All I want for Christmas is **an algorithm to detect a Sturmian word accepted by an ω-automaton**

Pierre BÉAUR,

joint works with Benjamin HELLOUIN de MENIBUS

LISN, Université Paris-Saclay

1 - Infinite words

An infinite word

- a finite alphabet $\mathcal{A}$ (e.g. $\{a, b\}$)
- an infinite succession of letters: $abaaabbbaaaaababaabb\ldots$
- the set of all infinite words: $\mathcal{A}^{\mathbb{N}}$

(technically works on $\mathbb{Z}$, but tedious)

1 - Infinite words

An infinite word

- a finite alphabet $\mathcal{A}$ (e.g. $\{a, b\}$)
- an infinite succession of letters: $abaaabbbaaaaababaabb\ldots$
- the set of all infinite words: $\mathcal{A}^{\mathbb{N}}$

(technically works on $\mathbb{Z}$, but tedious)

Factors and complexity of a word

- a factor: $abbba \leq abaa\textcolor{red}{abbba}aaaababaabb\ldots$
- complexity: $p_x(n) = \#\{w \in \mathcal{A}^n \mid w \leq x\}$

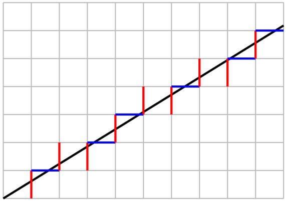
Morse-Hedlund theorem (1938)

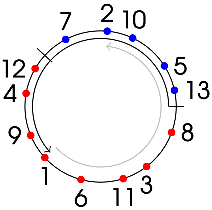
A word x is ultimately periodic iff there is n s.t. $p_x(n) \leq n$.

1- Sturmian words

Objectively the best words

- defined as words x s.t. $p_x(n) = n + 1$ for all n
- the **simplest aperiodic** words
- Fibonacci word: $abaababaabaababaababaabaababaabaab\ldots$
- many combinatorial and dynamical interpretations





2 - ω-automata

$$\frac{\text{finite automata} \implies \text{finite words}}{\omega\text{-automata} \implies \text{infinite words}}$$

2 - ω-automata

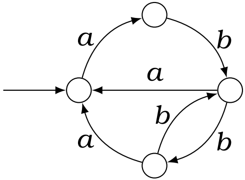
finite automata $\Longrightarrow$ finite words

ω-automata $\Longrightarrow$ infinite words

example: weak ω-automata, i.e. labeled graphs

$\mathfrak{A} = \langle Q, I, T \rangle$ with initial states I (no final state);

a word w is **accepted** if it labels an infinite walk on $\mathfrak{A}$



$$w = abbbba\ abbbba\ abbbba\ abbbba \ldots$$

Language of an ω-automaton

$\mathcal{L}_\infty(\mathfrak{A})$ = the set of infinite words accepted by $\mathfrak{A}$

2 - joining the two: decision problems

$\mathfrak{A}$ an ω-automaton, $\mathcal{L}_\infty(\mathfrak{A})$ its language

Problem 0

Is $\mathcal{L}_\infty(\mathfrak{A})$ non-empty?

2 - joining the two: decision problems

$\mathfrak{A}$ an ω-automaton, $\mathcal{L}_\infty(\mathfrak{A})$ its language

Problem 0

Is $\mathcal{L}_\infty(\mathfrak{A})$ non-empty?

Equivalent problem

Does $\mathcal{L}_\infty(\mathfrak{A})$ contain a (ult.) periodic word?

Trivial

Decidable: find a cycle.

2 - joining the two: decision problems

$\mathfrak{A}$ an ω-automaton, $\mathcal{L}_\infty(\mathfrak{A})$ its language

Problem 0

Is $\mathcal{L}_\infty(\mathfrak{A})$ non-empty?

Equivalent problem

Does $\mathcal{L}_\infty(\mathfrak{A})$ contain a (ult.) periodic word?

Trivial

Decidable: find a cycle.

Aperiodic walk

Does $\mathcal{L}_\infty(\mathfrak{A})$ contain an aperiodic word?

Less trivial, folk.

Decidable: check every cycle.

2 - joining the two: decision problems

$\mathfrak{A}$ an ω-automaton, $\mathcal{L}_\infty(\mathfrak{A})$ its language

Problem 0

Is $\mathcal{L}_\infty(\mathfrak{A})$ non-empty?

Equivalent problem

Does $\mathcal{L}_\infty(\mathfrak{A})$ contain a (ult.) periodic word?

Trivial

Decidable: find a cycle.

Aperiodic walk

Does $\mathcal{L}_\infty(\mathfrak{A})$ contain an aperiodic word?

Less trivial, folk.

Decidable: check every cycle.

Fibonacci walk

Does $\mathcal{L}_\infty(\mathfrak{A})$ contain the Fibo. word?

Sturmian walk

Does $\mathcal{L}_\infty(\mathfrak{A})$ contain a Sturmian word?

The rest of this presentation

In the weak case, decidable; in the Büchi case, probably as well!

