About the k-binomial equivalence and the associated complexity

March 07, 2019

Marie Lejeune (FNRS grantee)

Plan

Plan

1. Introduction
   - Morphisms and infinite words
   - Factors and subwords
   - Factor complexity function
   - Other complexity functions
2. Some results about the k-binomial complexity
   - Sturmian words
   - The Thue–Morse word
   - The Tribonacci word
3. Better understanding of $\sim_k$

Definition

A *morphism* on the alphabet A is an application

$$\sigma : A^* \to A^*$$

such that, for every word $u = u_1 \cdots u_n \in A^*$,

$$\sigma(u) = \sigma(u_1) \cdots \sigma(u_n).$$

Morphisms

Definition

A *morphism* on the alphabet A is an application

$$\sigma : A^* \to A^*$$

such that, for every word $u = u_1 \cdots u_n \in A^*$,

$$\sigma(u) = \sigma(u_1) \cdots \sigma(u_n).$$

If there exists a letter $a \in A$ such that $\sigma(a)$ begins by a, and if $\lim_{n\to+\infty} |\sigma^n(a)| = +\infty$, then one can define

$$\sigma^\omega(a) = \lim_{n\to+\infty} \sigma^n(a).$$

This infinite word is called a *fixed point* of the morphism σ.

Example (Thue–Morse)

Let us define the *Thue–Morse morphism*

$$\varphi : \{0,1\}^* \to \{0,1\}^* : \begin{cases} 0 \mapsto 01; \\ 1 \mapsto 10. \end{cases}$$

Example (Thue–Morse)

Let us define the *Thue–Morse morphism*

$$\varphi : \{0,1\}^* \to \{0,1\}^* : \begin{cases} 0 \mapsto 01; \\ 1 \mapsto 10. \end{cases}$$

We have

$$\begin{aligned} \varphi(0) &= 01, \\ \varphi^2(0) &= 0110, \\ \varphi^3(0) &= 01101001, \\ &\cdots \end{aligned}$$

Example (Thue–Morse)

Let us define the *Thue–Morse morphism*

$$\varphi : \{0,1\}^* \to \{0,1\}^* : \begin{cases} 0 \mapsto 01; \\ 1 \mapsto 10. \end{cases}$$

We have

$$\begin{aligned} \varphi(0) &= 01, \\ \varphi^2(0) &= 0110, \\ \varphi^3(0) &= 01101001, \\ &\dots \end{aligned}$$

We can thus define the *Thue–Morse word* as one of the fixed points of the morphism φ :

$$\mathbf{t} := \varphi^\omega(0) = 0110100110010110\cdots$$

Plan

Factors and subwords

Definition

Let $u = u_1 \cdots u_m \in A^m$ be a word ($m \in \mathbb{N}^+ \cup \{\infty\}$).
A *(scattered) subword of* u is a finite subsequence of the sequence $(u_j)_{j=1}^m$.
A *factor of* u is a subword made with consecutive letters.
Otherwise stated, every (non empty) factor of u is of the form
$u_i u_{i+1} \cdots u_{i+\ell}$, with $1 \leq i \leq m$, $0 \leq \ell \leq m - i$.

Factors and subwords

Definition

Let $u = u_1 \cdots u_m \in A^m$ be a word ($m \in \mathbb{N}^+ \cup \{\infty\}$).
A *(scattered) subword of* u is a finite subsequence of the sequence $(u_j)_{j=1}^m$.
A *factor of* u is a subword made with consecutive letters.
Otherwise stated, every (non empty) factor of u is of the form $u_i u_{i+1} \cdots u_{i+\ell}$, with $1 \leq i \leq m$, $0 \leq \ell \leq m - i$.

Example

Let us consider the alphabet $\{0, 1, 2\}$. Let $u = 0102010$.
The word 021 is a subword of u, but it is not a factor of u.

