Additive word complexity and `Walnut`

Pierre Popoli, Jeffrey O. Shallit and Manon Stipulanti

Université de Liège

ESI *Uniform Distribution of Sequences*
April, 23th 2025

Summary

Objects of study: Alphabet, word

- *Alphabet* A: finite set of symbols, called *letters*.
  $\rightarrow \{a, b, c, \ldots, z\}, \{0, 1\}, \{\cup, \wedge, \Gamma\} \ldots$

- *Word* over A: finite or infinite list of letters chosen in A.
  $\rightarrow A^*$: set of finite words with letters in A: 010, kayak, ...
  $\rightarrow A^\omega$: set of (right)-infinite word: $010101\cdots$, $001101010001\cdots$, ...
  $\rightarrow$ For a word w, $|w|$ is the length of w and $|w|_i$ is the number of occurrences of i in w.
  $\rightarrow \varepsilon$ is the empty word, i.e. the only word of length 0.
  $\rightarrow$ words $\Leftrightarrow$ sequences.

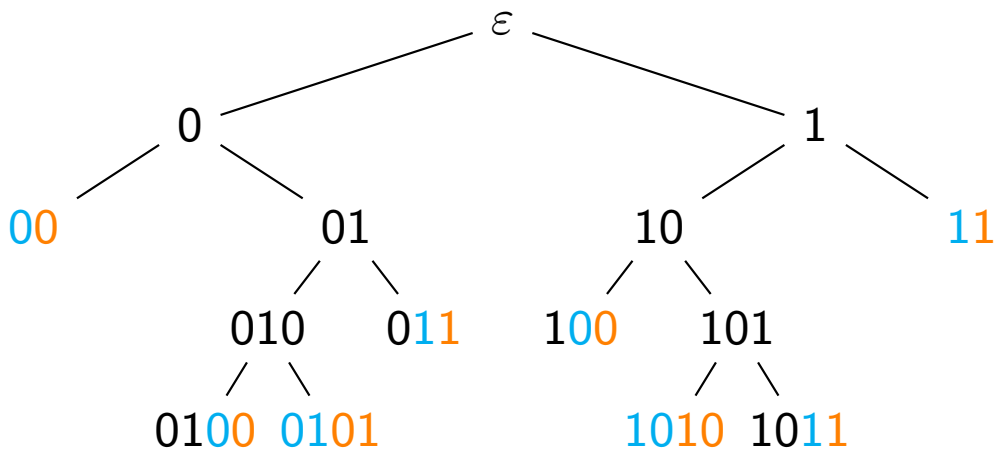
- Natural non-commutative operation on finite words: concatenation

$$(ab)(ba) = abba \neq baab.$$

  $\rightarrow A^*$ is a monoid equipped with the concatenation and neutral element ε.
  $\rightarrow$ For $n \geq 1$, $w^n = w \cdots w$.
  $\rightarrow$ A morphism h is a map $h : A^* \rightarrow B^*$ such that $h(xy) = h(x)h(y)$.

Pattern avoidance: birth of Combinatorics on Words

Thue noticed in 1906 that over a 2-letter alphabet, every word of length ≥ 4 contains a square, i.e. a consecutive subblock of the form xx, $|x| \geq 1$:



And in 1912, Thue construct an **infinite** word avoiding squares over a 3-letter alphabet

$$012021012102012\cdots,$$

that is based on the Thue–Morse sequence $(s_2(n) \bmod 2)_n$.

Erdös problem about abelian squares

28. There exists a sequence $\{s_n\}$ of 0's, 1's and 2's such that no two adjacent blocks from $\{s_n\}$ are the same. Presumably, the first (unpublished) proof of this was obtained by Rose Peltesohn and J. W. Sutherland.

Let N(k) be the least number N with the property that each sequence $\{s_n\}_{n=1}^{N}$ of numbers taken from the set $\{1, 2, \cdots, k\}$ contains two adjacent blocks such that each is a rearrangement of the other. My earliest conjecture, that $N(k) = 2^k - 1$, has been disproved by Bruijn and myself. It is not even known whether $N(4) < \infty$.

Figure: Some unsolved problems, Erdös (1957).

First point was also proved by Thue.

$x, y \in \Sigma^*$ are *abelian equivalent*, $x \sim_{\text{ab}} y$, if $|x| = |y|$ and $|x|_a = |y|_a$ for every letter a.
An abelian square is a word of the form xy,with $x \sim_{\text{ab}} y$.
Then second point can be reformulated in terms of abelian squares. One can easily check that every word over a 3-letter alphabet of length ≥ 8 contains an abelian square and Erdös suggested that the problem over 4 letters is worth studying.

Over alphabets of size ≥ 4, the story is long and with many intermediate results. The final answer is given by Keränen in 1992, where an infinite word abelian squarefree over an alphabet of four letters is given.

Additive framework

One can generalize this study to any equivalence relation: Given an equivalence relation $\sim$, what is the least number of letters that are needed to avoid $\sim$-squares ? $\sim$-cubes ?

Fix an alphabet $\Sigma = \{a_1, a_2, \ldots, a_\ell\} \subset \mathbb{N}$.
$u, v \in \Sigma^*$ are *additively equivalent*, $u \sim_{\text{add}} v$, if $|u| = |v|$ and $\sum_{i=1}^{\ell} a_i |u|_i = \sum_{i=1}^{\ell} a_i |v|_i$.

$$020 \sim_{\text{add}} 101, \quad 020 \nsim_{\text{ab}} 101, \quad 020 \nsim_{\text{add}} 11.$$

$\rightarrow$ Over an alphabet of size 2, abelian and additive problems are the same.