3 - Defining Sturmian words with substitutions

The four elementary Sturmian substitutions

$$L_a : \begin{cases} a \mapsto a \\ b \mapsto ab \end{cases} \quad L_b : \begin{cases} a \mapsto ba \\ b \mapsto b \end{cases} \quad R_a : \begin{cases} a \mapsto a \\ b \mapsto ba \end{cases} \quad R_b : \begin{cases} a \mapsto ab \\ b \mapsto b \end{cases}$$

Infinitely desubstitutable words ($\sim$ S-adic representations)

w is inf. desubstitutable by (σ_n) if for all n, $w = \sigma_0 \circ \cdots \circ \sigma_n(w_n)$ (for some w_n).

Directive sequences for Sturmian words (Arnoux, Rauzy, 1991)

With $\mathcal{S}_{St} = \{L_a, R_a, L_b, R_b\}$:

w is Sturmian

$\iff$

w is infinitely desubstitutable by $(\sigma_n) \subseteq \mathcal{S}_{St}$,
and (σ_n) alternates inf. between $\{L_a, R_a\}$ and $\{L_b, R_b\}$

3 - An example!

$$Fib = \quad a\ b\ a\ a\ b\ a\ b\ a\ a\ b\ a\ a\ b\ a\ b\ a\ a\ b\ a\ b\ a\ a\ b \cdots$$

$$L_a : \begin{cases} a \mapsto a \\ b \mapsto ab \end{cases} \qquad L_b : \begin{cases} a \mapsto ba \\ b \mapsto b \end{cases}$$

3 - An example!

$$Fib = \quad a\ b\ a\ a\ b\ a\ b\ a\ a\ b\ a\ a\ b\ a\ b\ a\ a\ b\ a\ b\ a\ a\ b \cdots \quad \big) L_a^{-1}$$

$$L_a : \begin{cases} a \mapsto a \\ b \mapsto ab \end{cases} \qquad L_b : \begin{cases} a \mapsto ba \\ b \mapsto b \end{cases}$$

3 - An example!

$$Fib \;=\; \underbrace{a\,b}\,\underbrace{a}\,\underbrace{a\,b}\,\underbrace{a\,b}\,\underbrace{a}\,\underbrace{a\,b}\,\underbrace{a}\,\underbrace{a\,b}\,\underbrace{a\,b}\,\underbrace{a}\,\underbrace{a\,b}\,\underbrace{a\,b}\,\underbrace{a}\,\underbrace{a\,b}\cdots \quad \big\downarrow L_a^{-1}$$

$$L_a : \begin{cases} a \mapsto a \\ b \mapsto ab \end{cases} \qquad\qquad L_b : \begin{cases} a \mapsto ba \\ b \mapsto b \end{cases}$$

$$Fib = a\,b\,a\,a\,b\,a\,b\,a\,a\,b\,a\,a\,b\,a\,b\,a\,a\,b\,a\,b\,a\,a\,b \cdots$$

$$\ b\ \ a\ \ b\ \ b\ \ a\ \ b\ \ a\ \ b\ \ b\ \ a\ \ b\ \ b\ \ a\ \ b \cdots \qquad \circlearrowright L_a^{-1}$$

$$L_a : \begin{cases} a \mapsto a \\ b \mapsto ab \end{cases} \qquad\qquad L_b : \begin{cases} a \mapsto ba \\ b \mapsto b \end{cases}$$

3 - An example!

$$Fib = \underbrace{a\,b}\underbrace{a}\underbrace{a\,b}\underbrace{a\,b}\underbrace{a}\underbrace{a\,b}\underbrace{a}\underbrace{a\,b}\underbrace{a\,b}\underbrace{a}\underbrace{a\,b}\underbrace{a\,b}\underbrace{a}\underbrace{a\,b}\cdots$$

$$b\ a\ b\ b\ a\ b\ a\ b\ b\ a\ b\ b\ a\ b \cdots$$

L_a^{-1}, L_b^{-1}

$$L_a : \begin{cases} a \mapsto a \\ b \mapsto ab \end{cases} \qquad L_b : \begin{cases} a \mapsto ba \\ b \mapsto b \end{cases}$$

3 - An example!

$$Fib = a\,b\,a\,a\,b\,a\,b\,a\,a\,b\,a\,a\,b\,a\,b\,a\,a\,b\,a\,b\,a\,a\,b\cdots \quad \big\downarrow L_a^{-1}$$

$$b\,a\,b\,b\,a\,b\,a\,b\,b\,a\,b\,b\,a\,b\cdots \quad \big\downarrow L_b^{-1}$$

$$L_a : \begin{cases} a \mapsto a \\ b \mapsto ab \end{cases} \qquad L_b : \begin{cases} a \mapsto ba \\ b \mapsto b \end{cases}$$

$$\begin{array}{ll} Fib = & a\ b\ a\ a\ b\ a\ b\ a\ a\ b\ a\ a\ b\ a\ b\ a\ a\ b\ a\ b\ a\ a\ b \cdots \\ & \quad \big\downarrow L_a^{-1} \\ & b\ a\ b\ b\ a\ b\ a\ b\ b\ a\ b\ b\ a\ b \cdots \\ & \quad \big\downarrow L_b^{-1} \\ & a\ b\ a\ a\ b\ a\ b\ a\ a \cdots \end{array}$$

$$L_a : \begin{cases} a \mapsto a \\ b \mapsto ab \end{cases} \qquad L_b : \begin{cases} a \mapsto ba \\ b \mapsto b \end{cases}$$