Factors and subwords

Definition

Let $u = u_1 \cdots u_m \in A^m$ be a word ($m \in \mathbb{N}^+ \cup \{\infty\}$).
A *(scattered) subword of* u is a finite subsequence of the sequence $(u_j)_{j=1}^m$.
A *factor of* u is a subword made with consecutive letters.
Otherwise stated, every (non empty) factor of u is of the form
$u_i u_{i+1} \cdots u_{i+\ell}$, with $1 \leq i \leq m$, $0 \leq \ell \leq m - i$.

Example

Let us consider the alphabet $\{0, 1, 2\}$. Let $u = 0102010$.
The word 021 is a subword of u, but it is not a factor of u.
The word 0201 is a factor of u, thus also a subword of u.

Factors and subwords

Definition

Let $u = u_1 \cdots u_m \in A^m$ be a word ($m \in \mathbb{N}^+ \cup \{\infty\}$).
A *(scattered) subword of* u is a finite subsequence of the sequence $(u_j)_{j=1}^m$.
A *factor of* u is a subword made with consecutive letters.
Otherwise stated, every (non empty) factor of u is of the form $u_i u_{i+1} \cdots u_{i+\ell}$, with $1 \leq i \leq m$, $0 \leq \ell \leq m - i$.

Example

Let us consider the alphabet $\{0, 1, 2\}$. Let $u = 0102010$.
The word 021 is a subword of u, but it is not a factor of u.
The word 0201 is a factor of u, thus also a subword of u.

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

Plan

1. Introduction
   - Morphisms and infinite words
   - Factors and subwords
   - Factor complexity function
   - Other complexity functions

2. Some results about the k-binomial complexity
   - Sturmian words
   - The Thue–Morse word
   - The Tribonacci word

3. Better understanding of $\sim_k$

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 $\mathbf{w}$.

The simplest complexity function is the following. Here, $\mathbb{N} = \{0, 1, 2, \ldots\}$.

Definition

The *factor complexity* of the word $\mathbf{w}$ is the function

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

Factor complexity of the Thue–Morse word

Example

Let us compute the first values of the Thue–Morse's factor complexity.
We have
$$\mathbf{t} = 0110100110010110\cdots$$

Factor complexity of the Thue–Morse word

Example

Let us compute the first values of the Thue–Morse's factor complexity. We have

$$\mathbf{t} = 0110100110010110\cdots$$

and

n	0	1	2	3	$\cdots$
$p_{\mathbf{t}}(n)$	1				

Factor complexity of the Thue–Morse word

Example

Let us compute the first values of the Thue–Morse's factor complexity. We have

$$\mathbf{t} = 0110100110010110\cdots$$

and

n	0	1	2	3	$\cdots$
$p_{\mathbf{t}}(n)$	1	2			

Factor complexity of the Thue–Morse word

Example

Let us compute the first values of the Thue–Morse's factor complexity. We have

$$\mathbf{t} = 0110100110010110\cdots$$

and

n	0	1	2	3	$\cdots$
$p_{\mathbf{t}}(n)$	1	2	4		

Factor complexity of the Thue–Morse word

Example

Let us compute the first values of the Thue–Morse's factor complexity. We have

$$\mathbf{t} = 01101001100101101001011001101001\cdots$$

and

n	0	1	2	3	$\cdots$
$p_{\mathbf{t}}(n)$	1	2	4	6	

Factor complexity of the Thue–Morse word

Example

Let us compute the first values of the Thue–Morse's factor complexity. We have

$$\mathbf{t} = 0110100110010110\cdots$$

and

n	0	1	2	3	$\cdots$
$p_{\mathbf{t}}(n)$	1	2	4	6	$\cdots$

Then, for every $n \geq 3$, it is known that

$$p_{\mathbf{t}}(n) = \begin{cases} 4n - 2 \cdot 2^m - 4, & \text{if } 2 \cdot 2^m < n \leq 3 \cdot 2^m; \\ 2n + 4 \cdot 2^m - 2, & \text{if } 3 \cdot 2^m < n \leq 4 \cdot 2^m. \end{cases}$$

Plan

1. Introduction
   - Morphisms and infinite words
   - Factors and subwords
   - Factor complexity function
   - Other complexity functions
2. Some results about the k-binomial complexity
   - Sturmian words
   - The Thue–Morse word
   - The Tribonacci word
3. Better understanding of $\sim_k$

From factor to k-binomial complexity

Let us rewrite the definition.