Conjecture: Brown–Freedman (1987)

There is no infinite word over the alphabet $\{1, \ldots, K\}$ that is additive squarefree.

$\rightarrow$ Still open. The conjecture is solved if we get rid of the assumption of equal length.

- Cassaigne, Currie, Schaeffer and Shallit (2014): Infinite word over the alphabet $\{0, 1, 3, 4\}$ with no additive cubes. Still open over $\{0, 1, 2, 3\}$.
- Rao (2015): infinite word over an alphabet of size 3 with no additive cubes (use the morphism of CCSS).

Summary

Factor complexity/Subword complexity

A *factor* of a word is one of its (contiguous) subblocks.

$$01101001100101101001011001101001\cdots$$

For a given word $\mathbf{x}$, for all $n \geq 0$, we let $\mathcal{L}_n(\mathbf{x})$ denote the set of length-n factors of $\mathbf{x}$.

FACTOR COMPLEXITY: Let $\mathbf{x}$ be an infinite word on Σ, $\rho_{\mathbf{x}} : \mathbb{N} \to \mathbb{N}, n \mapsto \#(\mathcal{L}_n(\mathbf{x}))$.

Theorem: Morse–Hedlund (1938)

The factor complexity of $\mathbf{x}$ is bounded if and only if $\mathbf{x}$ is ultimately periodic.

Standard complexity in many works in combinatorics on words, subshifts, ...

Abelian complexity

Recall that $u, v \in \Sigma^*$ are *abelian equivalent*, $u \sim_{\text{ab}} v$, if $|u| = |v|$ and $|u|_i = |v|_i$.

ABELIAN COMPLEXITY: Let $\mathbf{x}$ be an infinite word on Σ, the abelian complexity $\rho_{\mathbf{x}}^{\text{ab}}$ of $\mathbf{x}$ is

$$\rho_{\mathbf{x}}^{\text{ab}} : \mathbb{N} \to \mathbb{N}, n \mapsto \#(\mathcal{L}_n(\mathbf{x})/\sim_{\text{ab}}).$$

ABELIAN k-POWER: word $w = x_1 x_2 \cdots x_k$ such that $x_i \sim_{\text{ab}} x_j$ for every i, j.

Theorem: Richomme, Saari and Zamboni (2011)

If the abelian complexity of $\mathbf{x}$ is bounded, then $\mathbf{x}$ contains an abelian k-power for every $k \geq 1$.

$\to$ proof is based on van der Waerden's theorem.

Additive complexity

Fix an alphabet $\Sigma = \{a_1, a_2, \ldots, a_\ell\} \subset \mathbb{N}$.
$u, v \in \Sigma^*$ are *additively equivalent*, $u \sim_{\text{add}} v$, if $|u| = |v|$ and $\sum_{i=1}^{\ell} a_i |u|_i = \sum_{i=1}^{\ell} a_i |v|_i$.
$\rightarrow$ Depends on the **values** of the alphabet: not common in combinatorics on words.

ADDITIVE COMPLEXITY: Let $\mathbf{x}$ be an infinite word on Σ, the additive complexity $\rho_{\mathbf{x}}^{\text{add}}$ of $\mathbf{x}$ is

$$\rho_{\mathbf{x}}^{\text{add}} : \mathbb{N} \to \mathbb{N},\ n \mapsto \#(\mathcal{L}_n(\mathbf{x})/\sim_{\text{add}}).$$

Since $u \sim_{\text{ab}} v \implies u \sim_{\text{add}} v$, we have $\rho_{\mathbf{x}}^{\text{add}}(n) \leq \rho_{\mathbf{x}}^{\text{ab}}(n)$.

ADDITIVE k-POWER: word $w = x_1 x_2 \cdots x_k$ such that $x_i \sim_{\text{add}} x_j$ for every i, j.

Theorem: Ardal, Brown, Jungić and Sahasrabudhe (2012)

If the additive complexity of $\mathbf{x}$ is bounded, then $\mathbf{x}$ contains an additive k-power for every $k \geq 1$.

$\rightarrow$ proof also based on van der Waerden's theorem.

Summary

Classical automatic sequences

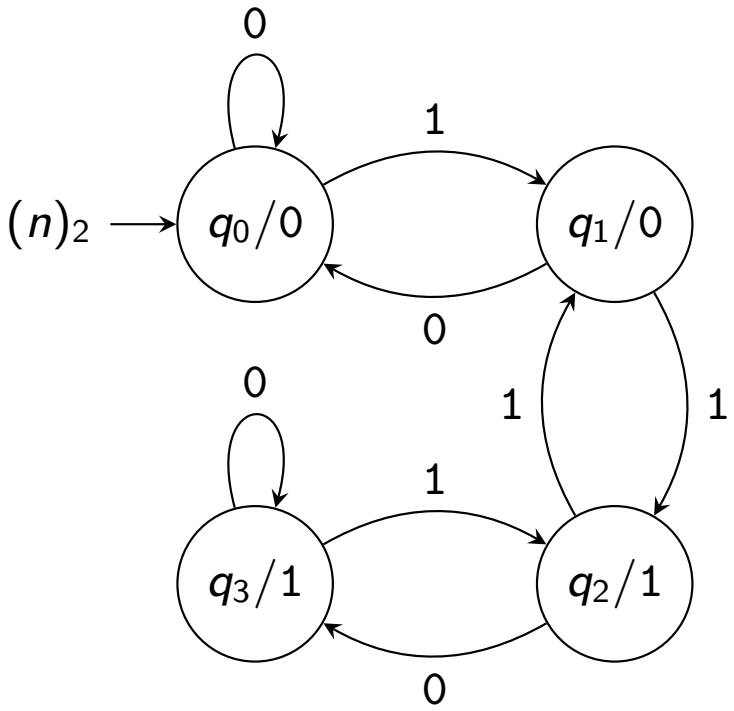


Figure: DFAO generating the Rudin–Shapiro sequence $\mathcal{R} = (r_n)_n$.

