String attractors, or how to capture text combinatorics

France Gheeraert

Octobre 14, 2025

Lempel-Ziv 77

- Loseless text compression algorithm
- Many variants used in ZIP, PNG, GIF, etc.

Lempel-Ziv 77

- Loseless text compression algorithm
- Many variants used in ZIP, PNG, GIF, etc.

Example:

abracadabra banana

Lempel-Ziv 77

- Loseless text compression algorithm
- Many variants used in ZIP, PNG, GIF, etc.

Example:

a.bracadabra banana

(0,0,a)

Lempel-Ziv 77

- Loseless text compression algorithm
- Many variants used in ZIP, PNG, GIF, etc.

Example:

a.b.racadabra banana

(0,0,a)(0,0,b)

Lempel-Ziv 77

- Loseless text compression algorithm
- Many variants used in ZIP, PNG, GIF, etc.

Example:

a.b.r.acadabra banana

(0,0,a)(0,0,b)(0,0,r)

Lempel-Ziv 77

- Loseless text compression algorithm
- Many variants used in ZIP, PNG, GIF, etc.

Example:

a.b.r.ac.adabra banana

(0,0,a)(0,0,b)(0,0,r)(1,1,c)

Lempel-Ziv 77

- Loseless text compression algorithm
- Many variants used in ZIP, PNG, GIF, etc.

Example:

a.b.r.ac.ad.abra banana

(0,0,a)(0,0,b)(0,0,r)(1,1,c)(1,1,d)

Lempel-Ziv 77

- Loseless text compression algorithm
- Many variants used in ZIP, PNG, GIF, etc.

Example:

a.b.r.ac.ad.abra .banana

(0,0,a)(0,0,b)(0,0,r)(1,1,c)(1,1,d)(1,4,)

Lempel-Ziv 77

- Loseless text compression algorithm
- Many variants used in ZIP, PNG, GIF, etc.

Example:

a.b.r.ac.ad.abra .ba.nana

(0,0,a)(0,0,b)(0,0,r)(1,1,c)(1,1,d)(1,4,)(2,1,a)

Lempel-Ziv 77

- Loseless text compression algorithm
- Many variants used in ZIP, PNG, GIF, etc.

Example:

a.b.r.ac.ad.abra .ba.n.ana

(0,0,a)(0,0,b)(0,0,r)(1,1,c)(1,1,d)(1,4,)(2,1,a)(0,0,n)

Lempel-Ziv 77

- Loseless text compression algorithm
- Many variants used in ZIP, PNG, GIF, etc.

Example:

a.b.r.ac.ad.abra .ba.n.ana

(0,0,a)(0,0,b)(0,0,r)(1,1,c)(1,1,d)(1,4,)(2,1,a)(0,0,n)(14,3,EOF)

Burrows-Wheeler Transform

- Another loseless text compression algorithm
- More efficient on shorter text

Burrows-Wheeler Transform

- Another loseless text compression algorithm
- More efficient on shorter text

Example:

b a n a n a

Burrows-Wheeler Transform

- Another loseless text compression algorithm
- More efficient on shorter text

Example:

b a n a n a

a n a n a b

Burrows-Wheeler Transform

- Another loseless text compression algorithm
- More efficient on shorter text

Example:

b a n a n a
a n a n a b
n a n a b a
a n a b a n
n a b a n a
a b a n a n

Burrows-Wheeler Transform

- Another loseless text compression algorithm
- More efficient on shorter text

Example:

b a n a n a	a b a n a **n**
a n a n a b	a n a b a **n**
n a n a b a	a n a n a **b**
a n a b a n	b a n a n **a**
n a b a n a	n a b a n **a**
a b a n a n	n a n a b **a**

Burrows-Wheeler Transform

- Another loseless text compression algorithm
- More efficient on shorter text

Example:

b a n a n a	a b a n a **n**
a n a n a b	a n a b a **n**
n a n a b a	a n a n a **b**
a n a b a n	b a n a n **a**
n a b a n a	n a b a n **a**
a b a n a n	n a n a b **a**

banana $\rightsquigarrow$ nnbaaa

Burrows-Wheeler Transform

- Another loseless text compression algorithm
- More efficient on shorter text

Example:

b a n a n a	a b a n a **n**
a n a n a b	a n a b a **n**
n a n a b a	a n a n a **b**
a n a b a n	b a n a n **a**
n a b a n a	n a b a n **a**
a b a n a n	n a n a b **a**

$$\text{banana} \rightsquigarrow \text{nnbaaa} \rightsquigarrow \text{(n,2)(b,1)(a,3)}$$

Burrows-Wheeler Transform

- Another loseless text compression algorithm
- More efficient on shorter text

Example:

b a n a n a	a b a n a **n**
a n a n a b	a n a b a **n**
n a n a b a	a n a n a **b**
a n a b a n	b a n a n **a**
n a b a n a	n a b a n **a**
a b a n a n	n a n a b **a**

$$\text{banana} \rightsquigarrow \text{nnbaaa} \rightsquigarrow (\text{n},2)(\text{b},1)(\text{a},3)$$

Can we measure the repetitiveness of a text?

String attractors

Definition (Kempa & Prezza 2017)

A *string attractor* of a text of length n is a set $\Gamma \subseteq \{1, \ldots, n\}$ of positions such that every substring (factor) of the text has an occurrence crossing a position in Γ.

String attractors

Definition (Kempa & Prezza 2017)

A *string attractor* of a text of length n is a set $\Gamma \subseteq \{1, \ldots, n\}$ of positions such that every substring (factor) of the text has an occurrence crossing a position in Γ.

Examples:

$$\underline{b}a\underline{n}an\underline{a}$$
$$\Gamma = \{1, 3, 6\}$$

String attractors

Definition (Kempa & Prezza 2017)

A *string attractor* of a text of length n is a set $\Gamma \subseteq \{1, \ldots, n\}$ of positions such that every substring (factor) of the text has an occurrence crossing a position in Γ.

Examples:

$$\underline{b}a\underline{n}an\underline{a} \qquad \Gamma = \{1, 3, 6\}$$

$$01\underline{10}010\underline{1}00\underline{1} \qquad \Gamma = \{3, 4, 8, 11\}$$

String attractors

Definition (Kempa & Prezza 2017)

A *string attractor* of a text of length n is a set $\Gamma \subseteq \{1, \ldots, n\}$ of positions such that every substring (factor) of the text has an occurrence crossing a position in Γ.

Examples:

$$\underline{b}a\underline{n}a\underline{n}a \qquad \Gamma = \{1, 3, 6\}$$

$$01\underline{10}010\underline{1}00\underline{1} \qquad \Gamma = \{3, 4, 8, 11\}$$

Definition

$\gamma^*(t) =$ minimal size of a string attractor of t.

