Is the right relaxation normal form for braids automatic?

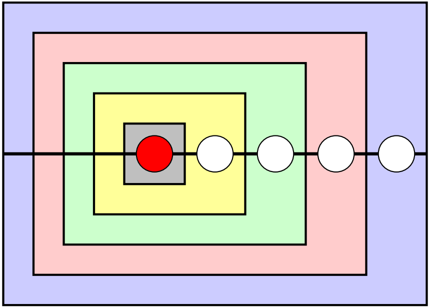
Vincent Jugé

LSV (CNRS & ENS Cachan, Université Paris-Saclay)

28/10/2016

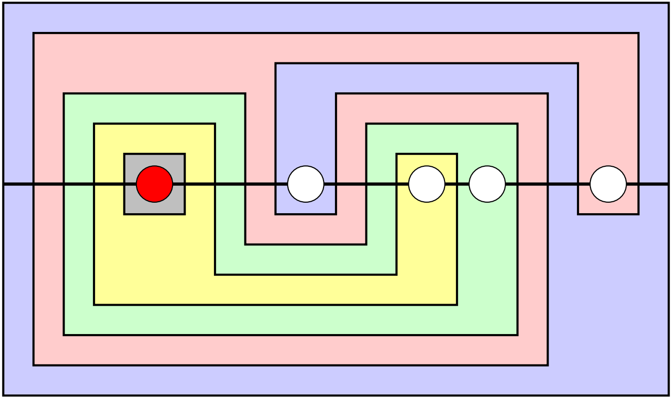
Contents

Curve Diagrams



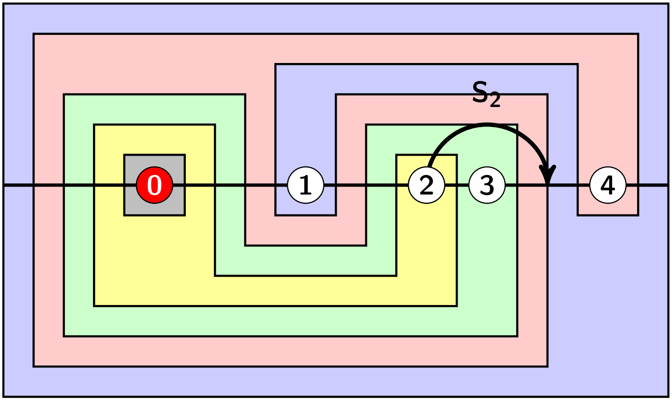
Trivial diagram

Curve Diagrams

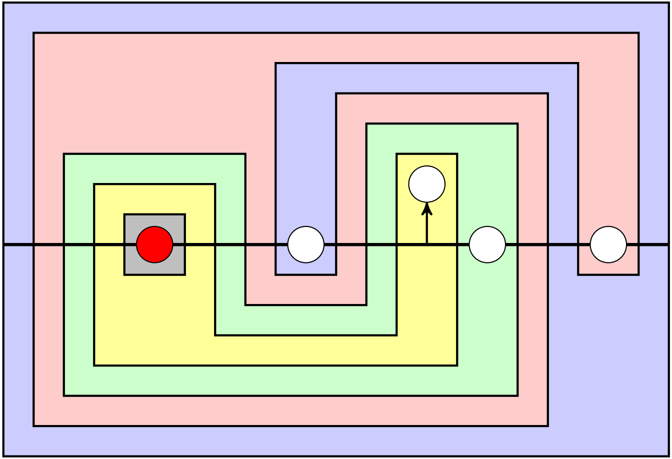


Non-trivial diagram

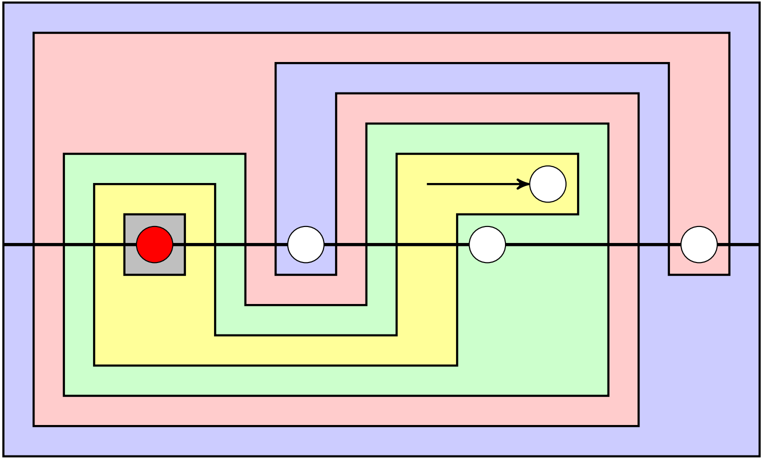
Deformations a Curve Diagram: Example



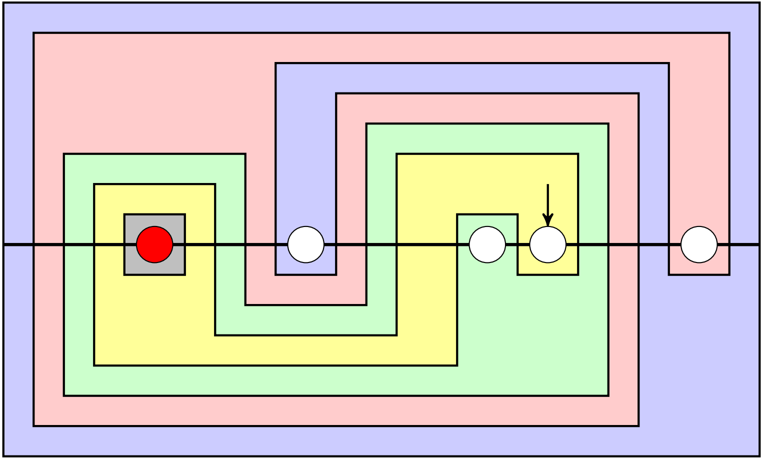
Deformations a Curve Diagram: Example



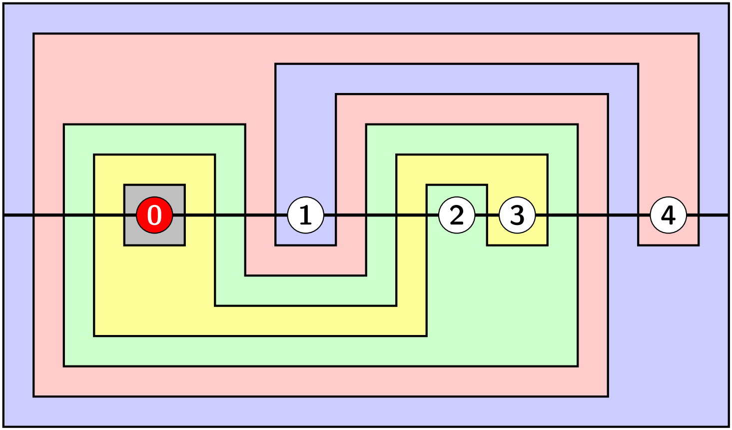
Deformations a Curve Diagram: Example



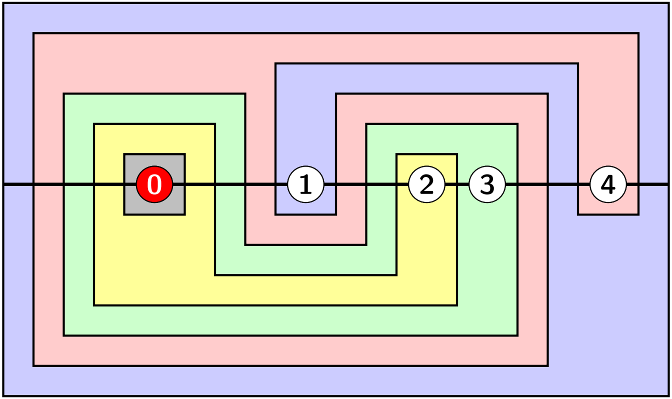
Deformations a Curve Diagram: Example



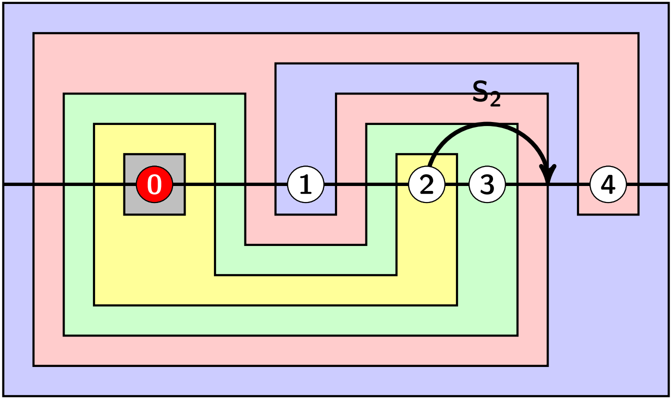
Deformations a Curve Diagram: Example



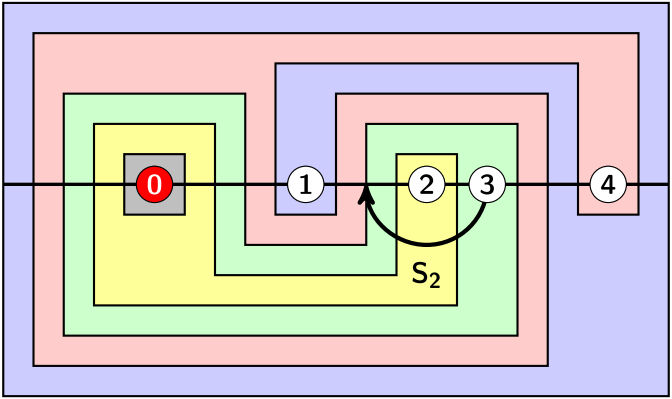
Deformations of a Curve Diagram: Artin Moves