Definition

The *factor complexity* of the word $\mathbf{w}$ is the function

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

Let us rewrite the definition.

Definition

The *factor complexity* of the word $\mathbf{w}$ is the function

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

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

Let us rewrite the definition.

Definition

The *factor complexity* of the word $\mathbf{w}$ is the function

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

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

The relation $\sim_=$ can be replaced by other equivalence relations.

From factor to k-binomial complexity

For example, let us define,

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

From factor to k-binomial complexity

For example, let us define,
if $k \in \mathbb{N}^+$,

- Abelian equivalence: $u \sim_{ab,1} v \Leftrightarrow |u|_a = |v|_a \ \forall a \in A$
- k-abelian equivalence: $u \sim_{ab,k} v \Leftrightarrow |u|_x = |v|_x \ \forall x \in A^{\leq k}$

From factor to k-binomial complexity

For example, let us define,
if $k \in \mathbb{N}^+$,

- Abelian equivalence: $u \sim_{ab,1} v \Leftrightarrow |u|_a = |v|_a \ \forall a \in A$
- k-abelian equivalence: $u \sim_{ab,k} v \Leftrightarrow |u|_x = |v|_x \ \forall x \in A^{\leq k}$
- k-binomial equivalence: $u \sim_k v \Leftrightarrow \binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}$

From factor to k-binomial complexity

For example, let us define,
if $k \in \mathbb{N}^+$,

- Abelian equivalence: $u \sim_{ab,1} v \Leftrightarrow |u|_a = |v|_a \ \forall a \in A$
- k-abelian equivalence: $u \sim_{ab,k} v \Leftrightarrow |u|_x = |v|_x \ \forall x \in A^{\leq k}$
- k-binomial equivalence: $u \sim_k v \Leftrightarrow \binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}$

Let us illustrate the last one.

Binomial coefficients

Definition (Reminder)

Let u and x be two words. The *binomial coefficient* $\binom{u}{x}$ is the number of times that x appears as a subword in u.

Binomial coefficients

Definition (Reminder)

Let u and x be two words. The *binomial coefficient* $\binom{u}{x}$ is the number of times that x appears as a subword in u.

Example

If $u = aababa$,

$$\binom{u}{ab} = ?$$

Binomial coefficients

Definition (Reminder)

Let u and x be two words. The *binomial coefficient* $\binom{u}{x}$ is the number of times that x appears as a subword in u.

Example

If $u = aababa$,

$$\binom{u}{ab} = 1.$$

Binomial coefficients

Definition (Reminder)

Let u and x be two words. The *binomial coefficient* $\binom{u}{x}$ is the number of times that x appears as a subword in u.

Example

If $u = aababa$,

$$\binom{u}{ab} = 2.$$

Binomial coefficients

Definition (Reminder)

Let u and x be two words. The *binomial coefficient* $\binom{u}{x}$ is the number of times that x appears as a subword in u.

Example

If $u = aababa$,

$$\binom{u}{ab} = 3.$$

Binomial coefficients

Definition (Reminder)

Let u and x be two words. The *binomial coefficient* $\binom{u}{x}$ is the number of times that x appears as a subword in u.

Example

If $u = aababa$,

$$\binom{u}{ab} = 4.$$

Binomial coefficients

Definition (Reminder)

Let u and x be two words. The *binomial coefficient* $\binom{u}{x}$ is the number of times that x appears as a subword in u.

