Probabilistic studies in Number Theory and Word Combinatorics: instances of dynamical analysis

Pablo Rotondo
IRIF, Paris 7 Diderot,
Universidad de la República, Uruguay
GREYC, associate

Joint works with
Valérie Berthé, Eda Cesaratto, Brigitte Vallée, Alfredo Viola

Some key words

Probabilistic studies

Recurrence Function of Sturmian Words

Continued Logarithm Algorithm

Continued Fractions Dynamical Systems

Dynamical Analysis
This talk

1. General Introduction: continued fractions and dynamical systems
   - Continued Fractions
   - Euclidean dynamical system

2. The recurrence function of a random Sturmian word
   - Sturmian words and recurrence
   - Our models and results

3. The Continued Logarithm
   - Origins and algorithm
   - The CL dynamical system
   - Extended system and results
   - Conclusions and extensions
Section

1. General Introduction: continued fractions and dynamical systems
   - Continued Fractions
   - Euclidean dynamical system

2. The recurrence function of a random Sturmian word
   - Sturmian words and recurrence
   - Our models and results

3. The Continued Logarithm
   - Origins and algorithm
   - The CL dynamical system
   - Extended system and results
   - Conclusions and extensions
Continued Fractions

Every irrational number $\alpha \in (0, 1)$ has a unique representation

$$\alpha = \frac{1}{m_1 + \frac{1}{m_2 + \cdots}}$$

where $m_1, m_2, \ldots \geq 1$ are integers called the digits or quotients.
Continued Fractions

Every irrational number $\alpha \in (0, 1)$ has a unique representation

$$\alpha = \frac{1}{m_1 + \frac{1}{m_2 + \cdots}}$$

where $m_1, m_2, \ldots \geq 1$ are integers called the digits or quotients.

Truncating the expansion at depth $k$ we get a convergent

$$\frac{p_k(\alpha)}{q_k(\alpha)} = \frac{1}{m_1 + \frac{1}{m_2 + \cdots \frac{1}{m_k}}}.$$  

The denominators $q_k(\alpha)$ are called the continuants of $\alpha$. 
Euclidean Algorithm and Continued Fractions

Property

Given integers $x$ and $y$ with $x \geq y \geq 0$

$$\gcd(x, y) = \gcd(y, x \text{ mod } y).$$

In conjunction with $\gcd(x, 0) = x$, we get the Euclidean Algorithm.
Euclidean Algorithm and Continued Fractions

Property
Given integers $x$ and $y$ with $x \geq y \geq 0$

$$\text{gcd}(x, y) = \text{gcd}(y, x \mod y).$$

In conjunction with $\text{gcd}(x, 0) = x$, we get the Euclidean Algorithm.

This algorithm is equivalent to the continued fraction expansion:

- given the integer division $y = mx + r$,

  $$\frac{x}{y} = \frac{1}{m + \frac{r}{x}},$$

  and the process continues with $\frac{r}{x}$. 
Euclidean Algorithm and Continued Fractions

Property

Given integers $x$ and $y$ with $x \geq y \geq 0$

$$\gcd(x, y) = \gcd(y, x \mod y).$$

In conjunction with $\gcd(x, 0) = x$, we get the Euclidean Algorithm.

This algorithm is equivalent to the continued fraction expansion:

- given the integer division $y = mx + r$,

  $$\frac{x}{y} = \frac{1}{m + \frac{r}{x}},$$

  and the process continues with $\frac{r}{x}$. 
Euclidean dynamical system

To get the digits of the continued fraction expansion observe

\[ \alpha = \cfrac{1}{m_1 + \cfrac{1}{m_2 + \ddots}}. \]

\[ \implies m_1 = \left\lfloor \frac{1}{\alpha} \right\rfloor, \quad \cfrac{1}{m_2 + \cfrac{1}{m_3 + \ddots}} = \left\{ \frac{1}{\alpha} \right\}. \]

The map

\[ T: (0, 1) \to (0, 1), \quad x \mapsto \left\{ \frac{1}{x} \right\}, \]

is known as the Gauss map.
Euclidean dynamical system

To get the digits of the continued fraction expansion observe

$$\alpha = \frac{1}{m_1 + \frac{1}{m_2 + \cdots}}.$$ 

$$\Rightarrow m_1 = \left\lfloor \frac{1}{\alpha} \right\rfloor,$$

$$\frac{1}{m_2 + \frac{1}{m_3 + \cdots}} = \left\{ \frac{1}{\alpha} \right\}.$$ 

The map

$$T: (0, 1) \rightarrow (0, 1), \quad x \mapsto \left\{ \frac{1}{x} \right\},$$

is known as the Gauss map.
**Gauss map**

**Branches**

\[ T_m(x) := \frac{1}{x} - m, \quad x \in \left( \frac{1}{m+1}, \frac{1}{m+0} \right) . \]

**Inverse branches**

\[ h_m(x) := \frac{1}{m + x}, \quad \mathcal{H} := \{ h_m : m \in \mathbb{N} \} , \]

and at depth \( k \)

\[ \mathcal{H}^k := \{ h_{m_1} \circ \cdots \circ h_{m_k} : m_1, \ldots, m_k \in \mathbb{N} \} . \]
Branches

\[ T_m(x) := \frac{1}{x} - m, \quad x \in \left( \frac{1}{m+1}, \frac{1}{m+0} \right). \]

