Arithmetical subword complexity of automatic sequences

Jakub Konieczny

Department of Computer Science
University of Oxford

One World Combinatorics on Words Seminar
23 Jan 2024

How complex is a sequence?

Setup: Let $a\colon \mathbb{N} \to \Omega$ be a sequence.
$\Omega =$ finite alphabet, e.g. $\Omega = \{0, 1\}$.

General question: How complex is a? $\longrightarrow$ hopelessly vague...

- Computational complexity of finding $a(n)$.
- Subword complexity = number of length-ℓ subwords that appear in a:

$$p_a(\ell) = \#\left\{ w \in \Omega^\ell \;:\; (\exists\, n)\ a(n+i) = w(i) \text{ for } 0 \le i < \ell \right\}.$$

- Arithmetical subword complexity complexity = number of length-ℓ subwords that appear in *a along an arithmetic progression*:

$$p_a^{\mathrm{AP}}(\ell) = \#\left\{ w \in \Omega^\ell \;:\; (\exists\, n, m)\ a(n+im) = w(i) \text{ for } 0 \le i < \ell \right\}.$$

- polynomial subword complexity, d-complexity, (maximal) pattern complexity, asymptotic subword complexity, etc., etc.… $\longrightarrow$ we will not discuss those

How complex is a sequence?

Setup: Let $a\colon \mathbb{N} \to \Omega$ be a sequence.
$\Omega =$ finite alphabet, e.g. $\Omega = \{0, 1\}$.

General question: How complex is a? $\longrightarrow$ hopelessly vague...

- Computational complexity of finding $a(n)$.
- Subword complexity = number of length-ℓ subwords that appear in a:

$$p_a(\ell) = \# \left\{ w \in \Omega^\ell \,:\, (\exists\, n)\ a(n+i) = w(i) \text{ for } 0 \leq i < \ell \right\}.$$

- Arithmetical subword complexity complexity = number of length-ℓ subwords that appear in *a along an arithmetic progression*:

$$p_a^{\mathrm{AP}}(\ell) = \# \left\{ w \in \Omega^\ell \,:\, (\exists\, n, m)\ a(n+im) = w(i) \text{ for } 0 \leq i < \ell \right\}.$$

- polynomial subword complexity, d-complexity, (maximal) pattern complexity, asymptotic subword complexity, etc., etc.. . . $\longrightarrow$ we will not discuss those

How complex is a sequence?

Setup: Let $a\colon \mathbb{N} \to \Omega$ be a sequence.
$\Omega =$ finite alphabet, e.g. $\Omega = \{0, 1\}$.

General question: How complex is a? $\longrightarrow$ hopelessly vague...

- Computational complexity of finding $a(n)$.
- Subword complexity = number of length-ℓ subwords that appear in a:

$$p_a(\ell) = \#\left\{w \in \Omega^\ell \,:\, (\exists\, n)\ a(n+i) = w(i) \text{ for } 0 \leq i < \ell\right\}.$$

- Arithmetical subword complexity complexity = number of length-ℓ subwords that appear in *a along an arithmetic progression*:

$$p_a^{\mathrm{AP}}(\ell) = \#\left\{w \in \Omega^\ell \,:\, (\exists\, n, m)\ a(n+im) = w(i) \text{ for } 0 \leq i < \ell\right\}.$$

- polynomial subword complexity, d-complexity, (maximal) pattern complexity, asymptotic subword complexity, etc., etc.... $\longrightarrow$ we will not discuss those

How complex is a sequence?

Setup: Let $a\colon \mathbb{N} \to \Omega$ be a sequence.
$\Omega =$ finite alphabet, e.g. $\Omega = \{0, 1\}$.

General question: How complex is a? $\longrightarrow$ hopelessly vague...

- Computational complexity of finding $a(n)$.
- Subword complexity = number of length-ℓ subwords that appear in a:

$$p_a(\ell) = \#\left\{ w \in \Omega^\ell \;:\; (\exists\, n)\ a(n+i) = w(i) \text{ for } 0 \le i < \ell \right\}.$$

- Arithmetical subword complexity complexity = number of length-ℓ subwords that appear in *a along an arithmetic progression*:

$$p_a^{\mathrm{AP}}(\ell) = \#\left\{ w \in \Omega^\ell \;:\; (\exists\, n, m)\ a(n+im) = w(i) \text{ for } 0 \le i < \ell \right\}.$$

- polynomial subword complexity, d-complexity, (maximal) pattern complexity, asymptotic subword complexity, etc., etc.... $\longrightarrow$ we will not discuss those

How complex is a sequence?

Setup: Let $a\colon \mathbb{N} \to \Omega$ be a sequence.
$\Omega =$ finite alphabet, e.g. $\Omega = \{0, 1\}$.

General question: How complex is a? $\longrightarrow$ hopelessly vague...

- Computational complexity of finding $a(n)$.
- Subword complexity = number of length-ℓ subwords that appear in a:

$$p_a(\ell) = \#\left\{w \in \Omega^\ell \ : \ (\exists\ n)\ a(n+i) = w(i) \text{ for } 0 \leq i < \ell\right\}.$$

- Arithmetical subword complexity complexity = number of length-ℓ subwords that appear in a *along an arithmetic progression*:

$$p_a^{\text{AP}}(\ell) = \#\left\{w \in \Omega^\ell \ : \ (\exists\ n, m)\ a(n+im) = w(i) \text{ for } 0 \leq i < \ell\right\}.$$

- polynomial subword complexity, d-complexity, (maximal) pattern complexity, asymptotic subword complexity, etc., etc.... $\longrightarrow$ we will not discuss those

How complex is a sequence?

Setup: Let $a\colon \mathbb{N} \to \Omega$ be a sequence.
$\Omega =$ finite alphabet, e.g. $\Omega = \{0, 1\}$.

General question: How complex is a? $\longrightarrow$ hopelessly vague...

- Computational complexity of finding $a(n)$.
- Subword complexity = number of length-ℓ subwords that appear in a:

$$p_a(\ell) = \#\left\{w \in \Omega^\ell \;:\; (\exists\, n)\ a(n+i) = w(i) \text{ for } 0 \le i < \ell\right\}.$$

- Arithmetical subword complexity complexity = number of length-ℓ subwords that appear in a *along an arithmetic progression*:

$$p_a^{\mathrm{AP}}(\ell) = \#\left\{w \in \Omega^\ell \;:\; (\exists\, n, m)\ a(n+im) = w(i) \text{ for } 0 \le i < \ell\right\}.$$

- polynomial subword complexity, d-complexity, (maximal) pattern complexity, asymptotic subword complexity, etc., etc.... $\longrightarrow$ we will not discuss those

Subword complexity

Trivial bound: If $a\colon \mathbb{N} \to \Omega$ then $1 \le p_a(\ell) \le \#\Omega^\ell$.

Fact (bounded complexity $\Leftrightarrow$ eventual periodicity)

If a is eventually periodic then p_a is bounded. Conversely, if $p_a(\ell) \le \ell$ for at least one ℓ then a is eventually periodic.

Fact (minimal complexity $\Leftrightarrow$ Sturmian)

If $a\colon \mathbb{N} \to \{0,1\}$ and $p_a(\ell) = \ell + 1$ for all ℓ then a is a Sturmian sequence:

$$a(n) = \lfloor \alpha(n+1) + \beta \rfloor - \lfloor \alpha n + \beta \rfloor \qquad n \in \mathbb{N},$$

where $\alpha \in [0,1) \setminus \mathbb{Q}$, $\beta \in [0,1)$. Conversely, if a is Sturmian then $p_a(\ell) = \ell + 1$.

Fact (linear complexity for automatic sequences)

If a is an automatic sequence then $p_a(\ell) = O(\ell)$, i.e., $p_a(\ell) \le C\ell$ for a constant C.

Subword complexity

Trivial bound: If $a\colon \mathbb{N} \to \Omega$ then $1 \le p_a(\ell) \le \#\Omega^\ell$.

Fact (bounded complexity $\Leftrightarrow$ eventual periodicity)

If a is eventually periodic then p_a is bounded. Conversely, if $p_a(\ell) \le \ell$ for at least one ℓ then a is eventually periodic.

Fact (minimal complexity $\Leftrightarrow$ Sturmian)

If $a\colon \mathbb{N} \to \{0,1\}$ and $p_a(\ell) = \ell + 1$ for all ℓ then a is a Sturmian sequence:

$$a(n) = \lfloor \alpha(n+1) + \beta \rfloor - \lfloor \alpha n + \beta \rfloor \qquad n \in \mathbb{N},$$

where $\alpha \in [0,1) \setminus \mathbb{Q}$, $\beta \in [0,1)$. Conversely, if a is Sturmian then $p_a(\ell) = \ell + 1$.

Fact (linear complexity for automatic sequences)

If a is an automatic sequence then $p_a(\ell) = O(\ell)$, i.e., $p_a(\ell) \le C\ell$ for a constant C.

Subword complexity

Trivial bound: If $a\colon \mathbb{N} \to \Omega$ then $1 \leq p_a(\ell) \leq \#\Omega^\ell$.

Fact (bounded complexity $\Leftrightarrow$ eventual periodicity)

If a is eventually periodic then p_a is bounded. Conversely, if $p_a(\ell) \leq \ell$ for at least one ℓ then a is eventually periodic.

Fact (minimal complexity $\Leftrightarrow$ Sturmian)

If $a\colon \mathbb{N} \to \{0,1\}$ and $p_a(\ell) = \ell + 1$ for all ℓ then a is a Sturmian sequence:

$$a(n) = \lfloor \alpha(n+1) + \beta \rfloor - \lfloor \alpha n + \beta \rfloor \qquad n \in \mathbb{N},$$

where $\alpha \in [0,1) \setminus \mathbb{Q}$, $\beta \in [0,1)$. Conversely, if a is Sturmian then $p_a(\ell) = \ell + 1$.

Fact (linear complexity for automatic sequences)

If a is an automatic sequence then $p_a(\ell) = O(\ell)$, i.e., $p_a(\ell) \leq C\ell$ for a constant C.

Subword complexity

Trivial bound: If $a\colon \mathbb{N} \to \Omega$ then $1 \le p_a(\ell) \le \#\Omega^\ell$.

Fact (bounded complexity $\Leftrightarrow$ eventual periodicity)

If a is eventually periodic then p_a is bounded. Conversely, if $p_a(\ell) \le \ell$ for at least one ℓ then a is eventually periodic.

Fact (minimal complexity $\Leftrightarrow$ Sturmian)

If $a\colon \mathbb{N} \to \{0,1\}$ and $p_a(\ell) = \ell + 1$ for all ℓ then a is a Sturmian sequence:

$$a(n) = \lfloor \alpha(n+1) + \beta \rfloor - \lfloor \alpha n + \beta \rfloor \qquad n \in \mathbb{N},$$

where $\alpha \in [0,1) \setminus \mathbb{Q}$, $\beta \in [0,1)$. Conversely, if a is Sturmian then $p_a(\ell) = \ell + 1$.

Fact (linear complexity for automatic sequences)

If a is an automatic sequence then $p_a(\ell) = O(\ell)$, i.e., $p_a(\ell) \le C\ell$ for a constant C.

The Thue–Morse(–Prouhet) sequence

$$+1, -1, -1, +1, -1, +1, +1, -1, -1, +1, +1, -1, +1, -1, -1, +1, \ldots$$

The Thue–Morse sequence (discovered by Prouhet) $t\colon \mathbb{N} \to \{+1, -1\}$ is the paradigmatic example of an automatic sequence. It can be defined in several ways:

1. Explicit formula: $t(n) = \begin{cases} +1 & \text{if } n \text{ is } \textit{evil} \text{ (i.e., sum of binary digits is even)}, \\ -1 & \text{if } n \text{ is } \textit{odious} \text{ (i.e., sum of binary digits is odd)}. \end{cases}$
2. Automatic sequence:

3. Recurrence: $t(0) = +1, \quad t(2n) = t(n), \quad t(2n+1) = -t(n).$
4. Fixed point of a substitution: $+1 \mapsto +1, -1; \quad -1 \mapsto -1, +1.$
5. Strongly 2-multiplicative sequence: $t(1) = -1$, and if $m < 2^\alpha$ then

$$t(2^\alpha n + m) = t(n)t(m).$$

The Thue–Morse(–Prouhet) sequence

$$+1, -1, -1, +1, -1, +1, +1, -1, -1, +1, +1, -1, +1, -1, -1, +1, \ldots$$

The Thue–Morse sequence (discovered by Prouhet) $t\colon \mathbb{N} \to \{+1, -1\}$ is the paradigmatic example of an automatic sequence. It can be defined in several ways:

1. Explicit formula: $t(n) = \begin{cases} +1 & \text{if } n \text{ is } \textit{evil} \text{ (i.e., sum of binary digits is even),} \\ -1 & \text{if } n \text{ is } \textit{odious} \text{ (i.e., sum of binary digits is odd).} \end{cases}$
2. Automatic sequence:

3. Recurrence: $t(0) = +1, \quad t(2n) = t(n), \quad t(2n+1) = -t(n).$
4. Fixed point of a substitution: $+1 \mapsto +1, -1; \quad -1 \mapsto -1, +1.$
5. Strongly 2-multiplicative sequence: $t(1) = -1$, and if $m < 2^\alpha$ then

$$t(2^\alpha n + m) = t(n)t(m).$$

The Thue–Morse(–Prouhet) sequence

$$+1, -1, -1, +1, -1, +1, +1, -1, -1, +1, +1, -1, +1, -1, -1, +1, \ldots$$

The Thue–Morse sequence (discovered by Prouhet) $t\colon \mathbb{N} \to \{+1, -1\}$ is the paradigmatic example of an automatic sequence. It can be defined in several ways:

1. Explicit formula: $t(n) = \begin{cases} +1 & \text{if } n \text{ is } \textit{evil} \text{ (i.e., sum of binary digits is even),} \\ -1 & \text{if } n \text{ is } \textit{odious} \text{ (i.e., sum of binary digits is odd).} \end{cases}$
2. Automatic sequence:

3. Recurrence: $t(0) = +1, \quad t(2n) = t(n), \quad t(2n+1) = -t(n).$
4. Fixed point of a substitution: $+1 \mapsto +1, -1; \quad -1 \mapsto -1, +1.$
5. Strongly 2-multiplicative sequence: $t(1) = -1$, and if $m < 2^\alpha$ then

$$t(2^\alpha n + m) = t(n)t(m).$$

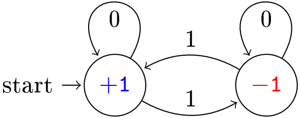
The Thue–Morse(–Prouhet) sequence

$$+1, -1, -1, +1, -1, +1, +1, -1, -1, +1, +1, -1, +1, -1, -1, +1, \dots$$

The Thue–Morse sequence (discovered by Prouhet) $t\colon \mathbb{N} \to \{+1, -1\}$ is the paradigmatic example of an automatic sequence. It can be defined in several ways:

1. Explicit formula: $t(n) = \begin{cases} +1 & \text{if } n \text{ is } \textit{evil} \text{ (i.e., sum of binary digits is even),} \\ -1 & \text{if } n \text{ is } \textit{odious} \text{ (i.e., sum of binary digits is odd).} \end{cases}$

2. Automatic sequence:



3. Recurrence: $t(0) = +1, \quad t(2n) = t(n), \quad t(2n+1) = -t(n).$

4. Fixed point of a substitution: $+1 \mapsto +1, -1; \quad -1 \mapsto -1, +1.$

5. Strongly 2-multiplicative sequence: $t(1) = -1$, and if $m < 2^\alpha$ then

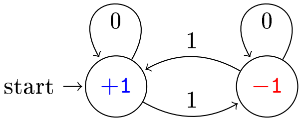
$$t(2^\alpha n + m) = t(n)t(m).$$

The Thue–Morse(–Prouhet) sequence

$$+1, -1, -1, +1, -1, +1, +1, -1, -1, +1, +1, -1, +1, -1, -1, +1, \ldots$$

The Thue–Morse sequence (discovered by Prouhet) $t\colon \mathbb{N} \to \{+1, -1\}$ is the paradigmatic example of an automatic sequence. It can be defined in several ways:

1. Explicit formula: $t(n) = \begin{cases} +1 & \text{if } n \text{ is } \textit{evil} \text{ (i.e., sum of binary digits is even),} \\ -1 & \text{if } n \text{ is } \textit{odious} \text{ (i.e., sum of binary digits is odd).} \end{cases}$
2. Automatic sequence:



3. Recurrence: $t(0) = +1, \quad t(2n) = t(n), \quad t(2n+1) = -t(n).$
4. Fixed point of a substitution: $+1 \mapsto +1, -1; \quad -1 \mapsto -1, +1.$
5. Strongly 2-multiplicative sequence: $t(1) = -1$, and if $m < 2^\alpha$ then

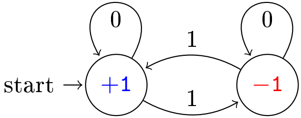
$$t(2^\alpha n + m) = t(n)t(m).$$

The Thue–Morse(–Prouhet) sequence

$$+1, -1, -1, +1, -1, +1, +1, -1, -1, +1, +1, -1, +1, -1, -1, +1, \dots$$

The Thue–Morse sequence (discovered by Prouhet) $t\colon \mathbb{N} \to \{+1, -1\}$ is the paradigmatic example of an automatic sequence. It can be defined in several ways:

1. Explicit formula: $t(n) = \begin{cases} +1 & \text{if } n \text{ is } \textit{evil} \text{ (i.e., sum of binary digits is even),} \\ -1 & \text{if } n \text{ is } \textit{odious} \text{ (i.e., sum of binary digits is odd).} \end{cases}$
2. Automatic sequence:



3. Recurrence: $t(0) = +1, \quad t(2n) = t(n), \quad t(2n+1) = -t(n).$
4. Fixed point of a substitution: $+1 \mapsto +1, -1; \quad -1 \mapsto -1, +1.$
5. Strongly 2-multiplicative sequence: $t(1) = -1$, and if $m < 2^\alpha$ then

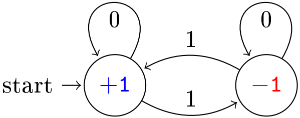
$$t(2^\alpha n + m) = t(n)t(m).$$

The Thue–Morse(–Prouhet) sequence

$$+1, -1, -1, +1, -1, +1, +1, -1, -1, +1, +1, -1, +1, -1, -1, +1, \dots$$

The Thue–Morse sequence (discovered by Prouhet) $t\colon \mathbb{N} \to \{+1, -1\}$ is the paradigmatic example of an automatic sequence. It can be defined in several ways:

1. Explicit formula: $t(n) = \begin{cases} +1 & \text{if } n \text{ is } \textit{evil} \text{ (i.e., sum of binary digits is even),} \\ -1 & \text{if } n \text{ is } \textit{odious} \text{ (i.e., sum of binary digits is odd).} \end{cases}$
2. Automatic sequence:



3. Recurrence: $t(0) = +1, \quad t(2n) = t(n), \quad t(2n+1) = -t(n).$
4. Fixed point of a substitution: $+1 \mapsto +1, -1; \quad -1 \mapsto -1, +1.$
5. Strongly 2-multiplicative sequence: $t(1) = -1$, and if $m < 2^\alpha$ then

$$t(2^\alpha n + m) = t(n)t(m).$$

Automatic sequences via finite automata

Some notation: We let k denote the base in which we work. $\longrightarrow$ e.g. $k = 10$ or $k = 2$

- $\Sigma_k = \{0, 1, \ldots, k-1\}$, the set of digits in base k;
- Σ_k^* is the set of words over Σ_k, monoid with concatenation;
- for $n \in \mathbb{N}$, $(n)_k \in \Sigma_k^*$ is the base-k expansion of n; $\longrightarrow$ no leading zeros
- for $w \in \Sigma_k^*$, $[w]_k \in \mathbb{N}$ is the integer encoded by w.

A finite k-automaton consists of:

- a finite set of states S with a distinguished initial state s_0;
- a transition function $\delta \colon S \times \Sigma_k \to S$;
- an output function $\tau \colon S \to \Omega$.

Computing the sequence:

- Extend δ to a map $S \times \Sigma_k^*$ with $\delta(s, uv) = \delta(\delta(s,u),v)$ or $\delta(\delta(s,v),u)$;
- The sequence computed by the automaton is given by $a(n) = \tau(\delta(s_0, (n)_k))$.
- The automaton above computes the Rudin–Shapiro sequence $(-1)^{\#\text{ of } 11 \text{ in } (n)_2}$.

Intuition: Automatic $\Longleftrightarrow$ Computable by a finite device.

Automatic sequences via finite automata

Some notation: We let k denote the base in which we work. $\longrightarrow$ e.g. $k = 10$ or $k = 2$

- $\Sigma_k = \{0, 1, \ldots, k-1\}$, the set of digits in base k;
- Σ_k^* is the set of words over Σ_k, monoid with concatenation;
- for $n \in \mathbb{N}$, $(n)_k \in \Sigma_k^*$ is the base-k expansion of n; $\longrightarrow$ no leading zeros
- for $w \in \Sigma_k^*$, $[w]_k \in \mathbb{N}$ is the integer encoded by w.

A finite k-automaton consists of:

- a finite set of states S with a distinguished initial state s_0;
- a transition function $\delta \colon S \times \Sigma_k \to S$;
- an output function $\tau \colon S \to \Omega$.

Computing the sequence:

- Extend δ to a map $S \times \Sigma_k^*$ with $\delta(s, uv) = \delta(\delta(s,u), v)$ or $\delta(\delta(s,v),u)$;
- The sequence computed by the automaton is given by $a(n) = \tau\left(\delta(s_0, (n)_k)\right)$.
- The automaton above computes the Rudin–Shapiro sequence $(-1)^{\#\text{ of }11\text{ in }(n)_2}$.

Intuition: Automatic $\Longleftrightarrow$ Computable by a finite device.

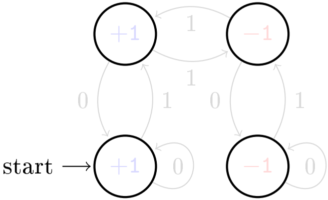
Automatic sequences via finite automata

Some notation: We let k denote the base in which we work. $\longrightarrow$ e.g. $k = 10$ or $k = 2$

- $\Sigma_k = \{0, 1, \ldots, k-1\}$, the set of digits in base k;
- Σ_k^* is the set of words over Σ_k, monoid with concatenation;
- for $n \in \mathbb{N}$, $(n)_k \in \Sigma_k^*$ is the base-k expansion of n; $\longrightarrow$ no leading zeros
- for $w \in \Sigma_k^*$, $[w]_k \in \mathbb{N}$ is the integer encoded by w.

A finite k-automaton consists of:

- a finite set of states S with a distinguished initial state s_0;
- a transition function $\delta \colon S \times \Sigma_k \to S$;
- an output function $\tau \colon S \to \Omega$.



Computing the sequence:

- Extend δ to a map $S \times \Sigma_k^*$ with $\delta(s, uv) = \delta(\delta(s,u),v)$ or $\delta(\delta(s,v),u)$;
- The sequence computed by the automaton is given by $a(n) = \tau\left(\delta(s_0, (n)_k)\right)$.
- The automaton above computes the Rudin–Shapiro sequence $(-1)^{\#\text{ of }11\text{ in }(n)_2}$.

Intuition: Automatic $\Longleftrightarrow$ Computable by a finite device.

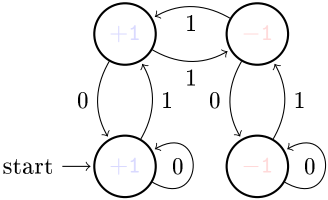
Automatic sequences via finite automata

Some notation: We let k denote the base in which we work. $\longrightarrow$ e.g. $k = 10$ or $k = 2$

- $\Sigma_k = \{0, 1, \ldots, k-1\}$, the set of digits in base k;
- Σ_k^* is the set of words over Σ_k, monoid with concatenation;
- for $n \in \mathbb{N}$, $(n)_k \in \Sigma_k^*$ is the base-k expansion of n; $\longrightarrow$ no leading zeros
- for $w \in \Sigma_k^*$, $[w]_k \in \mathbb{N}$ is the integer encoded by w.

A finite k-automaton consists of:

- a finite set of states S with a distinguished initial state s_0;
- a transition function $\delta\colon S \times \Sigma_k \to S$;
- an output function $\tau\colon S \to \Omega$.



Computing the sequence:

- Extend δ to a map $S \times \Sigma_k^*$ with $\delta(s, uv) = \delta(\delta(s,u), v)$ or $\delta(\delta(s,v), u)$;
- The sequence computed by the automaton is given by $a(n) = \tau\left(\delta(s_0, (n)_k)\right)$.
- The automaton above computes the Rudin–Shapiro sequence $(-1)^{\#\text{ of }11\text{ in }(n)_2}$.

Intuition: Automatic $\Longleftrightarrow$ Computable by a finite device.

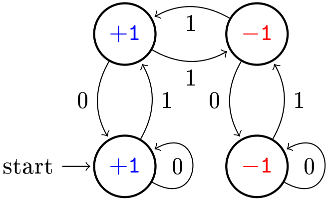
Automatic sequences via finite automata

Some notation: We let k denote the base in which we work. $\longrightarrow$ e.g. $k = 10$ or $k = 2$

- $\Sigma_k = \{0, 1, \dots, k-1\}$, the set of digits in base k;
- Σ_k^* is the set of words over Σ_k, monoid with concatenation;
- for $n \in \mathbb{N}$, $(n)_k \in \Sigma_k^*$ is the base-k expansion of n; $\longrightarrow$ no leading zeros
- for $w \in \Sigma_k^*$, $[w]_k \in \mathbb{N}$ is the integer encoded by w.

A finite k-automaton consists of:

- a finite set of states S with a distinguished initial state s_0;
- a transition function $\delta \colon S \times \Sigma_k \to S$;
- an output function $\tau \colon S \to \Omega$.



Computing the sequence:

- Extend δ to a map $S \times \Sigma_k^*$ with $\delta(s, uv) = \delta(\delta(s, u), v)$ or $\delta(\delta(s, v), u)$;
- The sequence computed by the automaton is given by $a(n) = \tau(\delta(s_0, (n)_k))$.
- The automaton above computes the Rudin–Shapiro sequence $(-1)^{\#\text{ of 11 in }(n)_2}$.

Intuition: Automatic $\Longleftrightarrow$ Computable by a finite device.

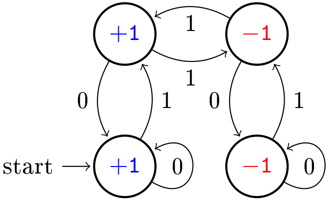
Automatic sequences via finite automata

Some notation: We let k denote the base in which we work. $\longrightarrow$ e.g. $k = 10$ or $k = 2$

- $\Sigma_k = \{0, 1, \ldots, k-1\}$, the set of digits in base k;
- Σ_k^* is the set of words over Σ_k, monoid with concatenation;
- for $n \in \mathbb{N}$, $(n)_k \in \Sigma_k^*$ is the base-k expansion of n; $\longrightarrow$ no leading zeros
- for $w \in \Sigma_k^*$, $[w]_k \in \mathbb{N}$ is the integer encoded by w.

A finite k-automaton consists of:

- a finite set of states S with a distinguished initial state s_0;
- a transition function $\delta \colon S \times \Sigma_k \to S$;
- an output function $\tau \colon S \to \Omega$.



Computing the sequence:

- Extend δ to a map $S \times \Sigma_k^*$ with $\delta(s, uv) = \delta(\delta(s, u), v)$ or $\delta(\delta(s, v), u)$;
- The sequence computed by the automaton is given by $a(n) = \tau\left(\delta(s_0, (n)_k)\right)$.
- The automaton above computes the Rudin–Shapiro sequence $(-1)^{\#\text{ of }11\text{ in }(n)_2}$.

Intuition: Automatic $\Longleftrightarrow$ Computable by a finite device.

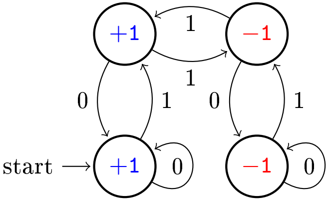
Automatic sequences via finite automata

Some notation: We let k denote the base in which we work. $\longrightarrow$ e.g. $k = 10$ or $k = 2$

- $\Sigma_k = \{0, 1, \ldots, k-1\}$, the set of digits in base k;
- Σ_k^* is the set of words over Σ_k, monoid with concatenation;
- for $n \in \mathbb{N}$, $(n)_k \in \Sigma_k^*$ is the base-k expansion of n; $\longrightarrow$ no leading zeros
- for $w \in \Sigma_k^*$, $[w]_k \in \mathbb{N}$ is the integer encoded by w.

A finite k-automaton consists of:

- a finite set of states S with a distinguished initial state s_0;
- a transition function $\delta \colon S \times \Sigma_k \to S$;
- an output function $\tau \colon S \to \Omega$.



Computing the sequence:

- Extend δ to a map $S \times \Sigma_k^*$ with $\delta(s, uv) = \delta(\delta(s,u), v)$ or $\delta(\delta(s,v), u)$;
- The sequence computed by the automaton is given by $a(n) = \tau\left(\delta(s_0, (n)_k)\right)$.
- The automaton above computes the Rudin–Shapiro sequence $(-1)^{\#\text{ of } 11 \text{ in } (n)_2}$.

Intuition: Automatic $\Longleftrightarrow$ Computable by a finite device.