Example

If $u = aababa$,

$$\binom{u}{ab} = 5.$$

Binomial coefficients

Definition (Reminder)

Let u and x be two words. The *binomial coefficient* $\binom{u}{x}$ is the number of times that x appears as a subword in u.

Example

If $u = aababa$,

$$\binom{u}{ab} = 5.$$

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

k-binomial equivalence

Definition (Reminder)

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}, \binom{u}{ba} = 3 = \binom{v}{ba}.$$

Definition (Reminder)

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

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

Example

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

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

Remark

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

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

Remark

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

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

Remark

For all words u, v,

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

Remark

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

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

Remark

For all words u, v,

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

Definition (Reminder)

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

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

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

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

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

Example

For the Thue–Morse word $\mathbf{t}$, we have $\mathbf{b}_{\mathbf{t}}^{(1)}(0) = 1$ and, for every $n \geq 1$,

$$\mathbf{b}_{\mathbf{t}}^{(1)}(n) = \begin{cases} 3, & \text{if } n \equiv 0 \pmod 2; \\ 2, & \text{otherwise.} \end{cases}$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

- If $n = 2\ell$, every factor u is either composed of ℓ blocks or is composed of $\ell - 1$ blocks with one letter before and one letter after.

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

- If $n = 2\ell$, every factor u is either composed of ℓ blocks or is composed of $\ell - 1$ blocks with one letter before and one letter after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

- If $n = 2\ell$, every factor u is either composed of ℓ blocks or is composed of $\ell - 1$ blocks with one letter before and one letter after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

- If $n = 2\ell$, every factor u is either composed of ℓ blocks or is composed of $\ell - 1$ blocks with one letter before and one letter after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

- If $n = 2\ell$, every factor u is either composed of ℓ blocks or is composed of $\ell - 1$ blocks with one letter before and one letter after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

- If $n = 2\ell$, every factor u is either composed of ℓ blocks or is composed of $\ell - 1$ blocks with one letter before and one letter after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

- If $n = 2\ell$, every factor u is either composed of ℓ blocks or is composed of $\ell - 1$ blocks with one letter before and one letter after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

- If $n = 2\ell$, every factor u is either composed of ℓ blocks or is composed of $\ell - 1$ blocks with one letter before and one letter after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell, \ell + 1\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

- If $n = 2\ell$, every factor u is either composed of ℓ blocks or is composed of $\ell - 1$ blocks with one letter before and one letter after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell, \ell + 1\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

- If $n = 2\ell$, every factor u is either composed of ℓ blocks or is composed of $\ell - 1$ blocks with one letter before and one letter after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell, \ell + 1\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

Since $\mathbf{t}$ is the fixed point of φ, we have $\mathbf{t} = \varphi(\mathbf{t})$.

- If $n = 2\ell$, every factor u is either composed of ℓ blocks or is composed of $\ell - 1$ blocks with one letter before and one letter after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell, \ell + 1\}.$$

Thus, $\mathbf{b}_t^{(1)}(n) = 3$.

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 1{\color{red}0 \cdot 01 \cdot 10 \cdot 01} \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 0{\color{red}1 \cdot 10 \cdot 01 \cdot 01} \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell\}.$$

Computing $\mathbf{b}_t^{(1)}(n)$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell\}.$$

Example (proof)

- If $n = 2\ell - 1$, every factor u is composed of $\ell - 1$ blocks with one letter before or after.

We thus have

$$\mathbf{t} = 01 \cdot 10 \cdot 10 \cdot 01 \cdot 10 \cdot 01 \cdot 01 \cdot 10 \cdots$$

We obtain that

$$\binom{u}{0} \in \{\ell - 1, \ell\}.$$

Thus, $\mathbf{b}_t^{(1)}(n) = 2$.

Plan

Plan

Sturmian words

Definition (Reminder)

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

Definition (Reminder)

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 $\mathbf{b}_{\mathbf{w}}^{(2)}(n) = p_{\mathbf{w}}(n) = n + 1$.

