THE ISOTOPY PROBLEM OF BRAIDS

Patrick Dehornoy

Laboratoire de Mathématiques
Nicolas Oresme, Caen

THE ISOTOPY PROBLEM OF BRAIDS

Patrick Dehornoy

Laboratoire de Mathématiques
Nicolas Oresme, Caen

- A problem of medium difficulty:

THE ISOTOPY PROBLEM OF BRAIDS

Patrick Dehornoy

Laboratoire de Mathématiques
Nicolas Oresme, Caen

• A problem of medium difficulty:
many efficient solutions,

THE ISOTOPY PROBLEM OF BRAIDS

Patrick Dehornoy

Laboratoire de Mathématiques
Nicolas Oresme, Caen

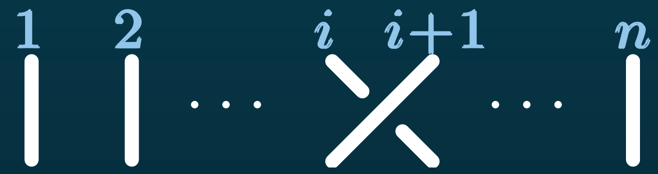
- A problem of medium difficulty:
  many efficient solutions,
  but all involving (requiring)
  some nontrivial theory behind.

- Artin's braid group B_n : $\left\langle \sigma_1, ..., \sigma_{n-1} \;\middle|\; \begin{array}{ll} \sigma_i\sigma_j = \sigma_j\sigma_i & \text{for } |i-j| \geqslant 2 \\ \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j & \text{for } |i-j| = 1 \end{array} \right\rangle$

- Artin's braid group B_n : $\left\langle \sigma_1, \ldots, \sigma_{n-1} \;\middle|\; \begin{array}{ll} \sigma_i\sigma_j = \sigma_j\sigma_i & \text{for } |i-j| \geqslant 2 \\ \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j & \text{for } |i-j| = 1 \end{array} \right\rangle$

  $\simeq \{$ braid diagrams $\}/$ isotopy

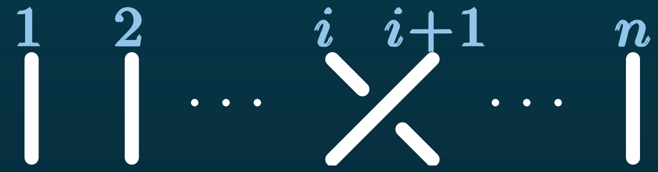
$\sigma_i \quad \leftrightsquigarrow$



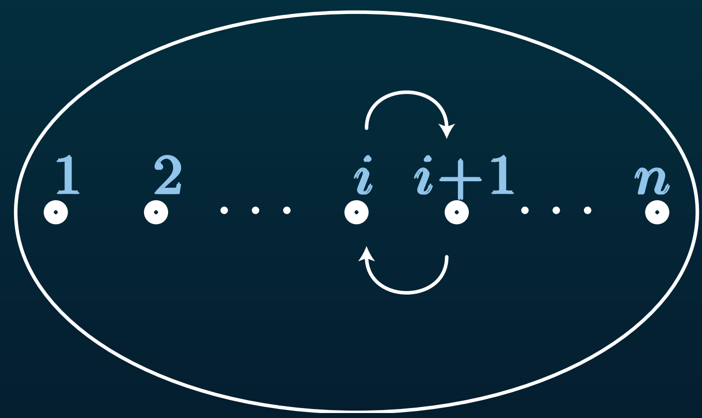
THE BRAID ISOTOPY PROBLEM

- Artin's braid group B_n : $\left\langle \sigma_1, ..., \sigma_{n-1} \;\middle|\; \begin{array}{ll} \sigma_i\sigma_j = \sigma_j\sigma_i & \text{for } |i-j| \geqslant 2 \\ \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j & \text{for } |i-j| = 1 \end{array} \right\rangle$

  $\simeq \{$ braid diagrams $\}/$ isotopy

  $\sigma_i \leftrightsquigarrow$ 

  $\simeq$ mapping class group (D_n)

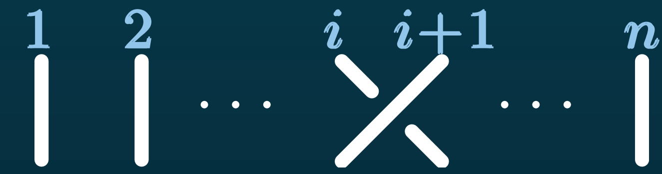
  $\sigma_i \leftrightsquigarrow$ 

THE BRAID ISOTOPY PROBLEM

- Artin's braid group B_n : $\left\langle \sigma_1, \ldots, \sigma_{n-1} \;\middle|\; \begin{array}{ll} \sigma_i\sigma_j = \sigma_j\sigma_i & \text{for } |i-j| \geqslant 2 \\ \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j & \text{for } |i-j| = 1 \end{array} \right\rangle$

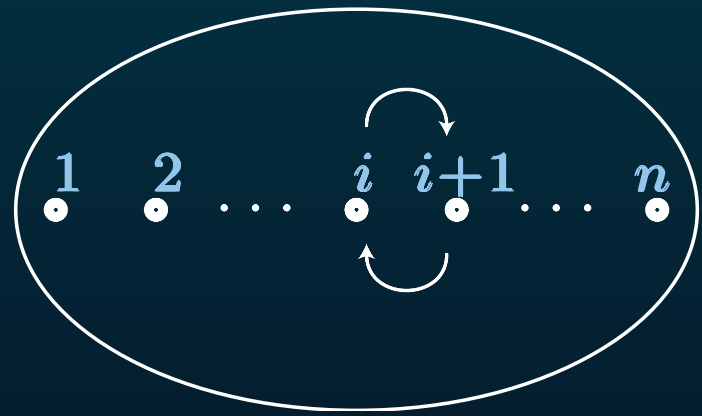
  $\simeq$ { braid diagrams }/ isotopy

  $\sigma_i \rightsquigarrow$



  $\simeq$ mapping class group (D_n)

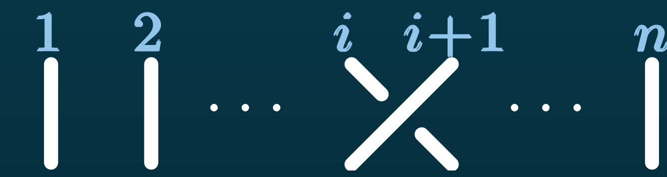
  $\sigma_i \rightsquigarrow$



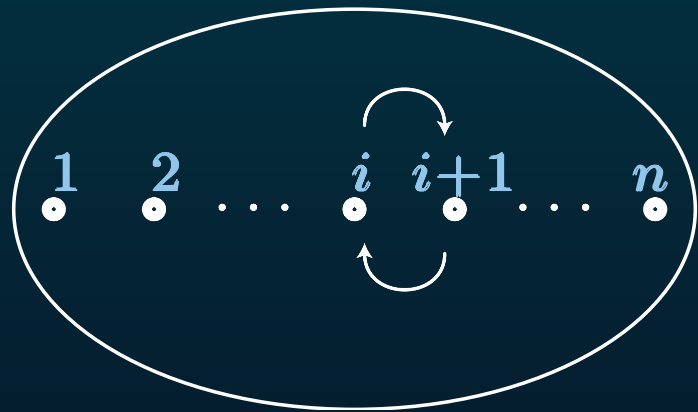
- Isotopy problem of braids = word problem of B_n = decide isotopy of 2 braid diagrams, ***i.e.***, equivalence of 2 braid words

- Artin's braid group B_n : $\left\langle \sigma_1, ..., \sigma_{n-1} \,\middle|\, \begin{array}{ll} \sigma_i\sigma_j = \sigma_j\sigma_i & \text{for } |i-j| \geqslant 2 \\ \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j & \text{for } |i-j| = 1 \end{array} \right\rangle$

