a word u is classicaly defined as

$$\Psi(u) := \left(\binom{u}{0} \quad \binom{u}{1} \quad \binom{u}{2}\right)^{\intercal} \in \mathbb{N}^3 .$$

Let us define the **extended Parikh vector** of a word u as

$$\Phi(u) := \left(\binom{u}{0} \quad \binom{u}{1} \quad \binom{u}{2} \quad \binom{u}{00} \quad \binom{u}{01} \quad \dots \quad \binom{u}{22}\right)^{\intercal} \in \mathbb{N}^{12} .$$

Its k-binomial complexity

The **Parikh vector** of a word u is classicaly defined as

$$\Psi(u) := \left(\binom{u}{0} \quad \binom{u}{1} \quad \binom{u}{2}\right)^{\mathsf{T}} \in \mathbb{N}^3 .$$

Let us define the **extended Parikh vector** of a word u as

$$\Phi(u) := \left(\binom{u}{0} \quad \binom{u}{1} \quad \binom{u}{2} \quad \binom{u}{00} \quad \binom{u}{01} \quad \dots \quad \binom{u}{22}\right)^{\mathsf{T}} \in \mathbb{N}^{12} .$$

Remark

We have $u \sim_2 v \;\Leftrightarrow\; \Phi(u) = \Phi(v) \;\Leftrightarrow\; \Phi(u) - \Phi(v) = 0.$

k-binomial complexity of Tribonacci

1 Preliminary definitions
- Words, factors and subwords
- Complexity functions
- k-binomial complexity

2 State of the art

3 Next result: the Tribonacci word
- Definition
- The theorem
- Introduction to templates and their parents
- Using templates to compute $\mathbf{b}_T^{(2)}$

Intuitive introduction to templates

We will be interested into values of $\Phi(u) - \Phi(v)$ for $u, v \in \mathrm{Fac}_{\mathcal{T}}$. We will thus express the difference using the notion of templates.

Intuitive introduction to templates

We will be interested into values of $\Phi(u) - \Phi(v)$ for $u, v \in \mathrm{Fac}_{\mathcal{T}}$. We will thus express the difference using the notion of templates.

Informally, we will associate to every pair of words several templates, which are 5-uples:

$$A^* \times A^* \leftrightsquigarrow \mathbb{Z}^{12} \times \mathbb{Z}^3 \times \mathbb{Z}^3 \times A \times A.$$

Intuitive introduction to templates

We will be interested into values of $\Phi(u) - \Phi(v)$ for $u, v \in \mathrm{Fac}_{\mathcal{T}}$. We will thus express the difference using the notion of templates.

Informally, we will associate to every pair of words several templates, which are 5-uples:

$$A^* \times A^* \leftrightsquigarrow \mathbb{Z}^{12} \times \mathbb{Z}^3 \times \mathbb{Z}^3 \times A \times A.$$

We will restrict this relation to factors of $\mathcal{T}$:

$$\mathrm{Fac}_{\mathcal{T}} \times \mathrm{Fac}_{\mathcal{T}} \leftrightsquigarrow \mathbb{Z}^{12} \times \mathbb{Z}^3 \times \mathbb{Z}^3 \times A \times A.$$

Intuitive introduction to templates

We will be interested into values of $\Phi(u) - \Phi(v)$ for $u, v \in \mathrm{Fac}_{\mathcal{T}}$. We will thus express the difference using the notion of templates.

Informally, we will associate to every pair of words several templates, which are 5-uples:

$$A^* \times A^* \leftrightsquigarrow \mathbb{Z}^{12} \times \mathbb{Z}^3 \times \mathbb{Z}^3 \times A \times A.$$

We will restrict this relation to factors of $\mathcal{T}$:

$$\mathrm{Fac}_{\mathcal{T}} \times \mathrm{Fac}_{\mathcal{T}} \leftrightsquigarrow \mathbb{Z}^{12} \times \mathbb{Z}^3 \times \mathbb{Z}^3 \times A \times A.$$

There exists a strong link between this notion and our thesis:

$$\mathbf{b}_{\mathcal{T}}^{(2)}(n) < p_{\mathcal{T}}(n)$$

$$\Leftrightarrow$$

$$\exists (u, v) \in \mathrm{Fac}_{\mathcal{T}} \times \mathrm{Fac}_{\mathcal{T}} \leftrightsquigarrow [\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b], a \neq b$$

Templates: a formal definition

Definition

A **template** is a 5-uple of the form $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ where $\mathbf{d} \in \mathbb{Z}^{12}$, $\mathbf{D_b}, \mathbf{D_e} \in \mathbb{Z}^3$ and $a_1, a_2 \in A$.

Templates: a formal definition

Definition

A **template** is a 5-uple of the form $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ where $\mathbf{d} \in \mathbb{Z}^{12}$, $\mathbf{D_b}, \mathbf{D_e} \in \mathbb{Z}^3$ and $a_1, a_2 \in A$.

The template t is said to be **realizable** by $(u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ if

Definition

A **template** is a 5-uple of the form $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ where $\mathbf{d} \in \mathbb{Z}^{12}$, $\mathbf{D_b}, \mathbf{D_e} \in \mathbb{Z}^3$ and $a_1, a_2 \in A$.

The template t is said to be **realizable** by $(u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ if

$$\begin{cases} \exists u' \in A^* \;:\; u = u' a_1, \\ \exists v' \in A^* \;:\; v = v' a_2, \\ \Phi(u) - \Phi(v) = \mathbf{d} + P_3\left(\mathbf{D_b} \otimes \Psi(u) + \Psi(u) \otimes \mathbf{D_e}\right), \end{cases}$$

Templates: a formal definition

Definition

A **template** is a 5-uple of the form $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ where $\mathbf{d} \in \mathbb{Z}^{12}$, $\mathbf{D_b}, \mathbf{D_e} \in \mathbb{Z}^3$ and $a_1, a_2 \in A$.
The template t is said to be **realizable** by $(u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ if

$$\begin{cases} \exists u' \in A^* \ : \ u = u' a_1, \\ \exists v' \in A^* \ : \ v = v' a_2, \\ \Phi(u) - \Phi(v) = \mathbf{d} + P_3 \left(\mathbf{D_b} \otimes \Psi(u) + \Psi(u) \otimes \mathbf{D_e} \right), \end{cases}$$

Templates: a formal definition

Definition

A **template** is a 5-uple of the form $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ where $\mathbf{d} \in \mathbb{Z}^{12}$, $\mathbf{D_b}, \mathbf{D_e} \in \mathbb{Z}^3$ and $a_1, a_2 \in A$.