3 - An example!

$$
\begin{aligned}
Fib &= a\,b\,a\,a\,b\,a\,b\,a\,a\,b\,a\,a\,b\,a\,b\,a\,a\,b\,a\,b\,a\,a\,b\cdots \\
& \quad b\;\;a\;\;b\;\;\;b\;\;a\;\;b\;\;a\;\;b\;\;\;b\;\;a\;\;b\;\;\;b\;\;a\;\;b\cdots \qquad \big\downarrow L_a^{-1} \\
Fib &= \quad a\quad b\quad a\quad a\quad b\quad a\quad b\quad a\quad a\cdots \qquad \big\downarrow L_b^{-1}
\end{aligned}
$$

$$
L_a : \begin{cases} a \mapsto a \\ b \mapsto ab \end{cases} \qquad\qquad L_b : \begin{cases} a \mapsto ba \\ b \mapsto b \end{cases}
$$

3 - An example!

$$Fib = a\ b\ a\ a\ b\ a\ b\ a\ a\ b\ a\ a\ b\ a\ b\ a\ a\ b\ a\ b\ a\ a\ b \cdots$$

$\Big\downarrow L_a^{-1}$

$$b\ a\ b\ \ b\ a\ b\ a\ b\ \ b\ a\ b\ \ b\ a\ b \cdots$$

$\Big\downarrow L_b^{-1}$

$$Fib = a\ b\ a\ a\ b\ a\ b\ a\ a \cdots$$

$\Big\downarrow L_a^{-1}$

$$b\ a\ b\ b\ a\ b \cdots$$

$$L_a : \begin{cases} a \mapsto a \\ b \mapsto ab \end{cases} \qquad L_b : \begin{cases} a \mapsto ba \\ b \mapsto b \end{cases}$$

3 - . . . why?

What about other classical definitions?

Classical definitions:

- with complexity function;
- with mechanical words;
- with interval exchanges; . . .

$\Rightarrow$ none are algorithmically practical!

What have we gained?

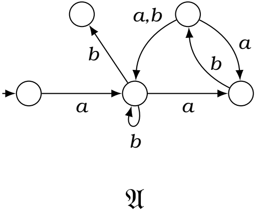
- substitutions: easier to make algorithms with;
- the language of directive sequences is simpler than the language of Sturmian words themselves

A classical result of formal language theory

Regular languages are stable under inverse morphism.

A classical result of formal language theory

Regular languages are stable under inverse morphism.



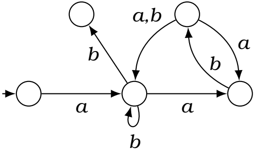
$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$

For weak ω-automata, desubstitution:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

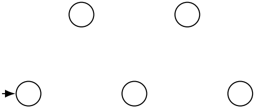
A classical result of formal language theory

Regular languages are stable under inverse morphism.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



$\sigma^{-1}(\mathfrak{A})$

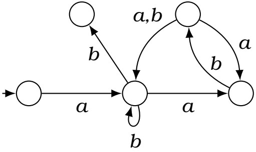
For weak ω-automata, desubstitution:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

3 - Substitutions and weak ω-automata

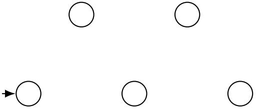
A classical result of formal language theory

Regular languages are stable under inverse morphism.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



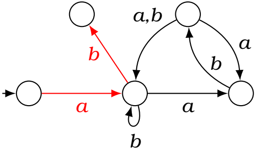
$\sigma^{-1}(\mathfrak{A})$

For weak ω-automata, desubstitution:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

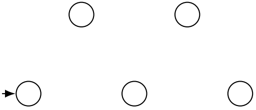
A classical result of formal language theory

Regular languages are stable under inverse morphism.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



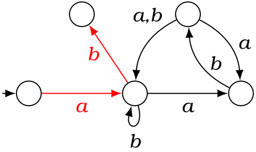
$\sigma^{-1}(\mathfrak{A})$

For weak ω-automata, **desubstitution**:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

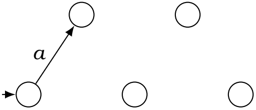
A classical result of formal language theory

Regular languages are stable under inverse morphism.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



$\sigma^{-1}(\mathfrak{A})$

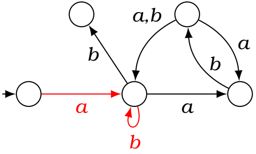
For weak ω-automata, **desubstitution**:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

3 - Substitutions and weak ω-automata

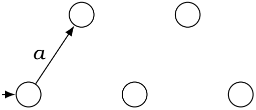
A classical result of formal language theory

Regular languages are stable under inverse morphism.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



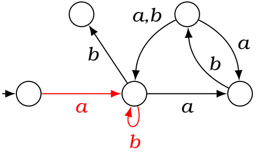
$\sigma^{-1}(\mathfrak{A})$

For weak ω-automata, **desubstitution**:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