Inverse branches

\[ h_m(x) := \frac{1}{m + x}, \quad \mathcal{H} := \{h_m : m \in \mathbb{N}\}, \]

and at depth \( k \)

\[ \mathcal{H}^k := \{h_{m_1} \circ \cdots \circ h_{m_k} : m_1, \ldots, m_k \in \mathbb{N}\}. \]

**Property.** Let \( h := h_{m_1} \circ \cdots \circ h_{m_k} \), if \( h(0) = \frac{p}{q} \) then \( |h'(0)| = \frac{1}{q^2} \).
Density transformer

**Question:** If \( g \in C^0(\mathcal{I}) \) were the density of \( x \mapsto \) density of \( T(x) \)?
Density transformer

Question: If $g \in C^0(\mathcal{I})$ were the density of $x \mapsto \text{density of } T(x)$?

\[ T(x) \]

\[ \text{1} \]

\[ \text{dy} \]

\[ \cdots \mid dh_3(y) \mid dh_2(y) \mid dh_1(y) \mid dh_0(y) \mid \]

\[ x \]

\[ 1 \]
Density transformer

**Question:** If \( g \in C^0(\mathcal{I}) \) were the density of \( x \mapsto \) density of \( T(x) \)?

**Answer:** The density is

\[
H[g](x) = \sum_{h \in \mathcal{H}} |h'(x)| \ g(h(x))
\]

\[
= \sum_{m \geq 0} \frac{1}{(m + x)^2} \ g \left( \frac{1}{m + x} \right).
\]
Density transformer

Question: If \( g \in C^0(\mathcal{I}) \) were the density of \( x \mapsto \) density of \( T(x) \)?

Answer: The density is

\[
H[g](x) = \sum_{h \in \mathcal{H}} |h'(x)| g(h(x))
\]

\[
= \sum_{m \geq 0} \frac{1}{(m + x)^2} g \left( \frac{1}{m + x} \right).
\]

In general \( T^k(x) \) has density

\[
H^k[g](x) = \sum_{h \in \mathcal{H}^k} |h'(x)| g(h(x)).
\]
Density transformer

Question: If $g \in C^0(I)$ were the density of $x \mapsto$ density of $T(x)$?

Answer: The density is

$$H[g](x) = \sum_{h \in H} |h'(x)| \ g(h(x))$$

$$= \sum_{m \geq 0} \frac{1}{(m + x)^2} \ g \left( \frac{1}{m + x} \right).$$

In general $T^k(x)$ has density

$$H^k[g](x) = \sum_{h \in H^k} |h'(x)| \ g(h(x)).$$

$\Rightarrow$ Transfer operator $H_s$ extends $H$, introducing a variable $s$

$$H_s[g](x) = \sum_{h \in H} |h'(x)|^s \ g(h(x)).$$
Principles of dynamical analysis [Vallée, Flajolet, Baladi, . . .]:

- Transfer operator $H_s \Rightarrow$ expressions for generating functions.
Principles of dynamical analysis [Vallée, Flajolet, Baladi, ...]:

- Transfer operator $H_s \Rightarrow$ expressions for generating functions.

Note

- $H_s$ describes all execution of depth 1.
- $H_s^2 = H_s \circ H_s$ describes all execution of depth 2.
- 
- and $(I - H_s)^{-1} = I + H_s + H_s^2 + \ldots$ describes all executions.
Principles of dynamical analysis [Vallée, Flajolet, Baladi, ...]:

- Transfer operator $H_s \Rightarrow$ expressions for generating functions.
- Dominant eigenvalue of $H_s \Rightarrow$ dominant singularities.

**Note**

- $H_s$ describes all execution of depth 1.
- $H_s^2 = H_s \circ H_s$ describes all execution of depth 2.
- ...
- and $(I - H_s)^{-1} = I + H_s + H_s^2 + \ldots$ describes all executions.
1. General Introduction: continued fractions and dynamical systems
   - Continued Fractions
   - Euclidean dynamical system

2. The recurrence function of a random Sturmian word
   - Sturmian words and recurrence
   - Our models and results

3. The Continued Logarithm
   - Origins and algorithm
   - The CL dynamical system
   - Extended system and results
   - Conclusions and extensions
Definition of Sturmian words

Definition

Complexity function of an infinite word $u \in \mathcal{A}^\mathbb{N}$

$$p_u : \mathbb{N} \to \mathbb{N}, \quad p_u(n) = \#\{\text{factors of length } n \text{ in } u\}.$$
Definition of Sturmian words

Definition

Complexity function of an infinite word $u \in \mathcal{A}^\mathbb{N}$

$$p_u : \mathbb{N} \to \mathbb{N}, \quad p_u(n) = \# \{ \text{factors of length } n \text{ in } u \}.$$
Definition of Sturmian words