Since $(6)_2 = 110$, then $r_6 = 1$.

Automatic sequence

$\mathcal{S} = (s_n)$ is k-automatic over A if there exists $M = (Q, \Sigma_k, \delta, q_0, A, \tau)$ with

- Q: set of states.
- q_0: initial state.
- $\Sigma_k = \{0, 1, \ldots, k-1\}$.
- δ: transition function
- A: Alphabet.
- $\tau : Q \to A$ coding.

such that $s_n = \tau(\delta(q_0, (n)_k))$, $n \geq 0$.

Generalized automatic sequence

Abstract Numeration System U: triple $(L, A, <)$ where A alphabet, $<$ total order on A and L an infinite regular language over A.

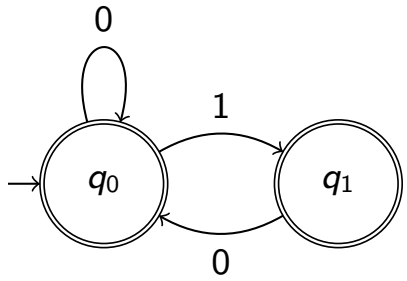


Figure: Zeckendorf numeration system

To each integer, we associate its unique representation, denoted $\mathrm{rep}_U(n)$, without leading zeros: $\varepsilon, 1, 10, 100, 101, 1000, 1001, 1010, \ldots$.

AUTOMATIC SEQUENCE: $\mathbf{x}$ is *U-automatic* if there exists a DFAO $\mathcal{A}$ such that, for all $n \geq 0$, the nth term $\mathbf{x}(n)$ of $\mathbf{x}$ is given by the output $\mathcal{A}(\mathrm{rep}_U(n))$ of $\mathcal{A}$.

A $\mathbf{x}$ is morphic if it is a fixed point of a prolongable morphism, under a coding.

Theorem: Rigo and Maes (2002)

A word is morphic if and only if it is U-automatic for some abstract numeration system U.

Ostrowski numeration system

Ostrowski numeration system (Ostrowski, 1922)

Let α be a real number with continued fraction $[a_0, a_1, \ldots]$ and $p_n/q_n = [a_0, \ldots, a_n]$. Then every natural number N can be written uniquely as $N = \sum_{k=0}^{\ell} b_k q_k$ where

- $0 \leq b_0 < a_1$,
- $0 \leq b_k \leq a_{k+1}$,
- If $b_k = a_{k+1}$, then $b_{k-1} = 0$.

Examples:

- For $\varphi = \frac{1+\sqrt{5}}{2} = [1, 1, \ldots]$, we get the Zeckendorf numeration system.
- For $\sqrt{2} = [1, 2, 2, \ldots]$, we get the Pell numeration system.

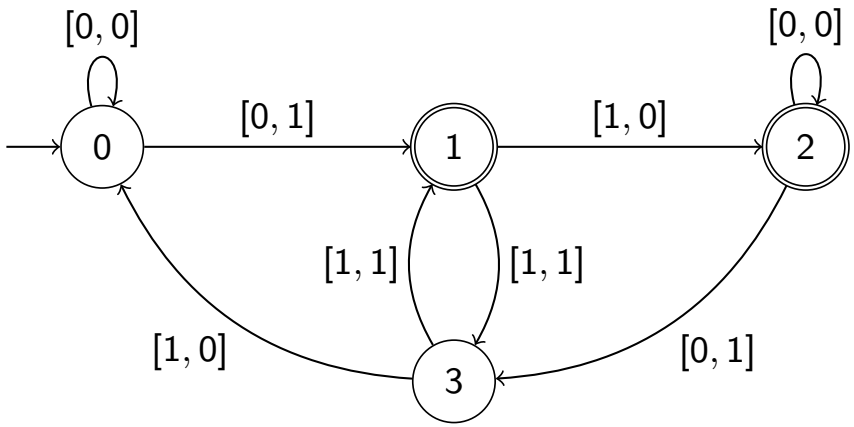
Theorem: Shallit (1994)

The Ostrowski numeration system of α is regular if and only if α is quadratic.

Synchronized sequences

SYNCHRONIZED SEQUENCE: $\mathbf{x}$ is *U-synchronized* if there exists a DFA that recognizes the language of U-representations of n and $\mathbf{x}(n)$ in parallel.

$\rightarrow$ Sequence of odious numbers o_n (n such that $t_n = 1$), ...



For instance the pair of words $[0110, 1101]$ is accepted, this means that $o_6 = 13$.
The odious numbers are $1, 2, 4, 7, 8, 11, 13, \ldots$.

$\rightarrow$ The factor complexity of any k-automatic sequence is synchronized.