The template t is said to be **realizable** by $(u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ if

$$\begin{cases} \exists u' \in A^* \ : \ u = u' a_1, \\ \exists v' \in A^* \ : \ v = v' a_2, \\ \Phi(u) - \Phi(v) = \mathbf{d} + P_3 \left(\mathbf{D_b} \otimes \Psi(u) + \Psi(u) \otimes \mathbf{D_e} \right), \end{cases}$$

Templates: a formal definition

Definition

A **template** is a 5-uple of the form $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ where $\mathbf{d} \in \mathbb{Z}^{12}$, $\mathbf{D_b}, \mathbf{D_e} \in \mathbb{Z}^3$ and $a_1, a_2 \in A$.

The template t is said to be **realizable** by $(u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ if

$$\begin{cases} \exists u' \in A^* \ : \ u = u' a_1, \\ \exists v' \in A^* \ : \ v = v' a_2, \\ \Phi(u) - \Phi(v) = \mathbf{d} + P_3 \left(\mathbf{D_b} \otimes \Psi(u) + \Psi(u) \otimes \mathbf{D_e} \right), \end{cases}$$

where the matrix P_3 is such that, for all $\mathbf{x} \in \mathbb{Z}^9$, $P_3 \cdot \mathbf{x} = \begin{pmatrix} 0 & 0 & 0 & \mathbf{x} \end{pmatrix}^\intercal$, and where $\otimes$ is the usual Kronecker product: if $A \in \mathbb{Z}^{m \times n}$ and $B \in \mathbb{Z}^{p \times q}$,

$$A \otimes B := \begin{pmatrix} a_{11} B & \cdots & a_{1n} B \\ \vdots & \ddots & \vdots \\ a_{m1} B & \cdots & a_{mn} B \end{pmatrix} \in \mathbb{Z}^{mp \times nq}.$$

Templates: a formal definition

Definition

A **template** is a 5-uple of the form $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ where $\mathbf{d} \in \mathbb{Z}^{12}$, $\mathbf{D_b}, \mathbf{D_e} \in \mathbb{Z}^3$ and $a_1, a_2 \in A$.
The template t is said to be **realizable** by $(u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ if

$$\begin{cases} \exists u' \in A^* \ : \ u = u' a_1, \\ \exists v' \in A^* \ : \ v = v' a_2, \\ \Phi(u) - \Phi(v) = \mathbf{d} + P_3 \left(\mathbf{D_b} \otimes \Psi(u) + \Psi(u) \otimes \mathbf{D_e} \right), \end{cases}$$

where the matrix P_3 is such that, for all $\mathbf{x} \in \mathbb{Z}^9$, $P_3 \cdot \mathbf{x} = \begin{pmatrix} 0 & 0 & 0 & \mathbf{x} \end{pmatrix}^\intercal$,
and where $\otimes$ is the usual Kronecker product: if $A \in \mathbb{Z}^{m \times n}$ and $B \in \mathbb{Z}^{p \times q}$,

$$A \otimes B := \begin{pmatrix} a_{11} B & \cdots & a_{1n} B \\ \vdots & \ddots & \vdots \\ a_{m1} B & \cdots & a_{mn} B \end{pmatrix} \in \mathbb{Z}^{mp \times nq}.$$

Templates: a formal definition

Definition

A **template** is a 5-uple of the form $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ where $\mathbf{d} \in \mathbb{Z}^{12}$, $\mathbf{D_b}, \mathbf{D_e} \in \mathbb{Z}^3$ and $a_1, a_2 \in A$.
The template t is said to be **realizable** by $(u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ if

$$\begin{cases} \exists u' \in A^* \ : \ u = u' a_1, \\ \exists v' \in A^* \ : \ v = v' a_2, \\ \Phi(u) - \Phi(v) = \mathbf{d} + P_3 \left(\mathbf{D_b} \otimes \Psi(u) + \Psi(u) \otimes \mathbf{D_e} \right), \end{cases}$$

where the matrix P_3 is such that, for all $\mathbf{x} \in \mathbb{Z}^9$, $P_3 \cdot \mathbf{x} = \begin{pmatrix} 0 & 0 & 0 & \mathbf{x} \end{pmatrix}^{\intercal}$, and where $\otimes$ is the usual Kronecker product: if $A \in \mathbb{Z}^{m \times n}$ and $B \in \mathbb{Z}^{p \times q}$,

$$A \otimes B := \begin{pmatrix} a_{11} B & \cdots & a_{1n} B \\ \vdots & \ddots & \vdots \\ a_{m1} B & \cdots & a_{mn} B \end{pmatrix} \in \mathbb{Z}^{mp \times nq}.$$

Templates: a formal definition

Definition

A **template** is a 5-uple of the form $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ where $\mathbf{d} \in \mathbb{Z}^{12}$, $\mathbf{D_b}, \mathbf{D_e} \in \mathbb{Z}^3$ and $a_1, a_2 \in A$.
The template t is said to be **realizable** by $(u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ if

$$\begin{cases} \exists u' \in A^* \; : \; u = u' a_1, \\ \exists v' \in A^* \; : \; v = v' a_2, \\ \Phi(u) - \Phi(v) = \mathbf{d} + P_3 \left(\mathbf{D_b} \otimes \Psi(u) + \Psi(u) \otimes \mathbf{D_e} \right), \end{cases}$$

where the matrix P_3 is such that, for all $\mathbf{x} \in \mathbb{Z}^9$, $P_3 \cdot \mathbf{x} = \begin{pmatrix} 0 & 0 & 0 & \mathbf{x} \end{pmatrix}^\intercal$, and where $\otimes$ is the usual Kronecker product: if $A \in \mathbb{Z}^{m \times n}$ and $B \in \mathbb{Z}^{p \times q}$,

$$A \otimes B := \begin{pmatrix} a_{11} B & \cdots & a_{1n} B \\ \vdots & \ddots & \vdots \\ a_{m1} B & \cdots & a_{mn} B \end{pmatrix} \in \mathbb{Z}^{mp \times nq}.$$

Why are templates useful?