Subword complexity of automatic sequences

Proposition

The subword complexity of the Thue–Morse sequence is given by:

$$p_t(\ell) = \begin{cases} 3 \cdot 2^k + 4(r-1) & \text{if } \ell = 2^k + r \text{ with } 1 \le r \le 2^{k-1}, \\ 4 \cdot 2^k + 2(r-1) & \text{if } \ell = 2^k + r \text{ with } 2^{k-1} < r \le 2^k. \end{cases}$$

Proposition

If a is automatic then $p_a(\ell) = O(\ell)$.

Sketch of proof:

- Pick any $\alpha, n \in \mathbb{N}$. For $0 \le i < k^\alpha$, $a(k^\alpha n + i)$ is determined by $\delta(s_0, (n)_2)$.
- If $w \in \{0,1\}^{k^\alpha}$ then w appears in a between $k^\alpha n + r$ and $k^\alpha(n+1) + r$ for some $n, r \in \mathbb{N}$, $r < k^\alpha$. Thus, w is determined by $\delta(s_0, (n)_k), \delta(s_0, (n+1)_k)$ and r, and

$$p_a(k^\alpha) \le \#S \times \#S \times k^\alpha.$$

Subword complexity of automatic sequences

Proposition

The subword complexity of the Thue–Morse sequence is given by:

$$p_t(\ell) = \begin{cases} 3 \cdot 2^k + 4(r-1) & \text{if } \ell = 2^k + r \text{ with } 1 \leq r \leq 2^{k-1}, \\ 4 \cdot 2^k + 2(r-1) & \text{if } \ell = 2^k + r \text{ with } 2^{k-1} < r \leq 2^k. \end{cases}$$

Proposition

If a is automatic then $p_a(\ell) = O(\ell)$.

Sketch of proof:

- Pick any $\alpha, n \in \mathbb{N}$. For $0 \leq i < k^\alpha$, $a(k^\alpha n + i)$ is determined by $\delta(s_0, (n)_2)$.
- If $w \in \{0,1\}^{k^\alpha}$ then w appears in a between $k^\alpha n + r$ and $k^\alpha(n+1) + r$ for some $n, r \in \mathbb{N}$, $r < k^\alpha$. Thus, w is determined by $\delta(s_0, (n)_k), \delta(s_0, (n+1)_k)$ and r, and

$$p_a(k^\alpha) \leq \#S \times \#S \times k^\alpha.$$

Subword complexity of automatic sequences

Proposition

The subword complexity of the Thue–Morse sequence is given by:

$$p_t(\ell) = \begin{cases} 3 \cdot 2^k + 4(r-1) & \text{if } \ell = 2^k + r \text{ with } 1 \leq r \leq 2^{k-1}, \\ 4 \cdot 2^k + 2(r-1) & \text{if } \ell = 2^k + r \text{ with } 2^{k-1} < r \leq 2^k. \end{cases}$$

Proposition

If a is automatic then $p_a(\ell) = O(\ell)$.

Sketch of proof:

- Pick any $\alpha, n \in \mathbb{N}$. For $0 \leq i < k^\alpha$, $a(k^\alpha n + i)$ is determined by $\delta(s_0, (n)_2)$.
- If $w \in \{0,1\}^{k^\alpha}$ then w appears in a between $k^\alpha n + r$ and $k^\alpha(n+1) + r$ for some $n, r \in \mathbb{N}$, $r < k^\alpha$. Thus, w is determined by $\delta(s_0, (n)_k), \delta(s_0, (n+1)_k)$ and r, and

$$p_a(k^\alpha) \leq \#S \times \#S \times k^\alpha.$$

Thue–Morse along subsequences

Intuition: If $(m(n))_{n=0}^{\infty}$ is a sequence that *"does not know"* about binary expansions then $s(n) = t(m(n))$ looks *"random"*. In particular, we expect that:

- $s(n) = +1$ for half of n: $\frac{1}{N} \sum_{n<N} s(n) \to 0$ as $N \to \infty$;
- s is normal: for $h_1 < h_2 < \cdots < h_r$, $\frac{1}{N} \sum_{n<N} \prod_{i=1}^{r} s(n+h_i) \to 0$ as $N \to \infty$;
- maximal subword complexity: $p_s(\ell) = 2^{\ell}$.

Theorem (Drmota, Mauduit & Rivat)

The sequence $t(n^2)$ is normal (in particular, it has maximal subword complexity).

Theorem (Müllner & Spiegelhofer)

The sequence $t(\lfloor n^c \rfloor)$ is normal for $1 < c < 3/2$.

Fact: The restriction $t(An+B)$ of the Thue–Morse sequence to a given arithmetic progression is automatic (and hence very non-random).

Hope: The restriction of $t(n)$ to a *random* arithmetic progression looks random.

Thue–Morse along subsequences

Intuition: If $(m(n))_{n=0}^{\infty}$ is a sequence that *"does not know"* about binary expansions then $s(n) = t(m(n))$ looks *"random"*. In particular, we expect that:

- $s(n) = +1$ for half of n: $\displaystyle\frac{1}{N}\sum_{n<N} s(n) \to 0$ as $N \to \infty$;
- s is normal: for $h_1 < h_2 < \cdots < h_r$, $\displaystyle\frac{1}{N}\sum_{n<N}\prod_{i=1}^{r} s(n+h_i) \to 0$ as $N \to \infty$;
- maximal subword complexity: $p_s(\ell) = 2^{\ell}$.

Theorem (Drmota, Mauduit & Rivat)

The sequence $t(n^2)$ is normal (in particular, it has maximal subword complexity).

Theorem (Müllner & Spiegelhofer)

The sequence $t(\lfloor n^c \rfloor)$ is normal for $1 < c < 3/2$.

Fact: The restriction $t(An + B)$ of the Thue–Morse sequence to a given arithmetic progression is automatic (and hence very non-random).

Hope: The restriction of $t(n)$ to a *random* arithmetic progression looks random.

Thue–Morse along subsequences

Intuition: If $(m(n))_{n=0}^{\infty}$ is a sequence that *"does not know"* about binary expansions then $s(n) = t(m(n))$ looks *"random"*. In particular, we expect that:

- $s(n) = +1$ for half of n: $\frac{1}{N} \sum_{n<N} s(n) \to 0$ as $N \to \infty$;
- s is normal: for $h_1 < h_2 < \cdots < h_r$, $\frac{1}{N} \sum_{n<N} \prod_{i=1}^{r} s(n+h_i) \to 0$ as $N \to \infty$;
- maximal subword complexity: $p_s(\ell) = 2^\ell$.

Theorem (Drmota, Mauduit & Rivat)

The sequence $t(n^2)$ is normal (in particular, it has maximal subword complexity).

Theorem (Müllner & Spiegelhofer)

The sequence $t(\lfloor n^c \rfloor)$ is normal for $1 < c < 3/2$.

Fact: The restriction $t(An + B)$ of the Thue–Morse sequence to a given arithmetic progression is automatic (and hence very non-random).

Hope: The restriction of $t(n)$ to a *random* arithmetic progression looks random.

Thue–Morse along subsequences

Intuition: If $(m(n))_{n=0}^{\infty}$ is a sequence that *"does not know"* about binary expansions then $s(n) = t(m(n))$ looks *"random"*. In particular, we expect that:

- $s(n) = +1$ for half of n: $\frac{1}{N} \sum_{n<N} s(n) \to 0$ as $N \to \infty$;
- s is normal: for $h_1 < h_2 < \cdots < h_r$, $\frac{1}{N} \sum_{n<N} \prod_{i=1}^{r} s(n + h_i) \to 0$ as $N \to \infty$;
- maximal subword complexity: $p_s(\ell) = 2^{\ell}$.

Theorem (Drmota, Mauduit & Rivat)

The sequence $t(n^2)$ is normal (in particular, it has maximal subword complexity).

Theorem (Müllner & Spiegelhofer)

The sequence $t(\lfloor n^c \rfloor)$ is normal for $1 < c < 3/2$.

Fact: The restriction $t(An + B)$ of the Thue–Morse sequence to a given arithmetic progression is automatic (and hence very non-random).

Hope: The restriction of $t(n)$ to a *random* arithmetic progression looks random.

Thue–Morse along subsequences

Intuition: If $(m(n))_{n=0}^{\infty}$ is a sequence that *"does not know"* about binary expansions then $s(n) = t(m(n))$ looks *"random"*. In particular, we expect that:

- $s(n) = +1$ for half of n: $\frac{1}{N} \sum_{n<N} s(n) \to 0$ as $N \to \infty$;
- s is normal: for $h_1 < h_2 < \cdots < h_r$, $\frac{1}{N} \sum_{n<N} \prod_{i=1}^{r} s(n + h_i) \to 0$ as $N \to \infty$;
- maximal subword complexity: $p_s(\ell) = 2^\ell$.

Theorem (Drmota, Mauduit & Rivat)

The sequence $t(n^2)$ is normal (in particular, it has maximal subword complexity).

Theorem (Müllner & Spiegelhofer)

The sequence $t(\lfloor n^c \rfloor)$ is normal for $1 < c < 3/2$.

Fact: The restriction $t(An + B)$ of the Thue–Morse sequence to a given arithmetic progression is automatic (and hence very non-random).

Hope: The restriction of $t(n)$ to a *random* arithmetic progression looks random.

Thue–Morse along subsequences

Intuition: If $(m(n))_{n=0}^{\infty}$ is a sequence that *"does not know"* about binary expansions then $s(n) = t(m(n))$ looks *"random"*. In particular, we expect that:

- $s(n) = +1$ for half of n: $\frac{1}{N}\sum_{n<N} s(n) \to 0$ as $N \to \infty$;
- s is normal: for $h_1 < h_2 < \cdots < h_r$, $\frac{1}{N}\sum_{n<N}\prod_{i=1}^{r} s(n+h_i) \to 0$ as $N \to \infty$;
- maximal subword complexity: $p_s(\ell) = 2^\ell$.

Theorem (Drmota, Mauduit & Rivat)

The sequence $t(n^2)$ is normal (in particular, it has maximal subword complexity).

Theorem (Müllner & Spiegelhofer)

The sequence $t(\lfloor n^c \rfloor)$ is normal for $1 < c < 3/2$.

Fact: The restriction $t(An + B)$ of the Thue–Morse sequence to a given arithmetic progression is automatic (and hence very non-random).

Hope: The restriction of $t(n)$ to a *random* arithmetic progression looks random.

Thue–Morse along subsequences

Intuition: If $(m(n))_{n=0}^{\infty}$ is a sequence that *"does not know"* about binary expansions then $s(n) = t(m(n))$ looks *"random"*. In particular, we expect that:

- $s(n) = +1$ for half of n: $\frac{1}{N} \sum_{n<N} s(n) \to 0$ as $N \to \infty$;
- s is normal: for $h_1 < h_2 < \cdots < h_r$, $\frac{1}{N} \sum_{n<N} \prod_{i=1}^{r} s(n + h_i) \to 0$ as $N \to \infty$;
- maximal subword complexity: $p_s(\ell) = 2^\ell$.

Theorem (Drmota, Mauduit & Rivat)

The sequence $t(n^2)$ is normal (in particular, it has maximal subword complexity).

Theorem (Müllner & Spiegelhofer)

The sequence $t(\lfloor n^c \rfloor)$ is normal for $1 < c < 3/2$.

Fact: The restriction $t(An + B)$ of the Thue–Morse sequence to a given arithmetic progression is automatic (and hence very non-random).

Hope: The restriction of $t(n)$ to a *random* arithmetic progression looks random.

Thue–Morse along subsequences

Intuition: If $(m(n))_{n=0}^{\infty}$ is a sequence that *"does not know"* about binary expansions then $s(n) = t(m(n))$ looks *"random"*. In particular, we expect that:

- $s(n) = +1$ for half of n: $\frac{1}{N}\sum_{n<N} s(n) \to 0$ as $N \to \infty$;
- s is normal: for $h_1 < h_2 < \cdots < h_r$, $\frac{1}{N}\sum_{n<N}\prod_{i=1}^{r} s(n+h_i) \to 0$ as $N \to \infty$;
- maximal subword complexity: $p_s(\ell) = 2^\ell$.

Theorem (Drmota, Mauduit & Rivat)

The sequence $t(n^2)$ is normal (in particular, it has maximal subword complexity).

Theorem (Müllner & Spiegelhofer)

The sequence $t(\lfloor n^c \rfloor)$ is normal for $1 < c < 3/2$.

Fact: The restriction $t(An+B)$ of the Thue–Morse sequence to a given arithmetic progression is automatic (and hence very non-random).

Hope: The restriction of $t(n)$ to a *random* arithmetic progression looks random.

Higher order Fourier analysis: first glance

Definition (Gowers norm)

Fix $d \geq 2$. Let $f\colon [N] := \{0, 1, \ldots, N-1\} \to \mathbb{C}$. Then $\|f\|_{U^d[N]} \geq 0$ is defined by:

$$\|f\|_{U^d[N]}^{2^d} = \mathop{\mathbb{E}}_{\mathbf{n}} \prod_{\omega \in \{0,1\}^d} \mathsf{C}^{|\omega|} f\left(n_0 + \omega_1 n_1 + \ldots \omega_d n_d\right),$$

where the average is taken over all parallelepipeds in $[N]$, i.e., over all $\mathbf{n} = (n_0, \ldots, n_d) \in \mathbb{Z}^{d+1}$ such that $n_0 + \omega_1 n_1 + \ldots \omega_d n_d \in [N]$ for all $\omega \in \{0,1\}^d$.

Theorem (Generalised von Neumann Theorem)

Fix $d \geq 2$ and let $f_0, f_1, \ldots, f_d\colon [N] \to \mathbb{C}$ be 1-bounded. Then

$$\left| \mathop{\mathbb{E}}_{n,m} f_0(n) f_1(n+m) f_2(n+2m) \ldots f_d(n+dm) \right| \ll \min_{0 \leq i \leq d} \|f_i\|_{U^d[N]}.$$

Higher order Fourier analysis: first glance

Definition (Gowers norm)

Fix $d \geq 2$. Let $f\colon [N] := \{0, 1, \ldots, N-1\} \to \mathbb{C}$. Then $\|f\|_{U^d[N]} \geq 0$ is defined by:

$$\|f\|_{U^d[N]}^{2^d} = \mathop{\mathbb{E}}_{\mathbf{n}} \prod_{\omega \in \{0,1\}^d} \mathsf{C}^{|\omega|} f\left(n_0 + \omega_1 n_1 + \ldots \omega_d n_d\right),$$

where the average is taken over all parallelepipeds in $[N]$, i.e., over all $\mathbf{n} = (n_0, \ldots, n_d) \in \mathbb{Z}^{d+1}$ such that $n_0 + \omega_1 n_1 + \ldots \omega_d n_d \in [N]$ for all $\omega \in \{0,1\}^d$.

