Effective computations of abelian complexities

Manon Stipulanti

Joint work with
J.-M. Couvreur, M. Delacourt, N. Ollinger
P. Popoli, J. Shallit

Numeration and Substitutions 2025
Tsukuba (Japan)
September 8–12, 2025

Notation in combinatorics on words

- infinite sequences in **bold**
- $|w|_a = \#$ letters a in w
- factor = subsequence of consecutive letters
- $|w|_x = \#$ occurrences of the factor x in w

Example: $|\texttt{reappear}|_{\texttt{a}} = 2 = |\texttt{reappear}|_{\texttt{e}}$

factor
re**appear**
re**app**ear
reap**ear**
reap**pear**

$|00100|_{00} = 2$ $|000100|_{00} = 3$

- length-n factors of $\mathbf{x}$: $\mathrm{Fac}_n(\mathbf{x})$

How to study combinatorial structure?

factor complexity $\rho_{\mathbf{x}} : \mathbb{N} \to \mathbb{N}, n \mapsto \#\mathrm{Fac}_n(\mathbf{x})$

Example: Fibonacci seq. $\mathbf{f} = 0100101001 \cdots$ (f.p. of $0 \mapsto 01, 1 \mapsto 0$) [Sloane, A003849]

n	$\mathrm{Fac}_n(\mathbf{f})$	$\rho_{\mathbf{f}}(n)$
0	ε	1
1	$0, 1$	2
2	$00, 01, 10$	3
3	$001, 010, 100, 101$	4
4	$0010, 0100, 0101, 1001, 1010$	5

Theorem (Morse–Hedlund 1938)

$\mathbf{x}$ with ℓ distinct letters

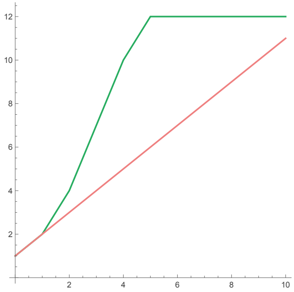
$\mathbf{x}$ ultimately periodic

iff $\rho_{\mathbf{x}}$ bounded

iff $\exists n \in \mathbb{N}$ s.t. $\rho_{\mathbf{x}}(n) < n + \ell - 1$

$\mathbf{x}$ Sturmian iff $\rho_{\mathbf{x}}(n) = n + 1 \quad \forall n$

(binary, aperiodic, minimal factor complexity)



$\mathbf{x} = y\, 01\, 01\, 01 \cdots$

$\mathbf{f} = 0100101001001 \cdots$

Variations

counting “different enough” factors

- with specific properties

  e.g. palindromes (Droubay–Pirillo 1999)

  privileged (Peltomäki 2013)

- extracted along specific subsequences

  e.g. arithmetical (Avgustinovich–Fon-Der-Flaass–Frid 2000)

  maximal pattern (Kamae–Zamboni 2002)

- with equivalence relations $u \sim v$

  $\mathbb{N} \to \mathbb{N}, n \mapsto \#(\mathrm{Fac}_n(\mathbf{x})/\sim)$

today: 2 equivalence relations

but many more

e.g. see intro. Allouche–Campbell–Li–Shallit–S 2025

Abelian context

- abelian equivalence: $u \sim_{\text{ab}} v$ if $|u|_a = |v|_a \ \forall a \in A$

  Example: `evil` $\sim_{\text{ab}}$ `live` $\sim_{\text{ab}}$ `veil` $\sim_{\text{ab}}$ `vile`

- abelian complexity $\rho^{\text{ab}}_{\mathbf{x}}$: $\mathbb{N} \to \mathbb{N}, n \mapsto \#(\text{Fac}_n(\mathbf{x})/\sim_{\text{ab}})$

  Example: Fibonacci sequence $\mathbf{f} = 0100101001\cdots$

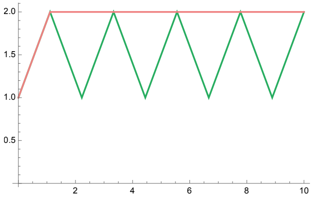
n	$\text{Fac}_n(\mathbf{f})$	$\rho_{\mathbf{f}}(n)$	$\{\cdot\}_{\sim_{\text{ab}}}$	$\rho^{\text{ab}}_{\mathbf{f}}(n)$
0	ε	1	$\{\varepsilon\}$	1
1	$0, 1$	2	$\{0\}$ $\{1\}$	2
2	$00, 01, 10$	3	$\{00\}$ $\{01, 10\}$	2
3	$001, 010, 100, 101$	4	$\{001, 010, 100\}$ $\{101\}$	2