  $\simeq$ $\{$ braid diagrams $\}/$ isotopy

  σ_i ⇝ 

  $\simeq$ mapping class group (D_n)

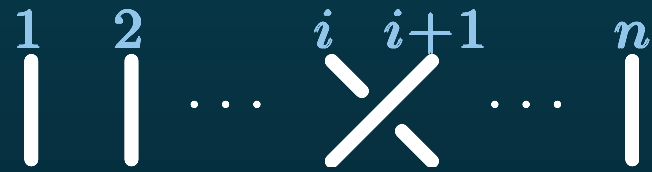
  σ_i ⇝ 

- Isotopy problem of braids = word problem of B_n =
  decide isotopy of 2 braid diagrams, ***i.e.***, equivalence of 2 braid words

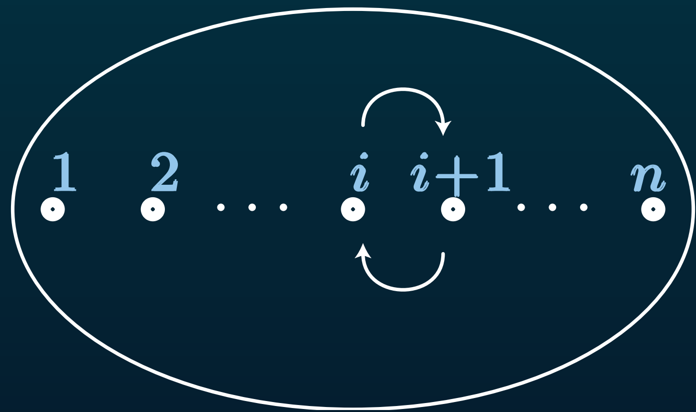
⇝ decide triviality of 1 braid diagram, ***i.e.***, equivalence of 1 braid word to ε

- Artin's braid group B_n : $\left\langle \sigma_1, ..., \sigma_{n-1} \;\middle|\; \begin{matrix} \sigma_i\sigma_j = \sigma_j\sigma_i & \text{for } |i-j| \geqslant 2 \\ \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j & \text{for } |i-j| = 1 \end{matrix} \right\rangle$

  $\simeq$ $\{$ braid diagrams $\}/$ isotopy

  σ_i ⟿ 

  $\simeq$ mapping class group (D_n)

  σ_i ⟿ 

- Isotopy problem of braids = word problem of B_n =
  decide isotopy of 2 braid diagrams, ***i.e.***, equivalence of 2 braid words

⟿ decide triviality of 1 braid diagram, ***i.e.***, equivalence of 1 braid word to ε

(E. Artin, $\sim$ 1925)

- Consider (pure) braids, and obtain a unique decomposition using combing:

(E. Artin, ∼ 1925)

- Consider (pure) braids, and obtain a unique decomposition using combing:

$$\begin{array}{c} \quad\ \uparrow \\ 1 \to PB_n \to B_n \to S_n \to 1 \end{array}$$

(E. Artin, $\sim$ 1925)

- Consider (pure) braids, and obtain a unique decomposition using combing:

$$1 \to PB_n \to B_n \to S_n \to 1$$

"Remove the nth strand, comb, then reintroduce the nth strand".

(E. Artin, ∼ 1925)

- Consider (pure) braids, and obtain a unique decomposition using combing:

$$\begin{array}{c}\uparrow\\ 1 \to PB_n \to B_n \to S_n \to 1\end{array}$$

"Remove the nth strand, comb, then reintroduce the nth strand".

- Behind: PB_n = semi-direct product of free groups:

$$PB_n \cong PB_{n-1} \rtimes \pi_1(D_{n-1})$$

SOLUTION 1: COMBING

(E. Artin, $\sim$ 1925)

- Consider (pure) braids, and obtain a unique decomposition using combing:

$$\begin{array}{c} \quad\uparrow \\ 1 \to PB_n \to B_n \to S_n \to 1 \end{array}$$

"Remove the nth strand, comb, then reintroduce the nth strand".

- Behind: PB_n = semi-direct product of free groups:

$$PB_n \cong PB_{n-1} \ltimes \pi_1(D_{n-1})$$

$\nwarrow$ a free group of rank $n-1$

SOLUTION 2: AUTOMORPHISMS OF A FREE GROUP

(E. Artin, $\sim$ 1925)

(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

SOLUTION 2: AUTOMORPHISMS OF A FREE GROUP

(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

  an n strand braid

  = the movie of the dance

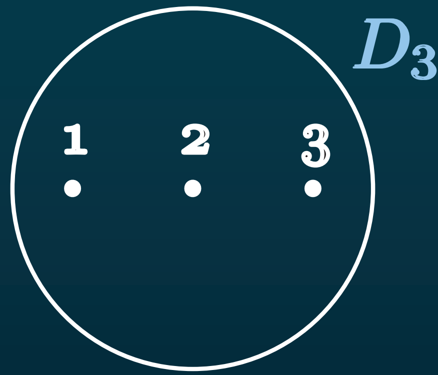
  of n points in a disk

SOLUTION 2: AUTOMORPHISMS OF A FREE GROUP

(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

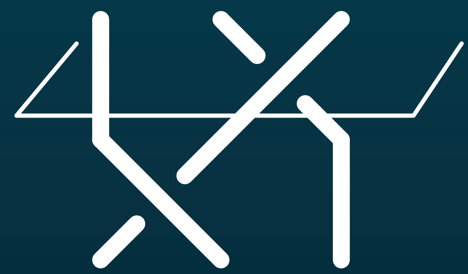
  an n strand braid
  = the movie of the dance
  of n points in a disk

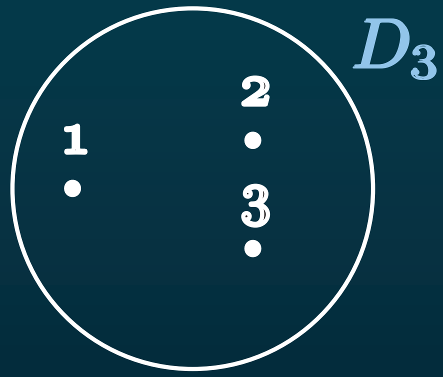


(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

an n strand braid
= the movie of the dance
of n points in a disk



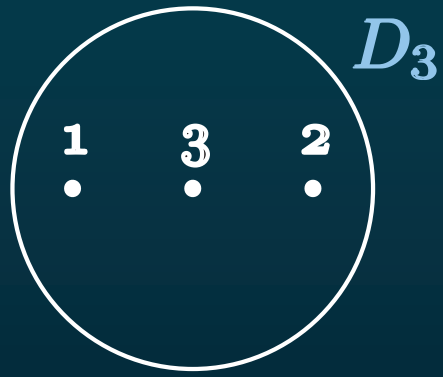


SOLUTION 2: AUTOMORPHISMS OF A FREE GROUP

(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

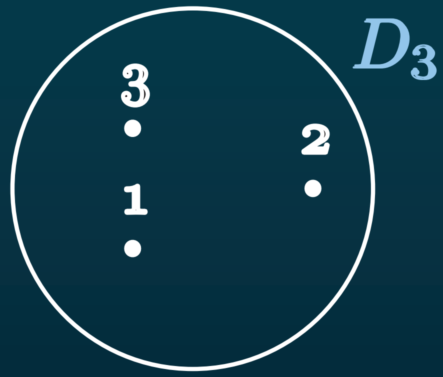
an n strand braid
= the movie of the dance
of n points in a disk



(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

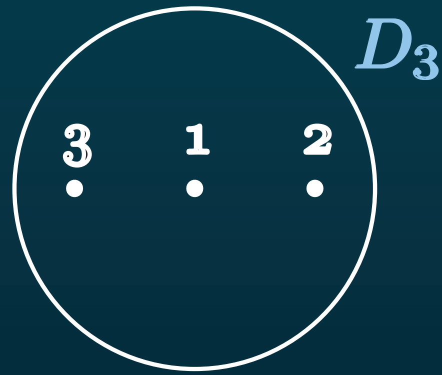
an n strand braid
= the movie of the dance
of n points in a disk