Link with compression algorithms

a.b.r.ac.ad.abra .ba.n.ana

Link with compression algorithms

a.b.r.ac.ad.abra .ba.n.ana $\rightsquigarrow$ abracadabra_banana

Link with compression algorithms

a.b.r.ac.ad.abra .ba.n.ana $\rightsquigarrow$ abracadabra_banana

Proposition (Kempa & Prezza)

If

- $lz(t)$ *is the number of phrases in the LZ77 compression of* t,

then

1. $\gamma^*(t) \leq lz(t)$,

Link with compression algorithms

a.b.r.ac.ad.abra .ba.n.ana $\rightsquigarrow$ abracadabra_banana

Proposition (Kempa & Prezza)

If

- *$lz(t)$ is the number of phrases in the LZ77 compression of t,*
- *$bwt(t)$ is the number of letter runs in the BWT of t,*

then

1. $\gamma^*(t) \leq lz(t)$,
2. $\gamma^*(t) \leq bwt(t)$,

Link with compression algorithms

a.b.r.ac.ad.abra .ba.n.ana $\rightsquigarrow$ abracadabra_banana

Proposition (Kempa & Prezza)

If

- *$lz(t)$ is the number of phrases in the LZ77 compression of t,*
- *$bwt(t)$ is the number of letter runs in the BWT of t,*

then

1. $\gamma^*(t) \leq lz(t)$,
2. $\gamma^*(t) \leq bwt(t)$,
3. *$\gamma^*(t)$ also gives upper bounds on $lz(t)$ and $bwt(t)$.*

Complexity

Proposition (Kempa & Prezza)

Let $k \geq 3$ and $n \in \mathbb{N}$.
Deciding whether a given text admits a k-attractor of size at most n is NP-complete.

Complexity

Proposition (Kempa & Prezza)

Let $k \geq 3$ and $n \in \mathbb{N}$.
Deciding whether a given text admits a k-attractor of size at most n is NP-complete.

k-attractor : attractor that "catches" all factors of length $\leq k$

Complexity

Proposition (Kempa & Prezza, Fuchs & Whittington)

Let $k \geq 2$ and $n \in \mathbb{N}$.
Deciding whether a given text admits a k-attractor of size at most n is NP-complete.

k-attractor : attractor that "catches" all factors of length $\leq k$

Complexity

Proposition (Kempa & Prezza, Fuchs & Whittington)

Let $k \geq 2$ and $n \in \mathbb{N}$.
Deciding whether a given text admits a k-attractor of size at most n is NP-complete.

k-attractor : attractor that "catches" all factors of length $\leq k$

Can we use combinatorial ideas to find optimal attractors for particular examples?

Combinatorial operations

Proposition

If Γ is a string attractor for t and Δ is a string attractor for s, then

Combinatorial operations

Proposition

If Γ is a string attractor for t and Δ is a string attractor for s, then

1. *is a string attractor for t^R (t reversed);*

Combinatorial operations

Proposition

If Γ is a string attractor for t and Δ is a string attractor for s, then

1. $\mathrm{len}(t) + 1 - \Gamma$ *is a string attractor for* t^R *(t reversed);*

Combinatorial operations

Proposition

If Γ is a string attractor for t and Δ is a string attractor for s, then

1. $\mathrm{len}(t) + 1 - \Gamma$ *is a string attractor for* t^R *(t reversed);*
2. *is a string attractor for ts;*

Combinatorial operations

Proposition

If Γ is a string attractor for t and Δ is a string attractor for s, then

1. $\mathrm{len}(t) + 1 - \Gamma$ *is a string attractor for* t^R *(t reversed);*
2. $\Gamma \cup (\Delta + \mathrm{len}(t)) \cup \{\mathrm{len}(t)\}$ *is a string attractor for* ts*;*

Combinatorial operations

Proposition

If Γ is a string attractor for t and Δ is a string attractor for s, then

1. $\mathrm{len}(t) + 1 - \Gamma$ *is a string attractor for* t^R *(t reversed);*
2. $\Gamma \cup (\Delta + \mathrm{len}(t)) \cup \{\mathrm{len}(t)\}$ *is a string attractor for* ts*;*
3. *is a string attractor for* tp *for any prefix* p *of* $ttttt\cdots$.

Combinatorial operations

Proposition

If Γ is a string attractor for t and Δ is a string attractor for s, then

1. $\mathrm{len}(t) + 1 - \Gamma$ *is a string attractor for* t^R *(t reversed);*
2. $\Gamma \cup (\Delta + \mathrm{len}(t)) \cup \{\mathrm{len}(t)\}$ *is a string attractor for* ts*;*
3. $\Gamma \cup \{\mathrm{len}(t)\}$ *is a string attractor for* tp *for any prefix* p *of* $ttttt\cdots$*.*

Combinatorial operations

Proposition

If Γ is a string attractor for t and Δ is a string attractor for s, then

1. $\mathrm{len}(t) + 1 - \Gamma$ *is a string attractor for* t^R *(t reversed);*
2. $\Gamma \cup (\Delta + \mathrm{len}(t)) \cup \{\mathrm{len}(t)\}$ *is a string attractor for* ts*;*
3. $\Gamma \cup \{\mathrm{len}(t)\}$ *is a string attractor for* tp *for any prefix* p *of* $ttttt\cdots$*.*

Corollary

1. $\gamma^*(t^R) = \gamma^*(t)$*;*
2. $\gamma^*(ts) \leq \gamma^*(t) + \gamma^*(s) + 1$*;*
3. $\gamma^*(tp) \leq \gamma^*(t) + 1$ *for every prefix* p *of* $tttt\cdots$*.*

Monotonicity

- Do we have $\gamma^*(t) \leq \gamma^*(ts)$?

Monotonicity

- Do we have $\gamma^*(t) \leq \gamma^*(ts)$?
  No, example:

  01110001

Monotonicity

- Do we have $\gamma^*(t) \leq \gamma^*(ts)$?
  No, example:

$$01110001$$

Monotonicity

- Do we have $\gamma^*(t) \leq \gamma^*(ts)$?
  No, example:

  $$01110001$$

Monotonicity

- Do we have $\gamma^*(t) \leq \gamma^*(ts)$?
  No, example:

$$01110001 \rightsquigarrow 01110001\underline{1}$$

Monotonicity

- Do we have $\gamma^*(t) \leq \gamma^*(ts)$?
  No, example:

$$01110001 \rightsquigarrow 01110011$$

Monotonicity

- Do we have $\gamma^*(t) \leq \gamma^*(ts)$?
  No, example:

$$01110001 \rightsquigarrow 01110011$$

Monotonicity

- Do we have $\gamma^*(t) \leq \gamma^*(ts)$?
  No, example:

$$01110001 \rightsquigarrow 0111\underline{0}0\underline{0}11$$

- And if s is a prefix of $tttt\cdots$?

Monotonicity

- Do we have $\gamma^*(t) \leq \gamma^*(ts)$?
  No, example:
  $$01110001 \rightsquigarrow 01110011$$

- And if s is a prefix of $tttt\cdots$?
  No, there are infinitely many texts t s.t. $\gamma^*(tt) \sim \frac{1}{2}\gamma^*(t)$.
  (Mantaci et al.)

Profile function

Definition

Let $x \in A^{\mathbb{N}}$.

Profile function

Definition

Let $x \in A^{\mathbb{N}}$. The (string attractor) profile function of x is

$$s_x : \mathbb{N} \to \mathbb{N} \quad n \mapsto \gamma^*(x[1, n]).$$

Profile function

Definition

Let $x \in A^{\mathbb{N}}$. The (string attractor) profile function of x is

$$s_x : \mathbb{N} \to \mathbb{N} \quad n \mapsto \gamma^*(x[1, n]).$$

Upper bound:

- $s_x(n) \leq n$ for all n;

Profile function

Definition

Let $x \in A^{\mathbb{N}}$. The (string attractor) profile function of x is

$$s_x : \mathbb{N} \to \mathbb{N} \quad n \mapsto \gamma^*(x[1, n]).$$

Upper bound:

- $s_x(n) \leq n$ for all n;
- $s_x(n) \leq \mathrm{lz}(x[1, n])$

Profile function

Definition

Let $x \in A^{\mathbb{N}}$. The (string attractor) profile function of x is

$$s_x : \mathbb{N} \to \mathbb{N} \quad n \mapsto \gamma^*(x[1, n]).$$

Upper bound:

- $s_x(n) \leq n$ for all n;
- $s_x(n) \leq \mathrm{lz}(x[1, n]) \in O\left(\frac{n}{\log_{\#A}(n)}\right)$.