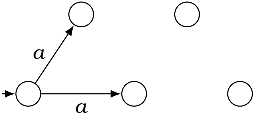
A classical result of formal language theory

Regular languages are stable under inverse morphism.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



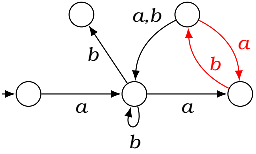
$\sigma^{-1}(\mathfrak{A})$

For weak ω-automata, **desubstitution**:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

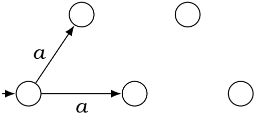
A classical result of formal language theory

Regular languages are stable under inverse morphism.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



$\sigma^{-1}(\mathfrak{A})$

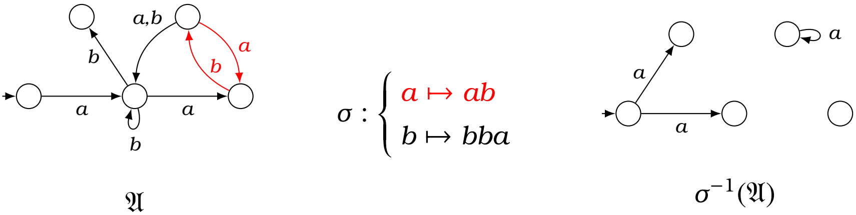
For weak ω-automata, **desubstitution**:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

3 - Substitutions and weak ω-automata

A classical result of formal language theory

Regular languages are stable under inverse morphism.

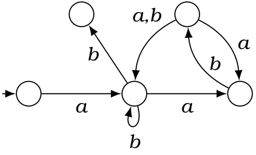


For weak ω-automata, **desubstitution**:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

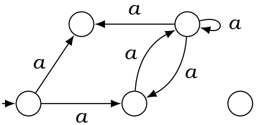
A classical result of formal language theory

Regular languages are stable under inverse morphism.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



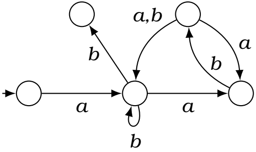
$\sigma^{-1}(\mathfrak{A})$

For weak ω-automata, **desubstitution**:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

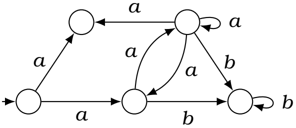
A classical result of formal language theory

Regular languages are stable under inverse morphism.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



$\sigma^{-1}(\mathfrak{A})$

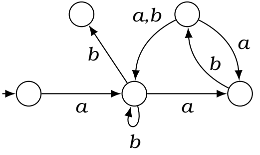
For weak ω-automata, **desubstitution**:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

3 - Substitutions and weak ω-automata

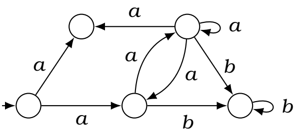
A classical result of formal language theory

Regular languages are stable under inverse morphism.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



$\sigma^{-1}(\mathfrak{A})$

For weak ω-automata, **desubstitution**:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

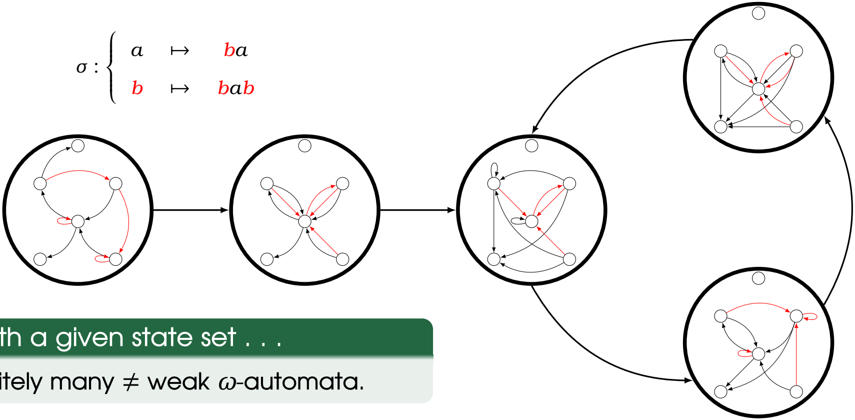
Key remark

The number of states is **constant** under desubstitution.

Key remark

The number of states is **constant** under desubstitution.

$$\sigma : \begin{cases} a & \mapsto & ba \\ b & \mapsto & bab \end{cases}$$



With a given state set . . .

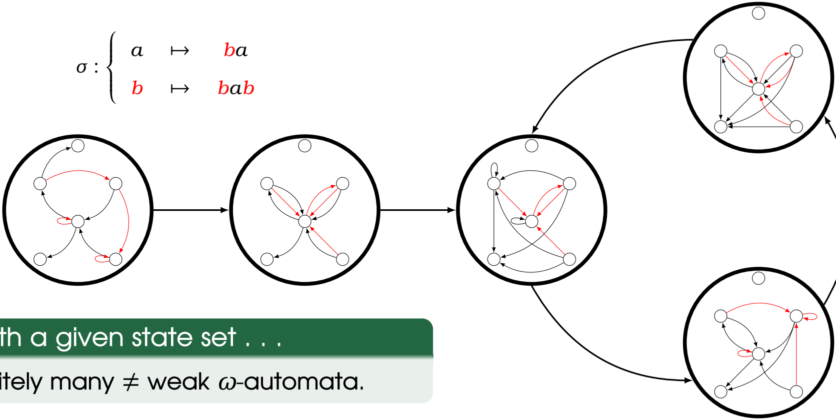
Finitely many $\neq$ weak ω-automata.