(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

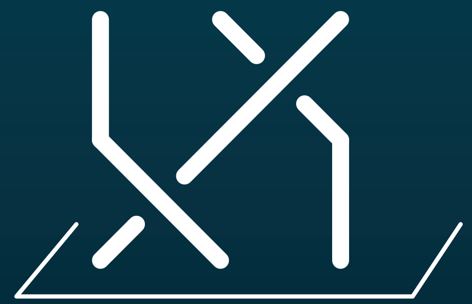
an n strand braid
= the movie of the dance
of n points in a disk

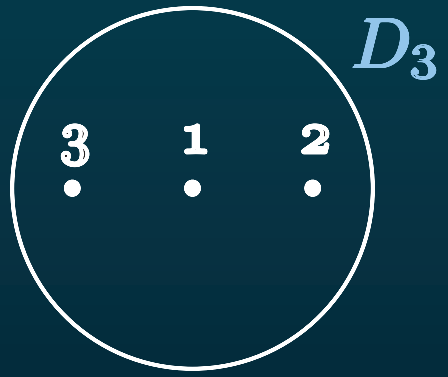


(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

an n strand braid
= the movie of the dance
of n points in a disk



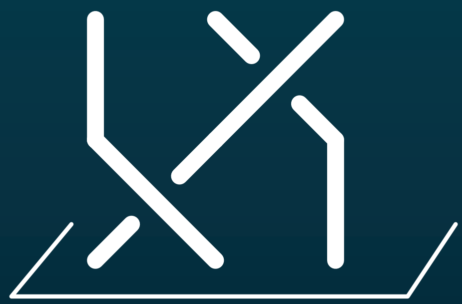


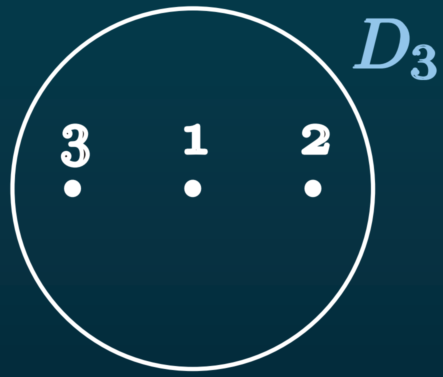
$\rightsquigarrow$ = a homeomorphism of D_n fixing ∂D_n and $\{$punctures$\}$, up to homotopy.

(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

an n strand braid
= the movie of the dance
of n points in a disk





⇝ = a homeomorphism of D_n fixing ∂D_n and $\{$punctures$\}$, up to homotopy.

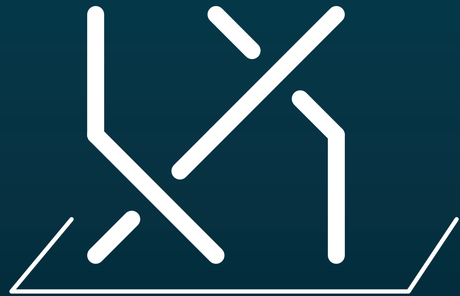
- Whence: action of B_n on $\pi_1(D_n)$,
a free group of rank n.

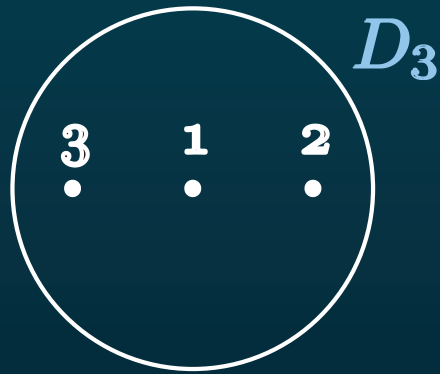
SOLUTION 2: AUTOMORPHISMS OF A FREE GROUP

(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

an n strand braid
= the movie of the dance
of n points in a disk





$\rightsquigarrow$ = a homeomorphism of D_n fixing ∂D_n and $\{$punctures$\}$, up to homotopy.

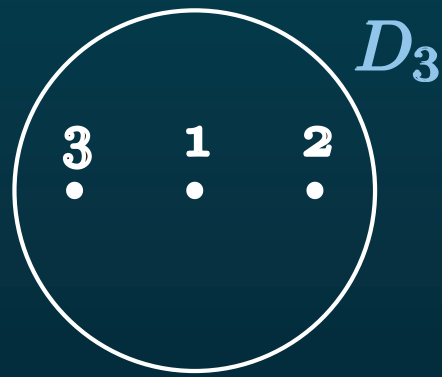
- Whence: action of B_n on $\pi_1(D_n)$,
a free group of rank n.

SOLUTION 2: AUTOMORPHISMS OF A FREE GROUP

(E. Artin, $\sim$ 1925)

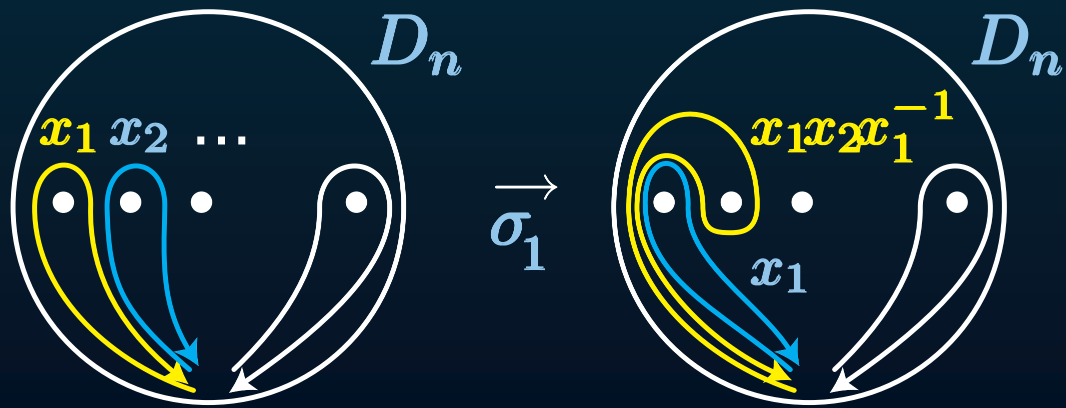
- $B_n \cong$ mapping class group of a disk with n punctures:

an n strand braid
= the movie of the dance
of n points in a disk



$\rightsquigarrow$ = a homeomorphism of D_n fixing ∂D_n and $\{$punctures$\}$, up to homotopy.

- Whence: action of B_n on $\pi_1(D_n)$,
a free group of rank n.