Definition
Complexity function of an infinite word $u \in \mathcal{A}^\mathbb{N}$

$$p_u : \mathbb{N} \rightarrow \mathbb{N}, \quad p_u(n) = \# \{ \text{factors of length } n \text{ in } u \}.$$ 

Important property

$u \in \mathcal{A}^\mathbb{N}$ is not eventually periodic

$$\iff p_u(n + 1) > p_u(n) \text{ for all } n \in \mathbb{N}.$$ 

Definition of Sturmian words

Definition

Complexity function of an infinite word $u \in \mathcal{A}^\mathbb{N}$

$$p_u : \mathbb{N} \rightarrow \mathbb{N}, \quad p_u(n) = \#\{\text{factors of length } n \text{ in } u\}.$$ 

Important property

$u \in \mathcal{A}^\mathbb{N}$ is not eventually periodic

$$\iff p_u(n + 1) > p_u(n) \text{ for all } n \in \mathbb{N}$$

$$\implies p_u(n) \geq n + 1.$$ 

Sturmian words are the “simplest” that are not eventually periodic.
Definition of Sturmian words

**Definition**

Complexity function of an infinite word \( u \in A^\mathbb{N} \)

\[
p_u : \mathbb{N} \to \mathbb{N}, \quad p_u(n) = \#\{\text{factors of length } n \text{ in } u\}.
\]

**Important property**

\( u \in A^\mathbb{N} \) is not eventually periodic \( \iff \) \( p_u(n + 1) > p_u(n) \) for all \( n \in \mathbb{N} \)

\( \implies p_u(n) \geq n + 1. \)

Sturmian words are the “simplest” that are not eventually periodic.

**Definition**

\( u \in \{0, 1\}^\mathbb{N} \) is Sturmian \( \iff \) \( p_u(n) = n + 1 \) for each \( n \geq 0. \)
Sturmian words and digital lines

Sturmian words correspond to discrete codings of lines, from below or above, by horizontal lines and diagonals.

\[
\begin{align*}
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0
\end{align*}
\]
Sturmian words and digital lines

*Sturmian words* correspond to discrete *codings of lines*, from below or above, by horizontal lines and diagonals.

The *slope* $\alpha$ plays a key role:

the *finite factors* are determined *exclusively* by $\alpha$. 
Recurrence of Sturmian words

Definition (Recurrence function)
Consider an infinite word $u$. Its recurrence function is:

$$R_u(n) = \inf \{ m \in \mathbb{N} : \text{every factor of length } m \text{ contains all the factors of length } n \}.$$
Recurrence of Sturmian words

**Definition (Recurrence function)**

Consider an infinite word $u$. Its recurrence function is:

$$ R_u(n) = \inf \{ m \in \mathbb{N} : \text{every factor of length } m \text{ contains all the factors of length } n \}.$$

- **Cost** we have to pay to discover the factors if we start from an arbitrary point in $u = u_1 u_2 \ldots$
Recurrence of Sturmian words

Definition (Recurrence function)
Consider an infinite word $u$. Its recurrence function is:

$$ R_u(n) = \inf \{ m \in \mathbb{N} : \text{every factor of length } m \text{ contains all the factors of length } n \}.$$

- **Cost** we have to pay to discover the factors if we start from an arbitrary point in $u = u_1 u_2 \ldots$

Theorem (Morse, Hedlund, 1940)
The recurrence function is piecewise affine and satisfies

$$ R_\alpha(n) = n - 1 + q_{k-1}(\alpha) + q_k(\alpha), \quad \text{for } q_{k-1}(\alpha) \leq n < q_k(\alpha). $$
Recurrence quotient and its parameters

\[ S(\alpha, n) := \frac{R_\alpha(n) + 1}{n} = 1 + \frac{q_{k-1}(\alpha) + q_k(\alpha)}{n}, \quad q_{k-1}(\alpha) \leq n < q_k(\alpha). \]
Recurrence quotient and its parameters

\[ S(\alpha, n) := \frac{R_\alpha(n) + 1}{n} = 1 + \frac{q_{k-1}(\alpha) + q_k(\alpha)}{n}, \quad q_{k-1}(\alpha) \leq n < q_k(\alpha). \]

Recurrence quotient \( \alpha = e^{-1} \).
Recurrence quotient and its parameters

\[ S(\alpha, n) := \frac{R_\alpha(n) + 1}{n} = 1 + \frac{q_{k-1}(\alpha) + q_k(\alpha)}{n}, \quad q_{k-1}(\alpha) \leq n < q_k(\alpha). \]
Studies of the recurrence function

Classical results concern the worst case scenarios for fixed $\alpha$: let $\epsilon > 0$, for almost every $\alpha$

$$\limsup_{n \to \infty} \frac{S(\alpha, n)}{\log n} = \infty, \quad \lim_{n \to \infty} \frac{S(\alpha, n)}{(\log n)^{1+\epsilon}} = 0.$$  

(Morse&Hedlund ’40)
Studies of the recurrence function

Classical results concern the worst case scenarios for fixed $\alpha$: let $\epsilon > 0$, for almost every $\alpha$

$$\limsup_{n \to \infty} \frac{S(\alpha, n)}{\log n} = \infty, \quad \lim_{n \to \infty} \frac{S(\alpha, n)}{(\log n)^{1+\epsilon}} = 0.$$

(Morse & Hedlund ’40)

We define two probabilistic models

in both cases $\alpha$ is drawn uniformly at random

1) fix the length $n \Rightarrow$ random variables $S_n(\alpha) := S(\alpha, n)$.

   in distribution and expectation as $n \to \infty$.