Regular sequences

REGULAR SEQUENCE: $\mathbf{x}$ is U-regular if it admits a *linear representation*, i.e. there exist a column vector λ, a row vector γ and matrix-valued morphism μ such that

$$\mathbf{x}(n) = \lambda\mu(\mathrm{rep}_U(n))\gamma.$$

$\rightarrow$ Sum of digits function $s_2(n)$ satisfies $\begin{cases} s_2(2n) = s_2(n), \\ s_2(2n+1) = s_2(n) + 1. \end{cases}$

And provides the following linear representation

$$\lambda = \begin{pmatrix} 1 & 0 \end{pmatrix}, \ \gamma = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \ \mu(0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ \mu(1) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Since $(13)_2 = 1101$, this gives that $s_2(13) = \lambda\mu(1)\mu(1)\mu(0)\mu(1)\gamma = 3$.

- Automatic $\Longrightarrow$ Synchronized $\Longrightarrow$ Regular.
- Regular and finitely many values $\Longrightarrow$ Automatic.

Long standing conjecture

Conjecture: Parreau, Rigo, Rowland and Vandomme (2015)

The abelian complexity of a U-automatic sequence is a U-regular sequence.

$\rightarrow$ Widely open. Some examples:

- The abelian complexity of the Thue–Morse sequence is automatic.
- The abelian complexity of the Rudin–Shapiro sequence is regular.
- Madill and Rampersad (2013): The abelian complexity of the paper-folding word is 2-regular (and unbounded).

This conjecture can be formulated for any complexity, including the additive complexity.

- Chen, Wen and Wu (2019): Let $\mathbf{x}$ be the fixed point of $0 \mapsto 01$, $1 \mapsto 12$, $2 \mapsto 20$. Then $\rho_{\mathbf{x}}^{\mathrm{add}}(n) = 2\lfloor \log_2 n \rfloor + 3$ for all $n \geq 1$. In particular, $(\rho_{\mathbf{x}}^{\mathrm{add}}(n))_{n \geq 0}$ is 2-regular.

Summary

General results

Let $\Sigma = \{a_1, \ldots, a_k\}$.

Parikh vector: for $w \in \Sigma^*$, $\Psi(w) = (|w|_{a_1}, \ldots, |w|_{a_k})$.

Theorem: P., Shallit and Stipulanti (2024)

Let $\mathbf{x}$ be an U-automatic sequence for some ANS U. Assume that

- the additive complexity $\rho_{\mathbf{x}}^{\text{add}}$ of $\mathbf{x}$ is bounded above by a constant,
- the Parikh vectors of length-n prefixes of $\mathbf{x}$ form a synchronized sequence.

Then $\rho_{\mathbf{x}}^{\text{add}}$ is U-automatic and the DFAO computing it is effectively computable.

The second condition is a strong assumption but this applies to well-known families.

Parikh-collinear morphism

A morphism $\varphi\colon \Sigma^* \to \Delta^*$ is *Parikh-collinear* if the Parikh vectors $\Psi(\varphi(b))$, $b \in \Sigma$, are collinear.

Lemma: Rigo, Stipulanti and Whiteland (2023)

Let $\mathbf{x} := \varphi^\omega(a)$ for $\varphi\colon \Sigma^* \to \Sigma^*$ a Parikh-collinear morphism prolongable on the letter a. For all $b \in \Sigma$, the sequence $(|\mathrm{pref}_n(\mathbf{x})|_b)_{n\geq 0}$ is k-synchronized for $k = \sum_{b\in\Sigma} |\varphi(b)|_b$.

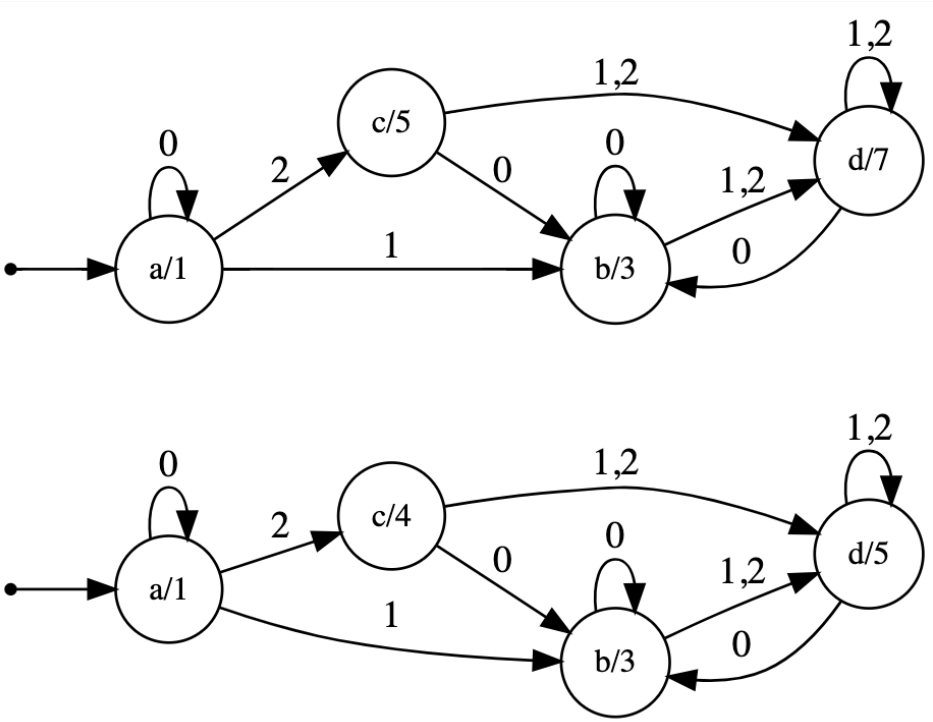
Corollary: P., Shallit and Stipulanti (2024)