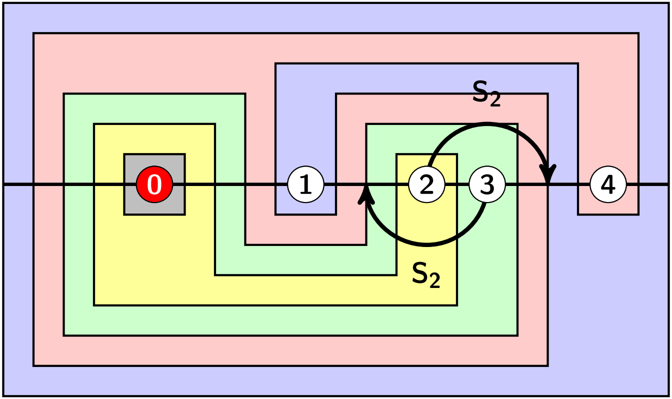
Deformations of a Curve Diagram: Artin Moves



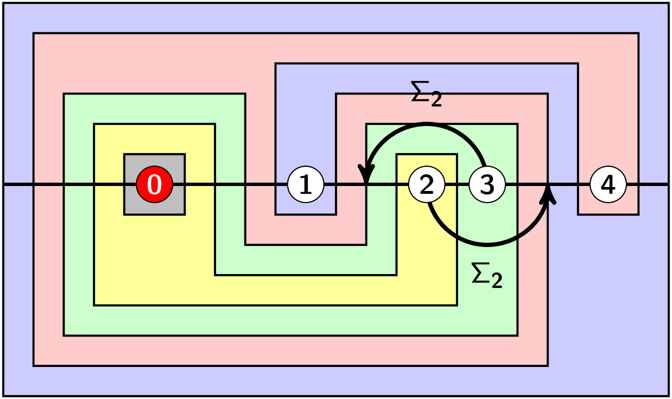
Deformations of a Curve Diagram: Artin Moves



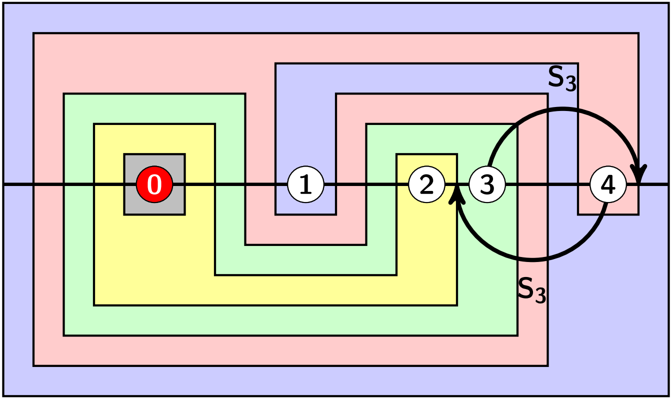
Deformations of a Curve Diagram: Artin Moves



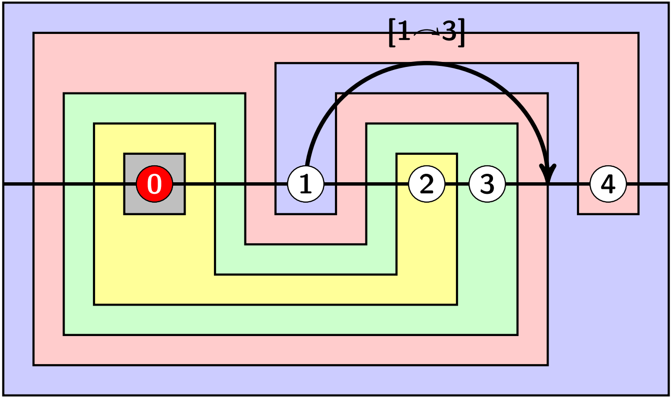
Deformations of a Curve Diagram: Artin Moves



Deformations of a Curve Diagram: Artin Moves

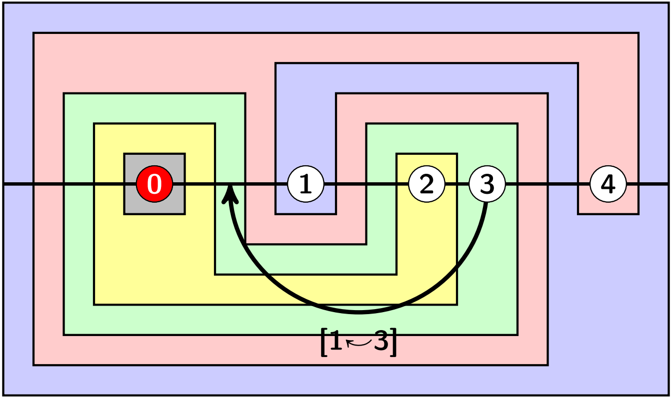


Deformations of a Curve Diagram: Semi-Circular Moves



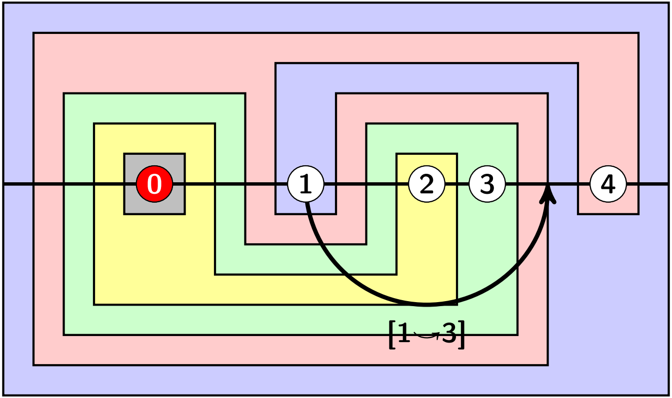
$[1 \curvearrowright 3] = \mathbf{S}_1 \mathbf{S}_2$

Deformations of a Curve Diagram: Semi-Circular Moves



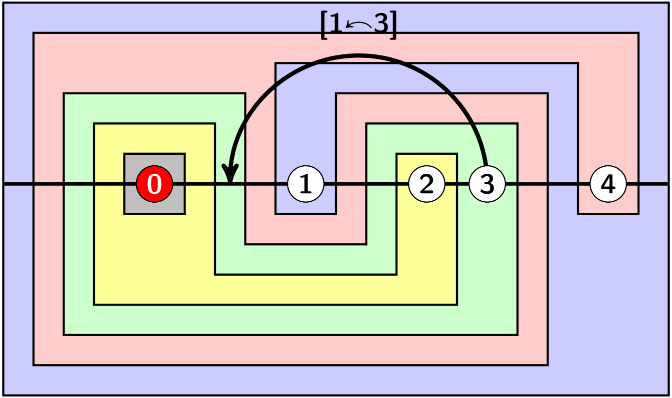
$$[1 \leftharpoondown 3] = \mathbf{S}_2\mathbf{S}_1 \neq \mathbf{S}_1\mathbf{S}_2 = [1 \rightharpoonup 3]$$

Deformations of a Curve Diagram: Semi-Circular Moves



$$[1 \curvearrowright 3] = \Sigma_1 \Sigma_2$$

Deformations of a Curve Diagram: Semi-Circular Moves



$$[1 \leftharpoondown 3] = \Sigma_2 \Sigma_1$$

Braid Group

Braid Group $\mathcal{B}_n$: Definition #1

- Monoid generated by the transformations $\mathbf{S}_i$ and Σ_i, $1 \leqslant i < n$
- $\mathbf{S}_i \Sigma_i = \Sigma_i \mathbf{S}_i = \mathbf{Id}$

Braid Group

Braid Group $\mathcal{B}_n$: Definition #1

- Monoid generated by the transformations $\mathbf{S}_i$ and Σ_i, $1 \leqslant i < n$
- $\mathbf{S}_i \Sigma_i = \Sigma_i \mathbf{S}_i = \mathbf{Id}$

Braid Group $\mathcal{B}_n$: Definition #2

$$\langle \mathbf{S}_1, \ldots, \mathbf{S}_{n-1} \mid \mathbf{S}_i \mathbf{S}_{i+1} \mathbf{S}_i = \mathbf{S}_{i+1} \mathbf{S}_i \mathbf{S}_{i+1}, \mathbf{S}_i \mathbf{S}_j = \mathbf{S}_j \mathbf{S}_i \text{ if } j \geqslant i+2 \rangle$$

Braid Group

Braid Group $\mathcal{B}_n$: Definition #1

- Monoid generated by the transformations $\mathbf{S}_i$ and Σ_i, $1 \leqslant i < n$
- $\mathbf{S}_i \Sigma_i = \Sigma_i \mathbf{S}_i = \mathbf{Id}$

Braid Group $\mathcal{B}_n$: Definition #2

$\langle \mathbf{S}_1, \ldots, \mathbf{S}_{n-1} \mid \mathbf{S}_i \mathbf{S}_{i+1} \mathbf{S}_i = \mathbf{S}_{i+1} \mathbf{S}_i \mathbf{S}_{i+1}, \mathbf{S}_i \mathbf{S}_j = \mathbf{S}_j \mathbf{S}_i \text{ if } j \geqslant i+2 \rangle$

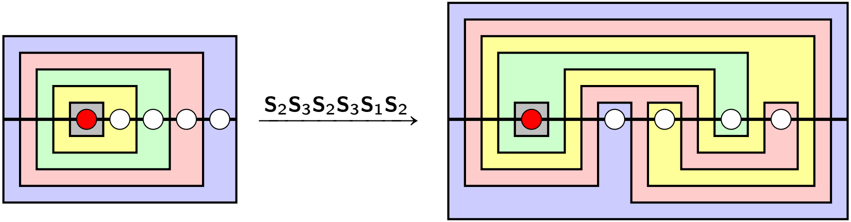
Systems of Generators

- Artin generators: $\mathbf{S}_i$, Σ_i
- Semi-circular generators: $[k \rightharpoondown \ell]$, $[k \rightharpoonup \ell]$, $[k \leftharpoondown \ell]$, $[k \leftharpoonup \ell]$

Braid ≈ Curve Diagram

Identification Theorem (Birman 74)

For all diagrams $\mathcal{D}$ and $\mathfrak{D}$, a unique braid maps $\mathcal{D}$ to $\mathfrak{D}$.