Theorem

There exists $n \in \mathbb{N}$ such that

$$\mathbf{b}_{\mathcal{T}}^{(2)}(n) < p_{\mathcal{T}}(n)$$

if and only if at least one template of the form $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ with $a \neq b$ is realizable by a pair of factors of $\mathcal{T}$.

Why are templates useful?

Theorem

There exists $n \in \mathbb{N}$ such that

$$\mathbf{b}_{\mathcal{T}}^{(2)}(n) < p_{\mathcal{T}}(n)$$

if and only if at least one template of the form $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ with $a \neq b$ is realizable by a pair of factors of $\mathcal{T}$.

Proof

Why are templates useful?

Theorem

There exists $n \in \mathbb{N}$ such that

$$\mathbf{b}_{\mathcal{T}}^{(2)}(n) < p_{\mathcal{T}}(n)$$

if and only if at least one template of the form $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ with $a \neq b$ is realizable by a pair of factors of $\mathcal{T}$.

Proof

$\Leftarrow$ $\exists (u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ realizing $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$.

Why are templates useful?

Theorem

There exists $n \in \mathbb{N}$ such that

$$\mathbf{b}_{\mathcal{T}}^{(2)}(n) < p_{\mathcal{T}}(n)$$

if and only if at least one template of the form $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ with $a \neq b$ is realizable by a pair of factors of $\mathcal{T}$.

Proof

$\Leftarrow$ $\exists (u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ realizing $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$. So, $\Phi(u) - \Phi(v) = 0$ and $u \neq v$.

Why are templates useful?

Theorem

There exists $n \in \mathbb{N}$ such that

$$\mathbf{b}_{\mathcal{T}}^{(2)}(n) < p_{\mathcal{T}}(n)$$

if and only if at least one template of the form $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ with $a \neq b$ is realizable by a pair of factors of $\mathcal{T}$.

Proof

$\Leftarrow$ $\exists (u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ realizing $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$. So, $\Phi(u) - \Phi(v) = 0$ and $u \neq v$.

$\Rightarrow$ $\exists u = u_1 \cdots u_n,\ v = v_1 \cdots v_n \in \mathrm{Fac}_{\mathcal{T}}$ such that $u \neq v$ and $u \sim_2 v$.

Why are templates useful?

Theorem

There exists $n \in \mathbb{N}$ such that

$$\mathbf{b}_{\mathcal{T}}^{(2)}(n) < p_{\mathcal{T}}(n)$$

if and only if at least one template of the form $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ with $a \neq b$ is realizable by a pair of factors of $\mathcal{T}$.

Proof

$\Leftarrow$ $\exists (u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ realizing $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$. So, $\Phi(u) - \Phi(v) = 0$ and $u \neq v$.

$\Rightarrow$ $\exists u = u_1 \cdots u_n,\ v = v_1 \cdots v_n \in \mathrm{Fac}_{\mathcal{T}}$ such that $u \neq v$ and $u \sim_2 v$.
Let $i \in \{1, \ldots, n\}$ be such that $u_i \neq v_i, u_{i+1} = v_{i+1}, \ldots, u_n = v_n$.

Why are templates useful?

Theorem

There exists $n \in \mathbb{N}$ such that

$$\mathbf{b}_{\mathcal{T}}^{(2)}(n) < p_{\mathcal{T}}(n)$$

if and only if at least one template of the form $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ with $a \neq b$ is realizable by a pair of factors of $\mathcal{T}$.

Proof

$\Leftarrow$ $\exists (u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$ realizing $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$. So, $\Phi(u) - \Phi(v) = 0$ and $u \neq v$.

$\Rightarrow$ $\exists u = u_1 \cdots u_n,\ v = v_1 \cdots v_n \in \mathrm{Fac}_{\mathcal{T}}$ such that $u \neq v$ and $u \sim_2 v$.
Let $i \in \{1, \ldots, n\}$ be such that $u_i \neq v_i, u_{i+1} = v_{i+1}, \ldots, u_n = v_n$.
Then $(u_1 \cdots u_i, v_1 \cdots v_i)$ realizes $[\mathbf{0}, \mathbf{0}, \mathbf{0}, u_i, v_i]$, because $u \sim_2 v \Rightarrow u_1 \cdots u_i \sim_2 v_1 \cdots v_i$.

Using this result

We want to verify that $\forall (u,v) \in (\mathrm{Fac}_{\mathcal{T}})^2$, the pair (u,v) doesn't realize any template of the form $[\mathbf{0},\mathbf{0},\mathbf{0},a,b]$ with $a \neq b$.

Using this result

We want to verify that $\forall (u,v) \in (\mathrm{Fac}_{\mathcal{T}})^2$, the pair (u,v) doesn't realize any template of the form $[\mathbf{0},\mathbf{0},\mathbf{0},a,b]$ with $a \neq b$.

There exists an infinite number of pairs to check. Hopefully, we have an interesting result:

Using this result

We want to verify that $\forall(u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$, the pair (u, v) doesn't realize any template of the form $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ with $a \neq b$.

There exists an infinite number of pairs to check. Hopefully, we have an interesting result:

Let us suppose that there exists a pair (u, v) realizing $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ and let fix $L > 0$. Then,

- either $\min(|u|, |v|) \leq L$, or
- there exists an *ancestor template* of $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ which is realized by a pair of words $(u', v') =$ "$((\tau^{-1})^j(u), (\tau^{-1})^j(v))$" such that $L \leq \min(|u'|, |v'|) \leq 2L$.

Using this result

We want to verify that $\forall (u, v) \in (\mathrm{Fac}_{\mathcal{T}})^2$, the pair (u, v) doesn't realize any template of the form $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ with $a \neq b$.

There exists an infinite number of pairs to check. Hopefully, we have an interesting result:

Let us suppose that there exists a pair (u, v) realizing $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ and let fix $L > 0$. Then,

- either $\min(|u|, |v|) \leq L$, or
- there exists an *ancestor template* of $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ which is realized by a pair of words $(\mathbf{u}', \mathbf{v}') = \text{"}((\tau^{-\mathbf{1}})^{\mathbf{j}}(\mathbf{u}), (\tau^{-\mathbf{1}})^{\mathbf{j}}(\mathbf{v}))\text{"}$ such that $L \leq \min(|u'|, |v'|) \leq 2L$.