Theorem (Generalised von Neumann Theorem)

Fix $d \geq 2$ and let $f_0, f_1, \ldots, f_d\colon [N] \to \mathbb{C}$ be 1-bounded. Then

$$\left| \mathop{\mathbb{E}}_{n,m} f_0(n) f_1(n+m) f_2(n+2m) \ldots f_d(n+dm) \right| \ll \min_{0 \leq i \leq d} \|f_i\|_{U^d[N]}.$$

Higher order Fourier analysis and arithmetical subword complexity

Corollary 1: If $f\colon [N] \to \{-1,+1\}$ and $\|f\|_{U^d[N]} \leq \varepsilon$ then f looks random along random $(d+1)$-term APs in $[N]$, meaning that for $w \in \{-1,+1\}^{d+1}$ we have:

$$\frac{\#\{(n,m) \,:\, n+im \in [N] \text{ and } f(n+im) = w(i) \text{ for } 0 \leq i \leq d\}}{\#\{(n,m) \,:\, n+im \in [N] \text{ for } 0 \leq i \leq d\}} = \frac{1}{2^{d+1}} + O(\varepsilon).$$

Corollary 2: In particular, if $f\colon \mathbb{N} \to \{-1,+1\}$ and $\|f\|_{U^d[N]} \to 0$ as $N \to \infty$ for each $d \geq 2$ then f has maximal arithmetical subword complexity, $p_f^{\mathrm{AP}}(\ell) = 2^\ell$.

Theorem (K.)

The Thue–Morse sequence is Gowers uniform of all orders. More precisely, for each $d \geq 2$ there exists $\kappa > 0$ such that

$$\|t\|_{U^d[N]} \ll N^{-\kappa}.$$

In particular, $t(n)$ has maximal arithmetical subword complexity, $p_t^{\mathrm{AP}}(\ell) = 2^\ell$.

Remark: The fact that $p_t^{\mathrm{AP}}(\ell) = 2^\ell$ has been proven several times.

Corollary 1: If $f\colon [N] \to \{-1,+1\}$ and $\|f\|_{U^d[N]} \le \varepsilon$ then f looks random along random $(d+1)$-term APs in $[N]$, meaning that for $w \in \{-1,+1\}^{d+1}$ we have:

$$\frac{\#\{(n,m)\,:\,n+im \in [N] \text{ and } f(n+im) = w(i) \text{ for } 0 \le i \le d\}}{\#\{(n,m)\,:\,n+im \in [N] \text{ for } 0 \le i \le d\}} = \frac{1}{2^{d+1}} + O(\varepsilon).$$

Corollary 2: In particular, if $f\colon \mathbb{N} \to \{-1,+1\}$ and $\|f\|_{U^d[N]} \to 0$ as $N \to \infty$ for each $d \ge 2$ then f has maximal arithmetical subword complexity, $p_f^{\mathrm{AP}}(\ell) = 2^\ell$.

Theorem (K.)

The Thue–Morse sequence is Gowers uniform of all orders. More precisely, for each $d \ge 2$ there exists $\kappa > 0$ such that

$$\|t\|_{U^d[N]} \ll N^{-\kappa}.$$

In particular, $t(n)$ has maximal arithmetical subword complexity, $p_t^{\mathrm{AP}}(\ell) = 2^\ell$.

Remark: The fact that $p_t^{\mathrm{AP}}(\ell) = 2^\ell$ has been proven several times.

Corollary 1: If $f\colon [N] \to \{-1,+1\}$ and $\|f\|_{U^d[N]} \leq \varepsilon$ then f looks random along random $(d+1)$-term APs in $[N]$, meaning that for $w \in \{-1,+1\}^{d+1}$ we have:

$$\frac{\#\{(n,m)\ :\ n+im \in [N] \text{ and } f(n+im) = w(i) \text{ for } 0 \leq i \leq d\}}{\#\{(n,m)\ :\ n+im \in [N] \text{ for } 0 \leq i \leq d\}} = \frac{1}{2^{d+1}} + O(\varepsilon).$$

Corollary 2: In particular, if $f\colon \mathbb{N} \to \{-1,+1\}$ and $\|f\|_{U^d[N]} \to 0$ as $N \to \infty$ for each $d \geq 2$ then f has maximal arithmetical subword complexity, $p_f^{\mathrm{AP}}(\ell) = 2^\ell$.

Theorem (K.)

The Thue–Morse sequence is Gowers uniform of all orders. More precisely, for each $d \geq 2$ there exists $\kappa > 0$ such that

$$\|t\|_{U^d[N]} \ll N^{-\kappa}.$$

In particular, $t(n)$ has maximal arithmetical subword complexity, $p_t^{\mathrm{AP}}(\ell) = 2^\ell$.

Remark: The fact that $p_t^{\mathrm{AP}}(\ell) = 2^\ell$ has been proven several times.

Uniformity of automatic sequences

Question: Which automatic sequences are Gowers uniform?

Theorem (Byszewski, K., Müllner)

For an automatic sequence $a\colon \mathbb{N} \to \mathbb{C}$, the following are equivalent:

- $\|a\|_{U^d[N]} \to 0$ *as* $N \to \infty$ *for each* $d \geq 2$*;*
- $\|a\|_{U^2[N]} \to 0$ *as* $N \to \infty$*;*
- $\frac{1}{N}\sum_{n=0}^{N-1} a(An+B) \to 0$ *as* $N \to \infty$ *for each* $A \geq 1,\ B \geq 0$.

Rationale: Only linear obstructions to uniformity, no quadratic structure possible.

Proposition

Let $a\colon \mathbb{N} \to \{0,1\}$ be an automatic sequence. Suppose for some $\alpha \in (0,1)$ we have:

$$\frac{1}{N}\sum_{n=0}^{N-1} a(An+B) \to \alpha \qquad \text{for each } A \geq 1,\ B \geq 0.$$

Then a has maximal arithmetical subword complexity: $p_a^{\mathrm{AP}}(\ell) = 2^{\ell}$.

Rationale: $a - \alpha 1_{\mathbb{N}}$ is Gowers uniform of all orders.

Uniformity of automatic sequences

Question: Which automatic sequences are Gowers uniform?

Theorem (Byszewski, K., Müllner)

For an automatic sequence $a\colon \mathbb{N} \to \mathbb{C}$, the following are equivalent:

- $\|a\|_{U^d[N]} \to 0$ *as* $N \to \infty$ *for each* $d \geq 2$*;*
- $\|a\|_{U^2[N]} \to 0$ *as* $N \to \infty$*;*
- $\frac{1}{N}\sum_{n=0}^{N-1} a(An+B) \to 0$ *as* $N \to \infty$ *for each* $A \geq 1,\ B \geq 0$.

Rationale: Only linear obstructions to uniformity, no quadratic structure possible.

Proposition

Let $a\colon \mathbb{N} \to \{0,1\}$ be an automatic sequence. Suppose for some $\alpha \in (0,1)$ we have:

$$\frac{1}{N}\sum_{n=0}^{N-1} a(An+B) \to \alpha \qquad \text{for each } A \geq 1,\ B \geq 0.$$

Then a has maximal arithmetical subword complexity: $p_a^{\mathrm{AP}}(\ell) = 2^\ell$.

Rationale: $a - \alpha 1_{\mathbb{N}}$ is Gowers uniform of all orders.

Uniformity of automatic sequences

Question: Which automatic sequences are Gowers uniform?

Theorem (Byszewski, K., Müllner)

For an automatic sequence $a\colon \mathbb{N} \to \mathbb{C}$, the following are equivalent:

- $\|a\|_{U^d[N]} \to 0$ *as* $N \to \infty$ *for each* $d \geq 2$*;*
- $\|a\|_{U^2[N]} \to 0$ *as* $N \to \infty$*;*
- $\frac{1}{N}\sum_{n=0}^{N-1} a(An+B) \to 0$ *as* $N \to \infty$ *for each* $A \geq 1,\ B \geq 0$.

Rationale: Only linear obstructions to uniformity, no quadratic structure possible.

Proposition

Let $a\colon \mathbb{N} \to \{0,1\}$ be an automatic sequence. Suppose for some $\alpha \in (0,1)$ we have:

$$\frac{1}{N}\sum_{n=0}^{N-1} a(An+B) \to \alpha \qquad \text{for each } A \geq 1,\ B \geq 0.$$

Then a has maximal arithmetical subword complexity: $p_a^{\mathrm{AP}}(\ell) = 2^\ell$.

Rationale: $a - \alpha 1_{\mathbb{N}}$ is Gowers uniform of all orders.

Non-uniform automatic sequences

If $a = [\text{constant}] + [\text{uniform}]$, we can apply generalised von Neumann.
Question: Which automatic sequences *do not* have this form?

Basic classes of non-uniform sequences:

1. *periodic*, such as $(-1)^n$;
2. *forwards synchronising*, such as $(-1)^{\nu_2(n)}$; $\longrightarrow 2^{\nu_2(n)} \parallel n$;
3. *backwards synchronising*, such as $(-1)^{\lfloor \log_2(n) \rfloor}$.

Definition (Synchronisation)

- An automaton $\mathcal{A} = (S, s_0, \Sigma_k, \delta, \Omega, \tau)$ is *synchronising* if there exists a word $w \in \Sigma_k^*$ which synchronises $\mathcal{A}$ to a state $s \in S$, meaning that:

$$\delta(s', w) = s \qquad \text{for all } s' \in S.$$

- A sequence $a \colon \mathbb{N} \to \Omega$ is *forwards synchronising* if it is computed by a synchronising automaton reading input starting with the *most* significant digit.
- A sequence $a \colon \mathbb{N} \to \Omega$ is *backwards synchronising* if it is computed by a synchronising automaton reading input starting with the *least* significant digit.

Non-uniform automatic sequences

If $a = [\text{constant}] + [\text{uniform}]$, we can apply generalised von Neumann.
Question: Which automatic sequences *do not* have this form?

Basic classes of non-uniform sequences:

1. *periodic*, such as $(-1)^n$;
2. *forwards synchronising*, such as $(-1)^{\nu_2(n)}$; $\longrightarrow 2^{\nu_2(n)} \parallel n$;
3. *backwards synchronising*, such as $(-1)^{\lfloor \log_2(n) \rfloor}$.

Definition (Synchronisation)

- An automaton $\mathcal{A} = (S, s_0, \Sigma_k, \delta, \Omega, \tau)$ is *synchronising* if there exists a word $w \in \Sigma_k^*$ which synchronises $\mathcal{A}$ to a state $s \in S$, meaning that:

$$\delta(s', w) = s \qquad \text{for all } s' \in S.$$

- A sequence $a \colon \mathbb{N} \to \Omega$ is *forwards synchronising* if it is computed by a synchronising automaton reading input starting with the *most* significant digit.
- A sequence $a \colon \mathbb{N} \to \Omega$ is *backwards synchronising* if it is computed by a synchronising automaton reading input starting with the *least* significant digit.

Non-uniform automatic sequences

If $a =$ [constant] + [uniform], we can apply generalised von Neumann.
Question: Which automatic sequences *do not* have this form?

Basic classes of non-uniform sequences:

1. *periodic*, such as $(-1)^n$;
2. *forwards synchronising*, such as $(-1)^{\nu_2(n)}$; $\longrightarrow 2^{\nu_2(n)} \parallel n$;
3. *backwards synchronising*, such as $(-1)^{\lfloor \log_2(n) \rfloor}$.

Definition (Synchronisation)

- An automaton $\mathcal{A} = (S, s_0, \Sigma_k, \delta, \Omega, \tau)$ is *synchronising* if there exists a word $w \in \Sigma_k^*$ which synchronises $\mathcal{A}$ to a state $s \in S$, meaning that:

$$\delta(s', w) = s \qquad \text{for all } s' \in S.$$

- A sequence $a \colon \mathbb{N} \to \Omega$ is *forwards synchronising* if it is computed by a synchronising automaton reading input starting with the *most* significant digit.
- A sequence $a \colon \mathbb{N} \to \Omega$ is *backwards synchronising* if it is computed by a synchronising automaton reading input starting with the *least* significant digit.

Non-uniform automatic sequences

Theorem (Deshouillers, Drmota, Müllner, Shubin & Spiegelhofer)

If $a\colon \mathbb{N} \to \Omega$ is a forwards synchronising automatic sequence then

$$p_a^{\mathrm{AP}}(\ell) = \exp(o(\ell)).$$

In fact, same estimate holds for polynomial subword complexity.

Remark: Similar estimates can be proved for backwards synchronising.

Synchronizing automatic sequences along Piatetski-Shapiro sequences

Arithmetic subword complexity of automatic sequences - part I

Clemens Müllner

TU Wien

Tuesday, January 9, 2024

Clemens Müllner | Arithmetic subword complexity | 9.1. 2024 | 1 / 29

Non-uniform automatic sequences

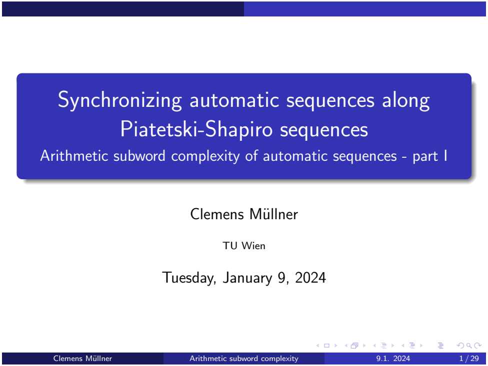
Theorem (Deshouillers, Drmota, Müllner, Shubin & Spiegelhofer)

If $a\colon \mathbb{N} \to \Omega$ is a forwards synchronising automatic sequence then

$$p_a^{\mathrm{AP}}(\ell) = \exp(o(\ell)).$$

In fact, same estimate holds for polynomial subword complexity.

Remark: Similar estimates can be proved for backwards synchronising.



Example: Uniform × Non-uniform

Let $a\colon \mathbb{N} \to \{\pm 1, \pm 2\}$ be defined by

$$a(n) = \begin{cases} t(n') & \text{if } n = 2n', \\ 2t(n') & \text{if } n = 2n' + 1. \end{cases}$$

$$(a(n))_{n=0}^{\infty} = (+1, +2, -1, -2, -1, -2, +1, +2, -1, -2, +1, +2, +1, +2, -1, -2, \\ -1, -2, +1, +2, +1, +2, -1, -2, +1, +2, -1, -2, -1, -2, +1, +2, \ldots)$$

There are three types of factors that appear in $a(n)$ along arithmetic progressions:

- factors (along arithmetic progressions) of $t(n)$;
- factors (along arithmetic progressions) of $2t(n)$;
- shuffles of factors of $t(n)$ and factors of $2t(n)$ (two ways to combine).