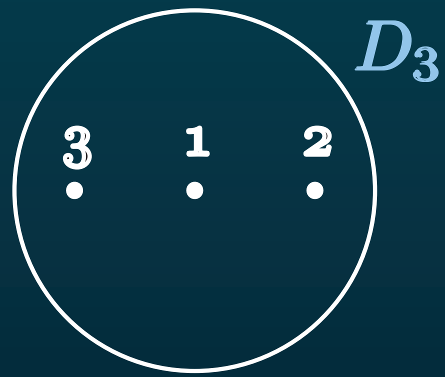


SOLUTION 2: AUTOMORPHISMS OF A FREE GROUP

(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

  an n strand braid
  = the movie of the dance
  of n points in a disk

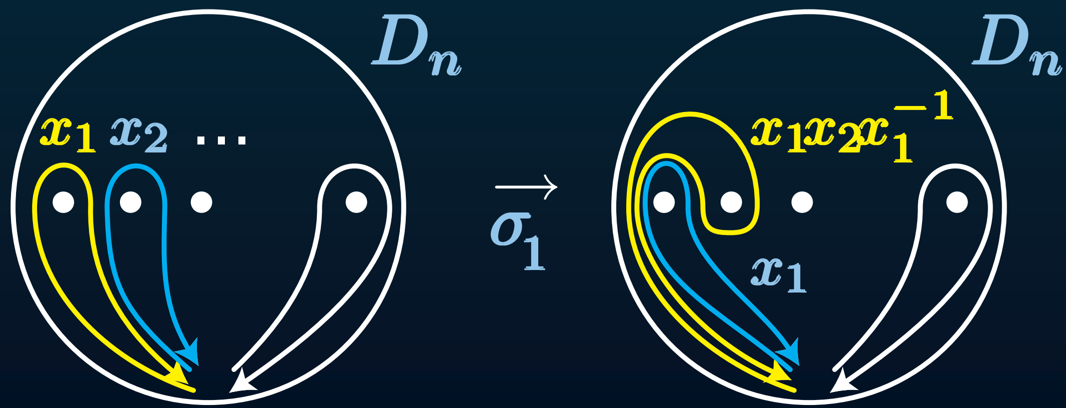


$\rightsquigarrow$ = a homeomorphism of D_n fixing ∂D_n and $\{$punctures$\}$, up to homotopy.

- Whence: action of B_n on $\pi_1(D_n)$,
  a free group of rank n.

$\rightsquigarrow$ read on loops:

$$\sigma_1 : x_i \mapsto \begin{cases} x_1 x_2 x_1^{-1} & \text{for } i = 1, \\ & \end{cases}$$

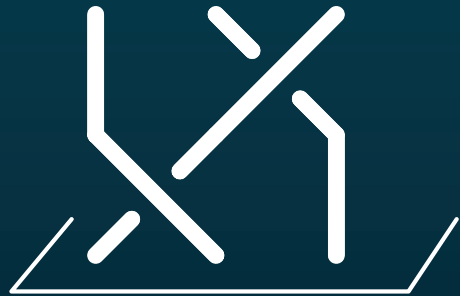


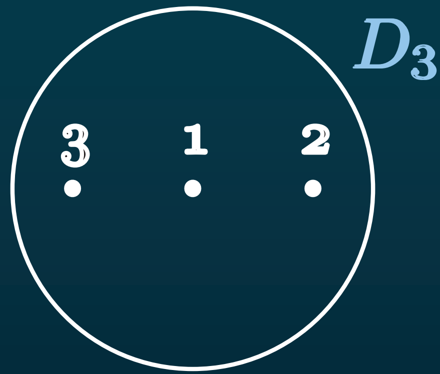
SOLUTION 2: AUTOMORPHISMS OF A FREE GROUP

(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

an n strand braid
= the movie of the dance
of n points in a disk



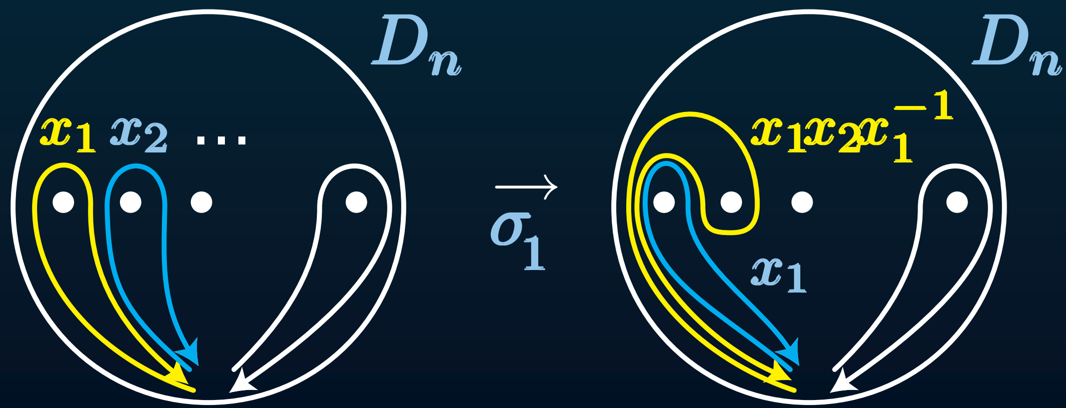


$\rightsquigarrow$ = a homeomorphism of D_n fixing ∂D_n and $\{$punctures$\}$, up to homotopy.

- Whence: action of B_n on $\pi_1(D_n)$,
a free group of rank n.

$\rightsquigarrow$ read on loops:

$$\sigma_1 : x_i \mapsto \begin{cases} x_1 x_2 x_1^{-1} & \text{for } i = 1, \\ x_1 & \text{for } i = 2, \end{cases}$$

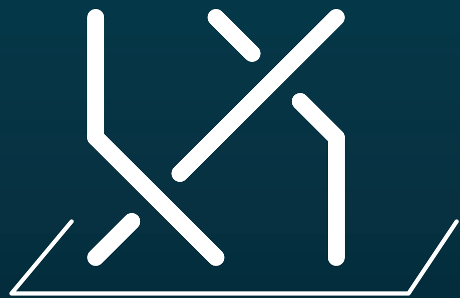


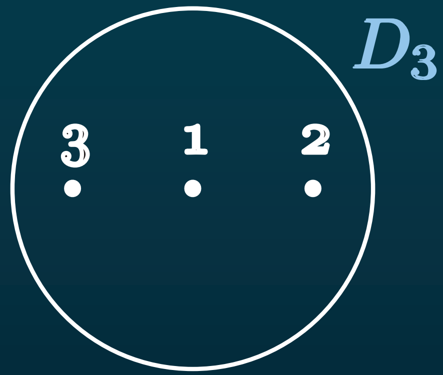
SOLUTION 2: AUTOMORPHISMS OF A FREE GROUP

(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

an n strand braid
= the movie of the dance
of n points in a disk



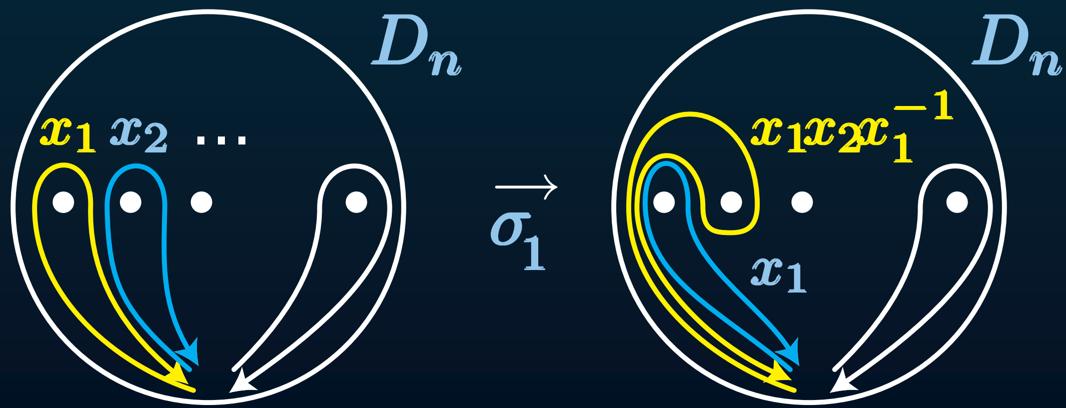


⇝ = a homeomorphism of D_n fixing ∂D_n and $\{$punctures$\}$, up to homotopy.

- Whence: action of B_n on $\pi_1(D_n)$,
a free group of rank n.