Using this result

We want to verify that $\forall (u,v) \in (\mathrm{Fac}_{\mathcal{T}})^2$, the pair (u,v) doesn't realize any template of the form $[\mathbf{0},\mathbf{0},\mathbf{0},a,b]$ with $a \neq b$.

There exists an infinite number of pairs to check. Hopefully, we have an interesting result:

Let us suppose that there exists a pair (u,v) realizing $[\mathbf{0},\mathbf{0},\mathbf{0},a,b]$ and let fix $L > 0$. Then,

- either $\min(|u|,|v|) \leq L$, or
- there exists an *ancestor template* of $[\mathbf{0},\mathbf{0},\mathbf{0},a,b]$ which is realized by a pair of words $(u',v') =$ "$((\tau^{-1})^j(u),(\tau^{-1})^j(v))$" such that $L \leq \min(|u'|,|v'|) \leq 2L$.

Preimages

Example: intuitive definition

Recall that

$$\mathcal{T} = 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdot 02 \cdot 01 \cdot 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdots$$

Preimages

Example: intuitive definition
Recall that

$$\mathcal{T} = 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdot 02 \cdot 01 \cdot 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdots$$

Let $u = 2010102010$.

Preimages

Example: intuitive definition

Recall that

$$\mathcal{T} = 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdot \overbrace{02}^{\tau(1)} \cdot \overbrace{01}^{\tau(0)} \cdot \overbrace{01}^{\tau(0)} \cdot \overbrace{02}^{\tau(1)} \cdot \overbrace{01}^{\tau(0)} \cdot \overbrace{0}^{\tau(2)} \cdot 01 \cdots$$

Let $u = 2010102010$.

Preimages

Example: intuitive definition

Recall that

$$\mathcal{T} = 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdot \overbrace{02}^{\tau(1)} \cdot \overbrace{01}^{\tau(0)} \cdot \overbrace{01}^{\tau(0)} \cdot \overbrace{02}^{\tau(1)} \cdot \overbrace{01}^{\tau(0)} \cdot \overbrace{0}^{\tau(2)} \cdot 01 \cdots$$

Let $u = 2010102010$. The word $u' = 100102$ is a *preimage* of u.

Preimages

Example: intuitive definition

Recall that

$$\mathcal{T} = 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdot \overbrace{02}^{\tau(1)} \cdot \overbrace{01}^{\tau(0)} \cdot \overbrace{01}^{\tau(0)} \cdot \overbrace{02}^{\tau(1)} \cdot \overbrace{01}^{\tau(0)} \cdot \overbrace{0}^{\tau(2)} \cdot 01 \cdots$$

Let $u = 2010102010$. The word $u' = 100102$ is a *preimage* of u.

Definition

Let u and u' be two words. The word u' is a **preimage** of u if

- u is a factor of $\tau(u')$, and
- u' is minimal: for all factors v of u', u is not a factor of $\tau(v)$.

Preimages (continued)

A word can have several preimages.

Example
Recall that

$$\mathcal{T} = 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdot 02 \cdot 01 \cdot 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdots$$

Take $u = 010$.

Preimages (continued)

A word can have several preimages.

Example
Recall that

$$\mathcal{T} = 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdot 02 \cdot 01 \cdot 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdots$$

Take $u = 010$. It has 00, 01 and 02 as preimages.

Preimages (continued)

A word can have several preimages.

Example
Recall that

$$\mathcal{T} = 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdot 02 \cdot \overbrace{01}^{\tau(0)} \cdot \overbrace{01}^{\tau(0)} \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdots$$

Take $u = 010$. It has 00, 01 and 02 as preimages.

Preimages (continued)

A word can have several preimages.

Example

Recall that

$$\mathcal{T} = \overbrace{01}^{\tau(0)} \cdot \overbrace{02}^{\tau(1)} \cdot 01 \cdot 0 \cdot 01 \cdot 02 \cdot 01 \cdot 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdots$$

Take $u = 010$. It has 00, 01 and 02 as preimages.

Preimages (continued)

A word can have several preimages.

Example
Recall that

$$\mathcal{T} = 01 \cdot 02 \cdot \overbrace{01}^{\tau(0)} \cdot \overbrace{0}^{\tau(2)} \cdot 01 \cdot 02 \cdot 01 \cdot 01 \cdot 02 \cdot 01 \cdot 0 \cdot 01 \cdots$$

Take $u = 010$. It has 00, 01 and 02 as preimages.

Templates have parents...

We will now introduce the notion of parents of a template.

Theorem

Let t be a template and let (u, v) be a pair of factors realizing t. Let u' (resp., v') be a preimage of u (resp., v).
There always exists a template t' which is realized by (u', v'). and which is, *in some way*, related to t.

$$(u, v) \quad \leftrightsquigarrow \quad [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2] = t$$

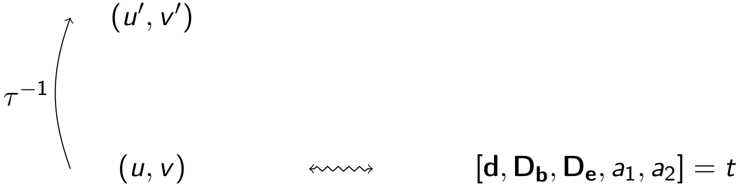
Templates have parents...

We will now introduce the notion of parents of a template.

Theorem

Let t be a template and let (u, v) be a pair of factors realizing t. Let u' (resp., v') be a preimage of u (resp., v).

There always exists a template t' which is realized by (u', v') and which is, *in some way*, related to t.



Templates have parents...

We will now introduce the notion of parents of a template.