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttt t\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

$$0$$

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

01

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

0110

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

01101001

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

$$0110100110010110\cdots$$

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

$$0110100110010110\cdots$$

$\rightarrow s_x(n) \leq 4$

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

$$0110100110010110\cdots$$

$\rightarrow s_x(n) \leq 4$

- Fibonacci sequence:

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

$$0110100110010110\cdots$$

$\rightarrow s_x(n) \leq 4$

- Fibonacci sequence:

$$01$$

0, 01

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

$$0110100110010110\cdots$$

$\rightarrow s_x(n) \leq 4$

- Fibonacci sequence:

$$010$$

0, 01, 010

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

$$0110100110010110\cdots$$

$\rightarrow s_x(n) \leq 4$

- Fibonacci sequence:

$$01001$$

0, 01, 010, 01001

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

$$0110100110010110\cdots$$

$\rightarrow s_x(n) \leq 4$

- Fibonacci sequence:

$$01001010$$

0, 01, 010, 01001, 01001010

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

$$0110100110010110\cdots$$

$\rightarrow s_x(n) \leq 4$

- Fibonacci sequence:

$$0100101001001\cdots$$

0, 01, 010, 01001, 01001010, 0100101001001, ...

The bounded case

- If $x = tttt\cdots$, then $s_x(n) \leq \mathrm{len}(t)$.
- If $x = sttt\cdots$, then $s_x(n) \leq \mathrm{len}(s) + \mathrm{len}(t)$.

Are there aperiodic sequences with bounded profile function?

- Thue-Morse sequence:

$$0110100110010110\cdots$$

$\rightarrow s_x(n) \leq 4$

- Fibonacci sequence:

$$0100101001001\cdots$$

0, 01, 010, 01001, 01001010, 0100101001001, ... $\rightarrow$
$s_x(n) \leq 2$

The bounded case (2)

In both examples, every factor appears with linearly bounded gaps.

The bounded case (2)

In both examples, every factor appears with linearly bounded gaps.

Proposition

- *If every factor appears with linearly bounded gaps, then s_x is bounded. (Schaeffer & Shallit)*

The bounded case (2)

In both examples, every factor appears with linearly bounded gaps.

Proposition

- *If every factor appears with linearly bounded gaps, then s_x is bounded. (Schaeffer & Shallit)*
- *If s_x is bounded, then either x is eventually periodic, or the number of different factors grows linearly. (Restivo, Romana & Sciortino)*

Fibonacci: first ideas

Construction of the string attractors:

$$0100101001001\cdots$$

String attractors:

Fibonacci: first ideas

Construction of the string attractors:

$$\underline{0}1001010010001\cdots$$

String attractors:

$$\{1\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$\underline{01}00101001001\cdots$$

String attractors:

$$\{1\}, \{1, 2\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$\underline{010}0101001001\cdots$$

String attractors:

$$\{1\}, \{1, 2\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$\underline{01}00101001001\cdots$$

String attractors:

$$\{1\}, \{1, 2\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$\underline{0100}1010010010\cdots$$

String attractors:

$$\{1\}, \{1, 2\}$$

Fibonacci: first ideas

Construction of the string attractors:

String attractors:

$$\{1\}, \{1, 2\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$0\underline{10}0101001001\cdots$$

String attractors:

$$\{1\}, \{1, 2\}, \{2, 3\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$0\underline{10}0101001001001\cdots$$

String attractors:

$$\{1\}, \{1, 2\}, \{2, 3\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$0\underline{10}0101001001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$0\underline{10}0101001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$0\underline{10}\underline{1}01001001\cdots$$

String attractors:

$$\{1\}, \{1, 2\}, \{2, 3\}$$

Fibonacci: first ideas

Construction of the string attractors:

String attractors:

$$\{1\}, \{1, 2\}, \{2, 3\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$01\underline{0}\underline{1}01001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}, \{3,5\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$01\underline{0}\underline{1}01001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}, \{3,5\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$01\underline{0}\underline{1}0100\color{gray}{1001\cdots}$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}, \{3,5\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$01\underline{0}\underline{1}01001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}, \{3,5\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$01\underline{0}\underline{1}0100101\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}, \{3,5\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$01\underline{0}\underline{1}01001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}, \{3,5\}$$

Fibonacci: first ideas

Construction of the string attractors:

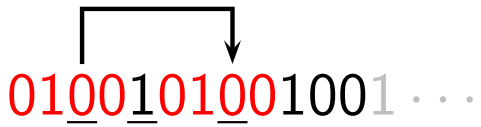
$$01\underline{0}\underline{1}01\underline{0}0100\textcolor{gray}{1\cdots}$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}, \{3,5\}$$

Fibonacci: first ideas

Construction of the string attractors:



String attractors:

$$\{1\}, \{1, 2\}, \{2, 3\}, \{3, 5\}$$

Fibonacci: first ideas

Construction of the string attractors:

$$0100\underline{1}01\underline{0}0100\textcolor{gray}{1\cdots}$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}, \{3,5\}, \{5,8\}, \ldots$$

Fibonacci: first ideas

Construction of the string attractors:

$$0100\underline{1}01\underline{0}01001\cdots$$

String attractors:

$$\{1\},\{1,2\},\{2,3\},\{3,5\},\{5,8\},\ldots$$

Fibonacci: first ideas

Construction of the string attractors:

$$0100\underline{1}01\underline{0}01001\cdots$$

String attractors:

$$\{1\},\{1,2\},\{2,3\},\{3,5\},\{5,8\},\ldots$$

- increase the length of the prefix

Fibonacci: first ideas

Construction of the string attractors:

$$0100\underline{1}01\underline{0}01001\cdots$$

String attractors:

$$\{1\},\{1,2\},\{2,3\},\{3,5\},\{5,8\},\ldots$$

- increase the length of the prefix
- if new factor, add a "well-chosen" position

Fibonacci: first ideas

Construction of the string attractors:

$$0100\underline{1}01\underline{0}01001\cdots$$

String attractors:

$$\{1\},\{1,2\},\{2,3\},\{3,5\},\{5,8\},\ldots$$

- increase the length of the prefix
- if new factor, add a "well-chosen" position
- remove the first position when it becomes redundant

Main result

Let

$$f_0 = 0, \quad f_1 = 01, \quad f_{n+2} = f_{n+1} f_n$$

and $F_n = \mathrm{len}(f_n)$.

Main result

Let

$$f_0 = 0, \quad f_1 = 01, \quad f_{n+2} = f_{n+1} f_n$$

and $F_n = \mathrm{len}(f_n)$.

Proposition

Every prefix has a string attractor of the form $\{F_n, F_{n+1}\}$.

We must show that:

Main result

Let

$$f_0 = 0, \quad f_1 = 01, \quad f_{n+2} = f_{n+1} f_n$$

and $F_n = \mathrm{len}(f_n)$.

Proposition

Every prefix has a string attractor of the form $\{F_n, F_{n+1}\}$.

We must show that:

1. the positions in $\{F_i : i \in \mathbb{N}\}$ are sufficient;

Main result

Let

$$f_0 = 0, \quad f_1 = 01, \quad f_{n+2} = f_{n+1} f_n$$

and $F_n = \mathrm{len}(f_n)$.

Proposition

Every prefix has a string attractor of the form $\{F_n, F_{n+1}\}$.