Contents

Goals: Why a Normal Form?

Desired Features

Write group elements as products of generators such that

- The product is "short"
- Algorithms are carried efficiently:
  - Group multiplication
  - Conjugation test
  - Many others...
- The factorization is "natural"

Goals: Why a Normal Form?

Desired Features

Write group elements as products of generators such that

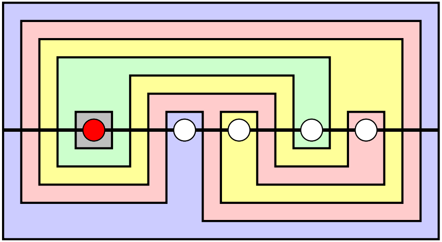
- The product is "short"
- Algorithms are carried efficiently:
  - Group multiplication
  - Conjugation test
  - Many others. . .
- The factorization is "natural"

Example: $(\mathbb{Z}, +)$

- Generators $= \{\pm 1\}$: base 1
- Generators $= \{\pm 2^n \mid n \in \mathbb{N}\}$: base 2 (but others are possible)

Right-Relaxation Algorithm: Example

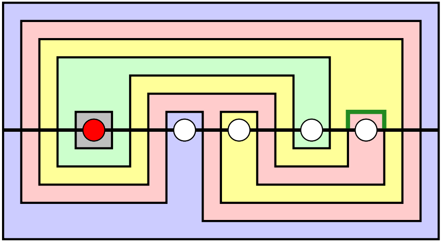
In a Non-Trivial Diagram. . .



Right-Relaxation Algorithm: Example

In a Non-Trivial Diagram...

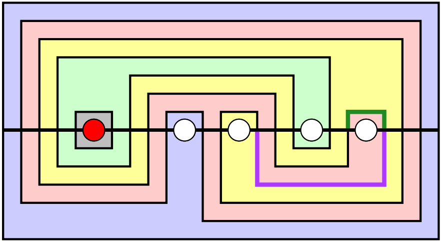
1. Pick the hole in the rightmost **bigon**



Right-Relaxation Algorithm: Example

In a Non-Trivial Diagram...

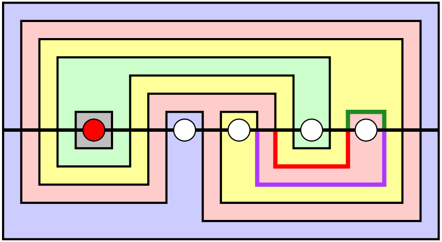
1. Pick the hole in the rightmost **bigon**
2. Slide it along its
   - **right branch** if possible



Right-Relaxation Algorithm: Example

In a Non-Trivial Diagram...

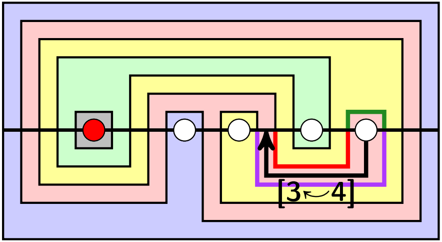
1. Pick the hole in the rightmost **bigon**
2. Slide it along its
   - **right branch** if possible
   - **left branch** otherwise



Right-Relaxation Algorithm: Example

In a Non-Trivial Diagram…

1. Pick the hole in the rightmost **bigon**
2. Slide it along its
   - **right branch** if possible
   - **left branch** otherwise
3. Iterate the process until you reach the trivial diagram

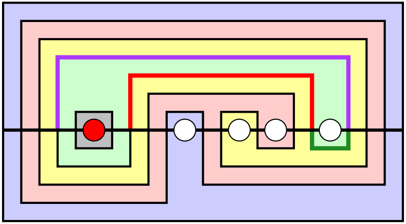


Move performed: [3↼4]

Right-Relaxation Algorithm: Example

In a Non-Trivial Diagram…

1. Pick the hole in the rightmost **bigon**
2. Slide it along its
   - **right branch** if possible
   - **left branch** otherwise
3. Iterate the process until you reach the trivial diagram

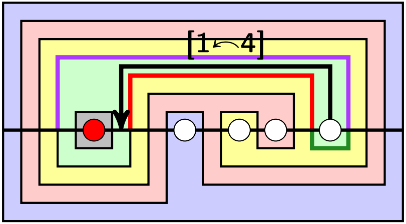


Move performed: [3↼4]

Right-Relaxation Algorithm: Example

In a Non-Trivial Diagram…

1. Pick the hole in the rightmost **bigon**
2. Slide it along its
   - **right branch** if possible
   - **left branch** otherwise
3. Iterate the process until you reach the trivial diagram

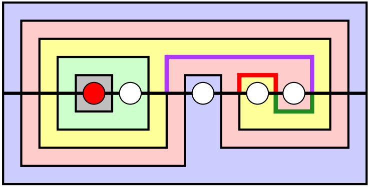


Moves performed: [3↼4][1↼4]

Right-Relaxation Algorithm: Example

In a Non-Trivial Diagram...

1. Pick the hole in the rightmost **bigon**
2. Slide it along its
   - **right branch** if possible
   - **left branch** otherwise
3. Iterate the process until you reach the trivial diagram

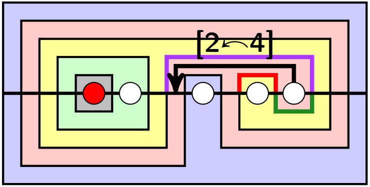


Moves performed: [3↼4][1↼4]

Right-Relaxation Algorithm: Example

In a Non-Trivial Diagram. . .

1. Pick the hole in the rightmost **bigon**
2. Slide it along its
   - **right branch** if possible
   - **left branch** otherwise
3. Iterate the process until you reach the trivial diagram

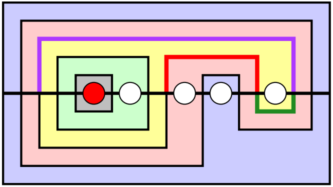


Moves performed: [3↜4][1↜4][2↜4]

Right-Relaxation Algorithm: Example

In a Non-Trivial Diagram...

1. Pick the hole in the rightmost **bigon**
2. Slide it along its
   - **right branch** if possible
   - **left branch** otherwise
3. Iterate the process until you reach the trivial diagram

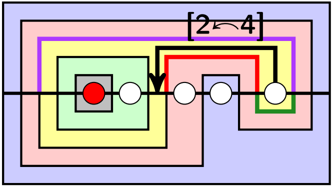


Moves performed: $[3 \curvearrowleft 4][1 \curvearrowleft 4][2 \curvearrowleft 4]$

Right-Relaxation Algorithm: Example

In a Non-Trivial Diagram...

1. Pick the hole in the rightmost **bigon**
2. Slide it along its
   - **right branch** if possible
   - **left branch** otherwise
3. Iterate the process until you reach the trivial diagram

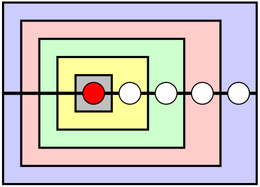


Moves performed: [3↽4][1↼4][2↼4][2↼4]

Right-Relaxation Algorithm: Example

In a Non-Trivial Diagram...

1. Pick the hole in the rightmost **bigon**
2. Slide it along its
   - **right branch** if possible
   - **left branch** otherwise
3. Iterate the process until you reach the trivial diagram



TRIVIAL DIAGRAM!