Theorem

Let t be a template and let (u, v) be a pair of factors realizing t. Let u' (resp., v') be a preimage of u (resp., v).
There always exists a template t' which is realized by (u', v') and which is, *in some way*, related to t.

$$(u', v') \quad \leftrightsquigarrow \quad [\mathbf{d}', \mathbf{D}'_{\mathbf{b}}, \mathbf{D}'_{\mathbf{e}}, a'_1, a'_2] = t'$$

$$\tau^{-1} \uparrow$$

$$(u, v) \quad \leftrightsquigarrow \quad [\mathbf{d}, \mathbf{D}_{\mathbf{b}}, \mathbf{D}_{\mathbf{e}}, a_1, a_2] = t$$

Templates have parents...

We will now introduce the notion of parents of a template.

Theorem

Let t be a template and let (u, v) be a pair of factors realizing t. Let u' (resp., v') be a preimage of u (resp., v).
There always exists a template t' which is realized by (u', v') and which is, *in some way*, related to t.

The template t' is called a **parent template** of t.

$$(u', v') \leftrightsquigarrow [\mathbf{d}', \mathbf{D}'_{\mathbf{b}}, \mathbf{D}'_{\mathbf{e}}, a'_1, a'_2] = t'$$

$$\tau^{-1} \uparrow \qquad\qquad \uparrow \mathrm{Par}(.)$$

$$(u, v) \leftrightsquigarrow [\mathbf{d}, \mathbf{D}_{\mathbf{b}}, \mathbf{D}_{\mathbf{e}}, a_1, a_2] = t$$

Templates have parents...

Remark

- Since a word can sometimes have several preimages, a template can also have several parents.

Templates have parents...

Remark

- Since a word can sometimes have several preimages, a template can also have several parents.
- There exists a formula allowing to compute all parents of a given template.

... and ancestors

Definition

Let t and t' be templates. We say that t' is an **(realizable) ancestor** of t if there exists a finite sequence of templates $t_0, \ldots, t_n$ such that

$$\begin{cases} t_0 = t', \\ t_n = t, \\ \forall i \in \{0, \ldots, n-1\}, t_i \text{ is a (realizable) parent of } t_{i+1}. \end{cases}$$

The formal theorem

We can now state the formal theorem:

We can now state the formal theorem:

Theorem

Let $L \geq 0$ be an integer, and let t be a template. If there exists a pair of factors (u, v) realizing t, then

The formal theorem

We can now state the formal theorem:

Theorem

Let $L \geq 0$ be an integer, and let t be a template. If there exists a pair of factors (u, v) realizing t, then

- either $\min(|u|, |v|) \leq L$, or

The formal theorem

We can now state the formal theorem:

Theorem

Let $L \geq 0$ be an integer, and let t be a template. If there exists a pair of factors (u, v) realizing t, then

- either $\min(|u|, |v|) \leq L$, or
- there exist an ancestor t' of t, and a pair (u', v') of factors realizing t', such that $L \leq \min(|u'|, |v'|) \leq 2L$.

k-binomial complexity of Tribonacci

1 Preliminary definitions
- Words, factors and subwords
- Complexity functions
- k-binomial complexity

2 State of the art

3 Next result: the Tribonacci word
- Definition
- The theorem
- Introduction to templates and their parents
- Using templates to compute $\mathbf{b}_T^{(2)}$

Verifying that no template $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ is realizable

To show that $\mathbf{b}_{\mathcal{T}}^{(2)} = p_{\mathcal{T}}$, we have to show that no template from

$$T := \{[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b] \,:\, a \neq b\}$$

is realizable.

Verifying that no template $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ is realizable

To show that $\mathbf{b}_{\mathcal{T}}^{(2)} = p_{\mathcal{T}}$, we have to show that no template from

$$T := \{[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b] \,:\, a \neq b\}$$

is realizable.

The following steps can be done using Mathematica:

Verifying that no template $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ is realizable

To show that $\mathbf{b}_{\mathcal{T}}^{(2)} = p_{\mathcal{T}}$, we have to show that no template from

$$T := \{[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b] \,:\, a \neq b\}$$

is realizable.

The following steps can be done using Mathematica:

1. We check that no pair of factors (u, v) with $\min(|u|, |v|) \leq L$ realizes a template of T.

Verifying that no template $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ is realizable

To show that $\mathbf{b}_{\mathcal{T}}^{(2)} = p_{\mathcal{T}}$, we have to show that no template from

$$T := \{[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b] \,:\, a \neq b\}$$

is realizable.

The following steps can be done using Mathematica:

1. We check that no pair of factors (u, v) with $\min(|u|, |v|) \leq L$ realizes a template of T.
2. We compute all the ancestors of T and we check that none of them is realized by a pair (u', v') with $L \leq \min(|u'|, |v'|) \leq 2L$.

Verifying that no template $[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b]$ is realizable

To show that $\mathbf{b}_{\mathcal{T}}^{(2)} = p_{\mathcal{T}}$, we have to show that no template from

$$T := \{[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b] \,:\, a \neq b\}$$

is realizable.

The following steps can be done using Mathematica:

1. We check that no pair of factors (u, v) with $\min(|u|, |v|) \leq L$ realizes a template of T.
2. We compute all the ancestors of T and we check that none of them is realized by a pair (u', v') with $L \leq \min(|u'|, |v'|) \leq 2L$.

<u>**Problem:**</u> there exists an infinite number of ancestors.

Keeping a finite number of templates

Instead of computing all the ancestors of T, we will focus on the **possibly** realizable ones.

Keeping a finite number of templates

Instead of computing all the ancestors of T, we will focus on the **possibly** realizable ones.

The last step is thus to find necessary conditions on templates to be realizable.

Keeping a finite number of templates

Instead of computing all the ancestors of T, we will focus on the **possibly** realizable ones.

The last step is thus to find necessary conditions on templates to be realizable.

That will leave us with a finite number of candidates.

It is then possible to verify with a computer that, in fact, none of them is realizable by a pair (u, v) with $L \leq \min(|u|, |v|) \leq 2L$.

A matrix associated to τ

Let us consider the matrix associated to τ : $\begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$

A matrix associated to τ

Let us consider the matrix associated to τ : $\begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$

It has the property that, for all $u \in \mathrm{Fac}_{\mathcal{T}}$, $M'_\tau \Psi(u) = \Psi(\tau(u))$.