Sturmian words

Definition (Reminder)

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 $\mathbf{b}_{\mathbf{w}}^{(2)}(n) = p_{\mathbf{w}}(n) = n + 1$.

Since for every infinite word $\mathbf{x}$,

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

we have $\mathbf{b}_{\mathbf{w}}^{(k)}(n) = p_{\mathbf{w}}(n) = n + 1$ for all $k \geq 2$.

Plan

Let **w** be a Sturmian word. We have

$$p_{\mathbf{w}}(n) < p_{\mathbf{t}}(n) \quad \forall n \geq 2.$$

This is not the case for the k-binomial complexity.

Theorem (M. Rigo, P. Salimov, 2015)

For every $k \geq 1$, there exists a constant $C_k > 0$ such that, for every $n \in \mathbb{N}$,

$$\mathbf{b}_{\mathbf{t}}^{(k)}(n) \leq C_k.$$

Why is the Thue–Morse word so interesting?

Let $\mathbf{w}$ be a Sturmian word. We have

$$p_{\mathbf{w}}(n) < p_{\mathbf{t}}(n) \quad \forall n \geq 2.$$

This is not the case for the k-binomial complexity.

Theorem (M. Rigo, P. Salimov, 2015)

For every $k \geq 1$, there exists a constant $C_k > 0$ such that, for every $n \in \mathbb{N}$,

$$\mathbf{b}_{\mathbf{t}}^{(k)}(n) \leq C_k.$$

In fact, this result holds for every infinite word which is a fixed point of a Parikh-constant morphism.

Parikh-constant morphisms

Definition

A morphism $\sigma : A^* \to A^*$ is *Parikh-constant* if, for all $a, b, c \in A$, $|\sigma(a)|_c = |\sigma(b)|_c$. Otherwise stated, images of the different letters have to be equal up to a permutation.

Parikh-constant morphisms

Definition

A morphism $\sigma : A^* \to A^*$ is *Parikh-constant* if, for all $a, b, c \in A$, $|\sigma(a)|_c = |\sigma(b)|_c$. Otherwise stated, images of the different letters have to be equal up to a permutation.

Example

The morphism

$$\sigma : \{0,1,2\}^* \to \{0,1,2\}^* : \begin{cases} 0 & \mapsto & 0112; \\ 1 & \mapsto & 1201; \\ 2 & \mapsto & 1120; \end{cases}$$

is Parikh-constant.

Back to Thue–Morse

We actually computed the exact value of $\mathbf{b}_t^{(k)}$ for all $n \in \mathbb{N}$.

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 $n \geq 2^k$,

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

Back to Thue–Morse

We actually computed the exact value of $\mathbf{b}_t^{(k)}$ for all $n \in \mathbb{N}$.

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 $n \geq 2^k$,

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

Open question : given $k \in \mathbb{N}$, can we find a word $\mathbf{w}$ which is a fixed point of a Parikh-constant morphism and such that there exists $N \in \mathbb{N}$ for which

$$\mathbf{b}_{\mathbf{w}}^{(k)}(n) < \mathbf{b}_{\mathbf{t}}^{(k)}(n) \quad \forall n \geq N \ ?$$

Plan

A ternary example: the Tribonacci word

Definition

The *Tribonacci word* is the fixed point $\mathbf{s} = \sigma^\omega(0)$ where σ is the morphism

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

$\mathbf{s} = 010201001020101 \cdots$.

A ternary example: the Tribonacci word

Definition

The *Tribonacci word* is the fixed point $\mathbf{s} = \sigma^\omega(0)$ where σ is the morphism

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

$\mathbf{s} = 010201001020101 \cdots$.

Once again, we computed the exact value of $\mathbf{b}_{\mathbf{s}}^{(k)}$.