We must show that:

1. the positions in $\{F_i : i \in \mathbb{N}\}$ are sufficient;
2. F_n becomes redundant with F_{n+2};

Main result

Let

$$f_0 = 0, \quad f_1 = 01, \quad f_{n+2} = f_{n+1} f_n$$

and $F_n = \mathrm{len}(f_n)$.

Proposition

Every prefix has a string attractor of the form $\{F_n, F_{n+1}\}$.

We must show that:

1. the positions in $\{F_i : i \in \mathbb{N}\}$ are sufficient;
2. F_n becomes redundant with F_{n+2};
3. F_n is already redundant when we need position F_{n+2}.

Step 1.

Proof that the positions in $\{F_i : i \in \mathbb{N}\}$ are sufficient:

Step 1.

Proof that the positions in $\{F_i : i \in \mathbb{N}\}$ are sufficient:

- already checked for the small prefixes

Step 1.

Proof that the positions in $\{F_i : i \in \mathbb{N}\}$ are sufficient:

- already checked for the small prefixes
- assume that $\Gamma \subseteq \{F_i : i \in \mathbb{N}\}$ is a s.a. of $x[1, F_n] = f_n$, $n \geq 2$

Step 1.

Proof that the positions in $\{F_i : i \in \mathbb{N}\}$ are sufficient:

- already checked for the small prefixes
- assume that $\Gamma \subseteq \{F_i : i \in \mathbb{N}\}$ is a s.a. of $x[1, F_n] = f_n$, $n \geq 2$
- f_{n-1} is a prefix of f_n

Step 1.

Proof that the positions in $\{F_i : i \in \mathbb{N}\}$ are sufficient:

- already checked for the small prefixes
- assume that $\Gamma \subseteq \{F_i : i \in \mathbb{N}\}$ is a s.a. of $x[1, F_n] = f_n$, $n \geq 2$
- f_{n-1} is a prefix of f_n
- so $\Gamma \cup \{F_n\}$ is a s.a. of $f_n p$ for every prefix p of f_{n-1}

Step 1.

Proof that the positions in $\{F_i : i \in \mathbb{N}\}$ are sufficient:

- already checked for the small prefixes
- assume that $\Gamma \subseteq \{F_i : i \in \mathbb{N}\}$ is a s.a. of $x[1, F_n] = f_n$, $n \geq 2$
- f_{n-1} is a prefix of f_n
- so $\Gamma \cup \{F_n\}$ is a s.a. of $f_n p$ for every prefix p of f_{n-1}
- hence, $\Gamma \cup \{F_n\}$ is a s.a. of $x[1, k]$ for any $F_n \leq k \leq F_{n+1}$

Step 1.

Proof that the positions in $\{F_i : i \in \mathbb{N}\}$ are sufficient:

- already checked for the small prefixes
- assume that $\Gamma \subseteq \{F_i : i \in \mathbb{N}\}$ is a s.a. of $x[1, F_n] = f_n$, $n \geq 2$
- f_{n-1} is a prefix of f_n
- so $\Gamma \cup \{F_n\}$ is a s.a. of $f_n p$ for every prefix p of f_{n-1}
- hence, $\Gamma \cup \{F_n\}$ is a s.a. of $x[1, k]$ for any $F_n \leq k \leq F_{n+1}$

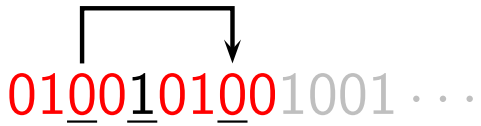
✓

Step 2.

Proof that F_n becomes redundant with F_{n+2}:

Step 2.

Proof that F_n becomes redundant with F_{n+2}:



Step 2.

Proof that F_n becomes redundant with F_{n+2}:

$$\mathbf{01\underline{0}0\underline{1}01\underline{0}}0\underline{1}001\cdots$$

- f_n is a suffix of f_{n+2};

Step 2.

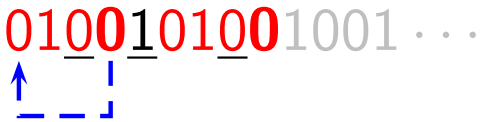
Proof that F_n becomes redundant with F_{n+2}:

$$01\underline{0}\mathbf{0}\underline{1}01\underline{0}\mathbf{0}1001\cdots$$

- f_n is a suffix of f_{n+2};
- f_{n+2} is followed by $x[F_n + 1, F_{n+1} - 1]$ in x;

Step 2.

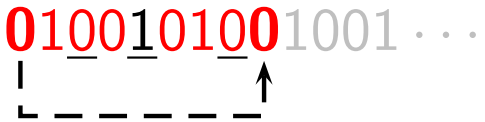
Proof that F_n becomes redundant with F_{n+2}:



- f_n is a suffix of f_{n+2};
- f_{n+2} is followed by $x[F_n + 1, F_{n+1} - 1]$ in x;
  - $x[F_n + 1, F_{n+1} - 1]$ is a prefix of f_n;

Step 2.

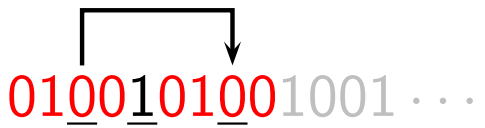
Proof that F_n becomes redundant with F_{n+2}:



- f_n is a suffix of f_{n+2};
- f_{n+2} is followed by $x[F_n + 1, F_{n+1} - 1]$ in x;
  - $x[F_n + 1, F_{n+1} - 1]$ is a prefix of f_n;
  - f_{n+2} is followed by f_n in x;

Step 2.

Proof that F_n becomes redundant with F_{n+2}:



- f_n is a suffix of f_{n+2};
- f_{n+2} is followed by $x[F_n + 1, F_{n+1} - 1]$ in x;
  - $x[F_n + 1, F_{n+1} - 1]$ is a prefix of f_n;
  - f_{n+2} is followed by f_n in x;

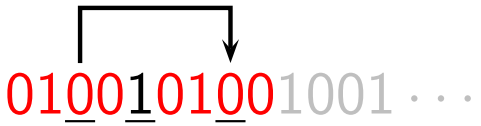
✓

Step 3.

Proof that F_n is redundant when we need F_{n+2}:

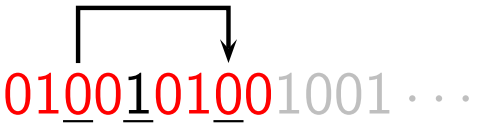
Step 3.

Proof that F_n is redundant when we need F_{n+2}:



Step 3.

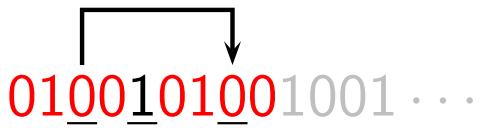
Proof that F_n is redundant when we need F_{n+2}:



- F_n is redundant starting with $x[1, F_{n+2} + F_{n+1} - 1 - F_n]$

Step 3.

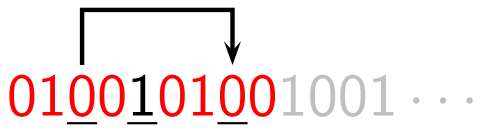
Proof that F_n is redundant when we need F_{n+2}:



- F_n is redundant starting with $x[1, F_{n+2} + F_{n+1} - 1 - F_n]$
- $x[F_{n+1} + 1, F_{n+2} + F_{n+1} - F_n - 2]$ is a prefix of f_{n+1}

Step 3.