A matrix associated to τ

Let us consider the matrix associated to $\tau : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$

It has the property that, for all $u \in \mathrm{Fac}_{\mathcal{T}}$, $M'_\tau \Psi(u) = \Psi(\tau(u))$.
We have

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

because

A matrix associated to τ

Let us consider the matrix associated to $\tau : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$

It has the property that, for all $u \in \mathrm{Fac}_{\mathcal{T}}$, $M'_\tau \Psi(u) = \Psi(\tau(u))$.
We have

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

because

$$M'_\tau \Psi(u) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \binom{u}{0} \\ \binom{u}{1} \\ \binom{u}{2} \end{pmatrix}$$

A matrix associated to τ

Let us consider the matrix associated to τ : $\begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$

It has the property that, for all $u \in \mathrm{Fac}_{\mathcal{T}}$, $M'_\tau \Psi(u) = \Psi(\tau(u))$.
We have

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

because

$$M'_\tau \Psi(u) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \binom{u}{0} \\ \binom{u}{1} \\ \binom{u}{2} \end{pmatrix} = \begin{pmatrix} \binom{u}{0} + \binom{u}{1} + \binom{u}{2} \\ \binom{u}{0} \\ \binom{u}{1} \end{pmatrix}$$

A matrix associated to τ

Let us consider the matrix associated to $\tau : \begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$

It has the property that, for all $u \in \mathrm{Fac}_{\mathcal{T}}$, $M'_\tau \Psi(u) = \Psi(\tau(u))$.
We have

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

because

$$M'_\tau \Psi(u) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \binom{u}{0} \\ \binom{u}{1} \\ \binom{u}{2} \end{pmatrix} = \begin{pmatrix} \binom{u}{0} + \binom{u}{1} + \binom{u}{2} \\ \binom{u}{0} \\ \binom{u}{1} \end{pmatrix} = \begin{pmatrix} \binom{\tau(u)}{0} \\ \binom{\tau(u)}{1} \\ \binom{\tau(u)}{2} \end{pmatrix}.$$

A matrix associated to τ

Let us consider the matrix associated to τ : $\begin{cases} 0 & \mapsto & 01; \\ 1 & \mapsto & 02; \\ 2 & \mapsto & 0. \end{cases}$

It has the property that, for all $u \in \mathrm{Fac}_{\mathcal{T}}$, $M'_\tau \Psi(u) = \Psi(\tau(u))$.
We have

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

because

$$M'_\tau \Psi(u) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \binom{u}{0} \\ \binom{u}{1} \\ \binom{u}{2} \end{pmatrix} = \begin{pmatrix} \binom{u}{0} + \binom{u}{1} + \binom{u}{2} \\ \binom{u}{0} \\ \binom{u}{1} \end{pmatrix} = \begin{pmatrix} \binom{\tau(u)}{0} \\ \binom{\tau(u)}{1} \\ \binom{\tau(u)}{2} \end{pmatrix}.$$

We define its extended version M_τ, such that, for all $u \in \mathrm{Fac}_{\mathcal{T}}$, we have $M_\tau \Phi(u) = \Phi(\tau(u))$.

The extended version

We have

$$M_\tau = \left(\begin{array}{ccc|ccc|ccc|ccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
\hline
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right)$$

The extended version

We have

$$M_\tau = \left(\begin{array}{ccc|ccc|ccc|ccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
\hline
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right)$$

The extended version

We have

$$M_\tau = \left(\begin{array}{ccc|ccc|ccc|ccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
\hline
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right)$$

For example, since $\tau(0) = 01$ and 1 is present only in $\tau(0)$ while 0 occurs once in every $\tau(a)$,

$$\binom{\tau(u)}{01} = \binom{u}{0} + \binom{u}{00} + \binom{u}{10} + \binom{u}{20}.$$

The extended version

We have

$$M_\tau \cdot \Phi(u) = \left(\begin{array}{ccc|ccc|ccc|ccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \color{red}1 & \color{red}0 & \color{red}0 & \color{red}1 & \color{red}0 & \color{red}0 & \color{red}1 & \color{red}0 & \color{red}0 & \color{red}1 & \color{red}0 & \color{red}0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right) \begin{pmatrix} \binom{u}{0} \\ \binom{u}{1} \\ \binom{u}{2} \\ \binom{u}{00} \\ \binom{u}{01} \\ \binom{u}{02} \\ \binom{u}{10} \\ \binom{u}{11} \\ \binom{u}{12} \\ \binom{u}{20} \\ \binom{u}{21} \\ \binom{u}{22} \end{pmatrix}$$

For example, since $\tau(0) = 01$ and 1 is present only in $\tau(0)$ while 0 occurs once in every $\tau(a)$,

$$\color{red}{\binom{\tau(u)}{01}} = \binom{u}{0} + \binom{u}{00} + \binom{u}{10} + \binom{u}{20}.$$

The extended version

We have

$$M_\tau \cdot \Phi(u) = \left(\begin{array}{ccc|ccc|ccc|ccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
\hline
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right) \begin{pmatrix} \binom{u}{0} \\ \binom{u}{1} \\ \binom{u}{2} \\ \binom{u}{00} \\ \binom{u}{01} \\ \binom{u}{02} \\ \binom{u}{10} \\ \binom{u}{11} \\ \binom{u}{12} \\ \binom{u}{20} \\ \binom{u}{21} \\ \binom{u}{22} \end{pmatrix}$$

For example, since $\tau(0) = 01$ and 1 is present only in $\tau(0)$ while 0 occurs once in every $\tau(a)$,

$$\binom{\tau(u)}{01} = \binom{u}{0} + \binom{u}{00} + \binom{u}{10} + \binom{u}{20}.$$

The extended version

We have

$$M_\tau \cdot \Phi(u) = \left(\begin{array}{ccc|ccc|ccc|ccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right)
\begin{pmatrix}
\binom{u}{0} \\ \binom{u}{1} \\ \binom{u}{2} \\ \binom{u}{00} \\ \binom{u}{01} \\ \binom{u}{02} \\ \binom{u}{10} \\ \binom{u}{11} \\ \binom{u}{12} \\ \binom{u}{20} \\ \binom{u}{21} \\ \binom{u}{22}
\end{pmatrix}$$

For example, since $\tau(0) = 01$ and 1 is present only in $\tau(0)$ while 0 occurs once in every $\tau(a)$,

$$\binom{\tau(u)}{01} = \binom{u}{0} + \binom{u}{00} + \binom{u}{10} + \binom{u}{20}.$$