⇝ read on loops:

$$\sigma_1 : x_i \mapsto \begin{cases} x_1 x_2 x_1^{-1} & \text{for } i = 1, \\ x_1 & \text{for } i = 2, \\ x_i & \text{for } i \geqslant 3. \end{cases}$$

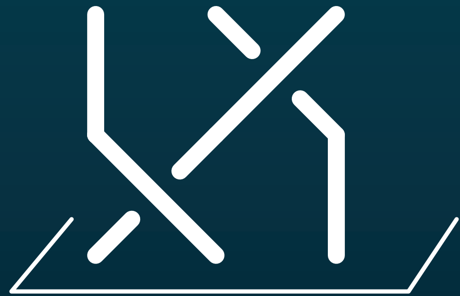


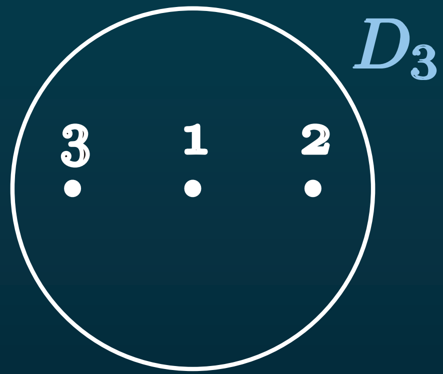
SOLUTION 2: AUTOMORPHISMS OF A FREE GROUP

(E. Artin, $\sim$ 1925)

- $B_n \cong$ mapping class group of a disk with n punctures:

an n strand braid
= the movie of the dance
of n points in a disk



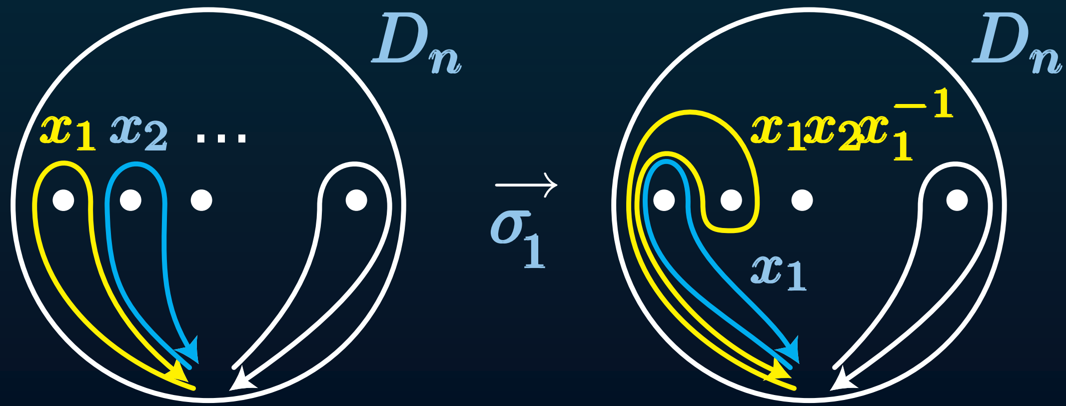


⇝ = a homeomorphism of D_n fixing ∂D_n and $\{$punctures$\}$, up to homotopy.

- Whence: action of B_n on $\pi_1(D_n)$,
a free group of rank n.

⇝ read on loops:

$$\sigma_1 : x_i \mapsto \begin{cases} x_1 x_2 x_1^{-1} & \text{for } i = 1, \\ x_1 & \text{for } i = 2, \\ x_i & \text{for } i \geqslant 3. \end{cases}$$



- Then: $B_n \hookrightarrow \mathtt{Aut}(F_n)$, hence solution to the braid isotopy problem.

SOLUTION 3: GREEDY NORMAL FORM

(Garside, 1967, Deligne, Adjan, Thurston, Morton-El Rifai...)

(Garside, 1967, Deligne, Adjan, Thurston, Morton-El Rifai...)

- The group B_n is a group of fractions for the monoid B_n^+ ($:= \langle \dots \rangle^+$)

(Garside, 1967, Deligne, Adjan, Thurston, Morton-El Rifai...)

- The group B_n is a group of fractions for the monoid B_n^+ ($:= \langle \dots \rangle^+$)
  $+$ there exists some "universal" denominator Δ_n.

(Garside, 1967, Deligne, Adjan, Thurston, Morton-El Rifai...)

- The group B_n is a group of fractions for the monoid B_n^+ ($:= \langle \dots \rangle^+$)
  $+$ there exists some "universal" denominator Δ_n.

$\rightsquigarrow$ more precisely: (B_n^+, Δ_n) s.t.
- B_n^+ is cancellative, has no invertible element, admits lcm's,
- left divisors (Δ_n) $=$right divisors (Δ_n), and generate B_n^+.

(Garside, 1967, Deligne, Adjan, Thurston, Morton-El Rifai...)

- The group B_n is a group of fractions for the monoid B_n^+ ($:= \langle \dots \rangle^+$)
  $+$ there exists some "universal" denominator Δ_n.

⇝ more precisely: (B_n^+, Δ_n) s.t.
- B_n^+ is cancellative, has no invertible element, admits lcm's,
- left divisors (Δ_n) $=$right divisors (Δ_n), and generate B_n^+.

"Δ_n is a Garside element in B_n^+"

⇝ B_n is a Garside group.

(Garside, 1967, Deligne, Adjan, Thurston, Morton-El Rifai...)

- The group B_n is a group of fractions for the monoid B_n^+ ($:= \langle \ldots \rangle^+$)
  $+$ there exists some "universal" denominator Δ_n.

  ⇝ more precisely: (B_n^+, Δ_n) s.t.
  - B_n^+ is cancellative, has no invertible element, admits lcm's,
  - left divisors (Δ_n) $=$right divisors (Δ_n), and generate B_n^+.

  "Δ_n is a Garside element in B_n^+"

  ⇝ B_n is a Garside group.

- Then: Every element of B_n has a unique expression $x_p^{-1}...x_1^{-1}y_1...y_q$ with

(Garside, 1967, Deligne, Adjan, Thurston, Morton-El Rifai...)

- The group B_n is a group of fractions for the monoid B_n^+ ($:= \langle \dots \rangle^+$)
  $+$ there exists some "universal" denominator Δ_n.

  ⇝ more precisely: (B_n^+, Δ_n) s.t.
  - B_n^+ is cancellative, has no invertible element, admits lcm's,
  - left divisors (Δ_n) $=$right divisors (Δ_n), and generate B_n^+.

  "Δ_n is a Garside element in B_n^+"

  ⇝ B_n is a Garside group.

- Then: Every element of B_n has a unique expression $x_p^{-1} ... x_1^{-1} y_1 ... y_q$ with
  - $x_1, ..., x_p, y_1, ..., y_p$ divisors of Δ_n,

(Garside, 1967, Deligne, Adjan, Thurston, Morton-El Rifai...)

- The group B_n is a group of fractions for the monoid B_n^+ ($:= \langle \ldots \rangle^+$)

  $+$ there exists some "universal" denominator Δ_n.

  ⇝ more precisely: (B_n^+, Δ_n) s.t.

  - B_n^+ is cancellative, has no invertible element, admits lcm's,
  - left divisors (Δ_n) $=$right divisors (Δ_n), and generate B_n^+.

  "Δ_n is a Garside element in B_n^+"

  ⇝ B_n is a Garside group.

- Then: Every element of B_n has a unique expression $x_p^{-1}...x_1^{-1}y_1...y_q$ with
  - $x_1, ..., x_p, y_1, ..., y_p$ divisors of Δ_n,
  - $x_i = \mathbf{gcd}(x_i x_{i+1}, \Delta_n)$ for each i, ***id.*** for y_j's, and $\mathbf{gcd}(x_1, y_1) = 1$.

(can be computed in quadratic time, for each fixed n)

SOLUTION 3: GREEDY NORMAL FORM

(Garside, 1967, Deligne, Adjan, Thurston, Morton-El Rifai...)

- The group B_n is a group of fractions for the monoid B_n^+ ($:= \langle \ldots \rangle^+$)
  $+$ there exists some "universal" denominator Δ_n.