Let $\mathbf{x} := \varphi^\omega(a)$ for $\varphi\colon \Sigma^* \to \Sigma^*$ a Parikh-collinear morphism prolongable on the letter a. Then the additive complexity function $\rho_{\mathbf{x}}^{\mathrm{add}}$ of $\mathbf{x}$ is k-automatic for $k = \sum_{b\in\Sigma} |\varphi(b)|_b$. Moreover, the automaton generating $\rho_{\mathbf{x}}^{\mathrm{add}}$ can be effectively computed given φ and a.

Example of Parikh-collinear morphism

Let $f : \{0,1,2\}^* \to \{0,1,2\}^*$ be defined by $0 \mapsto 012, 1 \mapsto 112002, 2 \mapsto \varepsilon$.

Consider $\mathbf{x} = f^\omega(0) = 012112002112002\cdots$ the fixed point of f starting by 0.



Abelian complexity (top) and Additive complexity (bot)

Then we have $\rho_{\mathbf{x}}^{\mathrm{ab}} = 135(377)^\omega$ and $\rho_{\mathbf{x}}^{\mathrm{add}} = 134(355)^\omega$.

$\to$ Performed with `Walnut`.

Tribonacci word 1/3: Definition

The *Tribonacci word* **tr** is the fixed point of the morphism $0 \mapsto 01$, $1 \mapsto 02$, $2 \mapsto 0$.

- Episturmian word: $\rho_{\mathbf{tr}}(n) = 2n + 1$ for every $n \geq 0$.
- Tribonacci numeration system: Based on the Tribonacci numbers $T_{n+3} = T_{n+2} + T_{n+1} + T_n$, $n \geq 0$, $T_0 = 0, T_1 = 1, T_2 = 1$. Supported by `Walnut`.

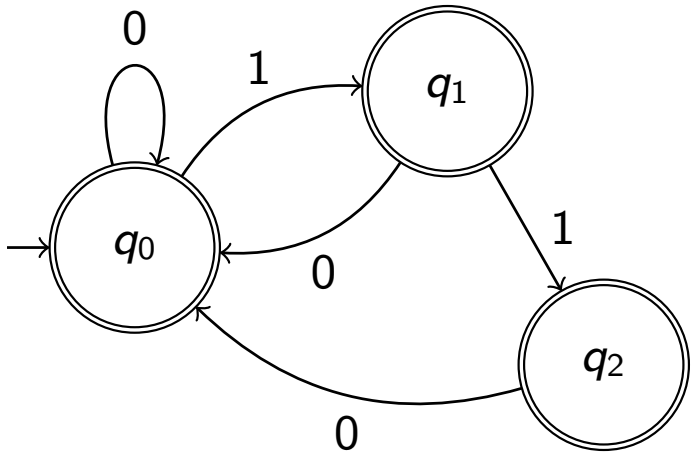


Figure: Tribonacci numeration system

$\varepsilon, 1, 10, 11, 100, 101, 110, 1000, 1001, 1010, 1011, 1100, 1101, 10000, \ldots$

$\rightarrow$ Tribonacci NS is not Ostrowski NS based on the dominant root of $X^3 = X^2 + X + 1$.

Tribonacci word 2/3: Abelian complexity

Theorem: Richomme, Saari and Zamboni (2010)

The abelian complexity function of **tr** satisfies $\rho^{\text{ab}}_{\mathbf{tr}}(n) \in \{3, 4, 5, 6, 7\}$, for $n \geq 1$.

These values are taken infinitely often:

- Values $\{3, 7\}$ by RSZ (2010),
- Values $\{4, 5, 6\}$ by Turek (2013).

Main strategy: find an infinite parametric family x_n that satisfies $\rho^{\text{ab}}_{\mathbf{tr}}(x_n) = k$.
For instance $\rho^{\text{ab}}_{\mathbf{tr}}(n) = 3 \Leftrightarrow n = 1$ or $n = \frac{1}{2}(T_{m+2} + T_m - 1)$ for some $m \geq 0$.

Theorem: Turek (2013), Shallit (2021)