2) fix index $k$ of interval $[q_{k-1}(\alpha), q_k(\alpha))$ and

   the relative position $\mu \Rightarrow$ sequence $(n_k(\alpha))_k$.

   $\Rightarrow$ random variables $S_k(\alpha) := S(\alpha, n_k(\alpha))$ as $k \to \infty$. 

\[ n_k \mu = \frac{1}{3} q_k - 1 \]
Studies of the recurrence function

Classical results concern the worst case scenarios for fixed $\alpha$: let $\epsilon > 0$, for almost every $\alpha$

$$\limsup_{n \to \infty} \frac{S(\alpha, n)}{\log n} = \infty, \quad \lim_{n \to \infty} \frac{S(\alpha, n)}{(\log n)^{1+\epsilon}} = 0.$$  

(Morse&Hedlund ’40)

We define two probabilistic models

in both cases $\alpha$ is drawn uniformly at random

1) fix the length $n \Rightarrow$ random variables $S_n(\alpha) := S(\alpha, n)$.
   in distribution and expectation as $n \to \infty$.
2) fix index $k$ of interval $[q_{k-1}(\alpha), q_k(\alpha))$ and
   the relative position $\mu \Rightarrow$ sequence $(n_k(\alpha))_k$.
   $\Rightarrow$ random variables $S_k(\alpha) := S(\alpha, n_k(\alpha))$ as $k \to \infty$.

Figure : Sequence of indices $(n_k(\alpha))_k$ for $\mu = 1/3$. 

$q_{k-1}$ \hspace{1cm} $\mu = 1/3$ \hspace{1cm} $q_k$ 

$n_k$
Results

Model (1): fixed $n$. [RV17]

- Limit distribution for
  $\mu_n, \rho_n, S_n$.
- Convergence of histograms to limit density.
- Conditional expectations
  $\mathbb{E}[S_n | \mu_n \geq \epsilon(n)] \sim |\log \epsilon(n)|$.

Figure: Limit density of $S_n$. 
Results

Model (1): fixed $n$. [RV17]
- Limit distribution for $\mu_n, \rho_n, S_n$.
- Convergence of histograms to limit density.
- Conditional expectations $\mathbb{E}[S_n|\mu_n \geq \epsilon(n)] \sim |\log \epsilon(n)|$.

Model (2): fixed $\mu$. [BCRVV15]
- Limit distribution/expectation of $S_k(\alpha) := S(\alpha, n_k)$ depending on $\mu$.
- Study of $\mathbb{E}[S_k(\alpha)]$ as $\mu := \mu_k \to 0$.

Figure: Limit density of $S_n$.

Figure: Limit of $\mathbb{E}[S_k] \text{ versus } \mu$. 
Elements of the proofs

Model (1): fixed $n$

- Distribution at $\lambda$ given by coprime Riemann sum of step $\frac{1}{n}$ of

$$\omega(x, y) = \frac{2}{y(x + y)},$$

over

$$\Delta_\lambda = \{(x, y) : 0 < x \leq 1 < y, 1 + x + y \leq \lambda\}.$$
Elements of the proofs

Model (1): fixed \( n \)

- Distribution at \( \lambda \) given by *coprime* Riemann sum of step \( \frac{1}{n} \) of

\[
\omega(x, y) = \frac{2}{y(x + y)},
\]

over

\[
\Delta_\lambda = \{(x, y) : 0 < x \leq 1 < y, 1 + x + y \leq \lambda\}.
\]

Model (2): fixed \( \mu \)

- Expected values given by Perron-Frobenius operator:

\[
\mathbb{E}[S_k] \sim 1 + \mathbf{H}^k[g](0),
\]

where

\[
g(\rho) = \left(\frac{1 + \rho}{\mu + \rho - \mu \rho}\right) / (1 + \rho).
\]
1. General Introduction: continued fractions and dynamical systems
   - Continued Fractions
   - Euclidean dynamical system

2. The recurrence function of a random Sturmian word
   - Sturmian words and recurrence
   - Our models and results

3. The Continued Logarithm
   - Origins and algorithm
   - The CL dynamical system
   - Extended system and results
   - Conclusions and extensions
The origins

Introduced by Gosper as a mutation of continued fractions:

- gives rise to a \texttt{gcd} algorithm akin to Euclid’s.
- quotients are powers of two:
  - small information parcel.
  - employs only shifts and subtractions.
- appears to be simple and efficient.
The origins

Introduced by Gosper as a mutation of continued fractions:

- gives rise to a gcd algorithm akin to Euclid’s.
- quotients are powers of two:
  - small information parcel.
  - employs only shifts and substractions.
- appears to be simple and efficient.