Theorem (Coven–Hedlund 1973)

$\mathbf{x}$ purely periodic iff $\exists n \in \mathbb{N}$ s.t. $\rho^{\text{ab}}_{\mathbf{x}}(n) = 1$

$\mathbf{x}$ Sturmian iff $\mathbf{x}$ binary aperiodic and $\rho^{\text{ab}}_{\mathbf{x}}(n) = 2 \ \forall n \geq 1$

+see survey by Fici–Puzynina 2023



$\mathbf{x} = 01\,01\,01\cdots$

$\mathbf{f} = 0100101001001\cdots$

General abelian context

$k \in \mathbb{N}_0$

- *k*-abelian equivalence: $u \sim_k v$ if $|u|_x = |v|_x \ \forall x$ s.t. $|x| \leq k$

  Example: 01001 $\sim_2$ 00101 but 01001 $\not\sim_3$ 00101

x	0	1	00	01	10	11	101
$\|01001\|_x$	3	2	1	2	1	0	0
$\|00101\|_x$	3	2	1	2	1	0	1

  Remark: $\sim_1 = \sim_{\mathrm{ab}}$

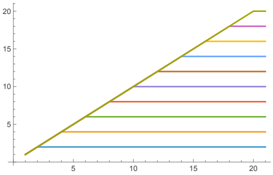
  $\sim_{k+1}$ refines $\sim_k$

- *k*-abelian complexity $\rho^k_{\mathbf{x}}\colon \mathbb{N} \to \mathbb{N}, n \mapsto \#(\mathrm{Fac}_n(\mathbf{x})/\sim_k)$

Theorem (Karhumäki–Saarela–Zamboni 2013)

$\mathbf{x}$ aperiodic

$\mathbf{x}$ Sturmian iff $\rho^k_{\mathbf{x}}(n) = \begin{cases} n+1 & \text{if } 0 \leq n \leq 2k-1 \\ 2k & \text{if } n \geq 2k \end{cases}$



$\mathbf{x}$ Sturmian $\Rightarrow \rho^k_{\mathbf{x}}$ ult. constant $\Rightarrow \rho^k_{\mathbf{x}}$ bounded

Bounded abelian complexities

$\mathbf{x}$ C-balanced if $||u|_a - |v|_a| \leq C$, $\forall\, u, v \in \mathrm{Fac}(\mathbf{x})$ with $|u| = |v|$, $\forall a \in A$
1-balanced = balanced

Characterization (Folklore)

$\rho^{\mathrm{ab}}_{\mathbf{x}}$ is bounded iff $\exists C > 0$ s.t. $\mathbf{x}$ is C-balanced

Example: Sturmian sequences (known constants)

$\mathbf{x}$ (k, C_k)-balanced if $||u|_w - |v|_w| \leq C_k$, $\forall\, u, v \in \mathrm{Fac}(\mathbf{x})$ s.t $|u| = |v|$, $\forall w \in A^k$

Characterization (Karhumäki–Saarela–Zamboni 2013)

$\rho^{k}_{\mathbf{x}}$ is bounded iff $\exists C_k > 0$ s.t. $\mathbf{x}$ is (k, C_k)-balanced

Remark: $\rho^k_{\mathbf{x}} \leq C_k \Rightarrow \mathbf{x}$ $(k, C_k - 1)$-balanced
$\mathbf{x}$ (k, C_k)-balanced $\Rightarrow \rho^k_{\mathbf{x}} \leq (C_k + 1)^k$
Bounds not optimal,
e.g. Sturmian sequences (k, k)-balanced (Fagnot–Vuillon 2002)
$\rho^k \leq 2k$ (Karhumäki–Saarela–Zamboni 2013)

Automatic sequences (since Büchi 1960)

- $S = (L, A, <)$ abstract numeration system (ANS) where A alphabet ordered by $<$ and $L \subseteq A^*$ (Lecomte–Rigo 2000)