Proof that F_n is redundant when we need F_{n+2}:



- F_n is redundant starting with $x[1, F_{n+2} + F_{n+1} - 1 - F_n]$
- $x[F_{n+1} + 1, F_{n+2} + F_{n+1} - F_n - 2]$ is a prefix of f_{n+1}

✓

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$\underline{01}\underline{0}2010010201010201001020\cdots$$

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$\underline{01}0\underline{2}0100102010201010201001020\cdots$$

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$\underline{01}0\underline{2}0100102010201002010 2 \cdots$$

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$\underline{0}\underline{1}0\underline{2}01\underline{0}0102010201001020\cdots$$

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$0\underline{1}0\underline{2}01\underline{0}0102010102010010\,2\cdots$$

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$0\underline{1}0\underline{2}01\underline{0}0102010\,1020100102\cdots$$

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$0\underline{1}0\underline{2}01\underline{0}0102010102020100102\cdots$$

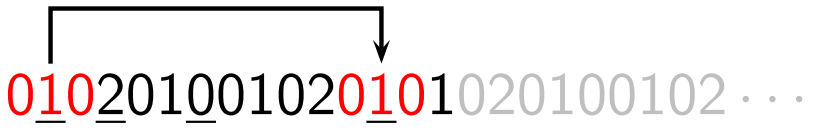
Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$0\underline{1}0\underline{2}01\underline{0}01020\underline{1}01020100102\cdots$$

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$



Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$0102\underline{0}1\underline{0}0102\underline{0}101020100102\cdots$$

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$0102\underline{0}1\underline{0}0102\underline{0}1\underline{0}1020100102\cdots$$

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$010\underline{2}01\underline{0}0102\underline{0}1010201001\underline{0}2\cdots$$

Proposition (Cassaigne et al.)

Every prefix has a s.a. made of (at most) 3 consecutive Tribonacci numbers.

Tribonacci sequence

$$t_0 = 0, \quad t_1 = 01, \quad t_2 = 0102, \quad t_{n+3} = t_{n+2}t_{n+1}t_n$$

$$0102\underline{0}1\underline{0}0102\underline{0}1\underline{0}1020100102\cdots$$

Proposition (Cassaigne et al.)

Every prefix has a s.a. made of (at most) 3 consecutive Tribonacci numbers.
More generally, if $x = \lim_{i\to\infty} u_i$ *with*

$$u_i = \begin{cases} u_{i-1}u_{i-2}\cdots u_0\ i & \text{if } i < k, \\ u_{i-1}u_{i-2}\cdots u_{i-k} & \text{if } i \geq k, \end{cases}$$

then every prefix of x *has a s.a. made of (at most)* k *consecutive elements of* $\{\mathrm{len}(u_i) : i \in \mathbb{N}\}$.

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases}$$

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases}$$

0

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases}$$

$$0 \mapsto 01$$

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases}$$

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 010 \end{aligned}$$

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases}$$

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 010 \\ &\mapsto 01001 \end{aligned}$$

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases}$$

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 010 \\ &\mapsto 01001 \\ &\mapsto 01001010 \end{aligned}$$

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases}$$

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 010 \\ &\mapsto 01001 \\ &\mapsto 01001010 \end{aligned}$$

Fibonacci!

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases} \qquad \varphi_{111} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 02, \\ 2 \mapsto 0 \end{cases}$$

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 010 \\ &\mapsto 01001 \\ &\mapsto 01001010 \end{aligned}$$

Fibonacci!

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases} \qquad \varphi_{111} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 02, \\ 2 \mapsto 0 \end{cases}$$

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 010 \\ &\mapsto 01001 \\ &\mapsto 01001010 \end{aligned} \qquad 0$$

Fibonacci!

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases} \qquad \varphi_{111} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 02, \\ 2 \mapsto 0 \end{cases}$$

$0 \mapsto 01$

$\mapsto 010$

$\mapsto 01001$

$\mapsto 01001010$

Fibonacci!

$0 \mapsto 01$

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases} \qquad \varphi_{111} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 02, \\ 2 \mapsto 0 \end{cases}$$

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 010 \\ &\mapsto 01001 \\ &\mapsto 01001010 \end{aligned} \qquad \begin{aligned} 0 &\mapsto 01 \\ &\mapsto 0102 \end{aligned}$$

Fibonacci!

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases} \qquad \varphi_{111} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 02, \\ 2 \mapsto 0 \end{cases}$$

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 010 \\ &\mapsto 01001 \\ &\mapsto 01001010 \end{aligned} \qquad \begin{aligned} 0 &\mapsto 01 \\ &\mapsto 0102 \\ &\mapsto 0102010 \end{aligned}$$

Fibonacci!

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases} \qquad \varphi_{111} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 02, \\ 2 \mapsto 0 \end{cases}$$

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 010 \\ &\mapsto 01001 \\ &\mapsto 01001010 \end{aligned}$$

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 0102 \\ &\mapsto 0102010 \\ &\mapsto 0102010010201 \end{aligned}$$

Fibonacci!

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases} \qquad \varphi_{111} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 02, \\ 2 \mapsto 0 \end{cases}$$

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 010 \\ &\mapsto 01001 \\ &\mapsto 01001010 \end{aligned}$$

Fibonacci!

$$\begin{aligned} 0 &\mapsto 01 \\ &\mapsto 0102 \\ &\mapsto 0102010 \\ &\mapsto 0102010010201 \end{aligned}$$

Tribonacci!

Transition to substitutive sequences

Consider the substitutions

$$\varphi_{11} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 0, \end{cases} \qquad \varphi_{111} : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 02, \\ 2 \mapsto 0 \end{cases}$$

$0 \mapsto 01$
$\mapsto 010$
$\mapsto 01001$
$\mapsto 01001010$

Fibonacci!

$0 \mapsto 01$
$\mapsto 0102$
$\mapsto 0102010$
$\mapsto 0102010010201$

Tribonacci!

Proposition (Cassaigne et al.)

If $x = \lim_{i\to\infty} \varphi_{1^k}^i(0)$, then every prefix of x has a s.a. made of (at most) k consecutive elements of $\{\mathrm{len}(\varphi_{1^k}^i(0)) : i \in \mathbb{N}\}$.

Generalizing

For some parameters $c_0, \ldots, c_{k-1} \in \mathbb{N}$ with $c_0, c_{k-1} \geq 1$, we define

$$\varphi_{c_0 \cdots c_{k-1}} : \begin{cases} 0 \mapsto 0^{c_0}1 \\ 1 \mapsto 0^{c_1}2 \\ \cdots \\ k-1 \mapsto 0^{c_{k-1}} \end{cases}$$

and

$$u_n = \varphi^n_{c_0 \cdots c_{k-1}}(0), \quad U_n = \mathrm{len}(u_n).$$

Generalizing

For some parameters $c_0, \ldots, c_{k-1} \in \mathbb{N}$ with $c_0, c_{k-1} \geq 1$, we define

$$\varphi_{c_0 \cdots c_{k-1}} : \begin{cases} 0 \mapsto 0^{c_0}1 \\ 1 \mapsto 0^{c_1}2 \\ \cdots \\ k-1 \mapsto 0^{c_{k-1}} \end{cases}$$

and

$$u_n = \varphi^n_{c_0 \cdots c_{k-1}}(0), \quad U_n = \mathrm{len}(u_n).$$

Does every prefix of $\lim_i \varphi^i_{c_0 \cdots c_{k-1}}(0)$ have a s.a. made of (at most) k consecutive U_n's?