More recently:
- Shallit studied its worst-case performance in 2016.
- We consider its average performance!
Continued Logarithm Algorithm

A sequence of binary “divisions” beginning from \((p, q)\):

\[
q = 2^a p + r, \quad 0 \leq r < 2^a p.
\]
Continued Logarithm Algorithm

A sequence of binary “divisions” beginning from \((p, q)\):

\[ q = 2^a p + r, \quad 0 \leq r < 2^a p. \]

Note. \( a = \max\{k \geq 0 : 2^k p \leq q\} \)

Example. Let us find \(\gcd(13, 31)\).

\[ \begin{array}{c|c|c}
\hline
p & q & r \\
\hline
1 & 13 & 5 \\
2 & 26 & 6 \\
3 & 32 & 8 \\
4 & 0 & 8 \\
\hline
\end{array} \]

\(\text{Ended with (0, 8)}, \) what is the \(\gcd\)?

\(\Rightarrow\) odd \(\gcd\) × parasitic powers of 2.
Continued Logarithm Algorithm

A sequence of binary “divisions” beginning from \((p, q)\):

\[
q = 2^a p + r, \quad 0 \leq r < 2^a p.
\]

**Note.** \(a = \max\{k \geq 0 : 2^k p \leq q\}\)

Continue with the new pair

\[
(p, q) \mapsto (p', q') = (r, 2^a p),
\]

until the remainder \(r\) equals 0.

Example. Let us find \(\gcd(13, 31)\).

\[
\begin{array}{c|c|c}
     & 13 & 31 \\
\hline
   2 & 6 & 26 \\
   2 & 8 & 20 \\
   2 & 4 & 12 \\
   2 & 0 & 8 \\
\end{array}
\]

 Ended with \((0, 8)\), what is the \(\gcd\)?

\[\Rightarrow \text{odd} \times \text{parasitic powers of } 2.\]
Continued Logarithm Algorithm

A sequence of binary “divisions” beginning from \((p, q)\):

\[ q = 2^a p + r, \quad 0 \leq r < 2^a p. \]

**Note.** \(a = \max\{k \geq 0 : 2^k p \leq q\}\)

Continue with the new pair

\[(p, q) \mapsto (p', q') = (r, 2^a p),\]

until the remainder \(r\) equals 0.

**Example.** Let us find \(\gcd(13, 31)\).

<table>
<thead>
<tr>
<th>(a)</th>
<th>(p)</th>
<th>(q)</th>
<th>(r)</th>
<th>(2^a p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>31</td>
<td>5</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>26</td>
<td>6</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>20</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>12</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>8</td>
</tr>
</tbody>
</table>
Continued Logarithm Algorithm

A sequence of binary “divisions” beginning from \((p, q)\):

\[ q = 2^a p + r, \quad 0 \leq r < 2^a p. \]

**Note.** \(a = \max\{k \geq 0 : 2^k p \leq q\}\)

Continue with the new pair

\[(p, q) \mapsto (p', q') = (r, 2^a p),\]

until the remainder \(r\) equals 0.

**Example.** Let us find \(\gcd(13, 31)\).

\[
\begin{array}{c|cc|cc}
 a & p & q & r & 2^a p \\
\hline
 1 & 13 & 31 & 5 & 26 \\
 2 & 5 & 26 & 6 & 20 \\
 1 & 6 & 20 & 8 & 12 \\
 0 & 8 & 12 & 4 & 8 \\
 1 & 4 & 8 & 0 & 8 \\
\end{array}
\]

\(\text{Ended with (0, 8), what is the gcd?}\)

\(\Rightarrow\) odd gcd \(\times\) parasitic powers of 2.
Consider
\[ \Omega_N = \{(p, q) \in \mathbb{N} \times \mathbb{N} : p \leq q \leq N\} . \]

Worst-case studied by Shallit (2016): \(2 \log_2 N + O(1)\) steps.
Consider
\[ \Omega_N = \{(p, q) \in \mathbb{N} \times \mathbb{N} : p \leq q \leq N\}. \]

Worst-case studied by Shallit (2016): \(2 \log_2 N + O(1)\) steps.

- Family \((p, q) = (1, 2^n - 1)\) gives the bound asymptotically.
Consider

$$\Omega_N = \{(p, q) \in \mathbb{N} \times \mathbb{N} : p \leq q \leq N\}.$$

Worst-case studied by Shallit (2016): $2 \log_2 N + O(1)$ steps.

○ Family $(p, q) = (1, 2^n - 1)$ gives the bound asymptotically.

We studied the **average** number of steps over $\Omega_N$, posed by Shallit.
Consider
\[\Omega_N = \{(p, q) \in \mathbb{N} \times \mathbb{N} : p \leq q \leq N\} .\]

Worst-case studied by Shallit (2016): \(2 \log_2 N + O(1)\) steps.

- Family \((p, q) = (1, 2^n - 1)\) gives the bound asymptotically.

We studied the average number of steps over \(\Omega_N\), posed by Shallit.

Main result (R., Vallée, '18).

Mean number of steps \(E_N[K]\) and shifts \(E_N[S]\) are \(\Theta(\log N)\).