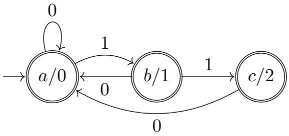
  Example: Tribonacci numeration system: forbid 111

n	0	1	2	3	4	5	6	7	8
$\mathrm{rep}_T(n)$	ε	1	10	11	100	101	110	1000	1001

- $\mathbf{x}$ S-automatic if $\exists$ S-DFAO $\mathcal{A}$ producing $\mathbf{x}$, i.e. $\mathbf{x}_n = \mathcal{A}(\mathrm{rep}_S(n))$

  Example: Tribonacci seq. $\mathbf{t} = 01020100102\cdots$ (f.p. of $0 \mapsto 01, 1 \mapsto 02, 2 \mapsto 0$) [Sloane, A080843]

n	0	1	2	3	4	5	6	7	8
$\mathrm{rep}_T(n)$	ε	1	10	11	100	101	110	1000	1001
$\mathbf{t}_n$	0	1	0	2	0	1	0	0	1



Regular sequences (since Allouche–Shallit 1992)

$S = (L, A, <)$ ANS

$\mathbf{x}$ S-regular if $\exists$ linear representation (λ, μ, γ) of $\mathbf{x}$, i.e., λ row vector, γ column vector, and μ matrix-valued morphism s.t. $\mathbf{x}_n = \lambda\mu(\mathrm{rep}_S(n))\gamma$

Example: sum-of-digits $\mathbf{s} = 0112113\cdots$ in base 2 [Sloane, A000120]

n	0	1	2	3	4	5	6	7
$\mathrm{rep}_2(n)$	ε	1	10	11	100	101	110	111
$\mathbf{s}_n$	0	1	1	2	1	2	2	3

$$\lambda = \begin{pmatrix} 0 & 1 \end{pmatrix} \quad \gamma = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\mu(0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \mu(1) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

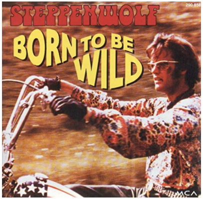
e.g. $\mathbf{s}_4 = \lambda\mu(1)\mu(0)\mu(0)\gamma$

Theorem

S-automatic $\Rightarrow$ S-regular

S-regular with finite image $\Rightarrow$ S-automatic

A (wild) conjecture



Conjecture (Parreau–Rigo–Rowland–Vandomme 2015)

The k-abelian complexity ρ^k of an S-automatic sequence is S-regular

Examples supporting this:

$\mathbf{x}$	what	who
Thue–Morse	$\rho^{\mathrm{ab}}_{\mathbf{x}} = 1(23)^\omega$	Richomme–Saari–Zamboni 2011
Sturmian	$\rho^k_{\mathbf{x}}$ ult. constant	Karhumäki–Saari–Zamboni 2013
paperfolding	$\rho^{\mathrm{ab}}_{\mathbf{x}}$ 2-regular	Madill–Rampersad 2013
Thue–Morse	$\rho^2_{\mathbf{x}}$ 2-regular	Parreau–Rigo–Rowland–Vandomme 2015
period-doubling	$\rho^2_{\mathbf{x}}$ 2-regular	Greinecker 2015 Parreau–Rigo–Rowland–Vandomme 2015
Tribonacci	$\rho^{\mathrm{ab}}_{\mathbf{x}}$ Trib.-aut.	Turek 2013, Shallit 2021
Rudin–Shapiro	$\rho^{\mathrm{ab}}_{\mathbf{x}}$ 2-regular	Lü–Chen–Wen–Wu 2017
Cantor	$\rho^k_{\mathbf{x}}$ 3-regular	Chen–Lü–Wu 2018
generalized Cantor	$\rho^2_{\mathbf{x}}$ regular	Lü–Chen–Wen 2022
Parikh-col. morphisms	$\rho^{\mathrm{ab}}_{\mathbf{x}}$ automatic	Rigo–S–Whiteland 2023

Our contributions

$S = (L, A, <)$ regular ANS, i.e. L and addition are regular languages

Contribution 1 (CDOPSS 2025+)

For $\mathbf{x}$ uniformly factor-balanced and S-automatic
its 2D generalized abelian complexity $(k, n) \mapsto \rho_{\mathbf{x}}^{k}(n)$ is S-regular