Theorem (M. L., M. Rigo, M. Rosenfeld, 2019)

For all $n \in \mathbb{N}$, for all $k \in \mathbb{N}^{\geq 2}$, we have

$$\mathbf{b}_{\mathbf{s}}^{(k)}(n) = p_{\mathbf{s}}(n) = 2n + 1.$$

What about Arnoux-Rauzy words?

The Tribonacci word is a particular Arnoux-Rauzy word.

Definition

An Arnoux-Rauzy word is an infinite word $\mathbf{w}$ having factorial complexity $p_{\mathbf{w}}(n) = dn + 1$ for some $d \in \mathbb{N}$, with some additional properties.

If such a d exists, then $\mathbf{w}$ is built on a $(d-1)$-letter alphabet.

What about Arnoux-Rauzy words?

The Tribonacci word is a particular Arnoux-Rauzy word.

Definition

An Arnoux-Rauzy word is an infinite word $\mathbf{w}$ having factorial complexity $p_{\mathbf{w}}(n) = dn + 1$ for some $d \in \mathbb{N}$, with some additional properties.

If such a d exists, then $\mathbf{w}$ is built on a $(d-1)$-letter alphabet.

Conjecture

Let $\mathbf{w}$ be an Arnoux-Rauzy word. Then,

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

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

What about Arnoux-Rauzy words?

Remark

The proof of the theorem seems complicated to adapt to the general case. Indeed, we used the fact that $\mathbf{s}$ is 2-balanced. Otherwise stated, for all factors u and v of $\mathbf{s}$ of the same length, we knew that

$$||u|_a - |v|_a| \leq 2,$$

for all $a \in \{0, 1, 2\}$.
This is not always the case with Arnoux-Rauzy words. We know that some of them are not N-balanced for any $N \in \mathbb{N}$.

Plan

When are two words equivalent?

To help us to solve these problems, we would like to better understand when two words are k-binomially equivalent.

When are two words equivalent?

To help us to solve these problems, we would like to better understand when two words are k-binomially equivalent.

As far as we know, there exist polynomial-time algorithms deciding whether two words are k-binomially equivalent or not.

When are two words equivalent?

To help us to solve these problems, we would like to better understand when two words are k-binomially equivalent.

As far as we know, there exist polynomial-time algorithms deciding whether two words are k-binomially equivalent or not.

We would like to obtain some characterizations of words belonging to the same equivalence class for $\sim_k$.

When are two words equivalent?

To help us to solve these problems, we would like to better understand when two words are k-binomially equivalent.

As far as we know, there exist polynomial-time algorithms deciding whether two words are k-binomially equivalent or not.

We would like to obtain some characterizations of words belonging to the same equivalence class for $\sim_k$.

Some characterizations exist for the Parikh-matrix equivalence.

Parikh-matrix equivalence

Definition

Let $A = \{a_1, \ldots, a_\ell\}$ be an ordered alphabet (*i.e.* $a_1 < a_2 < \ldots < a_\ell$). Two words u and v are Parikh-matrix equivalent ($u \sim_{PM} v$) if and only if $\binom{u}{x} = \binom{v}{x}$ for all x's that are factors of the word $a_1 \cdot a_2 \cdots a_\ell$.

Parikh-matrix equivalence

Definition

Let $A = \{a_1, \ldots, a_\ell\}$ be an ordered alphabet (*i.e.* $a_1 < a_2 < \ldots < a_\ell$). Two words u and v are Parikh-matrix equivalent ($u \sim_{PM} v$) if and only if $\binom{u}{x} = \binom{v}{x}$ for all x's that are factors of the word $a_1 \cdot a_2 \cdots a_\ell$.

Example

The words $u = 01120$ and $v = 01102$ are Parikh-matrix equivalent. Indeed, for any $z \in \{u, v\}$, we have $\binom{z}{0} = 2$, $\binom{z}{1} = 2$, $\binom{z}{2} = 1$, $\binom{z}{01} = 2$, $\binom{z}{12} = 2$ and $\binom{z}{012} = 2$.
However, they are not 2-binomially equivalent since $\binom{u}{02} = 1$ and $\binom{v}{02} = 2$.