We can estimate arithmetical subword complexity of $a(n)$:

$$p_a^{\mathrm{AP}}(\ell) \leq p_t^{\mathrm{AP}}(\ell) + p_{2t}^{\mathrm{AP}}(\ell) + 2 \times p_t^{\mathrm{AP}}(\ell/2) + p_{2t}^{\mathrm{AP}}(\ell/2) \leq 4 \times 2^{\ell}.$$

Takeaway: Passing to arithmetic progressions, we reduced alphabet size to 2.

Example: Uniform × Non-uniform

Let $a\colon \mathbb{N} \to \{\pm 1, \pm 2\}$ be defined by

$$a(n) = \begin{cases} t(n') & \text{if } n = 2n', \\ 2t(n') & \text{if } n = 2n'+1. \end{cases}$$

$$(a(n))_{n=0}^{\infty} = (+1, +2, -1, -2, -1, -2, +1, +2, -1, -2, +1, +2, +1, +2, -1, -2, -1, -2, +1, +2, +1, +2, -1, -2, +1, +2, -1, -2, -1, -2, +1, +2, \ldots)$$

There are three types of factors that appear in $a(n)$ along arithmetic progressions:

- factors (along arithmetic progressions) of $t(n)$;
- factors (along arithmetic progressions) of $2t(n)$;
- shuffles of factors of $t(n)$ and factors of $2t(n)$ (two ways to combine).

We can estimate arithmetical subword complexity of $a(n)$:

$$p_a^{\mathrm{AP}}(\ell) \le p_t^{\mathrm{AP}}(\ell) + p_{2t}^{\mathrm{AP}}(\ell) + 2 \times p_t^{\mathrm{AP}}(\ell/2) + p_{2t}^{\mathrm{AP}}(\ell/2) \le 4 \times 2^{\ell}.$$

Takeaway: Passing to arithmetic progressions, we reduced alphabet size to 2.

Example: Uniform × Non-uniform

Let $a\colon \mathbb{N} \to \{\pm 1, \pm 2\}$ be defined by

$$a(n) = \begin{cases} t(n') & \text{if } n = 2n', \\ 2t(n') & \text{if } n = 2n' + 1. \end{cases}$$

$$(a(n))_{n=0}^{\infty} = (+1, +2, -1, -2, -1, -2, +1, +2, -1, -2, +1, +2, +1, +2, -1, -2, -1, -2, +1, +2, +1, +2, -1, -2, +1, +2, -1, -2, -1, -2, +1, +2, \ldots)$$

There are three types of factors that appear in $a(n)$ along arithmetic progressions:

- factors (along arithmetic progressions) of $t(n)$;
- factors (along arithmetic progressions) of $2t(n)$;
- shuffles of factors of $t(n)$ and factors of $2t(n)$ (two ways to combine).

We can estimate arithmetical subword complexity of $a(n)$:

$$p_a^{\mathrm{AP}}(\ell) \le p_t^{\mathrm{AP}}(\ell) + p_{2t}^{\mathrm{AP}}(\ell) + 2 \times p_t^{\mathrm{AP}}(\ell/2) + p_{2t}^{\mathrm{AP}}(\ell/2) \le 4 \times 2^{\ell}.$$

Takeaway: Passing to arithmetic progressions, we reduced alphabet size to 2.

Effective alphabet size

Let $\mathcal{AP}_k$ denote the family of all sets P of the form

$$P = \left\{ n \in \mathbb{N} : \begin{array}{l} n \equiv r \bmod q, \\ u \text{ is a prefix of } (n)_k, \\ v \text{ is a suffix of } (n)_k, \\ \text{length of } (n)_k \equiv c \bmod \ell \end{array} \right\}.$$

Intuition: Like a residue class, plus base-k information.

Definition (Effective alphabet size)

The *effective alphabet size* of a k-automatic sequence $a \colon \mathbb{N} \to \Omega$ is the largest integer r_a such that there exists $P \in \mathcal{AP}_k$ such that for each $Q \in \mathcal{AP}_k$ with $Q \subset P$ we have

$$\# \{a(n) : n \in Q\} \geq r_a.$$

Remark: In fact, above we have $\# \{a(n) : n \in Q\} = r$.

Effective alphabet size

Let $\mathcal{AP}_k$ denote the family of all sets P of the form

$$P = \left\{ n \in \mathbb{N} : \begin{array}{l} n \equiv r \bmod q, \\ u \text{ is a prefix of } (n)_k, \\ v \text{ is a suffix of } (n)_k, \\ \text{length of } (n)_k \equiv c \bmod \ell \end{array} \right\}.$$

Intuition: Like a residue class, plus base-k information.

Definition (Effective alphabet size)

The *effective alphabet size* of a k-automatic sequence $a \colon \mathbb{N} \to \Omega$ is the largest integer r_a such that there exists $P \in \mathcal{AP}_k$ such that for each $Q \in \mathcal{AP}_k$ with $Q \subset P$ we have

$$\# \{a(n) \,:\, n \in Q\} \geq r_a.$$

Remark: In fact, above we have $\# \{a(n) \,:\, n \in Q\} = r$.

Effective alphabet size

Example

If a is periodic, forwards synchronising, or backwards synchronising then the effective alphabet size of a is $r_a = 1$.

Example

The effective alphabet size of the Thue–Morse sequence is $r_t = 2$.

Basic properties:

- For $a\colon \mathbb{N} \to \Omega$, $b\colon \mathbb{N} \to \Lambda$ we have $r_{a\times b} \leq r_a \times r_b$.
- For $a\colon \mathbb{N} \to \Omega$, $\phi\colon \Omega \to \Lambda$ we have $r_{\phi\circ a} \leq r_a$.

Thus, if a is constructed out of periodic and synchronising sequences then $r_a = 1$.

Fact: For each $\varepsilon > 0$ we can find a finite cover

$$\mathbb{N} = P_1 \cup P_2 \cup \cdots \cup P_N \cup Q_1 \cup Q_2 \cup \cdots \cup Q_M$$

by $P_i, Q_j \in \mathcal{AP}_k$ such that $\#\{a(n) : n \in P_i\} \leq r_a$ and $\bar{d}\,(Q_1 \cup Q_2 \cup \cdots \cup Q_M) < \varepsilon$.

Effective alphabet size

Example

If a is periodic, forwards synchronising, or backwards synchronising then the effective alphabet size of a is $r_a = 1$.

Example

The effective alphabet size of the Thue–Morse sequence is $r_t = 2$.

Basic properties:

- For $a\colon \mathbb{N} \to \Omega$, $b\colon \mathbb{N} \to \Lambda$ we have $r_{a\times b} \le r_a \times r_b$.
- For $a\colon \mathbb{N} \to \Omega$, $\phi\colon \Omega \to \Lambda$ we have $r_{\phi\circ a} \le r_a$.

Thus, if a is constructed out of periodic and synchronising sequences then $r_a = 1$.

Fact: For each $\varepsilon > 0$ we can find a finite cover

$$\mathbb{N} = P_1 \cup P_2 \cup \cdots \cup P_N \cup Q_1 \cup Q_2 \cup \cdots \cup Q_M$$

by $P_i, Q_j \in \mathcal{AP}_k$ such that $\#\{a(n) : n \in P_i\} \le r_a$ and $\bar{d}(Q_1 \cup Q_2 \cup \cdots \cup Q_M) < \varepsilon$.

Effective alphabet size

Example

If a is periodic, forwards synchronising, or backwards synchronising then the effective alphabet size of a is $r_a = 1$.

Example

The effective alphabet size of the Thue–Morse sequence is $r_t = 2$.

Basic properties:

- For $a\colon \mathbb{N} \to \Omega$, $b\colon \mathbb{N} \to \Lambda$ we have $r_{a\times b} \leq r_a \times r_b$.
- For $a\colon \mathbb{N} \to \Omega$, $\phi\colon \Omega \to \Lambda$ we have $r_{\phi\circ a} \leq r_a$.

Thus, if a is constructed out of periodic and synchronising sequences then $r_a = 1$.

Fact: For each $\varepsilon > 0$ we can find a finite cover

$$\mathbb{N} = P_1 \cup P_2 \cup \cdots \cup P_N \cup Q_1 \cup Q_2 \cup \cdots \cup Q_M$$

by $P_i, Q_j \in \mathcal{AP}_k$ such that $\#\{a(n) : n \in P_i\} \leq r_a$ and $\bar{d}(Q_1 \cup Q_2 \cup \cdots \cup Q_M) < \varepsilon$.

Effective alphabet size

Example

If a is periodic, forwards synchronising, or backwards synchronising then the effective alphabet size of a is $r_a = 1$.

Example

The effective alphabet size of the Thue–Morse sequence is $r_t = 2$.

Basic properties:

- For $a\colon \mathbb{N} \to \Omega$, $b\colon \mathbb{N} \to \Lambda$ we have $r_{a\times b} \leq r_a \times r_b$.
- For $a\colon \mathbb{N} \to \Omega$, $\phi\colon \Omega \to \Lambda$ we have $r_{\phi\circ a} \leq r_a$.

Thus, if a is constructed out of periodic and synchronising sequences then $r_a = 1$.

Fact: For each $\varepsilon > 0$ we can find a finite cover

$$\mathbb{N} = P_1 \cup P_2 \cup \cdots \cup P_N \cup Q_1 \cup Q_2 \cup \cdots \cup Q_M$$

by $P_i, Q_j \in \mathcal{AP}_k$ such that $\#\{a(n) : n \in P_i\} \leq r_a$ and $\bar{d}(Q_1 \cup Q_2 \cup \cdots \cup Q_M) < \varepsilon$.

Main result

Theorem (K., Müllner)

Let $a\colon \mathbb{N} \to \Omega$ be an automatic sequence with effective alphabet size r_a. Then

$$p_a^{\mathrm{AP}}(\ell) = (r_a + o(1))^{\ell}.$$

Remarks:

- In fact, in one direction we have exact inequality $p_a^{\mathrm{AP}}(\ell) \geq r_a^{\ell}$.
- We can find $\Lambda \subset \Omega$ with $\#\Lambda = r_a$ such that all words over Λ appear in a along an arithmetic progression.
- Same estimate for polynomial subword complexity.
- The value of r_a is computable, given a representation of a.

Corollary

Let $a\colon \mathbb{N} \to \Omega$ be an automatic sequence with maximal arithmetical subword complexity, i.e. $p_a^{\mathrm{AP}}(\ell) = \#\Omega^{\ell}$. Then for each $\ell \in \mathbb{N}$ and $w \in \Omega^{\ell}$ we have:

$$\liminf_{N\to\infty} \#\left\{(n,m)\,:\, n+im \in [N] \text{ and } a(n+im) = w(i) \text{ for } 0 \leq i < \ell\right\}/N^2 > 0$$

Main result

Theorem (K., Müllner)

Let $a\colon \mathbb{N} \to \Omega$ be an automatic sequence with effective alphabet size r_a. Then

$$p_a^{\mathrm{AP}}(\ell) = (r_a + o(1))^{\ell}.$$

Remarks:

- In fact, in one direction we have exact inequality $p_a^{\mathrm{AP}}(\ell) \geq r_a^{\ell}$.
- We can find $\Lambda \subset \Omega$ with $\#\Lambda = r_a$ such that all words over Λ appear in a along an arithmetic progression.
- Same estimate for polynomial subword complexity.
- The value of r_a is computable, given a representation of a.

Corollary

Let $a\colon \mathbb{N} \to \Omega$ be an automatic sequence with maximal arithmetical subword complexity, i.e. $p_a^{\mathrm{AP}}(\ell) = \#\Omega^{\ell}$. Then for each $\ell \in \mathbb{N}$ and $w \in \Omega^{\ell}$ we have:

$$\liminf_{N\to\infty} \#\left\{(n,m) \,:\, n + im \in [N] \text{ and } a(n+im) = w(i) \text{ for } 0 \leq i < \ell\right\}/N^2 > 0$$

Main result

Theorem (K., Müllner)

Let $a\colon \mathbb{N} \to \Omega$ be an automatic sequence with effective alphabet size r_a. Then

$$p_a^{\mathrm{AP}}(\ell) = (r_a + o(1))^{\ell}.$$

Remarks:

- In fact, in one direction we have exact inequality $p_a^{\mathrm{AP}}(\ell) \geq r_a^{\ell}$.
- We can find $\Lambda \subset \Omega$ with $\#\Lambda = r_a$ such that all words over Λ appear in a along an arithmetic progression.
- Same estimate for polynomial subword complexity.
- The value of r_a is computable, given a representation of a.

Corollary

Let $a\colon \mathbb{N} \to \Omega$ be an automatic sequence with maximal arithmetical subword complexity, i.e. $p_a^{\mathrm{AP}}(\ell) = \#\Omega^{\ell}$. Then for each $\ell \in \mathbb{N}$ and $w \in \Omega^{\ell}$ we have:

$$\liminf_{N\to\infty} \#\left\{(n,m) \,:\, n + im \in [N] \text{ and } a(n+im) = w(i) \text{ for } 0 \leq i < \ell\right\}/N^2 > 0$$

Arithmetic regularity lemma for automatic sequences

Definition (Structured sequences)

A k-automatic sequence $a\colon \mathbb{N} \to \Omega$ is *structured* if there exist automatic sequences $a_{\mathrm{per}}, a_{\mathrm{fs}}, a_{\mathrm{bs}} : \mathbb{N}_0 \to \Omega_{\mathrm{per}}, \Omega_{\mathrm{fs}}, \Omega_{\mathrm{bs}}$ which are periodic, forward synchronising and backward synchronising respectively and a map $F\colon \Omega_{\mathrm{per}} \times \Omega_{\mathrm{fs}} \times \Omega_{\mathrm{bs}} \to \mathbb{C}$ such that

$$a(n) = F\left(a_{\mathrm{per}}(n), a_{\mathrm{fs}}(n), a_{\mathrm{bs}}(n)\right).$$

Theorem (Byszewski, K. & Müllner)

Each automatic sequence $a\colon \mathbb{N} \to \mathbb{C}$ *has a decomposition* $a = a_{\mathrm{str}} + a_{\mathrm{uni}}$, *where*

- a_{str} *is structured in the sense defined above;*
- a_{uni} *is uniform in the sense that for each* $d \geq 2$ *there exists* $\kappa > 0$ *such that*

$$\|a_{\mathrm{uni}}\|_{U^d[N]} \ll N^{-\kappa}.$$

Arithmetic regularity lemma for automatic sequences

Definition (Structured sequences)

A k-automatic sequence $a\colon \mathbb{N} \to \Omega$ is *structured* if there exist automatic sequences $a_{\mathrm{per}}, a_{\mathrm{fs}}, a_{\mathrm{bs}}\colon \mathbb{N}_0 \to \Omega_{\mathrm{per}}, \Omega_{\mathrm{fs}}, \Omega_{\mathrm{bs}}$ which are periodic, forward synchronising and backward synchronising respectively and a map $F\colon \Omega_{\mathrm{per}} \times \Omega_{\mathrm{fs}} \times \Omega_{\mathrm{bs}} \to \mathbb{C}$ such that

$$a(n) = F\left(a_{\mathrm{per}}(n), a_{\mathrm{fs}}(n), a_{\mathrm{bs}}(n)\right).$$

Theorem (Byszewski, K. & Müllner)

Each automatic sequence $a\colon \mathbb{N} \to \mathbb{C}$ has a decomposition $a = a_{\mathrm{str}} + a_{\mathrm{uni}}$, where

- a_{str} *is structured in the sense defined above;*
- a_{uni} *is uniform in the sense that for each $d \geq 2$ there exists $\kappa > 0$ such that*

$$\|a_{\mathrm{uni}}\|_{U^d[N]} \ll N^{-\kappa}.$$

Arithmetic regularity lemma

Theorem (Green & Tao (2010))

Fix $s \geq 1$, $\varepsilon > 0$ and a growth function $\mathcal{F} \colon \mathbb{R}_+ \to \mathbb{R}_+$. Each sequence $a \colon [N] \to [0,1]$ has a decomposition $a = a_{\text{nil}} + a_{\text{sml}} + a_{\text{uni}}$, where $M = O(1)$ and

1. *a_{uni} is uniform in the sense that $\|a_{\text{uni}}\|_{U^{s+1}[N]} \leq 1/\mathcal{F}(M)$.*
2. *a_{sml} is small in the sense that $\|a_{\text{sml}}\|_{L^2[N]} \leq \varepsilon$.*
3. *a_{nil} is a $(\mathcal{F}(M), N)$-irrational virtual degree s nilsequence of complexity $\leq M$.*

Recall: If a is automatic, then $a_{\text{str}}(n) = F\left(a_{\text{per}}(n), a_{\text{fs}}(n), a_{\text{bs}}(n)\right)$, where

- a_{per} is periodic;
- a_{fs} is essentially periodic; $\longrightarrow a_{\text{fs}} = [k^i\text{-periodic}] + O(1/k^{i\eta})$ in $L^2[N]$
- a_{bs} is constant on long intervals.