Contribution 2 (CDOPSS 2025+)

For $\mathbf{x}$ fixed point of a prolongable primitive ultimately Pisot substitution τ,
its abelian complexity $\rho_{\mathbf{x}}^{\mathrm{ab}}$ is automatic in the ANS of τ
and the DFAO computing it is effectively computable

Contribution 3 (CDOPSS 2025+)

For $\mathbf{x}$ fixed point of a primitive substitution τ s.t. the 2-sliding-block-code substitution τ_2 is ultimately Pisot
its k-abelian complexity $\rho_{\mathbf{x}}^{k}$ is automatic in the ANS of τ
and the DFAO computing it is effectively computable

Contribution 1

Theorem (CDOPSS 2025+)

For $\mathbf{x}$ uniformly factor-balanced and S-automatic
its 2D generalized abelian complexity $(k, n) \mapsto \rho_{\mathbf{x}}^{k}(n)$ is S-regular

$\mathbf{x}$ uniformly factor-balanced if $\exists C$ s.t. $\mathbf{x}$ (k, C)-balanced $\forall k$

A family with this property:

Proposition (Fagnot–Vuillon 2002 + Vandeth 2000)

A Sturmian sequence $\mathbf{x}$ with slope $\alpha \in (0, 1)$ is uniformly factor-balanced if $\alpha/(1-\alpha)$ has bounded partial quotients

Example: Fibonacci sequence $\mathbf{f} = 0100101001\cdots$ $\quad \alpha = \frac{3-\sqrt{5}}{2} \quad \alpha/(1-\alpha) = \frac{\sqrt{5}-1}{2} = [0, \bar{1}]$

Uniform bounds: 3 (Fagnot–Vuillon 2002 + Vandeth 2000)
$\rightsquigarrow$ 2 (CDOPSS 2025+)

Implementation and experiments

Tools:

- licofage toolkit (deals with ANS and f.p.) `https://pypi.org/project/licofage/`
- `Walnut` (deals with first-order logic) `https://github.com/Walnut-Theorem-Prover`
- Awali C++ library (deals with weighted automata) `http://vaucanson-project.org/Awali/`

Pros: direct, naive, effective (produces a lin. rep.), $\forall k$

Cons: strong assumption, computer-intensive

Applications to:

- Fibonacci sequence (f.p. of $\phi\colon 0 \mapsto 01, 1 \mapsto 0$)
- Pell sequence (f.p. of $0 \mapsto 001, 1 \mapsto 0$)
- uniform fixed point (f.p. of $0 \mapsto 001, 1 \mapsto 010$)
- Tribonacci sequence (f.p. of $0 \mapsto 01, 1 \mapsto 02, 2 \mapsto 0$)

Repo: `https://github.com/nopid/abcomp/`

Tribonacci sequence t (f.p. of $0 \mapsto 01, 1 \mapsto 02, 2 \mapsto 0$)

Step 1: substitution, ANS, and fixed point

```
%%python
from licofage.kit import *
import os
setparams (True, True, os.environ ["WALNUT_HOME"] )

s = subst ('01/02/0')
ns = address(s, "tri")
ns. gen_ns ( )
ns. gen_word_automaton ()
```

Step 2: factor comparison in `Walnut`

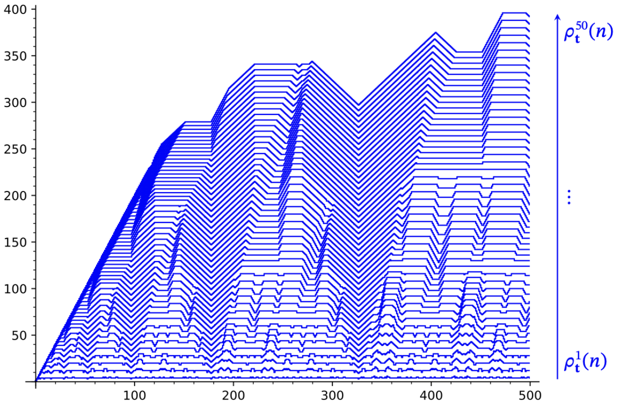
```
def cut "?msd_tri iu & jev & utj=v+i & unti & ven+j":
def feq_tri "?msd_tri ~(Eu,v $cut(i,j,n,u,v) & Trilu]!=Trilv])":
```