Moves performed: $[3 \curvearrowleft 4][1 \curvearrowleft 4][2 \curvearrowleft 4][2 \curvearrowleft 4]$

From the Algorithm to the Normal Form

Computing the Right-Relaxation Normal Form

1. Consider your braid $\mathbf{B}$ and the trivial diagram $\mathcal{D}_\varepsilon$
2. Apply $\mathbf{B}$ to $\mathcal{D}_\varepsilon$
   - You obtain a diagram $\mathcal{D}_\mathbf{B}$
3. Apply the right-relaxation algorithm to $\mathcal{D}_\mathbf{B}$
   - You perform moves $\mathbf{m}_1 \mathbf{m}_2 \ldots \mathbf{m}_k$ and obtain $\mathcal{D}_\varepsilon$ again
4. The right normal form of $\mathbf{B}$ is the product $\mathbf{m}_k^{-1} \mathbf{m}_{k-1}^{-1} \ldots \mathbf{m}_1^{-1}$

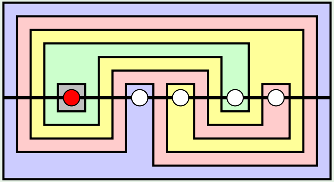
From the Algorithm to the Normal Form

Computing the Right-Relaxation Normal Form

1. Consider your braid **B** and the trivial diagram $\mathcal{D}_\varepsilon$
2. Apply **B** to $\mathcal{D}_\varepsilon$
   - You obtain a diagram $\mathcal{D}_{\mathbf{B}}$
3. Apply the right-relaxation algorithm to $\mathcal{D}_{\mathbf{B}}$
   - You perform moves $\mathbf{m}_1 \mathbf{m}_2 \ldots \mathbf{m}_k$ and obtain $\mathcal{D}_\varepsilon$ again
4. The right normal form of **B** is the product $\mathbf{m}_k^{-1} \mathbf{m}_{k-1}^{-1} \ldots \mathbf{m}_1^{-1}$

Example: $\mathbf{B} = \mathbf{S}_2 \mathbf{S}_3 \mathbf{S}_2 \mathbf{S}_3 \mathbf{S}_1 \mathbf{S}_2 = [2 \rightharpoonup 4][2 \rightharpoonup 4][1 \rightharpoonup 3]$

Diagram $\mathcal{D}_{\mathbf{B}}$:



Moves performed:

$[3 \leftharpoondown 4][1 \leftharpoonup 4][2 \leftharpoonup 4][2 \leftharpoonup 4]$

Right normal form:

$[2 \rightharpoonup 4][2 \rightharpoonup 4][1 \rightharpoonup 4][3 \rightharpoondown 4]$

Contents

Basic Properties

Satisfied Properties

- Prefix-closed language
- Efficient to compute
- Behaves nicely with the Dehornoy braid ordering

Unsatisfied Properties

- Not geodesic

Basic Properties

Satisfied Properties

- Prefix-closed language
- Efficient to compute
- Behaves nicely with the Dehornoy braid ordering

Unsatisfied Properties

- Not geodesic

Regularity & Automaticity Properties

Is the right-relaxation normal form

- regular?
- left automatic?
- right automatic?

Regularity

Regular Language

Language accepted by a NFA:

- Read your word from left to right
- Use a finite auxiliary memory
- Decide word membership

Regularity

Regular Language

Language accepted by a NFA:

- Read your word from left to right
- Use a finite auxiliary memory
- Decide word membership

Theorem (J. 2016^{+})

The right-relaxation normal form is regular!

Key Idea

- Use prefix-closure
- Split the real axis in intervals
- Remember which intervals are linked by **neighboring** arcs

Regularity (with a small automaton)

Regular Language

Language accepted by a NFA:

- Read your word from left to right
- Use a finite auxiliary memory
- Decide word membership

Theorem (J. 2016$^+$)

The right-relaxation normal form is regular!

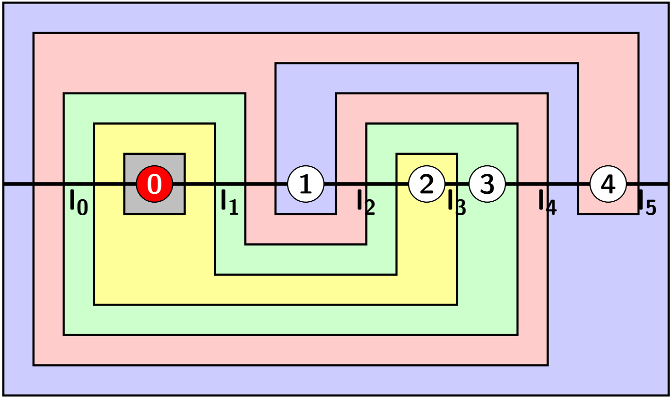
Key Idea

- Use prefix-closure
- Split the real axis in intervals
- Remember which intervals are linked by **neighboring** arcs

Regularity (with a small automaton)

Key Idea (continued)

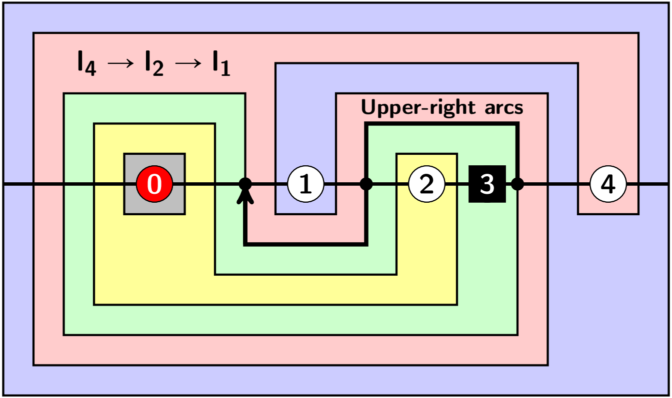
1. Intervals



Regularity (with a small automaton)

Key Idea (continued)

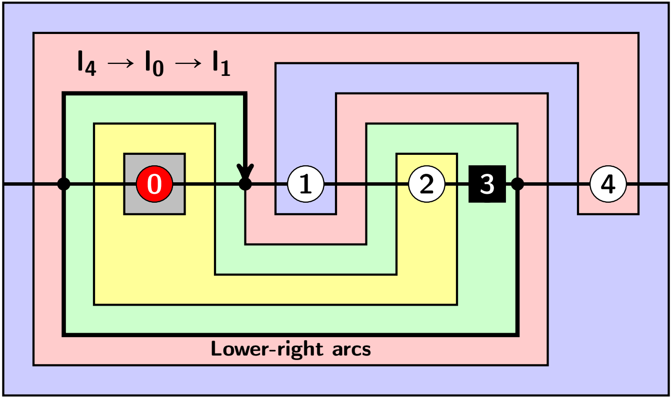
1. Intervals
2. Neighboring arcs



Regularity (with a small automaton)

Key Idea (continued)

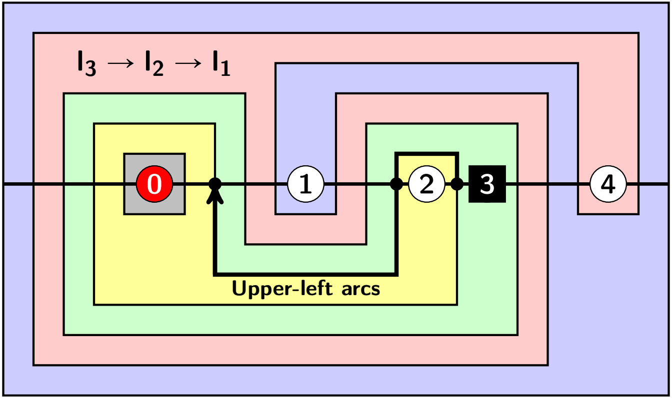
1. Intervals
2. Neighboring arcs



Regularity (with a small automaton)

Key Idea (continued)

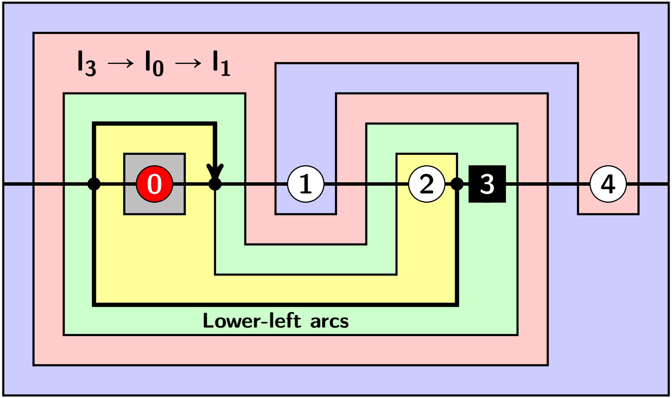
1. Intervals
2. Neighboring arcs



Regularity (with a small automaton)

Key Idea (continued)

1. Intervals
2. Neighboring arcs



Automaticity

Ingredients

- Group $\mathcal{G}$
- Finite generating set Σ
- Normal form $\mathbf{NF} \subseteq \Sigma^*$

Automaticity

Ingredients

- Group $\mathcal{G}$
- Finite generating set Σ
- Normal form $\mathbf{NF} \subseteq \Sigma^*$

Informal Meaning

- Regularity: **checking membership** in **NF** is easy
- Automaticity: **simulating multiplication** in $\mathcal{G}$ is easy

Automaticity

Ingredients

- Group $\mathcal{G}$
- Finite generating set Σ
- Normal form $\mathbf{NF} \subseteq \Sigma^*$

Informal Meaning

- Regularity: **checking membership** in **NF** is easy
- Automaticity: **simulating multiplication** in $\mathcal{G}$ is easy

Less Informal Meaning

- Use end-padding to encode pairs of words
- Consider the sets $R_x = \{(\mathbf{u}, \mathbf{v}) \mid \mathbf{u}, \mathbf{v} \in \mathbf{NF}, \mathbf{u}x \equiv_{\mathcal{G}} \mathbf{v}\}$ and $L_x = \{(\mathbf{u}, \mathbf{v}) \mid \mathbf{u}, \mathbf{v} \in \mathbf{NF}, x\mathbf{u} \equiv_{\mathcal{G}} \mathbf{v}\}$, for $x \in \Sigma$
  - **NF** is synchronously **right**-automatic if all R_x are regular
  - **NF** is synchronously **left**-automatic if all L_x are regular

Automaticity

Ingredients

- Group $\mathcal{G}$
- Finite generating set Σ
- Normal form $\mathbf{NF} \subseteq \Sigma^*$