Hence, $a_{\text{str}} = [\text{1-step nilsequence}] + [\text{small error}]$.

Key differences:

- For automatic sequences, 1-step nilsequences are enough.
- Quantitative bounds in the decomposition are reasonable.

Arithmetic regularity lemma

Theorem (Green & Tao (2010))

Fix $s \geq 1$, $\varepsilon > 0$ and a growth function $\mathcal{F}\colon \mathbb{R}_+ \to \mathbb{R}_+$. Each sequence $a\colon [N] \to [0,1]$ has a decomposition $a = a_{\mathrm{nil}} + a_{\mathrm{sml}} + a_{\mathrm{uni}}$, where $M = O(1)$ and

1. *a_{uni} is uniform in the sense that $\|a_{\mathrm{uni}}\|_{U^{s+1}[N]} \leq 1/\mathcal{F}(M)$.*
2. *a_{sml} is small in the sense that $\|a_{\mathrm{sml}}\|_{L^2[N]} \leq \varepsilon$.*
3. *a_{nil} is a $(\mathcal{F}(M), N)$-irrational virtual degree s nilsequence of complexity $\leq M$.*

Recall: If a is automatic, then $a_{\mathrm{str}}(n) = F\left(a_{\mathrm{per}}(n), a_{\mathrm{fs}}(n), a_{\mathrm{bs}}(n)\right)$, where

- a_{per} is periodic;
- a_{fs} is essentially periodic; $\longrightarrow a_{\mathrm{fs}} = [k^i\text{-periodic}] + O(1/k^{i\eta})$ in $L^2[N]$
- a_{bs} is constant on long intervals.

Hence, $a_{\mathrm{str}} = [\text{1-step nilsequence}] + [\text{small error}]$.

Key differences:

- For automatic sequences, 1-step nilsequences are enough.
- Quantitative bounds in the decomposition are reasonable.

Arithmetic regularity lemma

Theorem (Green & Tao (2010))

Fix $s \geq 1$, $\varepsilon > 0$ and a growth function $\mathcal{F}\colon \mathbb{R}_+ \to \mathbb{R}_+$. Each sequence $a\colon [N] \to [0,1]$ has a decomposition $a = a_{\mathrm{nil}} + a_{\mathrm{sml}} + a_{\mathrm{uni}}$, where $M = O(1)$ and

1. *a_{uni} is uniform in the sense that $\|a_{\mathrm{uni}}\|_{U^{s+1}[N]} \leq 1/\mathcal{F}(M)$.*
2. *a_{sml} is small in the sense that $\|a_{\mathrm{sml}}\|_{L^2[N]} \leq \varepsilon$.*
3. *a_{nil} is a $(\mathcal{F}(M), N)$-irrational virtual degree s nilsequence of complexity $\leq M$.*

Recall: If a is automatic, then $a_{\mathrm{str}}(n) = F\left(a_{\mathrm{per}}(n), a_{\mathrm{fs}}(n), a_{\mathrm{bs}}(n)\right)$, where

- a_{per} is periodic;
- a_{fs} is essentially periodic; $\longrightarrow a_{\mathrm{fs}} = [k^i\text{-periodic}] + O(1/k^{i\eta})$ in $L^2[N]$
- a_{bs} is constant on long intervals.

Hence, $a_{\mathrm{str}} = [\text{1-step nilsequence}] + [\text{small error}]$.

Key differences:

- For automatic sequences, 1-step nilsequences are enough.
- Quantitative bounds in the decomposition are reasonable.

Arithmetic regularity lemma and effective alphabet size

Observation: If $b\colon \mathbb{N} \to \Omega$ is a structured k-automatic sequence and $P \in \mathcal{AP}_k$ then there exists $Q \in \mathcal{AP}_k$ with $Q \subset P$ on which b is constant.

Lemma

Let $a\colon \mathbb{N} \to [0,1]$ be a k-automatic sequence and $P \in \mathcal{AP}_k$. Suppose that for all $Q \in \mathcal{AP}_k$ with $Q \subset P$ there exists $n \in Q$ with $a(n) > 0$. Then there exists $Q \in \mathcal{AP}_k$ with $Q \subset P$ such that a_{str} is constant and positive on Q.

Let $a\colon \mathbb{N} \to \Omega$ be a k-automatic sequence. For $x \in \Omega$, let $a^{(x)}(n) = \begin{cases} 1 & \text{if } a(n) = x, \\ 0 & \text{otherwise.} \end{cases}$

Lemma

Let $a\colon \mathbb{N} \to \Omega$ be a k-automatic sequence with effective alphabet size r_a. Then there exist $\Lambda \subset \Omega$ with $\#\Lambda = r_a$ and $P \in \mathcal{AP}_k$ such that $a^{(x)}_{\text{str}}$ is constant and positive on P for all $x \in \Lambda$.

Proof: Apply the previous lemma repeatedly.

Arithmetic regularity lemma and effective alphabet size

Observation: If $b\colon \mathbb{N} \to \Omega$ is a structured k-automatic sequence and $P \in \mathcal{AP}_k$ then there exists $Q \in \mathcal{AP}_k$ with $Q \subset P$ on which b is constant.

Lemma

Let $a\colon \mathbb{N} \to [0,1]$ be a k-automatic sequence and $P \in \mathcal{AP}_k$. Suppose that for all $Q \in \mathcal{AP}_k$ with $Q \subset P$ there exists $n \in Q$ with $a(n) > 0$. Then there exists $Q \in \mathcal{AP}_k$ with $Q \subset P$ such that a_{str} is constant and positive on Q.

Let $a\colon \mathbb{N} \to \Omega$ be a k-automatic sequence. For $x \in \Omega$, let $a^{(x)}(n) = \begin{cases} 1 & \text{if } a(n) = x, \\ 0 & \text{otherwise.} \end{cases}$

Lemma

Let $a\colon \mathbb{N} \to \Omega$ be a k-automatic sequence with effective alphabet size r_a. Then there exist $\Lambda \subset \Omega$ with $\#\Lambda = r_a$ and $P \in \mathcal{AP}_k$ such that $a^{(x)}_{\mathrm{str}}$ is constant and positive on P for all $x \in \Lambda$.

Proof: Apply the previous lemma repeatedly.

Arithmetic regularity lemma and effective alphabet size

Observation: If $b\colon \mathbb{N} \to \Omega$ is a structured k-automatic sequence and $P \in \mathcal{AP}_k$ then there exists $Q \in \mathcal{AP}_k$ with $Q \subset P$ on which b is constant.

Lemma

Let $a\colon \mathbb{N} \to [0,1]$ be a k-automatic sequence and $P \in \mathcal{AP}_k$. Suppose that for all $Q \in \mathcal{AP}_k$ with $Q \subset P$ there exists $n \in Q$ with $a(n) > 0$. Then there exists $Q \in \mathcal{AP}_k$ with $Q \subset P$ such that a_{str} is constant and positive on Q.

Let $a\colon \mathbb{N} \to \Omega$ be a k-automatic sequence. For $x \in \Omega$, let $a^{(x)}(n) = \begin{cases} 1 & \text{if } a(n) = x, \\ 0 & \text{otherwise.} \end{cases}$

Lemma

Let $a\colon \mathbb{N} \to \Omega$ be a k-automatic sequence with effective alphabet size r_a. Then there exist $\Lambda \subset \Omega$ with $\#\Lambda = r_a$ and $P \in \mathcal{AP}_k$ such that $a_{\mathrm{str}}^{(x)}$ is constant and positive on P for all $x \in \Lambda$.

Proof: Apply the previous lemma repeatedly.

Proof of the lower bound

We are now ready to show that $p_a^{\mathrm{AP}}(\ell) \geq r_a^\ell$ for all $\ell \geq 1$.

- Let $\Lambda \subset \Omega$ and $P \in \mathcal{AP}_k$ be like on the previous slide. Pick any $w \in \Lambda^\ell$. We claim that w appears in a along an arithmetic progression contained in P.
- Let N be a large integer. We will estimate the count of ℓ-term arithmetic progressions in $P \cap [N]$ where w appears in a:

$$C = \sum_{n,m=0}^{N-1} \prod_{i=0}^{\ell-1} 1_{P \cap [N]}(n+im) a^{(w(i))}(n+im).$$

- By generalised von Neumann theorem:

$$C \simeq \sum_{n,m=0}^{N-1} \prod_{i=0}^{\ell-1} 1_{P \cap [N]}(n+im) a_{\mathrm{str}}^{(w(i))}(n+im).$$

- Recall that $a_{\mathrm{str}}^{(w(i))}$ has constant value on P, say $\alpha_{w(i)} > 0$, so

$$C \simeq \prod_{i=0}^{\ell-1} \alpha_{w(i)} \cdot \#\{(n,m) \,:\, n+im \in P \cap [N] \text{ for } 0 \leq i < \ell\} \gg N^2.$$

Proof of the lower bound

We are now ready to show that $p_a^{\mathrm{AP}}(\ell) \geq r_a^\ell$ for all $\ell \geq 1$.

- Let $\Lambda \subset \Omega$ and $P \in \mathcal{AP}_k$ be like on the previous slide. Pick any $w \in \Lambda^\ell$. We claim that w appears in a along an arithmetic progression contained in P.
- Let N be a large integer. We will estimate the count of ℓ-term arithmetic progressions in $P \cap [N]$ where w appears in a:

$$C = \sum_{n,m=0}^{N-1} \prod_{i=0}^{\ell-1} 1_{P \cap [N]}(n+im) a^{(w(i))}(n+im).$$

- By generalised von Neumann theorem:

$$C \simeq \sum_{n,m=0}^{N-1} \prod_{i=0}^{\ell-1} 1_{P \cap [N]}(n+im) a_{\mathrm{str}}^{(w(i))}(n+im).$$

- Recall that $a_{\mathrm{str}}^{(w(i))}$ has constant value on P, say $\alpha_{w(i)} > 0$, so

$$C \simeq \prod_{i=0}^{\ell-1} \alpha_{w(i)} \cdot \# \{(n,m) : n+im \in P \cap [N] \text{ for } 0 \leq i < \ell\} \gg N^2.$$

Proof of the lower bound

We are now ready to show that $p_a^{\mathrm{AP}}(\ell) \geq r_a^\ell$ for all $\ell \geq 1$.

- Let $\Lambda \subset \Omega$ and $P \in \mathcal{AP}_k$ be like on the previous slide. Pick any $w \in \Lambda^\ell$. We claim that w appears in a along an arithmetic progression contained in P.
- Let N be a large integer. We will estimate the count of ℓ-term arithmetic progressions in $P \cap [N]$ where w appears in a:

$$C = \sum_{n,m=0}^{N-1} \prod_{i=0}^{\ell-1} 1_{P\cap[N]}(n+im)a^{(w(i))}(n+im).$$

- By generalised von Neumann theorem:

$$C \simeq \sum_{n,m=0}^{N-1} \prod_{i=0}^{\ell-1} 1_{P\cap[N]}(n+im)a_{\mathrm{str}}^{(w(i))}(n+im).$$

- Recall that $a_{\mathrm{str}}^{(w(i))}$ has constant value on P, say $\alpha_{w(i)} > 0$, so

$$C \simeq \prod_{i=0}^{\ell-1} \alpha_{w(i)} \cdot \#\{(n,m) : n+im \in P \cap [N] \text{ for } 0 \leq i < \ell\} \gg N^2.$$

Proof of the lower bound

We are now ready to show that $p_a^{\mathrm{AP}}(\ell) \geq r_a^\ell$ for all $\ell \geq 1$.

- Let $\Lambda \subset \Omega$ and $P \in \mathcal{AP}_k$ be like on the previous slide. Pick any $w \in \Lambda^\ell$. We claim that w appears in a along an arithmetic progression contained in P.
- Let N be a large integer. We will estimate the count of ℓ-term arithmetic progressions in $P \cap [N]$ where w appears in a:

$$C = \sum_{n,m=0}^{N-1} \prod_{i=0}^{\ell-1} 1_{P\cap[N]}(n+im)a^{(w(i))}(n+im).$$

- By generalised von Neumann theorem:

$$C \simeq \sum_{n,m=0}^{N-1} \prod_{i=0}^{\ell-1} 1_{P\cap[N]}(n+im)a_{\mathrm{str}}^{(w(i))}(n+im).$$

- Recall that $a_{\mathrm{str}}^{(w(i))}$ has constant value on P, say $\alpha_{w(i)} > 0$, so

$$C \simeq \prod_{i=0}^{\ell-1} \alpha_{w(i)} \cdot \#\{(n,m) : n+im \in P\cap[N] \text{ for } 0 \leq i < \ell\} \gg N^2.$$

Proof of the lower bound

We are now ready to show that $p_a^{\mathrm{AP}}(\ell) \geq r_a^\ell$ for all $\ell \geq 1$.