3 - The key remark

Key remark

The number of states is **constant** under desubstitution.

$$\sigma : \begin{cases} a \mapsto ba \\ b \mapsto bab \end{cases}$$



With a given state set . . .

Finitely many $\neq$ weak ω-automata.

B., Hellouin (2023)

For $\mathfrak{A}$ and σ, there is a **desubstitution graph** describing every possible desub.

3 - First result: finding a Fibonacci walk

Fibonacci walk

Does $\mathcal{L}_\infty(\mathfrak{A})$ contain the Fibonacci word?

Carton, Thomas (2001), Salo (2022), B., Hellouin (2023)

Decidable.

- build the desubstitution graph for ϕ the Fibonacci substitution;
- one of the following is true:
  - the whole loop has empty language $\implies \exists n, \phi^n(a) \notin \mathfrak{A} \implies$ NO
  - the whole loop has a true language $\implies \forall n, \phi^n(a) \in \mathfrak{A} \implies$ YES

3 - First result: finding a Fibonacci walk

Fibonacci walk

Does $\mathcal{L}_\infty(\mathfrak{A})$ contain the Fibonacci word?

Carton, Thomas (2001), Salo (2022), B., Hellouin (2023)

Decidable.

- build the desubstitution graph for ϕ the Fibonacci substitution;
- one of the following is true:
  - the whole loop has empty language $\implies \exists n, \phi^n(a) \notin \mathfrak{A} \implies$ NO
  - the whole loop has a true language $\implies \forall n, \phi^n(a) \in \mathfrak{A} \implies$ YES

Generalization: Carton, Thomas (2001), B., Hellouin (2023)

Given $\mathfrak{A}$ and σ, we decide whether $\mathfrak{A}$ accepts the sub. word generated by σ.

Proof

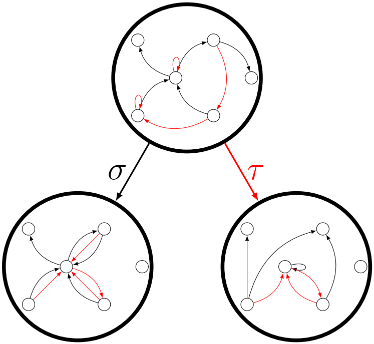
Gestion of growing letters, and topology: $\mathcal{L}_\infty(\mathfrak{A})$ is closed.

Extension: B., Hellouin (2023)

Given $\mathfrak{A}$ and σ, we decide whether $\mathfrak{A}$ accepts a fixed point of σ.

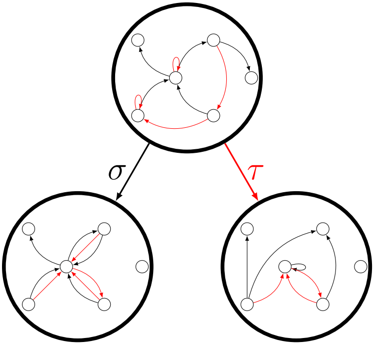
4 - The infinitely desubstitutable case

Why not do the same for multiple substitutions?



4 - The infinitely desubstitutable case

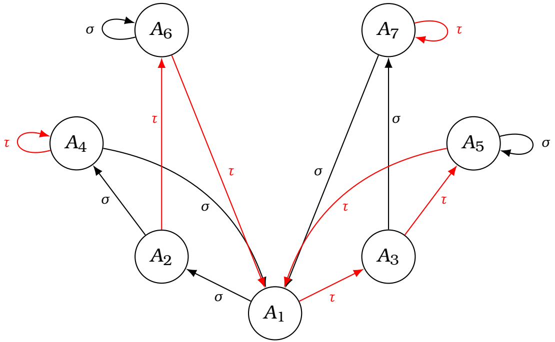
Why not do the same for multiple substitutions?



The key remark stands

The number of states is constant under desubstitution **no matter the substitution**.

4 - A meta ω-automaton



- A_i are weak ω-automata
- $A_i \xrightarrow{\varphi} A_j$

  $\Longleftrightarrow$

  $A_j = \varphi^{-1}(A_i)$

A new meta-ω-automaton

For $\mathfrak{A}$ and a set of substitutions $\mathcal{S}$, we build a meta-ω-automaton $\mathcal{S}^{-\infty}(\mathfrak{A})$ that describes the possible infinite desubstitutions with $\mathcal{S}$.

4 - Second results: the Sturmian case

B., Hellouin (2023)

Given $\mathfrak{A}$ a weak ω-automaton, it is decidable whether $\mathfrak{A}$ accepts a Sturmian word.