Informal Meaning

- Regularity: **checking membership** in **NF** is easy
- Automaticity: **simulating multiplication** in $\mathcal{G}$ is easy

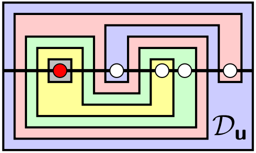
Less Informal Meaning

- Use end-padding to encode pairs of words
- Consider the sets $R_x = \{(\mathbf{u}, \mathbf{v}) \mid \mathbf{u}, \mathbf{v} \in \mathbf{NF}, \mathbf{u}x \equiv_{\mathcal{G}} \mathbf{v}\}$ and $L_x = \{(\mathbf{u}, \mathbf{v}) \mid \mathbf{u}, \mathbf{v} \in \mathbf{NF}, x\mathbf{u} \equiv_{\mathcal{G}} \mathbf{v}\}$, for $x \in \Sigma$
  - **NF** is synchronously **bi**-automatic if all R_x and L_x are regular

Automaticity

Example

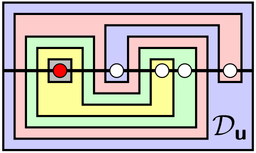
- $\mathcal{G} = \mathcal{B}_4$, $\Sigma = \{$semi-circular generators$\}$, $\mathbf{NF}$ = right-relaxation NF
- Set membership:
  - $\mathbf{u} = [2 \smile 4][1 \smile 3][1 \frown 4] \in \mathbf{NF}$

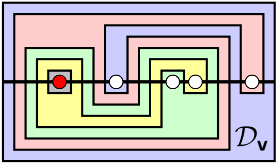


Automaticity

Example

- $\mathcal{G} = \mathcal{B}_4$, $\Sigma = \{$semi-circular generators$\}$, **NF** = right-relaxation NF
- Set membership:
  - $\mathbf{u} = [2 \hookrightarrow 4][1 \hookrightarrow 3][1 \curvearrowright 4] \in \mathbf{NF}$
  - $\mathbf{v} = [2 \hookrightarrow 4][1 \hookrightarrow 3][3 \curvearrowright 4][1 \curvearrowright 4] \in \mathbf{NF}$

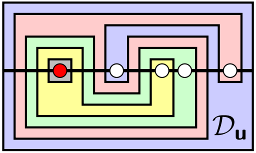


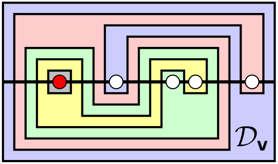


Automaticity

Example

- $\mathcal{G} = \mathcal{B}_4$, $\Sigma = \{$semi-circular generators$\}$, $\mathbf{NF}$ = right-relaxation NF
- Set membership:
  - $\mathbf{u} = [2 \smile 4][1 \smile 3][1 \frown 4] \in \mathbf{NF}$
  - $\mathbf{v} = [2 \smile 4][1 \smile 3][3 \frown 4][1 \frown 4] \in \mathbf{NF}$
  - $\mathbf{v} \equiv_{\mathcal{G}} \mathbf{u}\ \mathbf{S}_2$



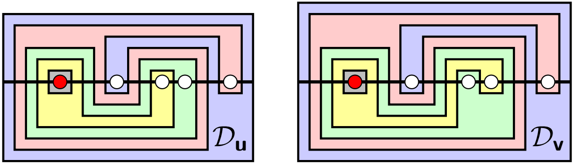


Automaticity

Example

- $\mathcal{G} = \mathcal{B}_4$, $\Sigma = \{$semi-circular generators$\}$, **NF** = right-relaxation NF
- Set membership:
  - $\mathbf{u} = [2 \smile 4][1 \smile 3][1 \frown 4] \in \mathbf{NF}$
  - $\mathbf{v} = [2 \smile 4][1 \smile 3][3 \frown 4][1 \frown 4] \in \mathbf{NF}$

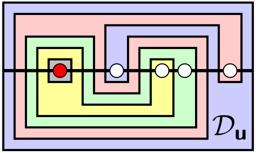
  - $(\mathbf{u}, \mathbf{v}) = \binom{[2 \smile 4]}{[2 \smile 4]}\binom{[1 \smile 3]}{[1 \smile 3]}\binom{[1 \frown 4]}{[3 \frown 4]}\binom{\varepsilon}{[1 \frown 4]} \in R_{\mathbf{S_2}}$

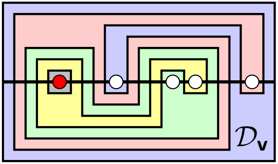


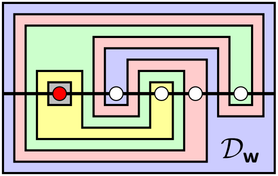
Automaticity

Example

- $\mathcal{G} = \mathcal{B}_4$, $\Sigma = \{\text{semi-circular generators}\}$, $\mathbf{NF}$ = right-relaxation NF
- Set membership:
  - $\mathbf{u} = [2 \smile 4][1 \smile 3][1 \frown 4] \in \mathbf{NF}$
  - $\mathbf{v} = [2 \smile 4][1 \smile 3][3 \frown 4][1 \frown 4] \in \mathbf{NF}$
  - $\mathbf{w} = [3 \smile 4][1 \smile 3][1 \frown 4] \in \mathbf{NF}$
  - $(\mathbf{u}, \mathbf{v}) = \binom{[2 \smile 4]}{[2 \smile 4]} \binom{[1 \smile 3]}{[1 \smile 3]} \binom{[1 \frown 4]}{[3 \frown 4]} \binom{\varepsilon}{[1 \frown 4]} \in R_{\mathbf{S_2}}$



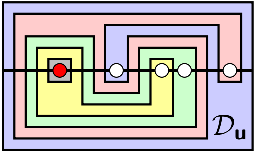


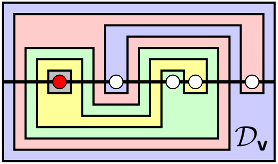


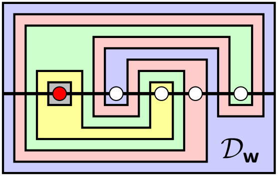
Automaticity

Example

- $\mathcal{G} = \mathcal{B}_4$, $\Sigma = \{$semi-circular generators$\}$, $\mathbf{NF} =$ right-relaxation NF
- Set membership:
  - $\mathbf{u} = [2 \smile 4][1 \smile 3][1 \frown 4] \in \mathbf{NF}$
  - $\mathbf{v} = [2 \smile 4][1 \smile 3][3 \frown 4][1 \frown 4] \in \mathbf{NF}$
  - $\mathbf{w} = [3 \smile 4][1 \smile 3][1 \frown 4] \in \mathbf{NF}$
  - $(\mathbf{u}, \mathbf{v}) = \binom{[2 \smile 4]}{[2 \smile 4]}\binom{[1 \smile 3]}{[1 \smile 3]}\binom{[1 \frown 4]}{[3 \frown 4]}\binom{\varepsilon}{[1 \frown 4]} \in R_{\mathbf{S}_2}$
  - $\mathbf{w} \equiv_{\mathcal{G}} \mathbf{S}_2\ \mathbf{u}$



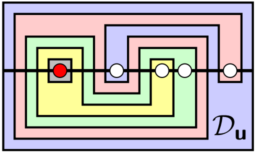


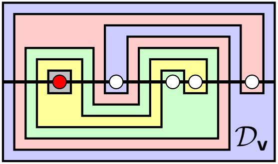


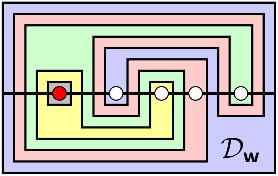
Automaticity

Example

- $\mathcal{G} = \mathcal{B}_4$, $\Sigma = \{$semi-circular generators$\}$, **NF** = right-relaxation NF
- Set membership:
  - $\mathbf{u} = [2 \smile 4][1 \smile 3][1 \frown 4] \in \mathbf{NF}$
  - $\mathbf{v} = [2 \smile 4][1 \smile 3][3 \frown 4][1 \frown 4] \in \mathbf{NF}$
  - $\mathbf{w} = [3 \smile 4][1 \smile 3][1 \frown 4] \in \mathbf{NF}$
  - $(\mathbf{u}, \mathbf{v}) = \binom{[2 \smile 4]}{[2 \smile 4]}\binom{[1 \smile 3]}{[1 \smile 3]}\binom{[1 \frown 4]}{[3 \frown 4]}\binom{\varepsilon}{[1 \frown 4]} \in R_{\mathbf{S_2}}$
  - $(\mathbf{u}, \mathbf{w}) = \binom{[2 \smile 4]}{[3 \frown 4]}\binom{[1 \smile 3]}{[1 \smile 3]}\binom{[1 \frown 4]}{[1 \frown 4]} \in L_{\mathbf{S_2}}$