- Let $\Lambda \subset \Omega$ and $P \in \mathcal{AP}_k$ be like on the previous slide. Pick any $w \in \Lambda^\ell$. We claim that w appears in a along an arithmetic progression contained in P.
- Let N be a large integer. We will estimate the count of ℓ-term arithmetic progressions in $P \cap [N]$ where w appears in a:

$$C = \sum_{n,m=0}^{N-1} \prod_{i=0}^{\ell-1} 1_{P \cap [N]}(n+im) a^{(w(i))}(n+im).$$

- By generalised von Neumann theorem:

$$C \simeq \sum_{n,m=0}^{N-1} \prod_{i=0}^{\ell-1} 1_{P \cap [N]}(n+im) a_{\mathrm{str}}^{(w(i))}(n+im).$$

- Recall that $a_{\mathrm{str}}^{(w(i))}$ has constant value on P, say $\alpha_{w(i)} > 0$, so

$$C \simeq \prod_{i=0}^{\ell-1} \alpha_{w(i)} \cdot \# \{(n,m) \,:\, n+im \in P \cap [N] \text{ for } 0 \leq i < \ell\} \gg N^2.$$

Height of a substitution

Let $a\colon \mathbb{N} \to \Omega$ be k-automatic, produced by automaton $\mathcal{A} = (S, s_0, \Sigma_k, \delta, \Omega, \tau)$.

Simplifying assumption: $\mathcal{A}$ is primitive (strongly connected, gcd of loop lengths $= 1$).

Recall: k-automatic $\Leftrightarrow$ coding of a k-uniform substitution.

Suppose $\eta\colon \Omega \to \Omega^k$ be a substitution with $\eta(a) = a$, i.e.:

$$(a(n))_{n=0}^{\infty} = \underbrace{a(0), a(1), \dots, a(k-1)}_{=\eta(a(0))}, \dots \underbrace{a(nk), a(nk+1), \dots, a((n+1)k-1)}_{=\eta(a(n))}, \dots$$

Definition

The height h of η is given by

$$\begin{aligned} h &= \max\{m : m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(d) = a(0)\} \\ &= \max\{m : m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(n+d) = a(n)\} \end{aligned}$$

Example: The Thue–Morse sequence has height 1.

Height of a substitution

Let $a\colon \mathbb{N} \to \Omega$ be k-automatic, produced by automaton $\mathcal{A} = (S, s_0, \Sigma_k, \delta, \Omega, \tau)$.

Simplifying assumption: $\mathcal{A}$ is primitive (strongly connected, gcd *of loop lengths $= 1$).*

Recall: k-automatic $\Leftrightarrow$ coding of a k-uniform substitution.

Suppose $\eta\colon \Omega \to \Omega^k$ be a substitution with $\eta(a) = a$, i.e.:

$$(a(n))_{n=0}^{\infty} = \underbrace{a(0), a(1), \dots, a(k-1)}_{=\eta(a(0))}, \dots \underbrace{a(nk), a(nk+1), \dots, a((n+1)k-1)}_{=\eta(a(n))}, \dots$$

Definition

The height h of η is given by

$$\begin{aligned} h &= \max\{m \,:\, m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(d) = a(0)\} \\ &= \max\{m \,:\, m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(n+d) = a(n)\} \end{aligned}$$

Example: The Thue–Morse sequence has height 1.

Height of a substitution

Let $a\colon \mathbb{N} \to \Omega$ be k-automatic, produced by automaton $\mathcal{A} = (S, s_0, \Sigma_k, \delta, \Omega, \tau)$.

Simplifying assumption: $\mathcal{A}$ is primitive (strongly connected, $\gcd$ of loop lengths $= 1$).

Recall: k-automatic $\Leftrightarrow$ coding of a k-uniform substitution.

Suppose $\eta\colon \Omega \to \Omega^k$ be a substitution with $\eta(a) = a$, i.e.:

$$(a(n))_{n=0}^{\infty} = \underbrace{a(0), a(1), \ldots, a(k-1)}_{=\eta(a(0))}, \ldots \underbrace{a(nk), a(nk+1), \ldots, a((n+1)k-1)}_{=\eta(a(n))}, \ldots$$

Definition

The height h of η is given by

$$\begin{aligned} h &= \max\{m : m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(d) = a(0)\} \\ &= \max\{m : m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(n+d) = a(n)\} \end{aligned}$$

Example: The Thue–Morse sequence has height 1.

Height of a substitution

Let $a\colon \mathbb{N} \to \Omega$ be k-automatic, produced by automaton $\mathcal{A} = (S, s_0, \Sigma_k, \delta, \Omega, \tau)$.

Simplifying assumption: $\mathcal{A}$ is primitive (strongly connected, gcd *of loop lengths $= 1$).*

Recall: k-automatic $\Leftrightarrow$ coding of a k-uniform substitution.

Suppose $\eta\colon \Omega \to \Omega^k$ be a substitution with $\eta(a) = a$, i.e.:

$$(a(n))_{n=0}^{\infty} = \underbrace{a(0), a(1), \ldots, a(k-1)}_{=\eta(a(0))}, \ldots \underbrace{a(nk), a(nk+1), \ldots, a((n+1)k-1)}_{=\eta(a(n))}, \ldots$$

Definition

The height h of η is given by

$$\begin{aligned} h &= \max\{m : m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(d) = a(0)\} \\ &= \max\{m : m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(n+d) = a(n)\} \end{aligned}$$

Example: The Thue–Morse sequence has height 1.

Height of a substitution

Let $a\colon \mathbb{N} \to \Omega$ be k-automatic, produced by automaton $\mathcal{A} = (S, s_0, \Sigma_k, \delta, \Omega, \tau)$.

Simplifying assumption: $\mathcal{A}$ is primitive (strongly connected, $\gcd$ of loop lengths $= 1$).

Recall: k-automatic $\Leftrightarrow$ coding of a k-uniform substitution.

Suppose $\eta\colon \Omega \to \Omega^k$ be a substitution with $\eta(a) = a$, i.e.:

$$(a(n))_{n=0}^{\infty} = \underbrace{a(0), a(1), \dots, a(k-1)}_{=\eta(a(0))}, \dots \underbrace{a(nk), a(nk+1), \dots, a((n+1)k-1)}_{=\eta(a(n))}, \dots$$

Definition

The height h of η is given by

$$\begin{aligned} h &= \max\{m : m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(d) = a(0)\} \\ &= \max\{m : m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(n+d) = a(n)\} \end{aligned}$$

Example: The Thue–Morse sequence has height 1.

Height of a substitution

Let $a\colon \mathbb{N} \to \Omega$ be k-automatic, produced by automaton $\mathcal{A} = (S, s_0, \Sigma_k, \delta, \Omega, \tau)$.

Simplifying assumption: $\mathcal{A}$ is primitive (strongly connected, $\gcd$ of loop lengths $= 1$).

Recall: k-automatic $\Leftrightarrow$ coding of a k-uniform substitution.

Suppose $\eta\colon \Omega \to \Omega^k$ be a substitution with $\eta(a) = a$, i.e.:

$$(a(n))_{n=0}^{\infty} = \underbrace{a(0), a(1), \dots, a(k-1)}_{=\eta(a(0))}, \dots \underbrace{a(nk), a(nk+1), \dots, a((n+1)k-1)}_{=\eta(a(n))}, \dots$$

Definition

The height h of η is given by

$$\begin{aligned} h &= \max\{m : m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(d) = a(0)\} \\ &= \max\{m : m \perp k,\ m \mid d \text{ for all } d \geq 0 \text{ such that } a(n+d) = a(n)\} \end{aligned}$$

Example: The Thue–Morse sequence has height 1.

Height and periodicity

Let $C_j := \{a(n) : n \equiv j \bmod h\}$.

1. For each i, j either $C_j = C_i$ or $C_i \cap C_j = \emptyset$.
2. The sequence $C_0, C_1, C_2, \ldots$ is periodic.
3. The height is the largest integer $h \perp k$ such that the above hold.

The sequence $C_0, C_1, C_2, \ldots$ contains all information about periodic behaviour of $a(n)$ that we would need in the argument.

Simplifying assumption: $h = 1$ (no periodic component).

Height and periodicity

Let $C_j := \{a(n) : n \equiv j \bmod h\}$.

1. For each i, j either $C_j = C_i$ or $C_i \cap C_j = \emptyset$.
2. The sequence $C_0, C_1, C_2, \ldots$ is periodic.
3. The height is the largest integer $h \perp k$ such that the above hold.

The sequence $C_0, C_1, C_2, \ldots$ contains all information about periodic behaviour of $a(n)$ that we would need in the argument.

Simplifying assumption: $h = 1$ (no periodic component).

Height and periodicity

Let $C_j := \{a(n) : n \equiv j \bmod h\}$.

1. For each i, j either $C_j = C_i$ or $C_i \cap C_j = \emptyset$.
2. The sequence $C_0, C_1, C_2, \ldots$ is periodic.
3. The height is the largest integer $h \perp k$ such that the above hold.

The sequence $C_0, C_1, C_2, \ldots$ contains all information about periodic behaviour of $a(n)$ that we would need in the argument.

Simplifying assumption: $h = 1$ (no periodic component).

Group extensions of automata

- Let $c = \min\{\#\delta(S, w) : w \in \Sigma_k^*\}$, where $\delta(S, w) = \{\delta(s, w) : s \in S\}$.
- Let $\mathcal{M} = \{M_0, M_1, \ldots, M_{p-1}\} = \{\delta(S, w) : w \in \Sigma_k^*, \ \#\delta(S, w) = c\}$. Without loss of generality: $s_0 \in M_0$.

Fact: For each $M \in \mathcal{M}$ and $u \in \Sigma_k^*$ we have $\delta(M, u) \in \mathcal{M}$.

- Let $s(n) = \delta(s_0, (n)_k) \in S$ where $n \in \mathbb{N}$.
- Let $i(n) \in \{0, 1, \ldots, p-1\}$ be such that $\delta(M_0, (n)_k) = M_{i(n)}$.

Fact: For each $n \in \mathbb{N}$ we have $s(n) \in M_{i(n)}$.

Proposition

The effective alphabet size is alternatively given by:

$$r_a = \max_{0 \le i < p} \#\tau(M_i).$$

Sketch of proof:

- $r_a \le \max_i \#\tau(M_i)$: If $P \in \mathcal{AP}_k$ then there are $Q \in \mathcal{AP}_k$, $Q \subset P$ and $M \in \mathcal{M}$, such that $\{a(n) : n \in Q\} \subseteq \{\tau(s) : s \in M\}$.
- $r_a \ge \max_i \#\tau(M_i)$: Given i, we can find $P \in \mathcal{AP}_k$ with $i(n) = i$ for all $n \in P$. Remains to show: for each $Q \in \mathcal{AP}_k$, $Q \subset P$ we have $\{s(n) : n \in Q\} = M_i$.

Group extensions of automata

- Let $c = \min \{\#\delta(S,w) : w \in \Sigma_k^*\}$, where $\delta(S,w) = \{\delta(s,w) : s \in S\}$.
- Let $\mathcal{M} = \{M_0, M_1, \ldots, M_{p-1}\} = \{\delta(S,w) : w \in \Sigma_k^*,\ \#\delta(S,w) = c\}$. Without loss of generality: $s_0 \in M_0$.

Fact: For each $M \in \mathcal{M}$ and $u \in \Sigma_k^*$ we have $\delta(M,u) \in \mathcal{M}$.

- Let $s(n) = \delta(s_0, (n)_k) \in S$ where $n \in \mathbb{N}$.
- Let $i(n) \in \{0,1,\ldots,p-1\}$ be such that $\delta(M_0,(n)_k) = M_{i(n)}$.

Fact: For each $n \in \mathbb{N}$ we have $s(n) \in M_{i(n)}$.

Proposition

The effective alphabet size is alternatively given by:

$$r_a = \max_{0 \le i < p} \#\tau(M_i).$$

Sketch of proof:

- $r_a \le \max_i \#\tau(M_i)$: If $P \in \mathcal{AP}_k$ then there are $Q \in \mathcal{AP}_k$, $Q \subset P$ and $M \in \mathcal{M}$, such that $\{a(n) : n \in Q\} \subseteq \{\tau(s) : s \in M\}$.
- $r_a \ge \max_i \#\tau(M_i)$: Given i, we can find $P \in \mathcal{AP}_k$ with $i(n) = i$ for all $n \in P$. Remains to show: for each $Q \in \mathcal{AP}_k$, $Q \subset P$ we have $\{s(n) : n \in Q\} = M_i$.

Group extensions of automata

- Let $c = \min\{\#\delta(S,w) : w \in \Sigma_k^*\}$, where $\delta(S,w) = \{\delta(s,w) : s \in S\}$.
- Let $\mathcal{M} = \{M_0, M_1, \ldots, M_{p-1}\} = \{\delta(S,w) : w \in \Sigma_k^*,\ \#\delta(S,w) = c\}$.
  Without loss of generality: $s_0 \in M_0$.

Fact: For each $M \in \mathcal{M}$ and $u \in \Sigma_k^*$ we have $\delta(M,u) \in \mathcal{M}$.

- Let $s(n) = \delta(s_0, (n)_k) \in S$ where $n \in \mathbb{N}$.
- Let $i(n) \in \{0, 1, \ldots, p-1\}$ be such that $\delta(M_0, (n)_k) = M_{i(n)}$.

Fact: For each $n \in \mathbb{N}$ we have $s(n) \in M_{i(n)}$.

Proposition

The effective alphabet size is alternatively given by:

$$r_a = \max_{0 \le i < p} \#\tau(M_i).$$

Sketch of proof:

- $r_a \le \max_i \#\tau(M_i)$: If $P \in \mathcal{AP}_k$ then there are $Q \in \mathcal{AP}_k$, $Q \subset P$ and $M \in \mathcal{M}$, such that $\{a(n) : n \in Q\} \subseteq \{\tau(s) : s \in M\}$.
- $r_a \ge \max_i \#\tau(M_i)$: Given i, we can find $P \in \mathcal{AP}_k$ with $i(n) = i$ for all $n \in P$. Remains to show: for each $Q \in \mathcal{AP}_k$, $Q \subset P$ we have $\{s(n) : n \in Q\} = M_i$.

Group extensions of automata

- Let $c = \min\{\#\delta(S,w) \,:\, w \in \Sigma_k^*\}$, where $\delta(S,w) = \{\delta(s,w) \,:\, s \in S\}$.
- Let $\mathcal{M} = \{M_0, M_1, \ldots, M_{p-1}\} = \{\delta(S,w) \,:\, w \in \Sigma_k^*,\ \#\delta(S,w) = c\}$. Without loss of generality: $s_0 \in M_0$.

Fact: For each $M \in \mathcal{M}$ and $u \in \Sigma_k^*$ we have $\delta(M,u) \in \mathcal{M}$.

- Let $s(n) = \delta(s_0, (n)_k) \in S$ where $n \in \mathbb{N}$.
- Let $i(n) \in \{0, 1, \ldots, p-1\}$ be such that $\delta(M_0, (n)_k) = M_{i(n)}$.