- build $\mathcal{S}_{St}^{-\infty}(\mathfrak{A})$
- find an infinite walk with no empty ω-aut. in it
- (it must also inf. alternate between $\{L_a, R_a\}$ and $\{L_b, R_b\}$)

4 - Second results: the Sturmian case

B., Hellouin (2023)

Given $\mathfrak{A}$ a weak ω-automaton, it is decidable whether $\mathfrak{A}$ accepts a Sturmian word.

- build $\mathcal{S}_{St}^{-\infty}(\mathfrak{A})$
- find an infinite walk with no empty ω-aut. in it
- (it must also inf. alternate between $\{L_a, R_a\}$ and $\{L_b, R_b\}$)

Generalization

Given $\mathfrak{A}$ and $\mathcal{S}$, we decide if there is an inf. desub. walk generated by $\mathcal{S}$ in $\mathfrak{A}$, even with ω-regular conditions on the directive sequence.

Similar consequences

We decide the existence of a walk labeled by:

- an Arnoux-Rauzy word (Arnoux, Rauzy, 1991);
- a minimal ternary dendric word (Gheeraert, Lejeune, Leroy, 2021).

4 - Some consequences

Allowed directive sequences in $\mathfrak{A}$ form an ω-regular language, so:

Structural consequences

- they form a closed subset of $\mathcal{S}^{\mathbb{N}}$;
- they contain a periodic sequence (if non empty).

4 - Some consequences

Allowed directive sequences in $\mathfrak{A}$ form an ω-regular language, so:

Structural consequences

- they form a closed subset of $\mathcal{S}^{\mathbb{N}}$;
- they contain a periodic sequence (if non empty).

Another application:

Codings of Sturmian words

Given $w_1, \ldots, w_k$ k finite words, does $(w_1 + \cdots + w_k)^\omega$ contain a Sturmian?

Decidable!

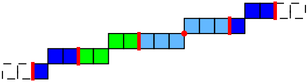
Decidable: just consider the equivalent ω-automaton.

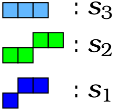
Combinatorial charac. for $k = 2$ (with Richomme), unknown for $k \geq 3$.

4 - Consequence in discrete geometry

Gluing discrete segments

Given some discrete segments $s_1, \ldots, s_n$, is there a way to concatenate copies of them to form an infinite discrete line?





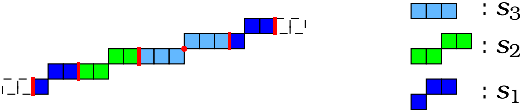
Regluing discrete surfaces

Given some discrete surfaces $s_1, \ldots, s_n$, is there a way to concatenate copies of them to form an infinite a discrete plane ?

4 - Consequence in discrete geometry

Gluing discrete segments

Given some discrete segments $s_1, \ldots, s_n$, is there a way to concatenate copies of them to form an infinite discrete line?



Regluing discrete surfaces

Given some discrete surfaces $s_1, \ldots, s_n$, is there a way to concatenate copies of them to form an infinite a discrete plane ?

A dimensional difference

Decidable for lines, undecidable for surfaces.

Proof

- for lines: find a Sturmian / Arnoux-Rauzy walk in a flower ω-automaton.
- for surfaces: reduction to the domino problem.

A little break

From weak ω-automata . . .
to Büchi automata!

(careful, the paint is fresh)

5 - another model of ω-automaton

- many results on weak ω-automata, cool!
- can we extend to a stronger model of ω-automata?

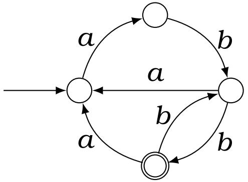
5 - another model of ω-automaton

- many results on weak ω-automata, cool!
- can we extend to a stronger model of ω-automata?

another example of ω-automata: Büchi automata

$\mathfrak{A} = \langle Q, I, F, T \rangle$ with initial states I and accepting states F;

a word w is **accepted** if it labels an infinite walk on $\mathfrak{A}$ which goes infinitely by F



$$w = abbbba\ abbbba\ abbbba \ldots : \checkmark$$
$$w' = aba\ aba\ aba\ aba\ aba \ldots : \times$$

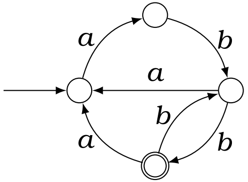
5 - another model of ω-automaton

- many results on weak ω-automata, cool!
- can we extend to a stronger model of ω-automata?

another example of ω-automata: Büchi automata

$\mathfrak{A} = \langle Q, I, F, T \rangle$ with initial states I and accepting states F;

a word w is **accepted** if it labels an infinite walk on $\mathfrak{A}$ which goes infinitely by F



$w = abbbba\ abbbba\ abbbba \ldots$: ✓

$w' = aba\ aba\ aba\ aba\ aba \ldots$: ✗

Büchi automata vs. weak ω-automata

Weak ω-automata can be simulated by Büchi automata, but not the converse: Büchi automata are **stronger** than weak ω-automata.