The extended version

We have

$$M_\tau \cdot \Phi(u) = \left(\begin{array}{ccc|ccc|ccc|ccc} 1&1&1&0&0&0&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0\\ \hline 0&0&0&1&1&1&1&1&1&1&1&1\\ \color{red}1&\color{red}0&\color{red}0&\color{red}1&\color{red}0&\color{red}0&\color{blue}1&\color{red}0&\color{red}0&\color{red}1&\color{red}0&\color{red}0\\ 0&1&0&0&1&0&0&1&0&0&1&0\\ \hline 0&0&0&1&1&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&0&1&1&1&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0 \end{array}\right) \begin{pmatrix} \binom{u}{0}\\ \binom{u}{1}\\ \binom{u}{2}\\ \binom{u}{00}\\ \binom{u}{01}\\ \binom{u}{02}\\ \color{blue}\binom{u}{10}\\ \binom{u}{11}\\ \binom{u}{12}\\ \binom{u}{20}\\ \binom{u}{21}\\ \binom{u}{22} \end{pmatrix}$$

For example, since $\tau(0) = 01$ and 1 is present only in $\tau(0)$ while 0 occurs once in every $\tau(a)$,

$$\color{red}\binom{\tau(u)}{01}\color{black} = \binom{u}{0} + \binom{u}{00} + \color{blue}\binom{u}{10}\color{black} + \binom{u}{20}.$$

The extended version

We have

$$M_\tau \cdot \Phi(u) = \left(\begin{array}{ccc|ccc|ccc|ccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \color{red}1 & \color{red}0 & \color{red}0 & \color{red}1 & \color{red}0 & \color{red}0 & \color{red}1 & \color{red}0 & \color{red}0 & \color{blue}1 & \color{red}0 & \color{red}0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right) \begin{pmatrix} \binom{u}{0} \\ \binom{u}{1} \\ \binom{u}{2} \\ \binom{u}{00} \\ \binom{u}{01} \\ \binom{u}{02} \\ \binom{u}{10} \\ \binom{u}{11} \\ \binom{u}{12} \\ \color{blue}\binom{u}{20} \\ \binom{u}{21} \\ \binom{u}{22} \end{pmatrix}$$

For example, since $\tau(0) = 01$ and 1 is present only in $\tau(0)$ while 0 occurs once in every $\tau(a)$,

$$\color{red}\binom{\tau(u)}{01} \color{black} = \binom{u}{0} + \binom{u}{00} + \binom{u}{10} + \color{blue}\binom{u}{20}\color{black}.$$

About its eigenvalues

The Perron-Frobenius eigenvalue of M'_τ is $\theta \approx 1.839$. The matrix M_τ has

- the eigenvalue θ once;
- the eigenvalue θ^2 once;
- two pairs of complex conjugate eigenvalues of modulus in $]1;\theta[$;
- a real eigenvalue of modulus less than 1, of geometric multiplicity 2;
- two pairs of complex conjugate eigenvalues of modulus less than 1.

About its eigenvalues

The Perron-Frobenius eigenvalue of M'_τ is $\theta \approx 1.839$. The matrix M_τ has

- the eigenvalue θ once;
- the eigenvalue θ^2 once;
- two pairs of complex conjugate eigenvalues of modulus in $]1;\theta[$;
- a real eigenvalue of modulus less than 1, of geometric multiplicity 2;
- two pairs of complex conjugate eigenvalues of modulus less than 1.

The bounds we will give on possibly realizable templates will concern projections of templates on the left eigenvectors associated to eigenvalues of modulus less than θ.

First restrictions

Theorem

Let λ be an eigenvalue of modulus less than 1. Let $\mathbf{r}$ be an associated eigenvector. If the template $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ is realizable, then

$$\min_{\boldsymbol{\delta} \in \Delta} |\mathbf{r} \cdot (\mathbf{d} + P_3 (\mathbf{D_b} \otimes \boldsymbol{\delta} + \boldsymbol{\delta} \otimes \mathbf{D_e}))| \leq 2C(\mathbf{r}),$$

where $C(\mathbf{r})$ is an easily computable constant such that, for all factors $w \in \mathrm{Fac}_{\mathcal{T}}$, we have

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

First restrictions

Theorem

Let λ be an eigenvalue of modulus less than 1. Let $\mathbf{r}$ be an associated eigenvector. If the template $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ is realizable, then

$$\min_{\boldsymbol{\delta} \in \Delta} |\mathbf{r} \cdot (\mathbf{d} + P_3 (\mathbf{D_b} \otimes \boldsymbol{\delta} + \boldsymbol{\delta} \otimes \mathbf{D_e}))| \leq 2C(\mathbf{r}),$$

where $C(\mathbf{r})$ is an easily computable constant such that, for all factors $w \in \mathrm{Fac}_{\mathcal{T}}$, we have

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

For the sake of notations, we wrote

$$\Delta = \left\{ \begin{pmatrix} \delta_0 \\ \delta_1 \\ \delta_2 \end{pmatrix} \in [-1.5; 1.5]^3 \, : \, \delta_0 + \delta_1 + \delta_2 = 0 \right\}.$$

First restrictions

Theorem

Let λ be an eigenvalue of modulus less than 1. Let $\mathbf{r}$ be an associated eigenvector. If the template $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ is realizable, then

$$\min_{\boldsymbol{\delta}\in\Delta} |\mathbf{r}\cdot(\mathbf{d} + P_3(\mathbf{D_b}\otimes\boldsymbol{\delta} + \boldsymbol{\delta}\otimes\mathbf{D_e}))| \leq 2C(\mathbf{r}),$$

where $C(\mathbf{r})$ is an easily computable constant such that, for all factors $w \in \mathrm{Fac}_{\mathcal{T}}$, we have

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

For the sake of notations, we wrote

$$\Delta = \left\{ \begin{pmatrix} \delta_0 \\ \delta_1 \\ \delta_2 \end{pmatrix} \in [-1.5; 1.5]^3 \;:\; \delta_0 + \delta_1 + \delta_2 = 0 \right\}.$$

Other restrictions

Theorem

Let λ be an eigenvalue of modulus in $]1,\theta[$. Let $\mathbf{r}$ be an associated eigenvector. If the template $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ is realizable by a pair (u, v) with $|u| \geq L$, then

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

where $C(\mathbf{r})$ is an easily computable constant such that, for all factors $w \in \mathrm{Fac}_{\mathcal{T}}$, we have

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

For the sake of notations, we wrote $\boldsymbol{\alpha} = \begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 \end{pmatrix}^{\mathsf{T}}$ the vector of densities of letters in $\mathcal{T}$.