Fact: For each $n \in \mathbb{N}$ we have $s(n) \in M_{i(n)}$.

Proposition

The effective alphabet size is alternatively given by:

$$r_a = \max_{0 \le i < p} \#\tau(M_i).$$

Sketch of proof:

- $r_a \le \max_i \#\tau(M_i)$: If $P \in \mathcal{AP}_k$ then there are $Q \in \mathcal{AP}_k$, $Q \subset P$ and $M \in \mathcal{M}$, such that $\{a(n) \,:\, n \in Q\} \subseteq \{\tau(s) \,:\, s \in M\}$.
- $r_a \ge \max_i \#\tau(M_i)$: Given i, we can find $P \in \mathcal{AP}_k$ with $i(n) = i$ for all $n \in P$. Remains to show: for each $Q \in \mathcal{AP}_k$, $Q \subset P$ we have $\{s(n) \,:\, n \in Q\} = M_i$.

Group extensions of automata

- Let $c = \min\{\#\delta(S,w) \,:\, w \in \Sigma_k^*\}$, where $\delta(S,w) = \{\delta(s,w) \,:\, s \in S\}$.
- Let $\mathcal{M} = \{M_0, M_1, \ldots, M_{p-1}\} = \{\delta(S,w) \,:\, w \in \Sigma_k^*,\ \#\delta(S,w) = c\}$.
  Without loss of generality: $s_0 \in M_0$.

Fact: For each $M \in \mathcal{M}$ and $u \in \Sigma_k^*$ we have $\delta(M,u) \in \mathcal{M}$.

- Let $s(n) = \delta(s_0, (n)_k) \in S$ where $n \in \mathbb{N}$.
- Let $i(n) \in \{0, 1, \ldots, p-1\}$ be such that $\delta(M_0, (n)_k) = M_{i(n)}$.

Fact: For each $n \in \mathbb{N}$ we have $s(n) \in M_{i(n)}$.

Proposition

The effective alphabet size is alternatively given by:

$$r_a = \max_{0 \le i < p} \#\tau(M_i).$$

Sketch of proof:

- $r_a \le \max_i \#\tau(M_i)$: If $P \in \mathcal{AP}_k$ then there are $Q \in \mathcal{AP}_k$, $Q \subset P$ and $M \in \mathcal{M}$, such that $\{a(n) \,:\, n \in Q\} \subseteq \{\tau(s) \,:\, s \in M\}$.
- $r_a \ge \max_i \#\tau(M_i)$: Given i, we can find $P \in \mathcal{AP}_k$ with $i(n) = i$ for all $n \in P$. Remains to show: for each $Q \in \mathcal{AP}_k$, $Q \subset P$ we have $\{s(n) \,:\, n \in Q\} = M_i$.

Group extensions of automata

- Let $c = \min\{\#\delta(S, w) \,:\, w \in \Sigma_k^*\}$, where $\delta(S, w) = \{\delta(s, w) \,:\, s \in S\}$.
- Let $\mathcal{M} = \{M_0, M_1, \ldots, M_{p-1}\} = \{\delta(S, w) \,:\, w \in \Sigma_k^*,\ \#\delta(S, w) = c\}$. Without loss of generality: $s_0 \in M_0$.

Fact: For each $M \in \mathcal{M}$ and $u \in \Sigma_k^*$ we have $\delta(M, u) \in \mathcal{M}$.

- Let $s(n) = \delta(s_0, (n)_k) \in S$ where $n \in \mathbb{N}$.
- Let $i(n) \in \{0, 1, \ldots, p-1\}$ be such that $\delta(M_0, (n)_k) = M_{i(n)}$.

Fact: For each $n \in \mathbb{N}$ we have $s(n) \in M_{i(n)}$.

Proposition

The effective alphabet size is alternatively given by:

$$r_a = \max_{0 \leq i < p} \#\tau(M_i).$$

Sketch of proof:

- $r_a \leq \max_i \#\tau(M_i)$: If $P \in \mathcal{AP}_k$ then there are $Q \in \mathcal{AP}_k$, $Q \subset P$ and $M \in \mathcal{M}$, such that $\{a(n) \,:\, n \in Q\} \subseteq \{\tau(s) \,:\, s \in M\}$.
- $r_a \geq \max_i \#\tau(M_i)$: Given i, we can find $P \in \mathcal{AP}_k$ with $i(n) = i$ for all $n \in P$. Remains to show: for each $Q \in \mathcal{AP}_k$, $Q \subset P$ we have $\{s(n) \,:\, n \in Q\} = M_i$.

Proof of upper bound

We are now ready to show that $p_a^{\mathrm{AP}}(\ell) \le r_a^\ell \exp(o(\ell))$ for all $\ell \ge 1$.

- Since $s(n) \in M_{i(n)}$ and $\#\tau(M_i) \le r_a$, we have

$$p_a^{\mathrm{AP}}(\ell) \le r_a^\ell \times p_i^{\mathrm{AP}}(\ell). \tag{1}$$

- Since $i\colon \mathbb{N} \to \{0, 1, \dots, p-1\}$ is k-automatic and synchronising, we have

$$p_i^{\mathrm{AP}}(\ell) = \exp(o(\ell)) \tag{2}$$

  by the Deshouillers, Drmota, Müllner, Shubin & Spiegelhofer.
- Combining (1) and (2) yields the claim.

Remarks:

- Recall that we have made simplifying assumptions. $\longrightarrow$ more work
- Same result for polynomial subword complexity.
- The factor $\exp(o(\ell))$ can be improved to $\exp(O(\ell^{1-\kappa}))$ with $\kappa > 0$.

Proof of upper bound

We are now ready to show that $p_a^{\mathrm{AP}}(\ell) \leq r_a^\ell \exp(o(\ell))$ for all $\ell \geq 1$.

- Since $s(n) \in M_{i(n)}$ and $\#\tau(M_i) \leq r_a$, we have

$$p_a^{\mathrm{AP}}(\ell) \leq r_a^\ell \times p_i^{\mathrm{AP}}(\ell). \tag{1}$$

- Since $i \colon \mathbb{N} \to \{0, 1, \ldots, p-1\}$ is k-automatic and synchronising, we have

$$p_i^{\mathrm{AP}}(\ell) = \exp(o(\ell)) \tag{2}$$

  by the Deshouillers, Drmota, Müllner, Shubin & Spiegelhofer.
- Combining (1) and (2) yields the claim.

Remarks:

- Recall that we have made simplifying assumptions. $\longrightarrow$ more work
- Same result for polynomial subword complexity.
- The factor $\exp(o(\ell))$ can be improved to $\exp(O(\ell^{1-\kappa}))$ with $\kappa > 0$.

Proof of upper bound

We are now ready to show that $p_a^{\mathrm{AP}}(\ell) \leq r_a^\ell \exp(o(\ell))$ for all $\ell \geq 1$.

- Since $s(n) \in M_{i(n)}$ and $\#\tau(M_i) \leq r_a$, we have

$$p_a^{\mathrm{AP}}(\ell) \leq r_a^\ell \times p_i^{\mathrm{AP}}(\ell). \tag{1}$$

- Since $i \colon \mathbb{N} \to \{0, 1, \ldots, p-1\}$ is k-automatic and synchronising, we have

$$p_i^{\mathrm{AP}}(\ell) = \exp(o(\ell)) \tag{2}$$

  by the Deshouillers, Drmota, Müllner, Shubin & Spiegelhofer.
- Combining (1) and (2) yields the claim.

Remarks:

- Recall that we have made simplifying assumptions. $\longrightarrow$ more work
- Same result for polynomial subword complexity.
- The factor $\exp(o(\ell))$ can be improved to $\exp(O(\ell^{1-\kappa}))$ with $\kappa > 0$.

Proof of upper bound

We are now ready to show that $p_a^{\mathrm{AP}}(\ell) \leq r_a^\ell \exp(o(\ell))$ for all $\ell \geq 1$.

- Since $s(n) \in M_{i(n)}$ and $\#\tau(M_i) \leq r_a$, we have

$$p_a^{\mathrm{AP}}(\ell) \leq r_a^\ell \times p_i^{\mathrm{AP}}(\ell). \tag{1}$$

- Since $i \colon \mathbb{N} \to \{0, 1, \ldots, p-1\}$ is k-automatic and synchronising, we have

$$p_i^{\mathrm{AP}}(\ell) = \exp(o(\ell)) \tag{2}$$

  by the Deshouillers, Drmota, Müllner, Shubin & Spiegelhofer.
- Combining (1) and (2) yields the claim.

Remarks:

- Recall that we have made simplifying assumptions. $\longrightarrow$ more work
- Same result for polynomial subword complexity.
- The factor $\exp(o(\ell))$ can be improved to $\exp(O(\ell^{1-\kappa}))$ with $\kappa > 0$.

Proof of upper bound

We are now ready to show that $p_a^{\mathrm{AP}}(\ell) \leq r_a^\ell \exp(o(\ell))$ for all $\ell \geq 1$.

- Since $s(n) \in M_{i(n)}$ and $\#\tau(M_i) \leq r_a$, we have

$$p_a^{\mathrm{AP}}(\ell) \leq r_a^\ell \times p_i^{\mathrm{AP}}(\ell). \tag{1}$$

- Since $i\colon \mathbb{N} \to \{0, 1, \ldots, p-1\}$ is k-automatic and synchronising, we have

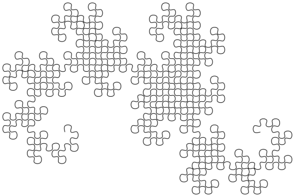
$$p_i^{\mathrm{AP}}(\ell) = \exp(o(\ell)) \tag{2}$$

  by the Deshouillers, Drmota, Müllner, Shubin & Spiegelhofer.
- Combining (1) and (2) yields the claim.

Remarks:

- Recall that we have made simplifying assumptions. $\longrightarrow$ more work
- Same result for polynomial subword complexity.
- The factor $\exp(o(\ell))$ can be improved to $\exp(O(\ell^{1-\kappa}))$ with $\kappa > 0$.

Thank you for your attention!



Bonus: Quantitative Cobham's theorem

Theorem (Cobham (1969))

Let $k, \ell \geq 2$ and let $a \colon \mathbb{N} \to \Omega$ be a sequence that is both k- and ℓ-automatic. Then

- *k and ℓ are multiplicatively dependent, i.e., $\log_k(\ell) \in \mathbb{Q}$; or*
- *a is eventually periodic (and hence automatic in every base).*

Question: How similar can a k-automatic sequence be to an ℓ-automatic sequence?

- We already know that they cannot be equal, or even asymptotically equal.
- We need to account for possible correlations with periodic sequences.

Theorem (Adamczewski, K., Müllner)

Let $k, \ell \geq 2$ be multiplicatively independent integers and let $a, b \colon \mathbb{N} \to \mathbb{C}$ be k- and ℓ-automatic, respectively. Then

$$\sum_{n<N} a(n)b(n) = \sum_{n<N} a_{\mathrm{str}}(n)b_{\mathrm{str}}(n) + O(N^{1-c}).$$

Corollary: Each Gowers uniform k-automatic sequence a is a orthogonal to each ℓ-automatic sequence b,

$$\sum_{n<N} a(n)b(n) = O(N^{1-c}).$$

Bonus: Quantitative Cobham's theorem

Theorem (Cobham (1969))

Let $k, \ell \geq 2$ and let $a\colon \mathbb{N} \to \Omega$ be a sequence that is both k- and ℓ-automatic. Then

- *k and ℓ are multiplicatively dependent, i.e., $\log_k(\ell) \in \mathbb{Q}$; or*
- *a is eventually periodic (and hence automatic in every base).*

Question: How similar can a k-automatic sequence be to an ℓ-automatic sequence?

- We already know that they cannot be equal, or even asymptotically equal.
- We need to account for possible correlations with periodic sequences.

Theorem (Adamczewski, K., Müllner)

Let $k, \ell \geq 2$ be multiplicatively independent integers and let $a, b\colon \mathbb{N} \to \mathbb{C}$ be k- and ℓ-automatic, respectively. Then

$$\sum_{n<N} a(n)b(n) = \sum_{n<N} a_{\text{str}}(n)b_{\text{str}}(n) + O(N^{1-c}).$$

Corollary: Each Gowers uniform k-automatic sequence a is a orthogonal to each ℓ-automatic sequence b,

$$\sum_{n<N} a(n)b(n) = O(N^{1-c}).$$

Bonus: Quantitative Cobham's theorem

Theorem (Cobham (1969))

Let $k, \ell \geq 2$ and let $a\colon \mathbb{N} \to \Omega$ be a sequence that is both k- and ℓ-automatic. Then

- *k and ℓ are multiplicatively dependent, i.e., $\log_k(\ell) \in \mathbb{Q}$; or*
- *a is eventually periodic (and hence automatic in every base).*

Question: How similar can a k-automatic sequence be to an ℓ-automatic sequence?

- We already know that they cannot be equal, or even asymptotically equal.
- We need to account for possible correlations with periodic sequences.

Theorem (Adamczewski, K., Müllner)

Let $k, \ell \geq 2$ be multiplicatively independent integers and let $a, b\colon \mathbb{N} \to \mathbb{C}$ be k- and ℓ-automatic, respectively. Then

$$\sum_{n<N} a(n)b(n) = \sum_{n<N} a_{\mathrm{str}}(n)b_{\mathrm{str}}(n) + O(N^{1-c}).$$

Corollary: Each Gowers uniform k-automatic sequence a is a orthogonal to each ℓ-automatic sequence b,

$$\sum_{n<N} a(n)b(n) = O(N^{1-c}).$$

Bonus: Quantitative Cobham's theorem

Theorem (Cobham (1969))

Let $k, \ell \geq 2$ and let $a\colon \mathbb{N} \to \Omega$ be a sequence that is both k- and ℓ-automatic. Then

- *k and ℓ are multiplicatively dependent, i.e., $\log_k(\ell) \in \mathbb{Q}$; or*
- *a is eventually periodic (and hence automatic in every base).*

Question: How similar can a k-automatic sequence be to an ℓ-automatic sequence?

- We already know that they cannot be equal, or even asymptotically equal.
- We need to account for possible correlations with periodic sequences.

Theorem (Adamczewski, K., Müllner)

Let $k, \ell \geq 2$ be multiplicatively independent integers and let $a, b\colon \mathbb{N} \to \mathbb{C}$ be k- and ℓ-automatic, respectively. Then

$$\sum_{n<N} a(n)b(n) = \sum_{n<N} a_{\mathrm{str}}(n)b_{\mathrm{str}}(n) + O(N^{1-c}).$$

Corollary: Each Gowers uniform k-automatic sequence a is a orthogonal to each ℓ-automatic sequence b,

$$\sum_{n<N} a(n)b(n) = O(N^{1-c}).$$