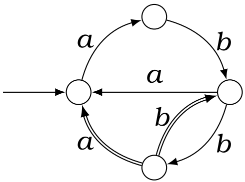
5 - about Büchi automata

- simpler for today: **edge** Büchi automata

edge Büchi automata

$\mathfrak{A} = \langle Q, I, T, F \rangle$ with initial states I and accepting **edges** F;

a word w is **accepted** if it labels an infinite walk on $\mathfrak{A}$ which goes infinitely by F



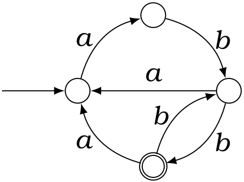
5 - about Büchi automata

- simpler for today: **edge** Büchi automata

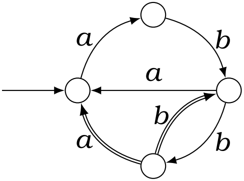
edge Büchi automata

$\mathfrak{A} = \langle Q, I, T, F \rangle$ with initial states I and accepting **edges** F;

a word w is **accepted** if it labels an infinite walk on $\mathfrak{A}$ which goes infinitely by F



$\Longleftrightarrow$



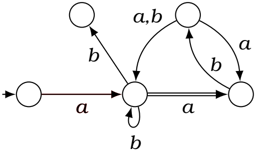
Equivalence with edge Büchi automata

(Vertex) Büchi automata are equivalent to edge Büchi automata.

5 - Desubstituting Büchi automata

For Büchi automata, desubstitution:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.

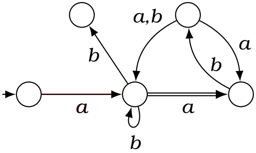


$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$

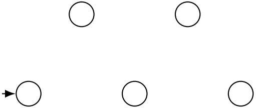
For Büchi automata, desubstitution:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.



$\mathfrak{A}$

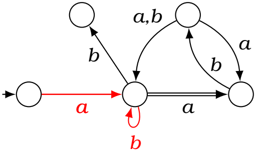
$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



$\sigma^{-1}(\mathfrak{A})$

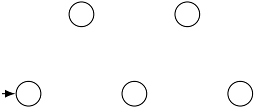
For Büchi automata, desubstitution:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.



$\mathfrak{A}$

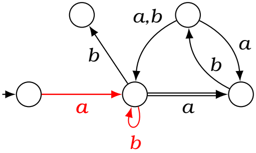
$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



$\sigma^{-1}(\mathfrak{A})$

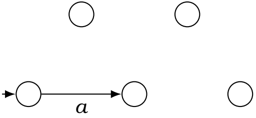
For Büchi automata, desubstitution:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$

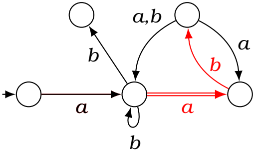


$\sigma^{-1}(\mathfrak{A})$

5 - Desubstituting Büchi automata

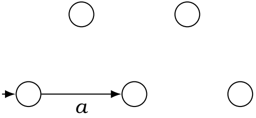
For Büchi automata, desubstitution:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.



$\mathfrak{A}$

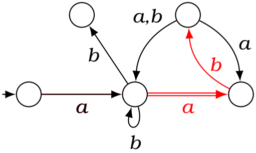
$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



$\sigma^{-1}(\mathfrak{A})$

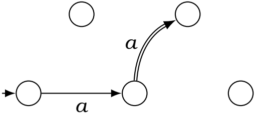
For Büchi automata, desubstitution:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.



$\mathfrak{A}$

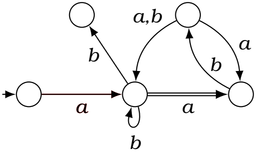
$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



$\sigma^{-1}(\mathfrak{A})$

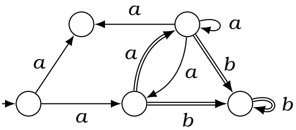
For Büchi automata, desubstitution:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.



$\mathfrak{A}$

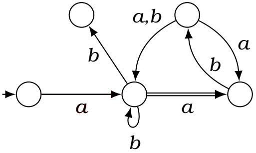
$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



$\sigma^{-1}(\mathfrak{A})$

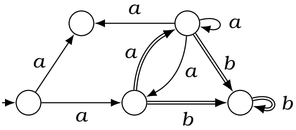
For Büchi automata, desubstitution:

Given $\mathfrak{A}$ and σ, can build $\sigma^{-1}(\mathfrak{A})$ s.t. $\mathcal{L}_\infty(\sigma^{-1}(\mathfrak{A})) = \{w \mid \sigma(w) \in \mathcal{L}_\infty(\mathfrak{A})\}$.



$\mathfrak{A}$

$$\sigma : \begin{cases} a \mapsto ab \\ b \mapsto bba \end{cases}$$



$\sigma^{-1}(\mathfrak{A})$

B., Hellouin (ongoing work)

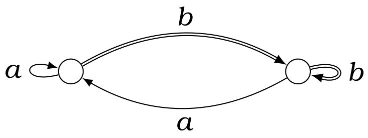
Given $\mathfrak{A}$ Büchi and σ, can build a desubstitution graph; given $\mathcal{S}$ a set of substitutions, can build a meta-ω-automaton.