Answer

Proposition (G., Romana & Stipulanti)

Let $x = \lim_i \varphi^i_{c_0 \cdots c_{k-1}}(0)$

- *If there exists* j *such that* $c_j \cdots c_{k-2}(c_{k-1}-1)c_0 \cdots c_{j-1} > c_0 \cdots c_{k-2}(c_{k-1}-1)$, *then there are prefixes of* x *having no s.a. included in* $\{U_i : i \in \mathbb{N}\}$.
  Step 1. fails

Answer

Proposition (G., Romana & Stipulanti)

Let $x = \lim_i \varphi^i_{c_0 \cdots c_{k-1}}(0)$

- *If there exists* j *such that* $c_j \cdots c_{k-2}(c_{k-1}-1)c_0 \cdots c_{j-1} > c_0 \cdots c_{k-2}(c_{k-1}-1)$, *then there are prefixes of* x *having no s.a. included in* $\{U_i : i \in \mathbb{N}\}$. *Step 1. fails*
- *Otherwise, every prefix of* x *has a s.a. made of (at most)* $k+1$ *consecutive* U_n*'s. Steps 1. and 2. succeed*

Answer

Proposition (G., Romana & Stipulanti)

Let $x = \lim_i \varphi^i_{c_0 \cdots c_{k-1}}(0)$

- *If there exists* j *such that* $c_j \cdots c_{k-2}(c_{k-1}-1)c_0 \cdots c_{j-1} > c_0 \cdots c_{k-2}(c_{k-1}-1)$, *then there are prefixes of* x *having no s.a. included in* $\{U_i : i \in \mathbb{N}\}$. *Step 1. fails*
- *Otherwise, every prefix of* x *has a s.a. made of (at most)* $k+1$ *consecutive* U_n*'s. Steps 1. and 2. succeed*
- *Moreover, if* $c_{k-1} = 1$ *and* $c_j \cdots c_{k-2}c_0 \cdots c_{j-1} \leq c_0 \cdots c_{k-2}$ *for every* j, *then every prefix of* x *has a s.a. made of (at most)* k *consecutive* U_n*'s. Step 3. succeeds*

Answer

Proposition (G., Romana & Stipulanti)

Let $x = \lim_i \varphi^i_{c_0 \cdots c_{k-1}}(0)$

- *If there exists* j *such that* $c_j \cdots c_{k-2}(c_{k-1}-1)c_0 \cdots c_{j-1} > c_0 \cdots c_{k-2}(c_{k-1}-1)$, *then there are prefixes of* x *having no s.a. included in* $\{U_i : i \in \mathbb{N}\}$. *Step 1. fails*
- *Otherwise, every prefix of* x *has a s.a. made of (at most)* $k+1$ *consecutive* U_n*'s. Steps 1. and 2. succeed*
- *Moreover, if* $c_{k-1} = 1$ *and* $c_j \cdots c_{k-2}c_0 \cdots c_{j-1} \leq c_0 \cdots c_{k-2}$ *for every* j, *then every prefix of* x *has a s.a. made of (at most)* k *consecutive* U_n*'s. Step 3. succeeds*

Proposition (Dvoraková & Moravcová)

If $c_j \cdots c_{k-2}(c_{k-1}-1)c_0 \cdots c_{j-1} \leq c_0 \cdots c_{k-2}(c_{k-1}-1)$ *for every* j, *then every prefix of* x *has a s.a. of size (at most)* k.

Other string attractors

Construction of the string attractors:

$$0100101001001\cdots$$

String attractors:

Other string attractors

Construction of the string attractors:

$$\underline{0}100101001001\cdots$$

String attractors:

$$\{1\}$$

Other string attractors

Construction of the string attractors:

$$\underline{01}00101001001\cdots$$

String attractors:

$$\{1\}, \{1, 2\}$$

Other string attractors

Construction of the string attractors:

$$\underline{010}0101001001\cdots$$

String attractors:

$$\{1\}, \{1, 2\}$$

Other string attractors

Construction of the string attractors:

$$\underline{010}0101001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}$$

Other string attractors

Construction of the string attractors:

$$0\underline{10}0101001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}$$

Other string attractors

Construction of the string attractors:

$$0\underline{10}0101001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}$$

Other string attractors

Construction of the string attractors:

$$0\underline{10}0101001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}$$

Other string attractors

Construction of the string attractors:

$$010\underline{01}0\,1001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}, \{4,5\}$$

Other string attractors

Construction of the string attractors:

$$0100\underline{1}101001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}, \{4,5\}$$

Other string attractors

Construction of the string attractors:

$$0100\underline{1}01001001\cdots$$

String attractors:

$$\{1\},\{1,2\},\{2,3\},\{4,5\}$$

Other string attractors

Construction of the string attractors:

$$0100\underline{1}0100\,1001\cdots$$

String attractors:

$$\{1\},\{1,2\},\{2,3\},\{4,5\}$$

Other string attractors

Construction of the string attractors:

$$0100\underline{1}01001001\cdots$$

String attractors:

$$\{1\}, \{1,2\}, \{2,3\}, \{4,5\}$$

Other string attractors

Construction of the string attractors:

$$0100\underline{1}0100101\cdots$$

String attractors:

$$\{1\},\{1,2\},\{2,3\},\{4,5\}$$

Other string attractors

Construction of the string attractors:

$$0100101\underline{0}0101\cdots$$

String attractors:

$$\{1\},\{1,2\},\{2,3\},\{4,5\},\{7,8\},\ldots$$

Other string attractors

Construction of the string attractors:

$$0100101\underline{0}0100\,1\cdots$$

String attractors:

$$\{1\},\{1,2\},\{2,3\},\{4,5\},\{7,8\},\ldots$$

Alternative construction of the sequence

0

Alternative construction of the sequence

0

0

Alternative construction of the sequence

0

01

Alternative construction of the sequence

0

010

Alternative construction of the sequence

0

010

010

Alternative construction of the sequence

0

010

0100

Alternative construction of the sequence

0

010

010010

Alternative construction of the sequence

0

010

010010

010010

Alternative construction of the sequence

0

010

010010

0100101

Alternative construction of the sequence

0

010

010010

01001010010

Alternative construction of the sequence

0

010

010010

01001010010

01001010010

Alternative construction of the sequence

0

010

010010

01001010010

0100101001010

Alternative construction of the sequence

0

010

010010

01001010010

0100101001001010010

...

Alternative construction of the sequence

0

010

010010

01001010010

0100101001001010010

...

Proposition (Restivo, Romana & Sciortino)

Every prefix has a string attractor made of 2 consecutive positions.

Alternative construction of the sequence

0

010

010010

01001010010

0100101001001010010

...

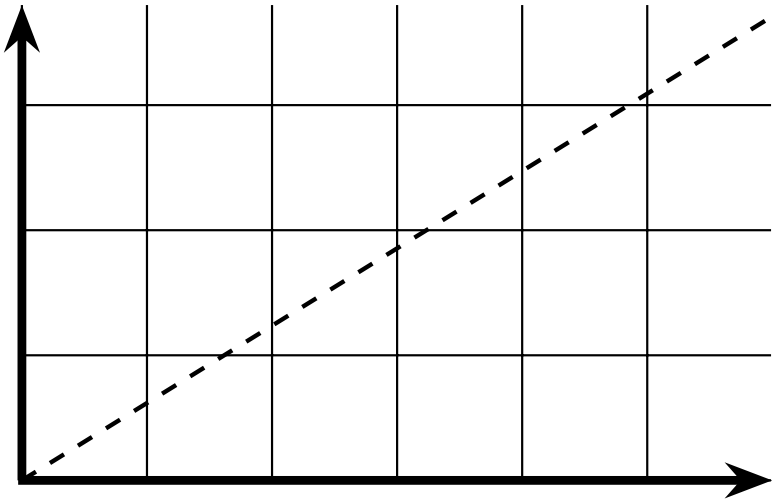
Proposition (Restivo, Romana & Sciortino)

Every prefix has a string attractor made of 2 consecutive positions.