Automaticity

Asynchronous Automaticity

- Consider a padding function φ
- Consider the sets $R_x^\varphi = \{(\mathbf{u},\mathbf{v})^\varphi \mid \mathbf{u},\mathbf{v} \in \mathbf{NF}, \mathbf{u}x \equiv_{\mathcal{G}} \mathbf{v}\}$ and $L_x^\varphi = \{(\mathbf{u},\mathbf{v})^\varphi \mid \mathbf{u},\mathbf{v} \in \mathbf{NF}, x\mathbf{u} \equiv_{\mathcal{G}} \mathbf{v}\}$, for $x \in \Sigma$
  - **NF** is asynchronously **right**-automatic if all R_x^φ are regular
  - **NF** is asynchronously **left**-automatic if all L_x^φ are regular

for **some** padding function φ

Automaticity

Asynchronous Automaticity

- Consider a padding function φ
- Consider the sets $R_x^\varphi = \{(\mathbf{u},\mathbf{v})^\varphi \mid \mathbf{u},\mathbf{v} \in \mathbf{NF}, \mathbf{u}x \equiv_\mathcal{G} \mathbf{v}\}$ and $L_x^\varphi = \{(\mathbf{u},\mathbf{v})^\varphi \mid \mathbf{u},\mathbf{v} \in \mathbf{NF}, x\mathbf{u} \equiv_\mathcal{G} \mathbf{v}\}$, for $x \in \Sigma$
  - **NF** is asynchronously **bi**-automatic if all R_x^φ and L_x^φ are regular

for **some** padding function φ

Automaticity

Asynchronous Automaticity

- Consider a padding function φ
- Consider the sets $R_x^\varphi = \{(\mathbf{u},\mathbf{v})^\varphi \mid \mathbf{u},\mathbf{v} \in \mathbf{NF}, \mathbf{u}x \equiv_{\mathcal{G}} \mathbf{v}\}$ and $L_x^\varphi = \{(\mathbf{u},\mathbf{v})^\varphi \mid \mathbf{u},\mathbf{v} \in \mathbf{NF}, x\mathbf{u} \equiv_{\mathcal{G}} \mathbf{v}\}$, for $x \in \Sigma$
  - **NF** is asynchronously **bi**-automatic if all R_x^φ and L_x^φ are regular

  for **some** padding function φ

Example

- $(\mathbf{u},\mathbf{v})^\varphi = \binom{[2 \hookrightarrow 4]}{[2 \hookrightarrow 4]}\binom{\varepsilon}{[1 \hookrightarrow 3]}\binom{[1 \hookrightarrow 3]}{\varepsilon}\binom{[1 \curvearrowright 4]}{\varepsilon}\binom{\varepsilon}{[3 \curvearrowright 4]}\binom{\varepsilon}{[1 \curvearrowright 4]} \in R_{\mathbf{S}_2}^\varphi$
- $(\mathbf{u},\mathbf{w})^\varphi = \binom{\varepsilon}{[3 \curvearrowright 4]}\binom{[2 \hookrightarrow 4]}{[1 \hookrightarrow 3]}\binom{[1 \hookrightarrow 3]}{\varepsilon}\binom{[1 \curvearrowright 4]}{[1 \curvearrowright 4]} \in L_{\mathbf{S}_2}^\varphi$

Automaticity

My Current Problem

The right-relaxation NF is regular. Is it automatic?

Automaticity

My Current Problem

The right-relaxation NF is regular. Is it automatic?

Theorem (J. 2016^+)

The right-relaxation NF is:

- synchronously **bi**-automatic if $n \leqslant 3$
- asynchronously **left**-automatic if $n = 4$
- **not** synchronously **left**-automatic if $n \geqslant 4$
- **not** asynchronously **right**-automatic if $n \geqslant 4$

Conjecture

The right-relaxation NF is asynchronously left-automatic for all $n \geqslant 1$

Automaticity

My Current Problem

The right-relaxation NF is regular. Is it automatic?

Theorem (J. 2016^+)

The right-relaxation NF is:

- synchronously **bi**-automatic if $n \leqslant 3$
- asynchronously **left**-automatic if $n = 4$
- **not** synchronously **left**-automatic if $n \geqslant 4$
- **not** asynchronously **right**-automatic if $n \geqslant 4$

Conjecture

The right-relaxation NF is asynchronously left-automatic for all $n \geqslant 1$

- Empirical tests suggest an asynchronous fellow-traveller property
- Computing automata for L_x^φ is memory-intensive

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \smile\!\!\to 4]}{[2 \smile\!\!\to 4]} \binom{\varepsilon}{[1 \smile\!\!\to 3]} \binom{[1 \smile\!\!\to 4]}{\varepsilon} \binom{[1 \frown\!\!\to 4]}{\varepsilon} \binom{\varepsilon}{[3 \frown\!\!\to 4]} \binom{\varepsilon}{[1 \frown\!\!\to 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \curvearrowright 4]}{[2 \curvearrowright 4]} \binom{\varepsilon}{[1 \curvearrowright 3]} \binom{[1 \curvearrowright 4]}{\varepsilon} \binom{[1 \curvearrowright 4]}{\varepsilon} \binom{\varepsilon}{[3 \curvearrowright 4]} \binom{\varepsilon}{[1 \curvearrowright 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni \quad [2 \curvearrowright 4]^{-1}[2 \curvearrowright 4]$

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \rightarrowtail 4]}{[2 \rightarrowtail 4]} \binom{\varepsilon}{[1 \rightarrowtail 3]} \binom{[1 \rightarrowtail 4]}{\varepsilon} \binom{[1 \rightarrowtail 4]}{\varepsilon} \binom{\varepsilon}{[3 \rightarrowtail 4]} \binom{\varepsilon}{[1 \rightarrowtail 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni$ $[2 \rightarrowtail 4]^{-1}[2 \rightarrowtail 4][1 \rightarrowtail 3]$

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \rightharpoonup 4]}{[2 \rightharpoonup 4]} \binom{\varepsilon}{[1 \rightharpoonup 3]} \binom{[1 \rightharpoonup 4]}{\varepsilon} \binom{[1 \rightharpoonup 4]}{\varepsilon} \binom{\varepsilon}{[3 \rightharpoonup 4]} \binom{\varepsilon}{[1 \rightharpoonup 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni \quad [1 \rightharpoonup 3]^{-1}[2 \rightharpoonup 4]^{-1}[2 \rightharpoonup 4][1 \rightharpoonup 3]$

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \rightarrowtail 4]}{[2 \rightarrowtail 4]}\binom{\varepsilon}{[1 \rightarrowtail 3]}\binom{[1 \rightarrowtail 4]}{\varepsilon}\binom{[1 \rightharpoonup 4]}{\varepsilon}\binom{\varepsilon}{[3 \rightharpoonup 4]}\binom{\varepsilon}{[1 \rightharpoonup 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \rightharpoonup 4]^{-1}[1 \rightarrowtail 3]^{-1}[2 \rightarrowtail 4]^{-1}[2 \rightarrowtail 4][1 \rightarrowtail 3]$

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \smile\!\!\!\to 4]}{[2 \smile\!\!\!\to 4]} \binom{\varepsilon}{[1 \smile\!\!\!\to 3]} \binom{[1 \smile\!\!\!\to 4]}{\varepsilon} \binom{[1 \frown\!\!\!\to 4]}{\varepsilon} \binom{\varepsilon}{[3 \frown\!\!\!\to 4]} \binom{\varepsilon}{[1 \frown\!\!\!\to 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \frown\!\!\!\to 4]^{-1}[1 \smile\!\!\!\to 3]^{-1}[2 \smile\!\!\!\to 4]^{-1}[2 \smile\!\!\!\to 4][1 \smile\!\!\!\to 3][3 \frown\!\!\!\to 4]$

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \smile 4]}{[2 \smile 4]} \binom{\varepsilon}{[1 \smile 3]} \binom{[1 \smile 4]}{\varepsilon} \binom{[1 \frown 4]}{\varepsilon} \binom{\varepsilon}{[3 \frown 4]} \binom{\varepsilon}{[1 \frown 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \frown 4]^{-1}[1 \smile 3]^{-1}[2 \smile 4]^{-1}[2 \smile 4][1 \smile 3][3 \frown 4][1 \frown 4]$

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \hookrightarrow 4]}{[2 \hookrightarrow 4]} \binom{\varepsilon}{[1 \hookrightarrow 3]} \binom{[1 \hookrightarrow 4]}{\varepsilon} \binom{[1 \curvearrowright 4]}{\varepsilon} \binom{\varepsilon}{[3 \curvearrowright 4]} \binom{\varepsilon}{[1 \curvearrowright 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \curvearrowright 4]^{-1}[1 \hookrightarrow 3]^{-1}[2 \hookrightarrow 4]^{-1}[2 \hookrightarrow 4][1 \hookrightarrow 3][3 \curvearrowright 4][1 \curvearrowright 4]$

Theorem (Epstein et al. 92)

R_x^φ is regular iff $\exists$ **finite** set S s.t. all words $\mathbf{w} \in R_x^\varphi$ right-travel in S.

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \rightharpoondown 4]}{[2 \rightharpoondown 4]} \binom{\varepsilon}{[1 \rightharpoondown 3]} \binom{[1 \rightharpoondown 4]}{\varepsilon} \binom{[1 \rightharpoonup 4]}{\varepsilon} \binom{\varepsilon}{[3 \rightharpoonup 4]} \binom{\varepsilon}{[1 \rightharpoonup 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \rightharpoonup 4]^{-1}[1 \rightharpoondown 3]^{-1}[2 \rightharpoondown 4]^{-1}[2 \rightharpoondown 4][1 \rightharpoondown 3][3 \rightharpoonup 4][1 \rightharpoonup 4]$
- **left**-travels in $T \subseteq \mathcal{G}$ if:

Theorem (Epstein et al. 92)

R_x^φ is regular iff $\exists$ **finite** set S s.t. all words $\mathbf{w} \in R_x^\varphi$ right-travel in S.