- Great! Same structures, so same algorithms . . . right?

Oopsie daisy

There exists $\mathfrak{A}$ a Büchi automata and σ such that:

- $\mathfrak{A} = \sigma^{-1}(\mathfrak{A})$ and $\mathcal{L}_\infty(\mathfrak{A}) \neq \emptyset$;
- but $\sigma^\infty(b) \notin \mathcal{L}_\infty(\mathfrak{A})$



$$\sigma : \begin{cases} a \mapsto a \\ b \mapsto aabaab \end{cases}$$

$$\sigma^\infty(b) = a^\infty$$

$$\sigma^{-1}(\mathfrak{A}) = \mathfrak{A}, \text{ but } \sigma^\infty(b) \notin \mathcal{L}_\infty(\mathfrak{A})$$

Why does this technique fail?

Topology: $\mathcal{L}_\infty(\mathfrak{A})$ is not necessarily closed.

$\Rightarrow$ need to be careful with the topology of the language

5 - on deterministic Büchi automata

$\Rightarrow$ my purpose: deciding whether $\mathcal{L}_\infty(\mathfrak{A})$ contains a Sturmian word

Carton, Thomas (2002)

Given $\mathfrak{A}$ a Büchi automaton and σ, we can decide whether $\sigma^\infty(a) \in \mathcal{L}_\infty(\mathfrak{A})$.

- adapt the proof of Carton & Thomas?

5 - on deterministic Büchi automata

$\Rightarrow$ my purpose: deciding whether $\mathcal{L}_\infty(\mathfrak{A})$ contains a Sturmian word

Carton, Thomas (2002)

Given $\mathfrak{A}$ a Büchi automaton and σ, we can decide whether $\sigma^\infty(a) \in \mathcal{L}_\infty(\mathfrak{A})$.

- adapt the proof of Carton & Thomas?

B., Hellouin (ongoing work)

The problem is decidable for deterministic Büchi automata.

Idea of the proof

Obscure and twisted: relies on making the congruence semi-groups used in their proof a proper group (thanks to determinism) to process the different substitutions.

5 - on deterministic Büchi automata

⇒ my purpose: deciding whether $\mathcal{L}_\infty(\mathfrak{A})$ contains a Sturmian word

Carton, Thomas (2002)

Given $\mathfrak{A}$ a Büchi automaton and σ, we can decide whether $\sigma^\infty(a) \in \mathcal{L}_\infty(\mathfrak{A})$.

- adapt the proof of Carton & Thomas?

B., Hellouin (ongoing work)

The problem is decidable for deterministic Büchi automata.

Idea of the proof

Obscure and twisted: relies on making the congruence semi-groups used in their proof a proper group (thanks to determinism) to process the different substitutions.

However . . .

Deterministic Büchi automata is a **weaker** model than Büchi automata.

5 - on non deterministic Büchi automata

B., Hellouin (ongoing work)

If $\mathfrak{A}$ non-deterministic Büchi accepts a Sturmian word, it accepts a substitutive Sturmian word.

Idea of the proof

Manipulate the accepted computation as a sequence of edges, i.e. an infinite word. Then, from it, create a substitutive accepted computation labeled by another Sturmian word. The labeled walk has to be substitutive too.

5 - on non deterministic Büchi automata

B., Hellouin (ongoing work)

If $\mathfrak{A}$ non-deterministic Büchi accepts a Sturmian word, it accepts a substitutive Sturmian word.

Idea of the proof

Manipulate the accepted computation as a sequence of edges, i.e. an infinite word. Then, from it, create a substitutive accepted computation labeled by another Sturmian word. The labeled walk has to be substitutive too.

Consequence

The problem is semi-decidable.

Proof

Enumerate the Sturmian substitutions, and apply the Carton-Thomas algorithm.

5 - the hope of Christmas magic

Conjecture

If $\mathfrak{A}$ Büchi accepts a Sturmian word, $\mathfrak{A}$ accepts a Sturmian word *whose computation is uniformly recurrent*.

- intuitive reasons: a Sturmian word is uniformly recurrent, so it should give bounds for the computations???

Conjecture

If $\mathfrak{A}$ Büchi accepts a Sturmian word, $\mathfrak{A}$ accepts a Sturmian word *whose computation is uniformly recurrent*.

- intuitive reasons: a Sturmian word is uniformly recurrent, so it should give bounds for the computations???

B., Hellouin (ongoing work)

Given $\mathfrak{A}$ Büchi, can decide whether $\mathfrak{A}$ accepts a Sturmian word whose computation is uniformly recurrent.

Idea of the algorithm

Build the meta-ω-automaton; forget the non-accepting edges in every desubstitution $\mathfrak{B}$; check if there is at least one remaining desubstitution which accepts a Sturmian word.

Conclusion and open questions

- new decidable problems for combinatorics on words;
- better understanding of structures for ω-automata and substitutive walks

- from inf. desub. words to S-adic words?
  $\Rightarrow$ technical difficulties on growingness
- proving or disproving the conjecture
  $\Rightarrow$ the space to explore is too large, need ideas
- possible extensions: pushdown automata, random substitutions, . . .

Thank you for your attention!