Fibonacci is the *characteristic Sturmian* sequence with directive sequence $010101\cdots$.

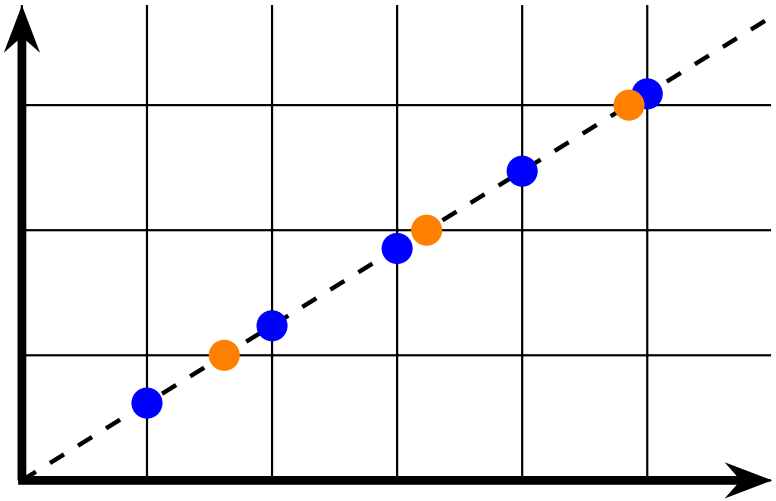
Sturmian sequences

Take a line with irrationnal slope,



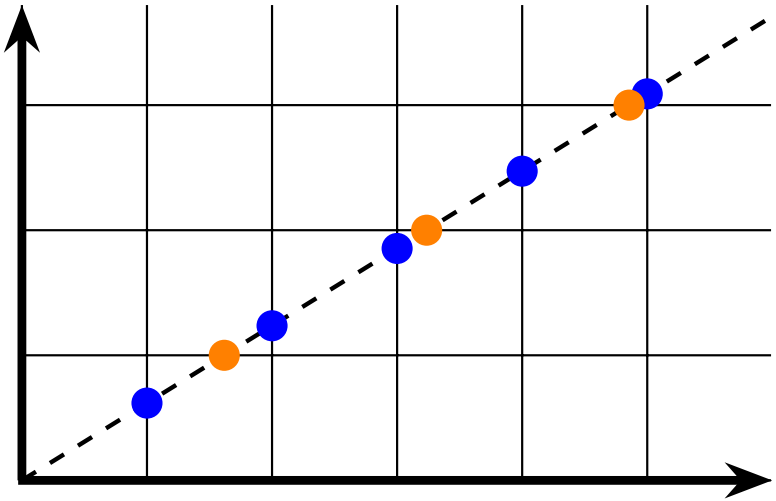
Sturmian sequences

Take a line with irrationnal slope,



Sturmian sequences

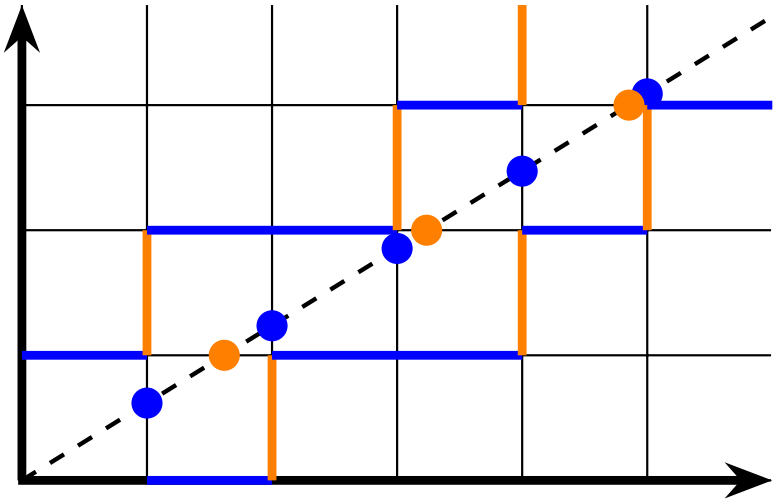
Take a line with irrationnal slope,



- intersections with the grid (billiard)

Sturmian sequences

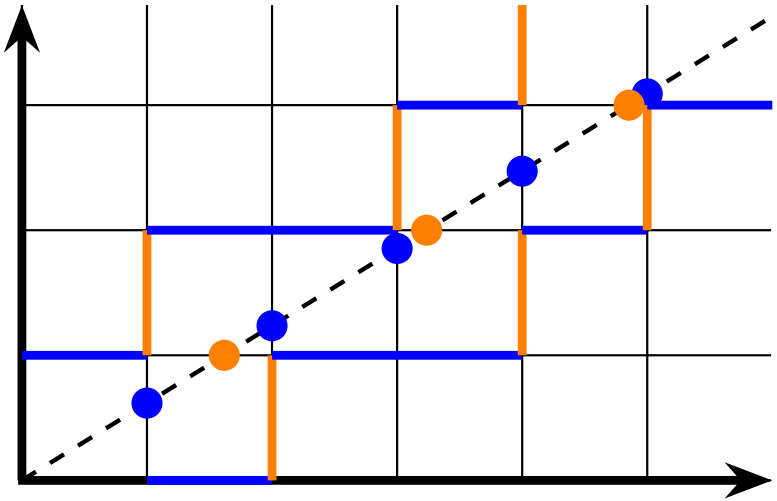
Take a line with irrationnal slope,



- intersections with the grid (billiard)
- approximation by grid lines

Sturmian sequences

Take a line with irrationnal slope,

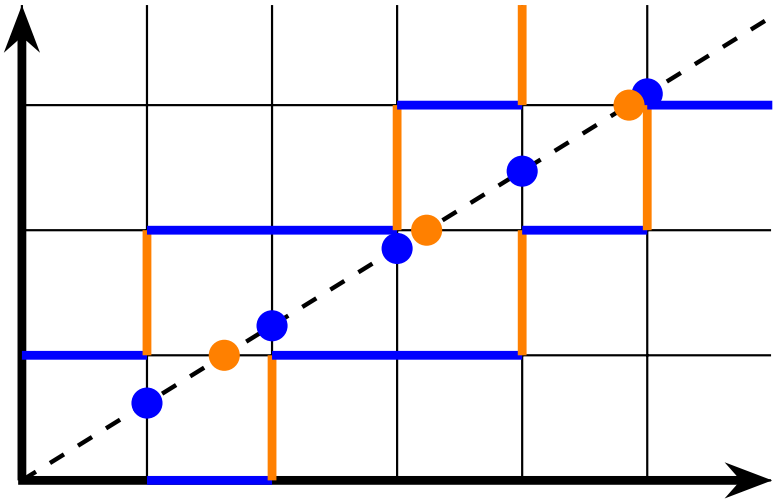


- intersections with the grid (billiard)
- approximation by grid lines

you get a binary sequence called *Sturmian*.

Sturmian sequences

Take a line with irrationnal slope,



- intersections with the grid (billiard)
- approximation by grid lines

you get a binary sequence called *Sturmian*.

If the intercept is zero, it is *characteristic Sturmian*.

Sturmian sequences and string attractors

Proposition (de Luca)

Characteristic Sturmian sequences are obtained by iterated palindromic closure for a directed sequence with 0's and 1's (infinitely many of each).

Sturmian sequences and string attractors

Proposition (de Luca)

Characteristic Sturmian sequences are obtained by iterated palindromic closure for a directed sequence with 0's and 1's (infinitely many of each).