More precisely
\[E_N[K] \sim k \log N, \quad E_N[S] \sim \frac{\log 3 - \log 2}{2 \log 2 - \log 3} \cdot E_N[K] \]

for an explicit constant \(k \approx 1.49283\ldots\) given by

\[k = \frac{2}{H}, \quad H = \frac{1}{\log(4/3)} \left( \frac{\pi^2}{6} + 2 \sum_{j} \frac{(-1)^j}{2^j j^2} - (\log 2) \frac{\log 27}{\log 16} \right)\]
Process depends **only** on $p/q$ rather than $(p, q)$.

- Map $p/q \mapsto p'/q'$ can be extended to $\mathcal{I} = (0, 1)$

  $$T: \mathcal{I} \rightarrow \mathcal{I}, \ T(x) = \frac{1}{2^a x} - 1,$$

  where $a = \lfloor \log_2(1/x) \rfloor$.

- Iteration gives a special continued fraction

  $$\frac{p}{q} = \frac{1}{2^a \left( 1 + \frac{p'}{q'} \right)}.$$
Process depends **only** on $p/q$ rather than $(p, q)$.

- Map $p/q \mapsto p'/q'$ can be extended to $I = (0, 1)$

  \[
  T: \mathcal{I} \rightarrow \mathcal{I}, \quad T(x) = \frac{1}{2^a x} - 1,
  \]

  where $a = \lfloor \log_2(1/x) \rfloor$.

- Iteration gives a special continued fraction

  \[
  \frac{p}{q} = \frac{1}{2^a \left(1 + \frac{p'}{q'}\right)}.
  \]

- For Euclid’s algorithm, we get the Gauss map

  \[
  T: \mathcal{I} \rightarrow \mathcal{I}, \quad T(x) = \frac{1}{x} - m,
  \]

  where $m = \lfloor 1/x \rfloor$.

- Iteration gives classical continued fractions

  \[
  \frac{p}{q} = \frac{1}{m + \frac{p'}{q'}}.
  \]
Process depends **only** on \( p/q \) rather than \((p, q)\).

- Map \( p/q \mapsto p'/q' \) can be extended to \( \mathcal{I} = (0, 1) \)

\[
T: \mathcal{I} \rightarrow \mathcal{I}, \quad T(x) = \frac{1}{2^a x} - 1,
\]

where \( a = \lfloor \log_2(1/x) \rfloor \).

- Iteration gives a special continued fraction

\[
\frac{p}{q} = \frac{1}{2^a \left(1 + \frac{p'}{q'}\right)}.
\]

- For Euclid’s algorithm, we get the Gauss map

\[
T: \mathcal{I} \rightarrow \mathcal{I}, \quad T(x) = \frac{1}{x} - m,
\]

where \( m = \lfloor 1/x \rfloor \).

- Iteration gives classical continued fractions

\[
\frac{p}{q} = \frac{1}{m + \frac{p'}{q'}}.
\]

The continued fraction expansion ends (is finite) when we get 0.
Dynamical system \((\mathcal{I}, T)\)

The map \(T : \mathcal{I} \to \mathcal{I}\)

**Branches**

For \(x \in \mathcal{I}_a := [2^{-a-1}, 2^{-a}]\)

\[
x \mapsto T_a(x) := \frac{2^{-a}}{x} - 1.
\]

where \(a(x) := \lceil \log_2(1/x) \rceil\).

**Inverse branches**

\[
h_a(x) := \frac{2^{-a}}{1 + x}, \quad \mathcal{H} := \{h_a : a \in \mathbb{N}\},
\]

and at depth \(k\)

\[
\mathcal{H}^k := \{h_{a_1} \circ \cdots \circ h_{a_k} : a_1, \ldots, a_k \in \mathbb{N}\}.
\]
Dynamical system \((\mathcal{I}, T)\)

The map \(T : \mathcal{I} \rightarrow \mathcal{I}\) 

The map for Euclid’s algorithm.
Reduced denominators and inverse branches

Observe for \( f(x) = \frac{ax+b}{cx+d} \):

\[
f'(x) = \frac{\det f}{(cx+d)^2}.
\]

Euclidean algorithm:

- **Homographies**
  
  \[
h_m(x) = \frac{1}{m+x},
  \]
  
  with \( \det h_m = -1 \).

- **For** \( h = h_{m_1} \circ \cdots \circ h_{m_k} \)
  
  \[
h(0) = \frac{p}{q} \Rightarrow |h'(0)| = \frac{1}{q^2},
  \]
  
  \( p/q \) reduced.

Problem: Denominator retrieved is engorged by powers of two.
Reduced denominators and inverse branches

Observe for \( f(x) = \frac{ax+b}{cx+d} \):

\[
f'(x) = \frac{\det f}{(cx + d)^2}.
\]

Euclidean algorithm:

- Homographies

\[
h_m(x) = \frac{1}{m + x},
\]

with \( \det h_m = -1 \).

- For \( h = h_{m_1} \circ \cdots \circ h_{m_k} \)

\[
h(0) = \frac{p}{q} \Rightarrow |h'(0)| = \frac{1}{q^2},
\]

\( p/q \) reduced.