On binary alphabets

On binary alphabets, there exists a simple characterization of words that are Parikh-matrix equivalent.

Theorem

Two words u and v over $\{0,1\}^*$ are Parikh-matrix equivalent if and only if we can go from u to v by applying a finite number of times the following transformation:

$$x01y10z \leftrightarrow x10y01z.$$

On binary alphabets

On binary alphabets, there exists a simple characterization of words that are Parikh-matrix equivalent.

Theorem

Two words u and v over $\{0,1\}^*$ are Parikh-matrix equivalent if and only if we can go from u to v by applying a finite number of times the following transformation:

$$x01y10z \ \leftrightarrow \ x10y01z.$$

Example

The words $u = 010110101$ and $v = 100011110$ are Parikh-matrix equivalent. Indeed,

$$u = 010110101$$

On binary alphabets

On binary alphabets, there exists a simple characterization of words that are Parikh-matrix equivalent.

Theorem

Two words u and v over $\{0,1\}^*$ are Parikh-matrix equivalent if and only if we can go from u to v by applying a finite number of times the following transformation:

$$x01y10z \leftrightarrow x10y01z.$$

Example

The words $u = 010110101$ and $v = 100011110$ are Parikh-matrix equivalent. Indeed,

$$u = \color{red}{01}01\color{red}{10}101 \leftrightarrow \color{red}{10}01\color{red}{01}101$$

On binary alphabets, there exists a simple characterization of words that are Parikh-matrix equivalent.

Theorem

Two words u and v over $\{0,1\}^*$ are Parikh-matrix equivalent if and only if we can go from u to v by applying a finite number of times the following transformation:

$$x01y10z \leftrightarrow x10y01z.$$

Example

The words $u = 010110101$ and $v = 100011110$ are Parikh-matrix equivalent. Indeed,

$$u = 010110101 \leftrightarrow 100101101$$

On binary alphabets

On binary alphabets, there exists a simple characterization of words that are Parikh-matrix equivalent.

Theorem

Two words u and v over $\{0,1\}^*$ are Parikh-matrix equivalent if and only if we can go from u to v by applying a finite number of times the following transformation:

$$x01y10z \leftrightarrow x10y01z.$$

Example

The words $u = 010110101$ and $v = 100011110$ are Parikh-matrix equivalent. Indeed,

$$u = 010110101 \leftrightarrow 100101101$$

On binary alphabets

On binary alphabets, there exists a simple characterization of words that are Parikh-matrix equivalent.

Theorem

Two words u and v over $\{0,1\}^*$ are Parikh-matrix equivalent if and only if we can go from u to v by applying a finite number of times the following transformation:

$$x01y10z \leftrightarrow x10y01z.$$

Example

The words $u = 010110101$ and $v = 100011110$ are Parikh-matrix equivalent. Indeed,

$$u = 010110101 \leftrightarrow 100101101 \leftrightarrow 100011110 = v$$

Link with $\sim_k$

On binary alphabets, $u \sim_{PM} v \iff u \sim_2 v$.

Link with $\sim_k$

On binary alphabets, $u \sim_{PM} v \Leftrightarrow u \sim_2 v$.

We thus have a characterization of binary words belonging to a particular equivalence class for $\sim_2$. Can we, in some way, generalize this characterization to deal with $\sim_k$, where k is arbitrary?

Link with $\sim_k$

On binary alphabets, $u \sim_{PM} v \Leftrightarrow u \sim_2 v$.

We thus have a characterization of binary words belonging to a particular equivalence class for $\sim_2$. Can we, in some way, generalize this characterization to deal with $\sim_k$, where k is arbitrary?

What about non-binary words? Even for $\sim_{PM}$, there is no complete characterization.