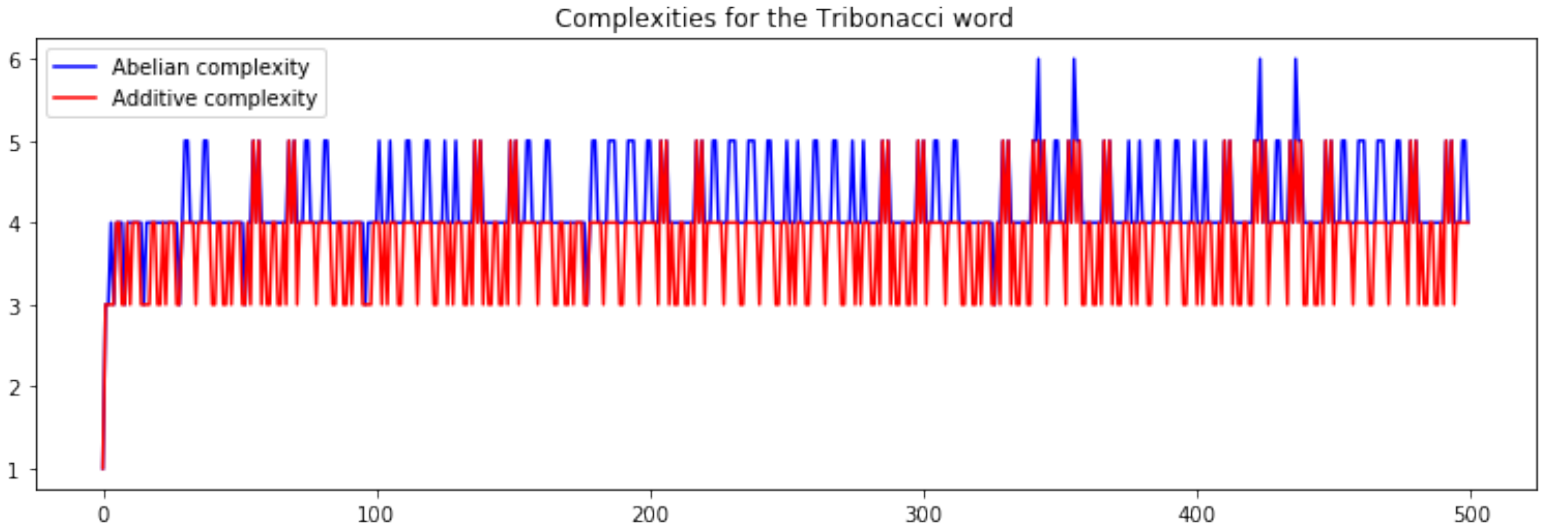
The abelian complexity function $\rho^{\text{ab}}_{\mathbf{tr}}$ is computed by a 78-state Tribonacci DFAO.

And this automaton allows to prove that each value is taken infinitely often **very easily**.

Tribonacci word 3/3: Additive complexity

Theorem: P., Shallit and Stipulanti (2024)

The additive complexity function of **tr** satisfies $\rho_{\mathbf{tr}}^{\mathrm{add}}(n) \in \{3, 4, 5\}$, for $n \geq 1$. Furthermore, each of the three values is taken infinitely often and $\rho_{\mathbf{tr}}^{\mathrm{add}}$ is computed by a 76-state Tribonacci DFAO.



→ Also performed with `Walnut`.

Thue–Morse over three letters: Bounded additive and abelian complexities

The *ternary Thue–Morse word* $\mathbf{t}_3$ is the fixed point of the morphism

$$0 \mapsto 012, \quad 1 \mapsto 120, \quad 2 \mapsto 201.$$

This word satisfies $\mathbf{t}_3(n) = s_3(n) \bmod 3$.

Theorem: Kaboré and Kientéga (2017)

The abelian complexity function $\rho^{\mathrm{ab}}_{\mathbf{t}_3}$ is the periodic infinite word $136(766)^\omega$.

Theorem: P., Shallit and Stipulanti (2024)

The additive complexity function $\rho^{\mathrm{add}}_{\mathbf{t}_3}$ is the periodic infinite word 135^ω.

Two possible proofs:

- by "hand". Not difficult, study the form of every factor of length n.
- Or by `Walnut` of course.

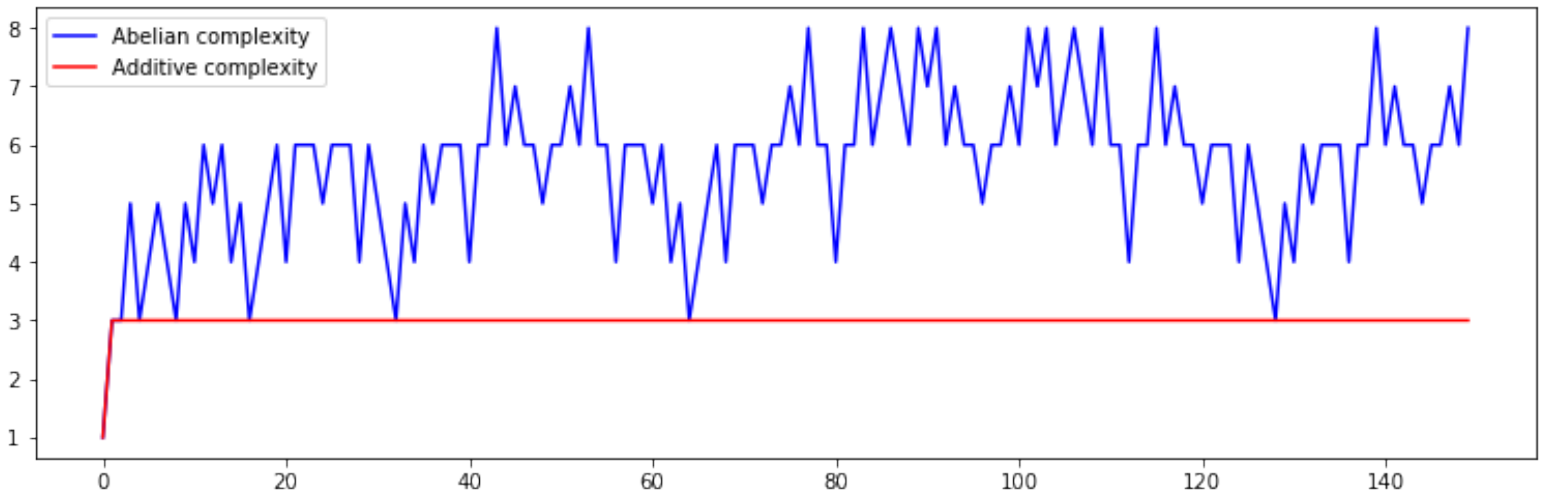
Variant of Thue–Morse: Bounded additive and unbounded abelian complexities

Let **vtm** be the fixed point of $f : 0 \mapsto 012, 1 \mapsto 02, 2 \mapsto 1$, starting with 0.