  ⇝ more precisely: (B_n^+, Δ_n) s.t.
  - B_n^+ is cancellative, has no invertible element, admits lcm's,
  - left divisors (Δ_n) $=$right divisors (Δ_n), and generate B_n^+.

  "Δ_n is a Garside element in B_n^+"

  ⇝ B_n is a Garside group.

- Then: Every element of B_n has a unique expression $x_p^{-1} ... x_1^{-1} y_1 ... y_q$ with
  - $x_1, ..., x_p, y_1, ..., y_p$ divisors of Δ_n,
  - $x_i = \mathbf{gcd}(x_i x_{i+1}, \Delta_n)$ for each i, ***id.*** for y_j's, and $\mathbf{gcd}(x_1, y_1) = 1$.

  (can be computed in quadratic time, for each fixed n)

- Behind: automatic structure of B_n (Cannon, Thurston)

"redresses to"

$\downarrow$

- Say that $w \curvearrowright w'$ holds if w' obtained from w by (iteratively)

"redresses to"

↓

- Say that $w \curvearrowright w'$ holds if w' obtained from w by (iteratively)
  - deleting some $\sigma_i^{-1}\sigma_i$, or

"redresses to"

$\downarrow$

- Say that $w \curvearrowright w'$ holds if w' obtained from w by (iteratively)
  - deleting some $\sigma_i^{-1}\sigma_i$, or
  - replacing some $\sigma_i^{-1}\sigma_j$ with $\sigma_j\sigma_i^{-1}$

$\uparrow$

case $|i - j| \geqslant 2$

"redresses to"

$\downarrow$

- Say that $w \curvearrowright w'$ holds if w' obtained from w by (iteratively)
  - deleting some $\sigma_i^{-1}\sigma_i$, or
  - replacing some $\sigma_i^{-1}\sigma_j$ with $\sigma_j\sigma_i^{-1}$, ***resp.*** $\sigma_j\sigma_i\sigma_j^{-1}\sigma_i^{-1}$.

$\uparrow$ case $|i-j| \geqslant 2$ $\uparrow$ case $|i-j| = 1$

"redresses to"

$\downarrow$

- Say that $w \curvearrowright w'$ holds if w' obtained from w by (iteratively)
  - deleting some $\sigma_i^{-1}\sigma_i$, or
  - replacing some $\sigma_i^{-1}\sigma_j$ with $\sigma_j\sigma_i^{-1}$, ***resp.*** $\sigma_j\sigma_i\sigma_j^{-1}\sigma_i^{-1}$.

$\uparrow$ case $|i-j| \geqslant 2$ $\uparrow$ case $|i-j| = 1$

- Then: - Every braid word redresses to a unique word of the form uv^{-1} with u, v positive (= no σ_i^{-1});

"redresses to"
↓
- Say that $w \curvearrowright w'$ holds if w' obtained from w by (iteratively)
  - deleting some $\sigma_i^{-1}\sigma_i$, or
  - replacing some $\sigma_i^{-1}\sigma_j$ with $\sigma_j\sigma_i^{-1}$, ***resp.*** $\sigma_j\sigma_i\sigma_j^{-1}\sigma_i^{-1}$.

↑ case $|i-j| \geqslant 2$ ↑ case $|i-j| = 1$

- Then: - Every braid word redresses to a unique word of the form uv^{-1} with u, v positive (= no σ_i^{-1});
  - The braid word w represents 1 in B_n iff, for some positive u, v, $w \curvearrowright uv^{-1}$ and $v^{-1}u \curvearrowright \varepsilon$ (= empty word).

"redresses to"
↓

- Say that $w \curvearrowright w'$ holds if w' obtained from w by (iteratively)
  - deleting some $\sigma_i^{-1}\sigma_i$, or
  - replacing some $\sigma_i^{-1}\sigma_j$ with $\sigma_j\sigma_i^{-1}$, ***resp.*** $\sigma_j\sigma_i\sigma_j^{-1}\sigma_i^{-1}$.

  ↑ case $|i-j| \geqslant 2$ ↑ case $|i-j| = 1$

- Then: - Every braid word redresses to a unique word of the form uv^{-1} with u, v positive (= no σ_i^{-1});
  - The braid word w represents 1 in B_n iff, for some positive u, v,
  $w \curvearrowright uv^{-1}$ and $v^{-1}u \curvearrowright \varepsilon$ (= empty word).

- Behind: Garside theory again.

- Free group reduction: delete $\sigma_i \sigma_i^{-1}$ or $\sigma_i^{-1} \sigma_i$;

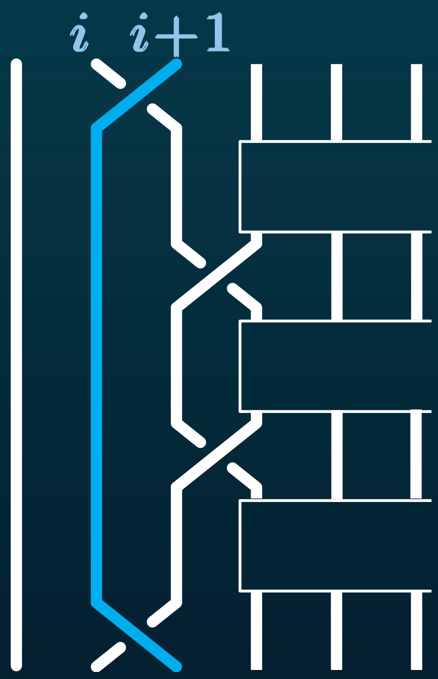
  ⇝ if w reduces to ε, then w represents 1; no converse, as B_n not free.

- Free group reduction: delete $\sigma_i \sigma_i^{-1}$ or $\sigma_i^{-1} \sigma_i$;

  ⇝ if w reduces to ε, then w represents 1; no converse, as B_n not free.

- Nevertheless:

  Define a σ_i-handle to be:

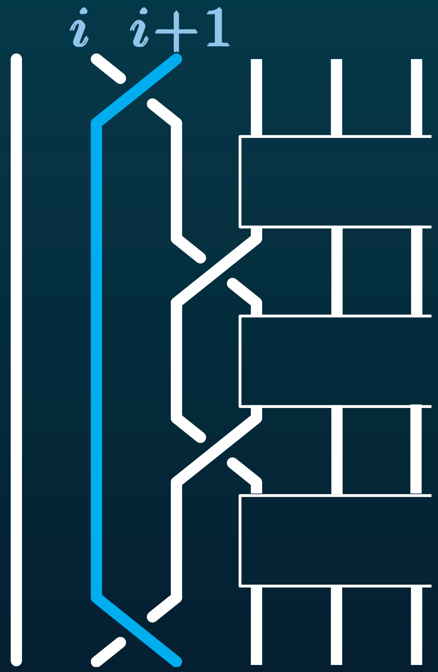


- Free group reduction: delete $\sigma_i\sigma_i^{-1}$ or $\sigma_i^{-1}\sigma_i$;

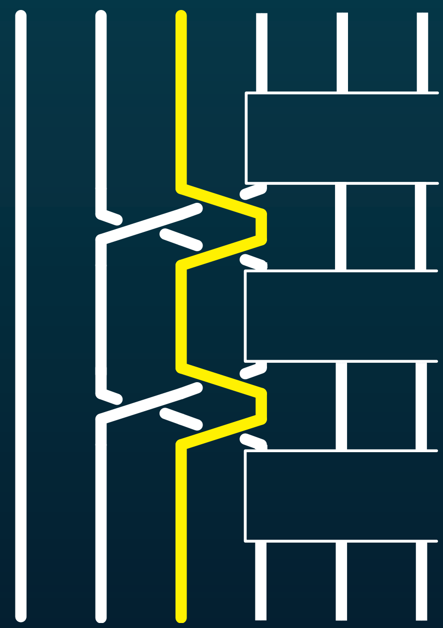
  ⇝ if w reduces to ε, then w represents 1; no converse, as B_n not free.