Step 3: obtain a linear representation for $\rho_{\mathbf{t}}^{k}$

```
def occ_tri "?msd_tri j1<=u & u<=j1+n & $feq_tri(i,u,k) & j2=j2":
def sametri "?msd_tri Dequitri[i][j1][j2][k][n] = @0":
def abeqextri "?msd_tri Ai $sametri(i,j1,j2,k,n)":
def abeqtri "?msd_tri (n<k & $feq_tri(i,j,n))
  | (n>=k & $feq_tri(i,j,k-1) & $abeqextri(i,j,k,n-k))":
def abfirsttri "?msd_tri k>0 & ~Ej j<i & $abeqtri(i,j,k,n)":
```

Output: linear representation of dimension 264 with entries in $\mathbb{Z}$

`https://github.com/nopid/abcomp/blob/main/section3/out/matri.sage`



Corollary (CDOPSS 2025+)

The Tribonacci sequence is uniformly factor balanced with constant $C = 2$

Drawback: 16h+ computation time (96 threads + 256 GB RAM)

Current work: a 5-minute blackboard proof

Contribution 2

Contribution 2 (CDOPSS 2025+)

For $\mathbf{x}$ fixed point of a prolongable primitive ultimately Pisot substitution τ, its abelian complexity $\rho^{\mathrm{ab}}_{\mathbf{x}}$ is automatic in the ANS of τ and the DFAO computing it is effectively computable

- substitution $\tau\colon A \to A^*$ and $\mathbf{x} = \tau^\omega(a)$
- τ prolongable if $\exists a \in A$ s.t. $\tau(a) = au$ and $\lim_{n\to+\infty} |\tau^n(a)| = +\infty$
- incidence matrix of τ: $(M_\tau)_{a,b} = |\tau(a)|_b \ \forall a, b \in A$
- τ primitive if M_τ primitive
- τ Pisot if the characteristic polynomial of M_τ is the minimal polynomial P_θ of θ Pisot number
- τ ultimately Pisot if the characteristic polynomial of M_τ is $X^m \cdot P_\theta$ for θ Pisot number

<u>Example</u>:

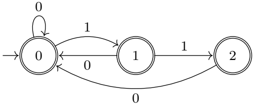
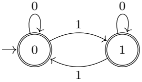
τ	M_τ	char. pol.	θ
Trib. $0 \mapsto 01, 1 \mapsto 02, 2 \mapsto 0$	$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$	$X^3 - X^2 - X - 1$	≈ 1.83929
T.-M. $0 \mapsto 01, 1 \mapsto 10$	$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$	$X(X-2)$	2

Contribution 2 (CDOPSS 2025+)

For $\mathbf{x}$ fixed point of a prolongable primitive ultimately Pisot substitution τ, its abelian complexity $\rho^{\mathrm{ab}}_{\mathbf{x}}$ is automatic in the ANS of τ and the DFAO computing it is effectively computable

- τ prolongable on $a \in A$
- $\Sigma_\tau = \{0, 1, \ldots, \max_b |\tau(b)| - 1\}$
- adressing automaton $\mathcal{A}_\tau$
  - states A; a initial; A final
  - alphabet Σ_τ
  - transition function $\delta(b, i) = i$th letter of $\tau(b)$
- L_τ = language accepted by $\mathcal{A}_\tau$ without leading 0's
- ANS of τ: $S_\tau = (L_\tau, \Sigma_\tau, <)$ (Dumont–Thomas NS)
- $\mathbf{x}$ is S_τ-automatic

Example:

τ	$\mathcal{A}_\tau$	L_τ
Trib. $\tau: 0 \mapsto 01, 1 \mapsto 02, 2 \mapsto 0$		binary forbidding 111
T.-M. $0 \mapsto 01, 1 \mapsto 10$		$1\{0,1\}^* \cup \{\varepsilon\}$

Contributions 3

Stronger assumption:

Contribution 3 (CDOPSS 2025+)

For $\mathbf{x}$ fixed point of a primitive substitution τ s.t. the 2-sliding-block-code substitution τ_2 is ultimately Pisot
its k-abelian complexity $\rho^k_{\mathbf{x}}$ is S_τ-automatic
and the DFAO computing it is effectively computable