Other restrictions

Theorem

Let λ be an eigenvalue of modulus in $]1, \theta[$. Let $\mathbf{r}$ be an associated eigenvector. If the template $t = [\mathbf{d}, \mathbf{D_b}, \mathbf{D_e}, a_1, a_2]$ is realizable by a pair (u, v) with $|u| \geq L$, then

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

where $C(\mathbf{r})$ is an easily computable constant such that, for all factors $w \in \mathrm{Fac}_{\mathcal{T}}$, we have

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

For the sake of notations, we wrote $\boldsymbol{\alpha} = \begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 \end{pmatrix}^{\mathsf{T}}$ the vector of densities of letters in $\mathcal{T}$.

Final algorithm

Recall that we want to compute the ancestors of T and check that none of them is realized by a pair (u, v) with

$$L \leq \min(|u|, |v|) \leq 2L. \qquad (\star)$$

Final algorithm

Recall that we want to compute the ancestors of T and check that none of them is realized by a pair (u, v) with

$$L \leq \min(|u|, |v|) \leq 2L. \qquad (\star)$$

- Initialize `toSee = T` and `seen = {}`

Final algorithm

Recall that we want to compute the ancestors of T and check that none of them is realized by a pair (u, v) with

$$L \leq \min(|u|, |v|) \leq 2L. \qquad (\star)$$

- Initialize `toSee = T` and `seen =` $\{\}$
- While `toSee` $\neq \{\}$, take a template `t` $\in$ `toSee`

Final algorithm

Recall that we want to compute the ancestors of T and check that none of them is realized by a pair (u, v) with

$$L \leq \min(|u|, |v|) \leq 2L. \qquad (\star)$$

- Initialize `toSee = T` and `seen = {}`
- While `toSee` $\neq \{\}$, take a template `t` $\in$ `toSee`
  - Compute its parents

Final algorithm

Recall that we want to compute the ancestors of T and check that none of them is realized by a pair (u, v) with

$$L \leq \min(|u|, |v|) \leq 2L. \qquad (\star)$$

- Initialize `toSee = T` and `seen =` $\{\}$
- While `toSee` $\neq \{\}$, take a template `t` $\in$ `toSee`
  - Compute its parents
  - Keep, among them, only the ones that verify the previous theorems for all eigenvalues of modulus less than θ

Final algorithm

Recall that we want to compute the ancestors of T and check that none of them is realized by a pair (u, v) with

$$L \leq \min(|u|, |v|) \leq 2L. \qquad (\star)$$

- Initialize `toSee = T` and `seen = {}`
- While `toSee` $\neq \{\}$, take a template `t` $\in$ `toSee`
  - Compute its parents
  - Keep, among them, only the ones that verify the previous theorems for all eigenvalues of modulus less than θ
  - Add them in `toSee`, if they are neither in `toSee`, nor in `seen`

Final algorithm

Recall that we want to compute the ancestors of T and check that none of them is realized by a pair (u, v) with

$$L \leq \min(|u|, |v|) \leq 2L. \qquad (\star)$$

- Initialize `toSee = T` and `seen =` $\{\}$
- While `toSee` $\neq \{\}$, take a template `t` $\in$ `toSee`
  - Compute its parents
  - Keep, among them, only the ones that verify the previous theorems for all eigenvalues of modulus less than θ
  - Add them in `toSee`, if they are neither in `toSee`, nor in `seen`
  - `toSee = toSee\{t}` and `seen = seen` $\cup$ `{t}`

Final algorithm

Recall that we want to compute the ancestors of T and check that none of them is realized by a pair (u, v) with

$$L \leq \min(|u|, |v|) \leq 2L. \qquad (\star)$$

- Initialize `toSee = T` and `seen = {}`
- While `toSee` $\neq \{\}$, take a template `t` $\in$ `toSee`
  - Compute its parents
  - Keep, among them, only the ones that verify the previous theorems for all eigenvalues of modulus less than θ
  - Add them in `toSee`, if they are neither in `toSee`, nor in `seen`
  - `toSee = toSee\{t}` and `seen = seen` $\cup$ `{t}`
- If the program stops, `seen` contains all the possibly realizable ancestors of T, which is a finite set

Final algorithm

Recall that we want to compute the ancestors of T and check that none of them is realized by a pair (u, v) with

$$L \leq \min(|u|, |v|) \leq 2L. \qquad (\star)$$

- Initialize `toSee = T` and `seen = {}`
- While `toSee` $\neq \{\}$, take a template `t` $\in$ `toSee`
  - Compute its parents
  - Keep, among them, only the ones that verify the previous theorems for all eigenvalues of modulus less than θ
  - Add them in `toSee`, if they are neither in `toSee`, nor in `seen`
  - `toSee = toSee\{t}` and `seen = seen` $\cup$ `{t}`
- If the program stops, `seen` contains all the possibly realizable ancestors of T, which is a finite set
- Check that none of them is realized by a pair (u, v) of factors of $\mathcal{T}$ satisfying $(\star)$

Our implementation

In our implementation, we took $L = 15$.

Computing all the possibly realizable ancestors gave us a list of 241544 elements, in more or less eight hours.

The program then checks in less than three hours that none of them is realized.

Our implementation

In our implementation, we took $L = 15$.

Computing all the possibly realizable ancestors gave us a list of 241544 elements, in more or less eight hours.

The program then checks in less than three hours that none of them is realized.

Thus, no template from $T = \{[\mathbf{0}, \mathbf{0}, \mathbf{0}, a, b] \; : \; a \neq b\}$ is realizable.

That implies that $p_{\mathcal{T}}(n) = \mathbf{b}_{\mathcal{T}}^{(2)}(n)$ for all $n \in \mathbb{N}$.

Thank you!