- Nevertheless:

  Define a σ_i-handle to be:



  and its reduction to be:

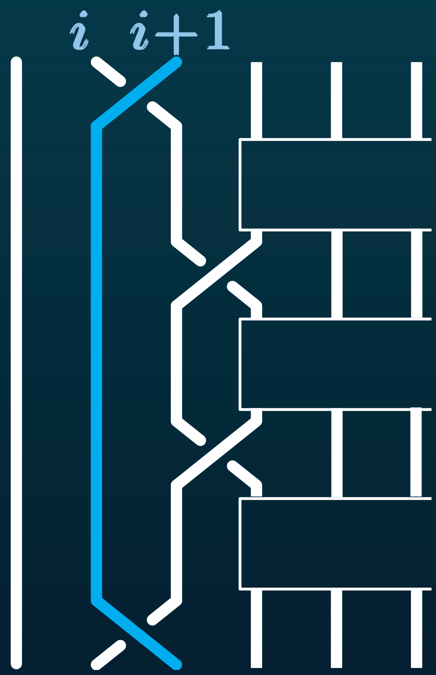


- Free group reduction: delete $\sigma_i \sigma_i^{-1}$ or $\sigma_i^{-1} \sigma_i$;

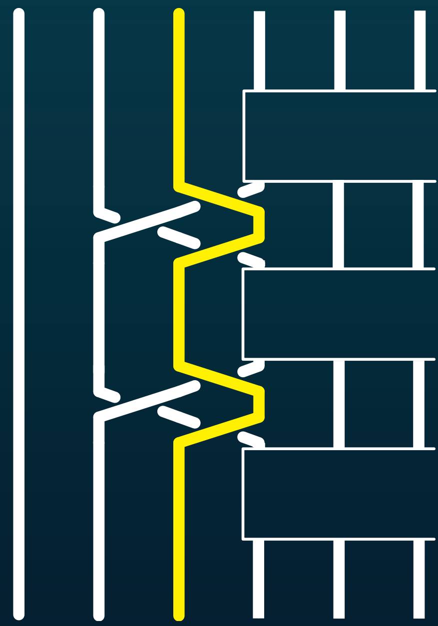
  ⇝ if w reduces to ε, then w represents 1; no converse, as B_n not free.

- Nevertheless:

  Define a σ_i-handle to be:



  and its reduction to be:



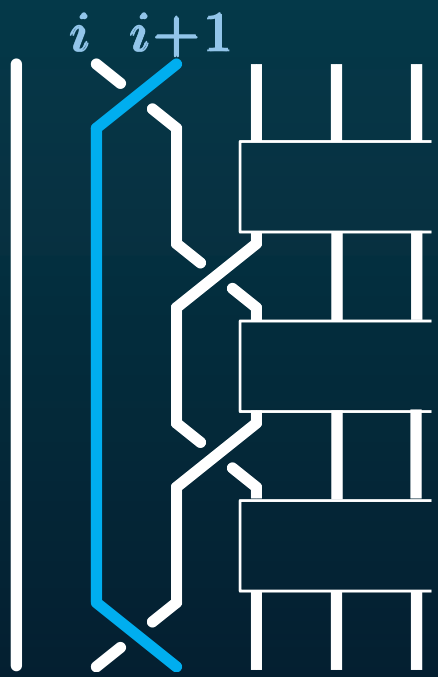
- Then: A braid word represents 1 iff it reduces to ε.

SOLUTION 5: HANDLE REDUCTION

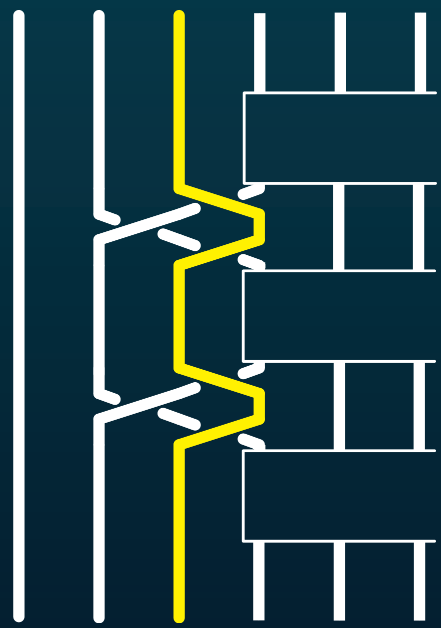
- Free group reduction: delete $\sigma_i \sigma_i^{-1}$ or $\sigma_i^{-1} \sigma_i$;
  ⇝ if w reduces to ε, then w represents 1; no converse, as B_n not free.

- Nevertheless:

  Define a σ_i-handle to be:



  and its reduction to be:



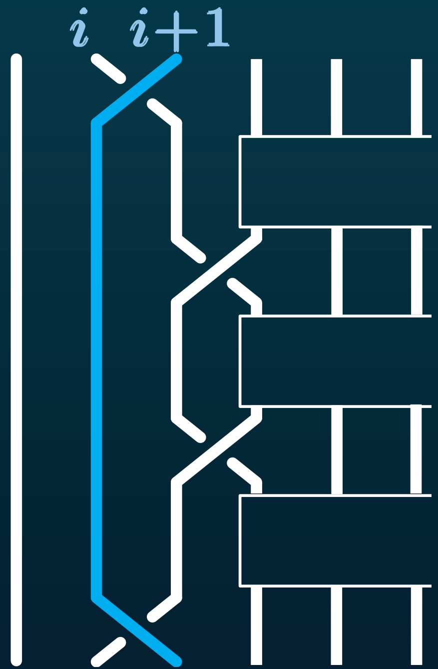
- Then: A braid word represents 1 iff it reduces to ε.

  (extremely efficient in practice)

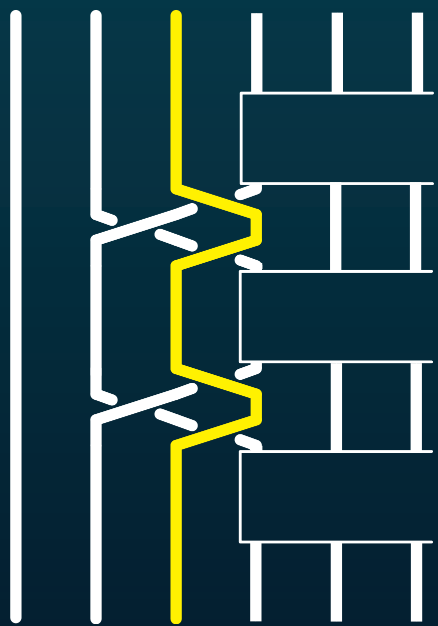
- Free group reduction: delete $\sigma_i \sigma_i^{-1}$ or $\sigma_i^{-1} \sigma_i$;
  $\rightsquigarrow$ if w reduces to ε, then w represents 1; no converse, as B_n not free.

- Nevertheless:

  Define a σ_i-handle to be:



  and its reduction to be:



- Then: A braid word represents 1 iff it reduces to ε.

  (extremely efficient in practice)

- Behind: Garside theory $+$ order properties.

- Cayley graph of B_n: vertices = braids; edges: $\overset{x}{\bullet}\xrightarrow{\sigma_i}\overset{y}{\bullet}$ for $y = x\sigma_i$.

- Cayley graph of B_n: vertices = braids; edges: $\overset{x}{\bullet} \xrightarrow{\sigma_i} \overset{y}{\bullet}$ for $y = x\sigma_i$.