Proposition (Restivo, Romana & Sciortino)

1. *If x is a characteristic Sturmian sequence, then every prefix has a string attractor made of 2 consecutive positions.*

Sturmian sequences and string attractors

Proposition (de Luca)

Characteristic Sturmian sequences are obtained by iterated palindromic closure for a directed sequence with 0's and 1's (infinitely many of each).

Proposition (Restivo, Romana & Sciortino)

1. *If x is a characteristic Sturmian sequence, then every prefix has a string attractor made of 2 consecutive positions.*
2. *If x is a Sturmian sequence, then infinitely many prefixes have a string attractor made of 2 consecutive positions.*

Sturmian sequences and string attractors

Proposition (de Luca)

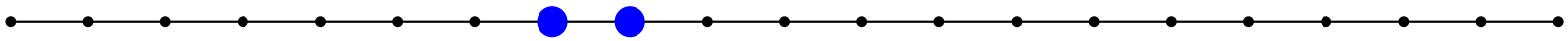
Characteristic Sturmian sequences are obtained by iterated palindromic closure for a directed sequence with 0's and 1's (infinitely many of each).

Proposition (Restivo, Romana & Sciortino)

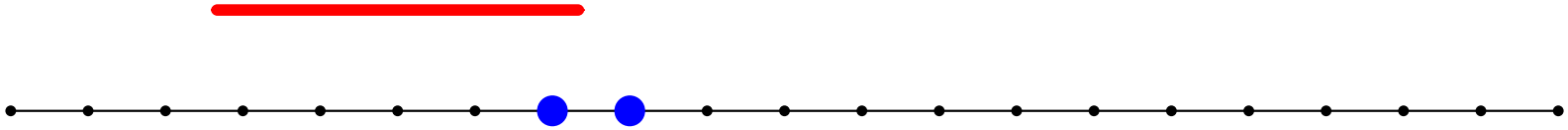
1. *If x is a characteristic Sturmian sequence, then every prefix has a string attractor made of 2 consecutive positions.*
2. *If x is a Sturmian sequence, then infinitely many prefixes have a string attractor made of 2 consecutive positions.*

What about the converse?

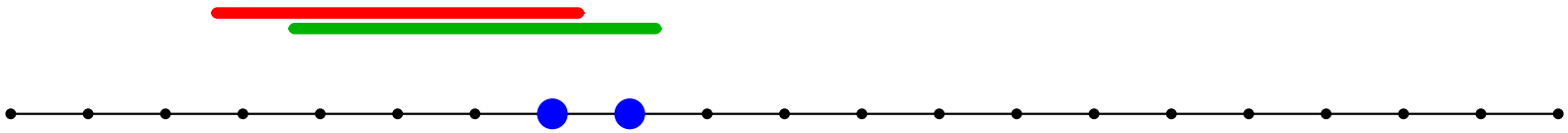
Sturmian sequences and string attractors (2)



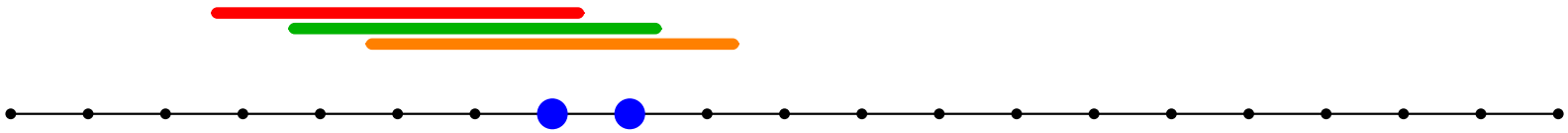
Sturmian sequences and string attractors (2)



Sturmian sequences and string attractors (2)

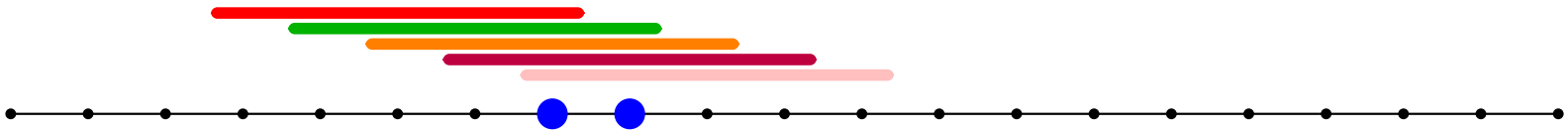


Sturmian sequences and string attractors (2)



Sturmian sequences and string attractors (2)

Sturmian sequences and string attractors (2)



Sturmian sequences and string attractors (2)

Sturmian sequences and string attractors (2)

$\rightarrow$ at most $n+1$ different factors of length n.

Sturmian sequences and string attractors (2)

$\rightarrow$ at most $n+1$ different factors of length n.

Theorem (Morse & Hedlund)

Let $p_x(n)$ denote the number of different length-n factors.

- *If x is aperiodic then $p_x(n) \geq n+1$ for all n.*

Sturmian sequences and string attractors (2)

$\rightarrow$ at most $n+1$ different factors of length n.

Theorem (Morse & Hedlund)

Let $p_x(n)$ denote the number of different length-n factors.

- *If x is aperiodic then $p_x(n) \geq n+1$ for all n.*
- *Sturmian sequences are exactly the sequences such that $p_x(n) = n+1$ for all n.*

Sturmian sequences and string attractors (2)

$\rightarrow$ at most $n+1$ different factors of length n.

Theorem (Morse & Hedlund)

Let $p_x(n)$ denote the number of different length-n factors.

- *If x is aperiodic then $p_x(n) \geq n+1$ for all n.*
- *Sturmian sequences are exactly the sequences such that $p_x(n) = n+1$ for all n.*

Proposition (Restivo, Romana & Sciortino)

A sequence is Sturmian if and only if it is aperiodic and infinitely many prefixes have a string attractor made of 2 consecutive positions.

Conclusion and open questions

Summary:

- combinatorial object related to text compression

Conclusion and open questions

Summary:

- combinatorial object related to text compression
- two ways of building string attractors for the Fibonacci sequence

Conclusion and open questions

Summary:

- combinatorial object related to text compression
- two ways of building string attractors for the Fibonacci sequence
  - using the substitutive structure

Conclusion and open questions

Summary:

- combinatorial object related to text compression
- two ways of building string attractors for the Fibonacci sequence
  - using the substitutive structure
  - using the palindromic structure

Conclusion and open questions

Summary:

- combinatorial object related to text compression
- two ways of building string attractors for the Fibonacci sequence
  - using the substitutive structure
  - using the palindromic structure

Open questions:

Conclusion and open questions

Summary:

- combinatorial object related to text compression
- two ways of building string attractors for the Fibonacci sequence
  - using the substitutive structure
  - using the palindromic structure

Open questions:

- Can we characterize the sequences such that s_x is bounded?

Conclusion and open questions

Summary:

- combinatorial object related to text compression
- two ways of building string attractors for the Fibonacci sequence
  - using the substitutive structure
  - using the palindromic structure

Open questions:

- Can we characterize the sequences such that s_x is bounded?
- Can we describe string attractors for other substitutive sequences?

Conclusion and open questions

Summary:

- combinatorial object related to text compression
- two ways of building string attractors for the Fibonacci sequence
  - using the substitutive structure
  - using the palindromic structure

Open questions:

- Can we characterize the sequences such that s_x is bounded?
- Can we describe string attractors for other substitutive sequences?
- Can we characterize other well-known families of sequences through string attractors?

The end

Thank you for listening!