The word **vtm** $= 012021012102012021012\cdots$ is squarefree.

Theorem: Blanchet-Sadri, Currie, Rampersad and Fox (2014)

The abelian complexity of **vtm** is $O(\log n)$ and it is $\Omega(1)$ with constant 3.



Theorem: P., Shallit and Stipulanti (2024)

The additive complexity of **vtm** is the periodic infinite word 13^ω.

$\rightarrow$ The proof is actually really simple and does not require `Walnut`.

Summary

First order logic in Combinatorics on Words

Many properties of words can be phrased in first-order logic. For instance, the following formula if the two factors in $\mathbf{x}$ starting at positions i and j of length n are equal:

$$\textsc{FactorEq}(i,j,n) := \forall t < n \implies \mathbf{x}[i+t] = \mathbf{x}[j+t].$$

For abelian/additive equivalence, this is not possible to write such a formula in general.

But with the extra assumption of synchronicity of occurrences of any given letter in a length-n prefix, we can make it work. Suppose that we have a DFA for $\textsc{Pref}_a(n,t)$, where t is the number of a in the prefix of length n. Then

$$\textsc{Fac}_a(i,n,s) := \exists t,u\ \textsc{Pref}_a(n,t) \wedge \textsc{Pref}_a(i+n,u) \wedge (t+s=u).$$

And then we can test the additive equivalence of two factors (over $0,1,2$ for example) as:

$$\begin{aligned}\textsc{AddEq}(i,j,n) := \exists p,q,r,s\ \textsc{Fac}_1(i,n,p) \wedge \textsc{Fac}_2(i,n,q)\\ \wedge \textsc{Fac}_1(j,n,r) \wedge \textsc{Fac}_2(j,n,s)\\ \wedge (p+2\times q = r+2\times s).\end{aligned}$$

Walnut: automatic theorem prover

`Walnut` is a free software program (in Java) designed by Mousavi (2016) that allows one to automatically decide the truth of assertions about many properties in CoW.

Let U be an abstract numeration system.

Theorem: Base of Walnut

There is an algorithm that, given a formula φ with no free variables, phrased in first-order logic, using only the universal and existential quantifiers, addition and subtraction of variables and constants, logical operations, comparisons, and indexing into a given U-automatic sequence $\mathbf{x}$, will decide the truth of that formula.
Furthermore, if φ has $t \geq 1$ free variables, the algorithm produces a DFA $\mathcal{M}$ that recognizes the language of all U-representations of t-tuples of natural numbers that make φ evaluate to true.

$\rightarrow$ Based on works of Büchi and Bruyère et al. about decidability on $\langle \mathbb{N}, +, V_k \rangle$.

`Walnut` essentially implements the decision procedure of this theorem. It can handle any abstract numeration systems such as

- The classical k-numeration systems.
- Tribonacci, Dumont–Thomas, ...
- Ostrowski numeration systems based on a quadratic number.

Use of Walnut

```
morphism h "0->01 1->02  2->0":
promote TR h:
```

The sequences $|\mathbf{tr}[0..n-1]|_a$ are synchronized for all $a \in \{0,1,2\}$.

```
def Fac0 "?msd_tri Aq Ar ($Pref0(i+n,q) & $Pref0(i,r)) => (q=r+s)":
```

And the additive equivalence of $\mathbf{tr}[i..i+n-1]$ and $\mathbf{tr}[j..j+n-1]$ is tested like this:

```
def triAddEq "?msd_tri Ep,q,r,s $Fac1(i,n,p) & $Fac2(i,n,q)
              & $Fac1(j,n,r) & $Fac2(j,n,s) & p+2*q=r+2*s":
```

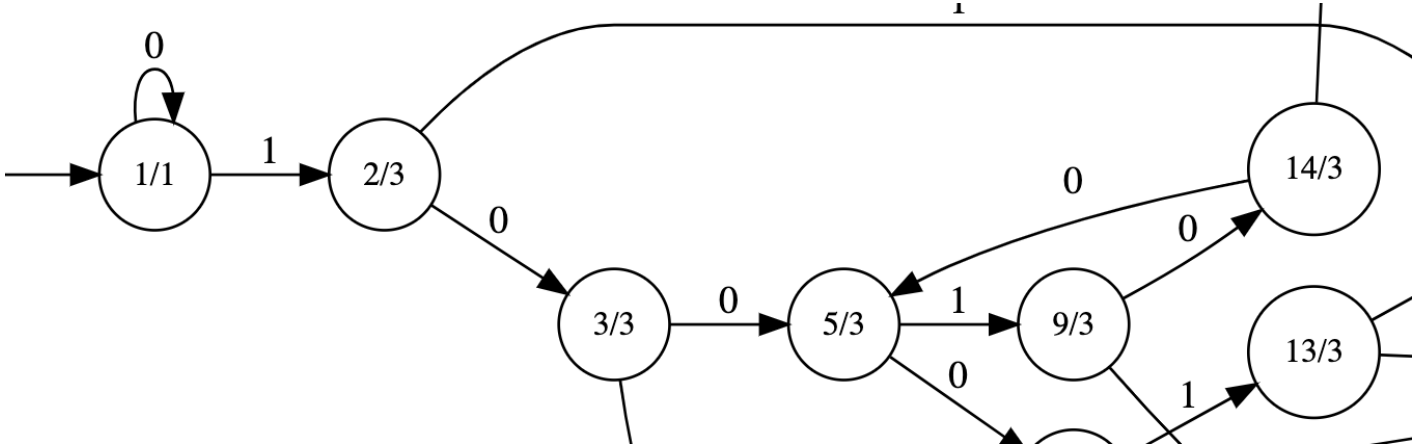
$\rightarrow$ This step can be very long for some examples (and even not end). Here, it has 4584 states
And, we obtain a linear representation of the additive complexity as follows

```
eval triAddComp n "?msd_tri Aj j<i => ~$triAddEq(i,j,n)":
```