2-sliding-block code

Example: Tribonacci sequence $\mathbf{t}$ $\quad \rho_{\mathbf{t}}(n) = 2n + 1$ $\quad$ (episturmian sequence)

Blocks $0\,1$, $0\,2$, $0\,1$, $0\,0$, $1\,0$, $2\,0$ (above): $1 \quad 3 \quad 1 \quad 5 \quad 2 \quad 4$

$$\mathbf{t} = 0 \quad 1 \quad 0 \quad 2 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 2 \quad 0 \quad \cdots$$

Blocks $1\,0$, $2\,0$, $1\,0$, $0\,1$, $0\,2$ (below): $2 \quad 4 \quad 2 \quad 1 \quad 3$

$$B_2(\mathbf{t}) = 1 \quad 2 \quad 3 \quad 4 \quad 1 \quad 2 \quad 5 \quad 1 \quad 2 \quad 3 \quad 4 \quad \cdots$$

Lemma (Quéffelec 1987)

$\mathbf{x}$ fixed point of $\tau \Rightarrow B_2(\mathbf{x})$ fixed point of τ_2 (constructive proof)

Example (continued): $\tau_2 : 1, 3, 5 \to 12; 2 \to 34; 4 \to 5$

Implementation and experiments

Tools:

- licofage toolkit (deal with ANS and f.p.) `https://pypi.org/project/licofage/`
- `Walnut` (deal with first-order logic) `https://github.com/Walnut-Theorem-Prover`

Pros: effective (produces a lin. rep.)

Cons: fixed k, relies on a blend of results (Quéffelec 1987, Adamczewski 2003, Shallit 2021, Carton–Couvreur–Delacourt–Ollinger 2025)

Applications to:

- Parikh-collinear substitutions (e.g. f.p. of $0 \mapsto 010011, 1 \mapsto 1001$)
- f.p. of $0 \mapsto 011, 1 \mapsto 01$ with Pisot $\theta = 1 + \sqrt{2}$
- f.p. of $0 \mapsto 0001011, 1 \mapsto 001011$ with Pisot $\theta = \frac{7+\sqrt{37}}{2}$
- f.p. of $0 \mapsto 001, 1 \mapsto 02, 2 \mapsto 002$ with Pisot root of $X^3 - 3X^2 + X - 1$
- f.p. of $0 \mapsto 010, 1 \mapsto 2, 2 \mapsto 02$ with Pisot root of $X^3 - 3X^2 + 2X - 1$
- twisted Tribonacci sequence, f.p. of $0 \mapsto 01, 1 \mapsto 20, 2 \mapsto 0$ with Pisot root of $X^3 - X^2 - X - 1$ [Sloane, A277735]
- Narayana sequence, f.p. of $0 \mapsto 01, 1 \mapsto 2, 2 \mapsto 0$ with Pisot root $\theta \approx 1.46557$ of $X^3 - X^2 - 1$ [Sloane, A105083]

Repo: `https://github.com/nopid/abcomp/`

Narayana sequence n (f.p. of $0 \mapsto 01, 1 \mapsto 2, 2 \mapsto 0$)

Step 1: substitution, ANS, and fixed point

```
%%python
from licofage.kit import *
import os
setparams(True, True, os.environ["WALNUT_HOME"])

s = subst('01/2/0')
ns = address(s, "nara")
ns.gen_ns()
ns.gen_word_automaton()
```

Step 2: factor comparison and special border condition (Fici–Puzynina 2023) in `Walnut`

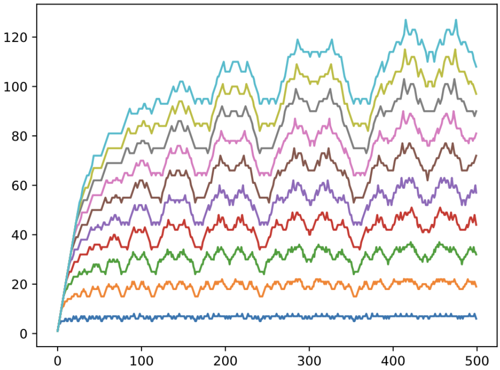
```
def cut "?msd_nara i<=u & j<=v & u+j=v+i & u<n+i & v<n+j":
def feq_nara "?msd_nara ~(Eu,v $cut(i,j,n,u,v) & Nara[u]!=Nara[v])":
eval comp_nara n "?msd_nara Aj $feq_nara(i,j,n) => i<=j":
def bordercond "?msd_nara (k<=n => $feq_nara(i,j,k-1))
 & (n<k => $feq_nara(i,j,n))":
```