CL algorithm:

- Homographies

\[
h_a(x) = \frac{1}{2^a(1 + x)},
\]

with \( \det h_a = -2^a \).

- For \( h = h_{m_1} \circ \cdots \circ h_{m_k} \)

\[
h(0) = \frac{p}{q} \Rightarrow |h'(0)| \text{ vs. } \frac{1}{q^2}?
\]

\( p/q \) reduced.
Reduced denominators and inverse branches

Observe for \( f(x) = \frac{ax+b}{cx+d} \):

\[
f'(x) = \frac{\det f}{(cx+d)^2}.
\]

Euclidean algorithm:

- **Homographies**
  \[
h_m(x) = \frac{1}{m+x},
\]
  with \( \det h_m = -1 \).
- **For** \( h = h_{m_1} \circ \cdots \circ h_{m_k} \)
  \[
h(0) = \frac{p}{q} \Rightarrow |h'(0)| = \frac{1}{q^2},
\]
  \( p/q \) reduced.

CL algorithm:

- **Homographies**
  \[
h_a(x) = \frac{1}{2^a(1+x)},
\]
  with \( \det h_a = -2^a \).
- **For** \( h = h_{m_1} \circ \cdots \circ h_{m_k} \)
  \[
h(0) = \frac{p}{q} \Rightarrow |h'(0)| \text{ vs. } \frac{1}{q^2}?
\]
  \( p/q \) reduced.

**Problem:** Denominator retrieved is engorged by powers of two.
Recording the dyadic behaviour

Dyadic behaviour is related to *divisibility*
Recording the dyadic behaviour

Dyadic behaviour is related to divisibility

⇒ ... but we employ analysis!
Recording the dyadic behaviour

Dyadic behaviour is related to divisibility

⇒ ... but we employ analysis!

Response: Dyadic numbers \( \mathbb{Q}_2 \)!

Dyadic topology = Divisibility by 2 constraints,

using the dyadic norm \(| \cdot |_2\).
Recording the dyadic behaviour

Dyadic behaviour is related to divisibility
\[ \implies \ldots \text{ but we employ analysis!} \]

**Response:** Dyadic numbers \( \mathbb{Q}_2 \)!

Dyadic topology = Divisibility by 2 constraints,
using the dyadic norm \( \cdot |_2 \).

- Introduce dyadic component
  \[ \implies \text{mixed dynamical system } (x, y) \in \mathcal{I} \times \mathbb{Q}_2 \]
Recording the dyadic behaviour

Dyadic behaviour is related to divisibility

⇒ ... but we employ analysis!

Response: Dyadic numbers $\mathbb{Q}_2$!

Dyadic topology $= \text{Divisibility by 2 constraints}$,

using the dyadic norm $|\cdot|_2$.

- Introduce dyadic component
  $\Rightarrow$ mixed dynamical system $(x, y) \in \mathcal{I} \times \mathbb{Q}_2$

- Incorporate $\mathbb{Q}_2$ into the Transfer Operator?
Recording the dyadic behaviour

Dyadic behaviour is related to divisibility
\[ \Rightarrow \ldots \text{ but we employ analysis!} \]

Response: Dyadic numbers \( \mathbb{Q}_2 \) !

Dyadic topology = Divisibility by 2 constraints,
using the dyadic norm \( | \cdot |_2 \).

- Introduce dyadic component
  \[ \Rightarrow \text{mixed dynamical system } (x, y) \in \mathcal{I} \times \mathbb{Q}_2 \]
- Incorporate \( \mathbb{Q}_2 \) into the Transfer Operator?

Idea works!
The extended dynamical system

Introduce $\mathcal{I} := \mathcal{I} \times \mathbb{Q}_2$ and $\mathcal{T} : \mathcal{I} \to \mathcal{I}$ as follows

$$\mathcal{T}(x, y) = (T_a(x), T_a(y)),$$

for $x \in \mathcal{I}_a = [2^{-a-1}, 2^{-a}]$. This gives inverse branches

$$h_a(x, y) = (h_a(x), h_a(y)),$$  \quad  (x, y) \in \mathcal{I}.$$
The extended dynamical system

Introduce $\mathcal{I} := \mathcal{I} \times \mathbb{Q}_2$ and $T: \mathcal{I} \to \mathcal{I}$ as follows

$$T(x, y) = (T_a(x), T_a(y)),$$

for $x \in \mathcal{I}_a = [2^{-a-1}, 2^{-a}]$. This gives inverse branches

$$h_a(x, y) = (h_a(x), h_a(y)),$$

$(x, y) \in \mathcal{I}$.

Evolution is led by the real component, which determines $a$. 
The extended dynamical system

Introduce $\mathcal{I} := \mathcal{I} \times \mathbb{Q}_2$ and $\mathcal{T}: \mathcal{I} \to \mathcal{I}$ as follows

$$\mathcal{T}(x, y) = (T_a(x), T_a(y)),$$

for $x \in \mathcal{I}_a = [2^{-a-1}, 2^{-a}]$. This gives inverse branches

$$h_a(x, y) = (h_a(x), h_a(y)),$$  \quad (x, y) \in \mathcal{I}.