$\rightarrow$ We count novel occurrences with such a formula. See Walnut's book of Shallit for more details about this procedure. This returns a linear representation of size 184, and of size 62 after minimization.
$\rightarrow$ Finally, we apply the "Semigroup trick algorithm" to deduce an 76-states automaton.

Each value is obtained infinitely often

First solution: look for loops in the automaton.



Then every n such that $(n)_T = 100(100)^k$ for $k \geq 0$ satisfies $\rho_{\mathbf{tr}}^{\mathsf{add}}(n) = 3$.

Second solution: with `Walnut` like this:

```
eval triAddComp_3 "?msd_tri An Em (m>n) & TAC[m]=@3":
eval triAddComp_4 "?msd_tri An Em (m>n) & TAC[m]=@4":
eval triAddComp_5 "?msd_tri An Em (m>n) & TAC[m]=@5":
```

and `Walnut` then returns `TRUE` each time.

Summary

Changing the letters: Thue–Morse word over three letters

The additive complexity **depends** on the values of the letters.
We can always suppose that one value is 0.

Let ℓ, m be integers such that $1 \leq \ell < m$.
The (ℓ, m)-*Thue–Morse word* $\mathbf{t}_{\ell,m}$ is the fixed point of the morphism

$$0 \mapsto 0\ell m, \quad \ell \mapsto \ell m 0, \quad m \mapsto m 0 \ell.$$

This does not change the abelian complexity: $\rho^{\mathsf{ab}}_{\mathbf{t}_{\ell,m}} = 136(766)^\omega$.

Theorem: P., Shallit and Stipulanti (2024)

Let ℓ, m be integers such that $1 \leq \ell < m$.

The additive complexity of $\mathbf{t}_{\ell,m}$ satisfies $\rho^{\mathsf{add}}_{\mathbf{t}_{\ell,m}} = \begin{cases} \rho^{\mathsf{ab}}_{\mathbf{t}_{\ell,m}} \text{ if } m \neq 2\ell \\ 135^\omega \text{ if } m = 2\ell. \end{cases}$

$\rightarrow$ Combinatorial proof, not possible with `Walnut`.

Changing the letters: Variant of Thue–Morse

Let $f_\lambda : \begin{cases} 0 \mapsto 01\lambda, \\ 1 \mapsto 0\lambda, \\ \lambda \mapsto 1, \end{cases}$ and $\mathbf{vtm}_\lambda = f_\lambda^\omega(0) = 01\lambda0\lambda101\lambda10\lambda\cdots$.

- For $\lambda = 2$, we have $\rho^{\mathsf{add}}_{\mathbf{vtm}_2}(n) = 3$ for all $n \geq 1$.
- For $\lambda = 3$, $\rho^{\mathsf{add}}_{\mathbf{vtm}_3}(n) = \rho^{\mathsf{ab}}_{\mathbf{vtm}_3}(n)$ for all $1 \leq n \leq 11$.
- For $\lambda = 4$, $\rho^{\mathsf{add}}_{\mathbf{vtm}_4}(n) = \rho^{\mathsf{ab}}_{\mathbf{vtm}_4}(n)$ for all $1 \leq n \leq 43$.

Theorem: P., Shallit and Stipulanti (2024)

We have $\rho^{\mathsf{add}}_{\mathbf{vtm}_\lambda}(n) = \rho^{\mathsf{ab}}_{\mathbf{vtm}_\lambda}(n)$ for every $\lambda \geq 5$.

$\rightarrow$ The value 5 forces that every two additive equivalent factors are actually abelian equivalent.

Summary

Extensions and open problems

Theorem: Couvreur, Delacourt, Ollinger, Shallit, P. and Stipulanti (2025+)

The abelian complexity of the fixed point of a prolongable Pisot type morphism is an automatic sequence in the associated Dumont–Thomas numeration system and the DFAO computing it is effectively computable.

$\rightarrow$ the case of k-abelian complexity is also treated under assumptions on the sliding-block code.

Recall that:

- Morse–Hedlund's theorem: Bounded factor complexity $\Longrightarrow$ there exists a k-power for all $k \geq 1$.
- RSZ theorem: Bounded abelian complexity $\Longrightarrow$ there exists an abelian k-power for all $k \geq 1$.
- ABJS theorem: Bounded additive complexity $\Longrightarrow$ there exists an additive k-power for all $k \geq 1$.

$\rightarrow$ this is also true for k-abelian complexity, cyclic complexity, ...

Open problem

Characterize complexities such that the latter property is true.

Thank you for your attention !