Step 3: substitution for the length-3 sliding-block of **n**, ANS, and fixed point

```
%%python
s3 = block(s, 3)
ns3 = address(s3, "narab3")
ns3.gen_ns()
(ns-ns3).gen_dfa("conv_nara_narab3")

def bordercond3 "?msd_narab3 (?msd_nara Eii,jj,kk,nn
 ($conv_nara_narab3(?msd_nara ii, ?msd_narab3 i) &
  $conv_nara_narab3(?msd_nara jj, ?msd_narab3 j) &
  $conv_nara_narab3(?msd_nara kk, ?msd_narab3 k) &
  $conv_nara_narab3(?msd_nara nn, ?msd_narab3 n) &
  $bordercond(ii,jj,kk,nn)))":
```

Step 4: obtain a DFAO for $\rho^3_{\mathbf{n}}$

```
%%python
for (m,a) in enumerate(ns3.alpha):
    w = {'_': 0}
    w[a] = 1
    parikh = address(s3, ns3.ns, **w)
    (parikh - ns3).gen_dfa(f"narab3p{m}")

def fac{m} "?msd_narab3 Ex,y $narab3p{m}(i,x) & $narab3p{m}(i+n,y)
 & z+x=y":
def min{m} "?msd_narab3 Ei $fac{m}(i,n,x) & Aj,y $fac{m}(j,n,y)
 => y>=x":
def diff{m} "?msd_narab3 Ex,y $min{m}(n,x) & $fac{m}(i,n,y) & z+x=y":

def abeq_narab3 "?msd_narab3 $bordercond3(i,j,3,n+2)
 & (Ez $diff0(i,n,z) &  $diff0(j,n,z))
 & (Ez $diff1(i,n,z) &  $diff1(j,n,z))
 & (Ez $diff2(i,n,z) &  $diff2(j,n,z))
 & (Ez $diff3(i,n,z) &  $diff3(j,n,z))
 & (Ez $diff4(i,n,z) &  $diff4(j,n,z))
 & (Ez $diff5(i,n,z) &  $diff5(j,n,z))
 & (Ez $diff6(i,n,z) &  $diff6(j,n,z))":

def abeq_nara3 "?msd_nara (n<2 & $feq_nara(i,j,n))
 | (n>=2 & (?msd_narab3 Ei,j,n
  ($conv_nara_narab3(?msd_nara ii, ?msd_narab3 i)
   & $conv_nara_narab3(?msd_nara jj, ?msd_narab3 j)
   & $conv_nara_narab3(?msd_nara nn, ?msd_narab3 n)
   & $abeq_narab3(ii,jj,nn-2))))":

eval comp_nara3 n "?msd_nara Aj $abeq_nara3(i,j,n) => i<=j":

%SGT comp_nara3 msd_nara Comp_nara3
```



k	DFAO size for $\rho_{\mathbf{n}}^k$
1	97
2	277
3	467
4	634
5	871
6	969
7	1218
8	1309
9	1646
10	1745

Remark: Why this sequence?

Recent combinatorial studies by Shallit (2025+), Letouzey (2025+)

Conclusion

Strategy (CDOPSS 2025+)

How to effectively compute $\rho_{\mathbf{x}}^k$?

- try the 1st approach
- if it ends, $\rho_{\mathbf{x}}^k$ $\forall k$
- if computation does not converge in reasonable time
  ($\mathbf{x}$ not uniformly-factor-balanced or computation too heavy),
  try the 2nd approach

Some related open problems

Problem 1 (CDOPSS 2025+)

Find a black-board proof of the 2-uniform factor balancedness of Fibonacci and Tribonacci.
True for Pisot substitutions in general? Under which conditions?

Problem 2 (CDOPSS 2025+)

Smallest $C_k^{(m)} \geq 1$ s.t. the m-bonacci word $\mathbf{x}_m$ is $(k, C_k^{(m)})$-balanced?
Bounds in Břinda–Pelantová–Turek 2014