Evolution is led by the real component, which determines $a$.

For Transfer operator $\Rightarrow$ need change of variables formula!
The extended dynamical system

Introduce $\mathcal{I} := \mathcal{I} \times \mathbb{Q}_2$ and $T: \mathcal{I} \to \mathcal{I}$ as follows

$$T(x, y) = (T_a(x), T_a(y)),$$

for $x \in \mathcal{I}_a = [2^{-a-1}, 2^{-a}]$. This gives inverse branches

$$h_a(x, y) = (h_a(x), h_a(y)), \quad (x, y) \in \mathcal{I}.$$

Evolution is led by the real component, which determines $a$.

For Transfer operator $\Rightarrow$ need change of variables formula!

Haar (translation invariant) measure $\nu$ on $\mathbb{Q}_2$ has one!
Functional space $\mathcal{F}$ for the extended operator $\mathbf{H}_s$

- **Real component** directs the dynamical system:
  - *sections* $F_y$ fixing $y \in \mathbb{Q}_2$ asked to be $C^1(\mathcal{I})$.
  - the **dyadic component** follows, demanding only integrability.
Functional space $\mathcal{F}$ for the extended operator $\mathbb{H}_s$

Real component directs the dynamical system:
- *sections* $F_y$ fixing $y \in \mathbb{Q}_2$ asked to be $C^1(\mathcal{I})$.
- the dyadic component follows, demanding only integrability.

Ensuing space $\mathcal{F}$ makes $\mathbb{H}_s$
- have a dominant eigenvalue and spectral gap
  relying strongly on the real component.
Functional space $\mathcal{F}$ for the extended operator $H_s$

Real component directs the dynamical system:
- Sections $F_y$ fixing $y \in \mathbb{Q}_2$ asked to be $C^1(I)$.
- The dyadic component follows, demanding only integrability.

Ensuing space $\mathcal{F}$ makes $H_s$
- Have a dominant eigenvalue and spectral gap, relying strongly on the real component.

We can finish the dynamical analysis!
Conclusion and further questions

- We have studied the average number of shifts and substractions for the CL algorithm.
- Study makes an interesting use of the dyadics in the framework of dynamical analysis.

Questions:
1. Conjecture: The successive pairs \((p_i, q_i)\) given by the algorithm satisfy \(\log_2 \gcd(p_i, q_i) \sim \frac{i}{2}\).
2. Comparison to other binary algorithms: binary GCD, LSB.
Conclusion and further questions

We have studied the average number of shifts and subtractions for the CL algorithm.

Study makes an interesting use of the dyadics in the framework of dynamical analysis.

Questions:

1. **Conjecture:** The successive pairs \((p_i, q_i)\) given by the algorithm satisfy

\[
\log_2 \gcd(p_i, q_i) \sim i/2.
\]

Back to (13, 31)

<table>
<thead>
<tr>
<th>(i)</th>
<th>(a_i)</th>
<th>(p_i)</th>
<th>(q_i)</th>
<th>(\gcd(p_i, q_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>13</td>
<td>31</td>
<td>2^0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>26</td>
<td>2^0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>6</td>
<td>20</td>
<td>2^1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>8</td>
<td>12</td>
<td>2^2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>2^2</td>
</tr>
</tbody>
</table>
Conclusion and further questions

- We have studied the average number of shifts and subtractions for the CL algorithm.
- Study makes an interesting use of the dyadics in the framework of dynamical analysis.

Questions:

1. **Conjecture:** The successive pairs \((p_i, q_i)\) given by the algorithm satisfy

\[
\log_2 \gcd(p_i, q_i) \sim i/2 .
\]

Back to \((13, 31)\)

<table>
<thead>
<tr>
<th>(i)</th>
<th>(a_i)</th>
<th>(p_i)</th>
<th>(q_i)</th>
<th>(\gcd(p_i, q_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>13</td>
<td>31</td>
<td>2^0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>26</td>
<td>2^0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>6</td>
<td>20</td>
<td>2^1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>8</td>
<td>12</td>
<td>2^2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>2^2</td>
</tr>
</tbody>
</table>

2. **Comparison to other binary algorithms:** binary GCD, LSB.
Conclusion and further questions

 spoilers: We have studied the average number of shifts and subtractions for the CL algorithm.
 spoilers: Study makes an interesting use of the dyadics in the framework of dynamical analysis.

Questions:

1. Conjecture: The successive pairs \((p_i, q_i)\) given by the algorithm satisfy

\[
\log_2 \gcd(p_i, q_i) \sim \frac{i}{2}.
\]

Back to \((13, 31)\)

<table>
<thead>
<tr>
<th>i</th>
<th>(a_i)</th>
<th>(p_i)</th>
<th>(q_i)</th>
<th>(\gcd(p_i, q_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>13</td>
<td>31</td>
<td>2^0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>26</td>
<td>2^0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>6</td>
<td>20</td>
<td>2^1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>8</td>
<td>12</td>
<td>2^2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>2^2</td>
</tr>
</tbody>
</table>

2. Comparison to other binary algorithms: binary GCD, LSB.