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \smile 4]}{[2 \smile 4]} \binom{\varepsilon}{[1 \smile 3]} \binom{[1 \smile 4]}{\varepsilon} \binom{[1 \frown 4]}{\varepsilon} \binom{\varepsilon}{[3 \frown 4]} \binom{\varepsilon}{[1 \frown 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \frown 4]^{-1}[1 \smile 3]^{-1}[2 \smile 4]^{-1}[2 \smile 4][1 \smile 3][3 \frown 4][1 \frown 4]$
- **left**-travels in $T \subseteq \mathcal{G}$ if:
  $T \ni \quad [1 \frown 4]^{-1}$

Theorem (Epstein et al. 92)

R_x^φ is regular iff $\exists$ **finite** set S s.t. all words $\mathbf{w} \in R_x^\varphi$ right-travel in S.

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \hookrightarrow 4]}{[2 \hookrightarrow 4]} \binom{\varepsilon}{[1 \hookrightarrow 3]} \binom{[1 \hookrightarrow 4]}{\varepsilon} \binom{[1 \curvearrowright 4]}{\varepsilon} \binom{\color{red}\varepsilon}{\color{blue}[3 \curvearrowright 4]} \binom{\varepsilon}{[1 \curvearrowright 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \curvearrowright 4]^{-1}[1 \hookrightarrow 3]^{-1}[2 \hookrightarrow 4]^{-1}[2 \hookrightarrow 4][1 \hookrightarrow 3][3 \curvearrowright 4][1 \curvearrowright 4]$
- **left**-travels in $T \subseteq \mathcal{G}$ if:
  $T \ni \qquad [1 \curvearrowright 4]^{-1}{\color{blue}[3 \curvearrowright 4]^{-1}}$

Theorem (Epstein et al. 92)

R_x^φ is regular iff $\exists$ **finite** set S s.t. all words $\mathbf{w} \in R_x^\varphi$ right-travel in S.

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \rightharpoondown 4]}{[2 \rightharpoondown 4]} \binom{\varepsilon}{[1 \rightharpoondown 3]} \binom{[1 \rightharpoondown 4]}{\varepsilon} \binom{[1 \rightharpoonup 4]}{\varepsilon} \binom{\varepsilon}{[3 \rightharpoonup 4]} \binom{\varepsilon}{[1 \rightharpoonup 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \rightharpoonup 4]^{-1}[1 \rightharpoondown 3]^{-1}[2 \rightharpoondown 4]^{-1}[2 \rightharpoondown 4][1 \rightharpoondown 3][3 \rightharpoonup 4][1 \rightharpoonup 4]$
- **left**-travels in $T \subseteq \mathcal{G}$ if:
  $T \ni [1 \rightharpoonup 4][1 \rightharpoonup 4]^{-1}[3 \rightharpoonup 4]^{-1}$

Theorem (Epstein et al. 92)

R_x^φ is regular iff $\exists$ **finite** set S s.t. all words $\mathbf{w} \in R_x^\varphi$ right-travel in S.

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2\hookrightarrow 4]}{[2\hookrightarrow 4]}\binom{\varepsilon}{[1\hookrightarrow 3]}\binom{[1\hookrightarrow 4]}{\varepsilon}\binom{[1\curvearrowright 4]}{\varepsilon}\binom{\varepsilon}{[3\curvearrowright 4]}\binom{\varepsilon}{[1\curvearrowright 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \curvearrowright 4]^{-1}[1 \hookrightarrow 3]^{-1}[2 \hookrightarrow 4]^{-1}[2 \hookrightarrow 4][1 \hookrightarrow 3][3 \curvearrowright 4][1 \curvearrowright 4]$
- **left**-travels in $T \subseteq \mathcal{G}$ if:
  $T \ni \quad [1 \hookrightarrow 3][1 \curvearrowright 4][1 \curvearrowright 4]^{-1}[3 \curvearrowright 4]^{-1}$

Theorem (Epstein et al. 92)

R_x^φ is regular iff $\exists$ **finite** set S s.t. all words $\mathbf{w} \in R_x^\varphi$ right-travel in S.

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \hookrightarrow 4]}{[2 \hookrightarrow 4]} \binom{\varepsilon}{[1 \hookrightarrow 3]} \binom{[1 \hookrightarrow 4]}{\varepsilon} \binom{[1 \rightharpoonup 4]}{\varepsilon} \binom{\varepsilon}{[3 \rightharpoonup 4]} \binom{\varepsilon}{[1 \rightharpoonup 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \rightharpoonup 4]^{-1}[1 \hookrightarrow 3]^{-1}[2 \hookrightarrow 4]^{-1}[2 \hookrightarrow 4][1 \hookrightarrow 3][3 \rightharpoonup 4][1 \rightharpoonup 4]$
- **left**-travels in $T \subseteq \mathcal{G}$ if:
  $T \ni \quad [1 \hookrightarrow 3][1 \rightharpoonup 4][1 \rightharpoonup 4]^{-1}[3 \rightharpoonup 4]^{-1}[1 \hookrightarrow 3]^{-1}$

Theorem (Epstein et al. 92)

R_x^φ is regular iff $\exists$ **finite** set S s.t. all words $\mathbf{w} \in R_x^\varphi$ right-travel in S.

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{\color{red}[2\hookrightarrow 4]}{\color{blue}[2\hookrightarrow 4]}\binom{\varepsilon}{[1\hookrightarrow 3]}\binom{[1\hookrightarrow 4]}{\varepsilon}\binom{[1\rightharpoonup 4]}{\varepsilon}\binom{\varepsilon}{[3\rightharpoonup 4]}\binom{\varepsilon}{[1\rightharpoonup 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \rightharpoonup 4]^{-1}[1 \hookrightarrow 3]^{-1}[2 \hookrightarrow 4]^{-1}[2 \hookrightarrow 4][1 \hookrightarrow 3][3 \rightharpoonup 4][1 \rightharpoonup 4]$
- **left**-travels in $T \subseteq \mathcal{G}$ if:
  $T \ni {\color{red}[2 \hookrightarrow 4]}[1 \hookrightarrow 3][1 \rightharpoonup 4][1 \rightharpoonup 4]^{-1}[3 \rightharpoonup 4]^{-1}[1 \hookrightarrow 3]^{-1}{\color{blue}[2 \hookrightarrow 4]^{-1}}$

Theorem (Epstein et al. 92)

R_x^φ is regular iff $\exists$ **finite** set S s.t. all words $\mathbf{w} \in R_x^\varphi$ right-travel in S.

Automaticity: Fellow-Traveller Property

Example

The word $\mathbf{w} = \binom{[2 \hookrightarrow 4]}{[2 \hookrightarrow 4]} \binom{\varepsilon}{[1 \hookrightarrow 3]} \binom{[1 \hookrightarrow 4]}{\varepsilon} \binom{[1 \rightharpoonup 4]}{\varepsilon} \binom{\varepsilon}{[3 \rightharpoonup 4]} \binom{\varepsilon}{[1 \rightharpoonup 4]}$

- **right**-travels in $S \subseteq \mathcal{G}$ if:
  $S \ni [1 \rightharpoonup 4]^{-1}[1 \hookrightarrow 3]^{-1}[2 \hookrightarrow 4]^{-1}[2 \hookrightarrow 4][1 \hookrightarrow 3][3 \rightharpoonup 4][1 \rightharpoonup 4]$
- **left**-travels in $T \subseteq \mathcal{G}$ if:
  $T \ni [2 \hookrightarrow 4][1 \hookrightarrow 3][1 \rightharpoonup 4][1 \rightharpoonup 4]^{-1}[3 \rightharpoonup 4]^{-1}[1 \hookrightarrow 3]^{-1}[2 \hookrightarrow 4]^{-1}$

Theorem (Epstein et al. 92)

R_x^φ is regular iff $\exists$ **finite** set S s.t. all words $\mathbf{w} \in R_x^\varphi$ right-travel in S.
L_x^φ is regular iff $\exists$ **finite** set T s.t. all words $\mathbf{w} \in L_x^\varphi$ right-travel in T.

Automaticity: Key Ideas

Synchronous bi-automaticity if $n \leqslant 3$

Computing an automaton for R_x or L_x (with $x \in \{\mathbf{S}_1, \mathbf{S}_2\}$):

- Direct computations are too expensive!

Automaticity: Key Ideas

Synchronous bi-automaticity if $n \leqslant 3$

Computing an automaton for R_x or L_x (with $x \in \{\mathbf{S}_1, \mathbf{S}_2\}$):

- Direct computations are too expensive!
- Replace sliding generators with Artin generators
- Recover the synchronous automaticity

Example

Sliding generators:

- $\mathbf{u} = [2 \rightharpoondown 3][2 \rightharpoondown 3][1 \rightharpoondown 3][1 \rightharpoondown 3][1 \hookrightarrow 3] \in \mathbf{NF}$
- $\mathbf{v} = [2 \rightharpoondown 3][2 \rightharpoondown 3][1 \rightharpoondown 3][2 \hookrightarrow 3][1 \rightharpoondown 3] \in \mathbf{NF}$

Artin generators:

- $\mathbf{u}' = \mathbf{S}_2\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\Sigma_1\Sigma_2 \in \mathbf{NF}'$
- $\mathbf{v}' = \mathbf{S}_2\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\Sigma_2\mathbf{S}_1\mathbf{S}_2 \in \mathbf{NF}'$

Automaticity: Key Ideas

Synchronous bi-automaticity if $n \leqslant 3$

Computing an automaton for R_x or L_x (with $x \in \{\mathbf{S}_1, \mathbf{S}_2\}$):

- Direct computations are too expensive!
- Replace sliding generators with Artin generators
- Recover the synchronous automaticity

Example

Sliding generators:

- $\mathbf{u} = [2 \rightharpoonup 3][2 \rightharpoonup 3][1 \rightharpoonup 3][1 \rightharpoonup 3][1 \hookrightarrow 3] \in \mathbf{NF}$
- $\mathbf{v} = [2 \rightharpoonup 3][2 \rightharpoonup 3][1 \rightharpoonup 3][2 \hookrightarrow 3][1 \rightharpoonup 3] \in \mathbf{NF}$

Artin generators:

- $\mathbf{u}' = \mathbf{S}_2\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\Sigma_1\Sigma_2 \in \mathbf{NF}'$
- $\mathbf{v}' = \mathbf{S}_2\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\Sigma_2\mathbf{S}_1\mathbf{S}_2 \in \mathbf{NF}'$

Automaticity: Key Ideas

Synchronous bi-automaticity if $n \leqslant 3$

Computing an automaton for R_x or L_x (with $x \in \{\mathbf{S}_1, \mathbf{S}_2\}$):

- Direct computations are too expensive!
- Replace sliding generators with Artin generators
- Recover the synchronous automaticity

Example

Sliding generators:

- $\mathbf{u} = [2 \rightharpoondown 3][2 \rightharpoondown 3][1 \rightharpoondown 3][1 \rightharpoondown 3][1 \hookrightarrow 3] \in \mathbf{NF}$
- $\mathbf{v} = [2 \rightharpoondown 3][2 \rightharpoondown 3][1 \rightharpoondown 3][2 \hookrightarrow 3][1 \rightharpoondown 3] \in \mathbf{NF}$

Artin generators:

- $\mathbf{u}' = \mathbf{S}_2\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\Sigma_1\Sigma_2 \in \mathbf{NF}'$
- $\mathbf{v}' = \mathbf{S}_2\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\Sigma_2\mathbf{S}_1\mathbf{S}_2 \in \mathbf{NF}'$

Automaticity: Key Ideas

Synchronous bi-automaticity if $n \leqslant 3$

Computing an automaton for R_x or L_x (with $x \in \{\mathbf{S}_1, \mathbf{S}_2\}$):

- Direct computations are too expensive!
- Replace sliding generators with Artin generators
- Recover the synchronous automaticity

Example

Sliding generators:

- $\mathbf{u} = [2 \rightharpoondown 3][2 \rightharpoondown 3][1 \rightharpoondown 3][1 \rightharpoondown 3][1 \hookrightarrow 3] \in \mathbf{NF}$
- $\mathbf{v} = [2 \rightharpoondown 3][2 \rightharpoondown 3][1 \rightharpoondown 3][2 \hookrightarrow 3][1 \rightharpoondown 3] \in \mathbf{NF}$

Artin generators:

- $\mathbf{u}' = \mathbf{S}_2\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\Sigma_1\Sigma_2 \in \mathbf{NF}'$
- $\mathbf{v}' = \mathbf{S}_2\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\Sigma_2\mathbf{S}_1\mathbf{S}_2 \in \mathbf{NF}'$

Automaticity: Key Ideas

Synchronous bi-automaticity if $n \leqslant 3$

Computing an automaton for R_x or L_x (with $x \in \{\mathbf{S}_1, \mathbf{S}_2\}$):

- Direct computations are too expensive!
- Replace sliding generators with Artin generators
- Recover the synchronous automaticity

Example

Sliding generators:

- $\mathbf{u} = [2 \rightharpoondown 3][2 \rightharpoondown 3][1 \rightharpoondown 3][1 \rightharpoondown 3][1 \hookrightarrow 3] \in \mathbf{NF}$
- $\mathbf{v} = [2 \rightharpoondown 3][2 \rightharpoondown 3][1 \rightharpoondown 3][2 \hookrightarrow 3][1 \rightharpoondown 3] \in \mathbf{NF}$

Artin generators:

- $\mathbf{u}' = \mathbf{S}_2\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\Sigma_1\Sigma_2 \in \mathbf{NF}'$
- $\mathbf{v}' = \mathbf{S}_2\mathbf{S}_2\mathbf{S}_1\mathbf{S}_2\Sigma_2\mathbf{S}_1\mathbf{S}_2 \in \mathbf{NF}'$

Automaticity: Key Ideas

Synchronous bi-automaticity if $n \leqslant 3$

Computing an automaton for R_x or L_x (with $x \in \{\mathbf{S}_1, \mathbf{S}_2\}$):

- Direct computations are too expensive!
- Replace sliding generators with Artin generators
- Recover the synchronous automaticity

Asynchronous left-automaticity if $n = 4$

Finding a suitable padding function φ:

- General strategies work but are computationnally too expensive!

Automaticity: Key Ideas

Synchronous bi-automaticity if $n \leqslant 3$

Computing an automaton for R_x or L_x (with $x \in \{\mathbf{S}_1, \mathbf{S}_2\}$):

- Direct computations are too expensive!
- Replace sliding generators with Artin generators
- Recover the synchronous automaticity

Asynchronous left-automaticity if $n = 4$

Finding a suitable padding function φ:

- General strategies work but are computationnally too expensive!
- Use heuristics and evaluate the efficiency of the strategy

Automaticity: Key Ideas

Synchronous bi-automaticity if $n \leqslant 3$

Computing an automaton for R_x or L_x (with $x \in \{\mathbf{S}_1, \mathbf{S}_2\}$):

- Direct computations are too expensive!
- Replace sliding generators with Artin generators
- Recover the synchronous automaticity

Asynchronous left-automaticity if $n \geqslant 4$

Finding a suitable padding function φ:

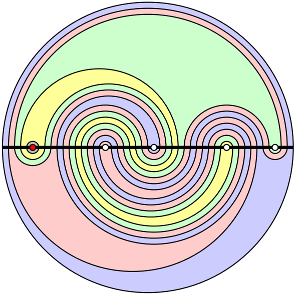
- General strategies work but are computationnally too expensive!
- Use heuristics and evaluate the efficiency of the strategy
- Space issues are intractable for $n = 5$

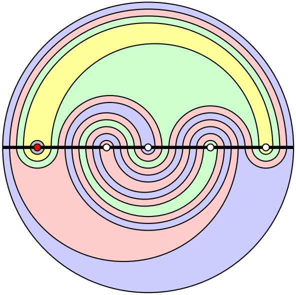
Automaticity: Key Ideas

No synchronous left automaticity if $n \geqslant 4$

Find witnesses falsifying the synchronous fellow-traveller property:

- $\mathbf{S}_1 \cdot [2 \rightharpoonup 4] \cdot ([1 \rightharpoonup 3] \cdot [1 \smile 4] \cdot [3 \rightharpoonup 4])^\ell = ([2 \rightharpoonup 4] \cdot [2 \smile 4])^\ell \cdot [1 \rightharpoonup 4]$





Automaticity: Key Ideas

No synchronous left automaticity if $n \geqslant 4$

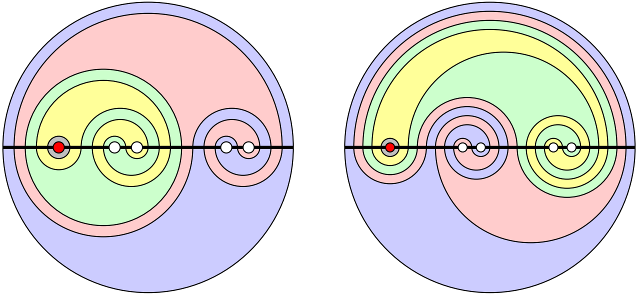
Find witnesses falsifying the synchronous fellow-traveller property:

- $\mathbf{S}_1 \cdot [2 \rightharpoondown 4] \cdot ([1 \rightharpoondown 3] \cdot [1 \smile 4] \cdot [3 \rightharpoondown 4])^\ell = ([2 \rightharpoondown 4] \cdot [2 \smile 4])^\ell \cdot [1 \rightharpoondown 4]$

No asynchronous right automaticity if $n \geqslant 4$

Find witnesses falsifying the asynchronous fellow-traveller property:

- $[1 \rightharpoondown 2]^\ell \cdot [3 \rightharpoondown 4]^\ell \cdot \Delta_3 = [3 \rightharpoondown 4]^{\ell+1} \cdot [1 \rightharpoondown 4]^2 \cdot [3 \rightharpoondown 4]^{\ell-1}$



Automaticity: Key Ideas

No synchronous left automaticity if $n \geqslant 4$

Find witnesses falsifying the synchronous fellow-traveller property:

- $\mathbf{S}_1 \cdot [2 \rightharpoondown 4] \cdot ([1 \rightharpoondown 3] \cdot [1 \smile 4] \cdot [3 \rightharpoondown 4])^\ell = ([2 \rightharpoondown 4] \cdot [2 \smile 4])^\ell \cdot [1 \rightharpoondown 4]$

No asynchronous right automaticity if $n \geqslant 4$

Find witnesses falsifying the asynchronous fellow-traveller property:

- $[1 \rightharpoondown 2]^\ell \cdot [3 \rightharpoondown 4]^\ell \cdot \Delta_3 = [3 \rightharpoondown 4]^{\ell+1} \cdot [1 \rightharpoondown 4]^2 \cdot [3 \rightharpoondown 4]^{\ell-1}$

$\Rightarrow$ Witnesses found while trying to compute automata L_x and R_x^φ!
$\Rightarrow$ Yet no witness disproving the asynchronous left automaticity for $n = 4$

Contents

Conclusion

Next Goals

- Prove or disprove the conjecture
- Get published
- Discuss with you

Conclusion

Next Goals

- Prove or disprove the conjecture
- Get published
- Discuss with you

Thank you!