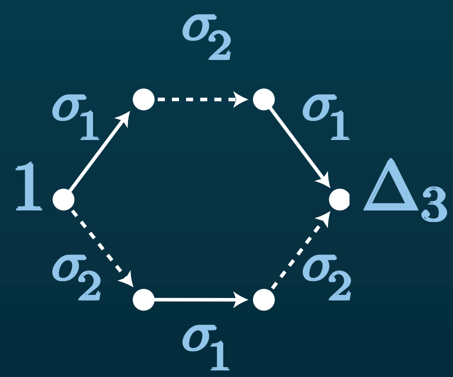
$\rightsquigarrow$ **Cayley(Δ_n^d)** := restriction of the Cayley graph of B_n to divisors of Δ_n^d

(in the sense of the monoid B_n^+)

- Cayley graph of B_n: vertices = braids; edges: $x \bullet \xrightarrow{\sigma_i} \bullet y$ for $y = x\sigma_i$.

⇝ **Cayley(Δ_n^d)** := restriction of the Cayley graph of B_n to divisors of Δ_n^d

(in the sense of the monoid B_n^+)

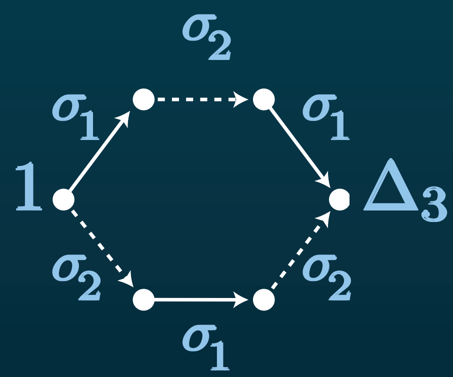
Example: Cayley(Δ_3) = 

- Cayley graph of B_n: vertices = braids; edges: $x \xrightarrow{\sigma_i} y$ for $y = x\sigma_i$.

⇝ **Cayley(Δ_n^d)** := restriction of the Cayley graph of B_n to divisors of Δ_n^d

(in the sense of the monoid B_n^+)

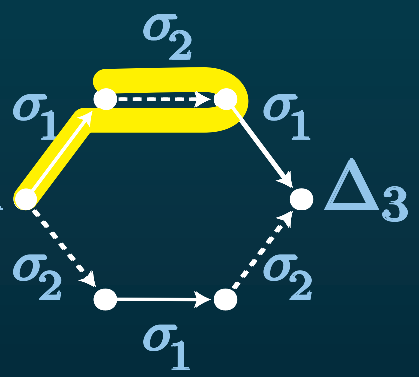
Example: Cayley(Δ_3) =



- Word drawn from x in Cayley(Δ_n^d):

- Cayley graph of B_n: vertices = braids; edges: $x \bullet \xrightarrow{\sigma_i} \bullet\, y$ for $y = x\sigma_i$.

⇝ **Cayley(Δ_n^d)** := restriction of the Cayley graph of B_n to divisors of Δ_n^d
(in the sense of the monoid B_n^+)

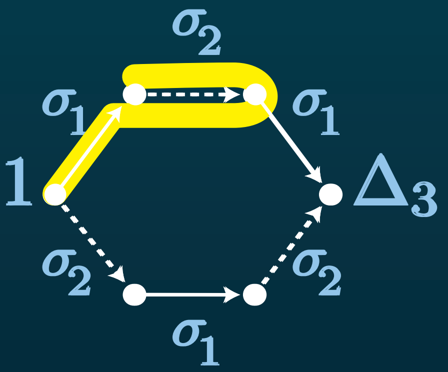
Example: Cayley(Δ_3) = 1 

- Word drawn from x in Cayley(Δ_n^d): *e.g.*, $\{$ $\sigma_1\sigma_2\sigma_2^{-1}$ drawn from 1,

- Cayley graph of B_n: vertices = braids; edges: $x \overset{\sigma_i}{\bullet\!\longrightarrow\!\bullet} y$ for $y = x\sigma_i$.

⇝ Cayley(Δ_n^d) := restriction of the Cayley graph of B_n to divisors of Δ_n^d

(in the sense of the monoid B_n^+)

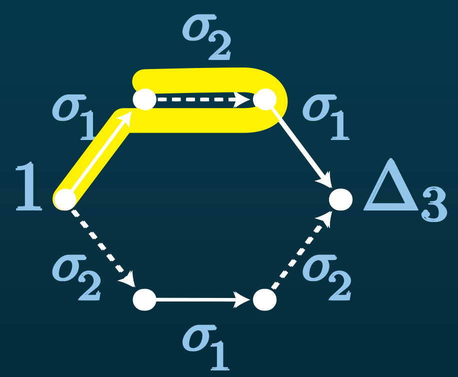
Example: Cayley(Δ_3) = 

- Word drawn from x in Cayley(Δ_n^d): *e.g.*, $\begin{cases} \sigma_1\sigma_2\sigma_2^{-1} \text{ drawn from } 1, \\ \sigma_1^2 \text{ not drawn from } 1. \end{cases}$

- Cayley graph of B_n: vertices = braids; edges: $x \bullet \xrightarrow{\sigma_i} \bullet y$ for $y = x\sigma_i$.

⇝ **Cayley(Δ_n^d)** := restriction of the Cayley graph of B_n to divisors of Δ_n^d (in the sense of the monoid B_n^+)

Example: Cayley(Δ_3) =



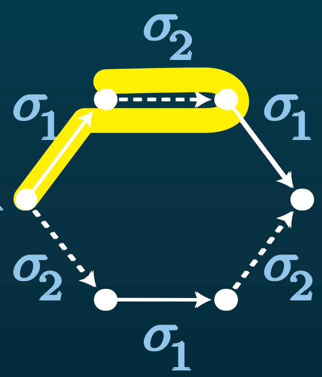
- Word drawn from x in Cayley(Δ_n^d): *e.g.*, $\begin{cases} \sigma_1\sigma_2\sigma_2^{-1} \text{ drawn from } 1, \\ \sigma_1^2 \text{ not drawn from } 1. \end{cases}$

- Lemma: (i) Every n strand braid word is drawn in Cayley(Δ_n^d) for $d \gg 0$;

• Cayley graph of B_n: vertices = braids; edges: $x \bullet \xrightarrow{\sigma_i} \bullet y$ for $y = x\sigma_i$.

⇝ Cayley(Δ_n^d) := restriction of the Cayley graph of B_n to divisors of Δ_n^d

(in the sense of the monoid B_n^+)

Example: Cayley(Δ_3) = 1  Δ_3

• Word drawn from x in Cayley(Δ_n^d): *e.g.*, $\begin{cases} \sigma_1\sigma_2\sigma_2^{-1} \text{ drawn from } 1, \\ \sigma_1^2 \text{ not drawn from } 1. \end{cases}$

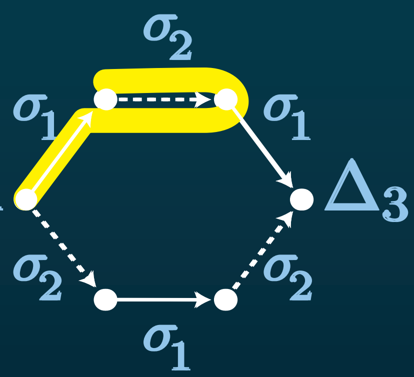
• Lemma: (i) Every n strand braid word is drawn in Cayley(Δ_n^d) for $d \gg 0$;
(ii) The set of words drawn in Cayley(Δ_n^d) is closed under handle reduction.

- Cayley graph of B_n: vertices = braids; edges: $x \bullet \xrightarrow{\sigma_i} \bullet\, y$ for $y = x\sigma_i$.

⇝ Cayley(Δ_n^d) := restriction of the Cayley graph of B_n to divisors of Δ_n^d

(in the sense of the monoid B_n^+)

Example: Cayley(Δ_3) = 1



- Word drawn from x in Cayley(Δ_n^d): *e.g.*, $\begin{cases} \sigma_1\sigma_2\sigma_2^{-1} \text{ drawn from } 1, \\ \sigma_1^2 \text{ not drawn from } 1. \end{cases}$

- Lemma: (i) Every n strand braid word is drawn in Cayley(Δ_n^d) for $d \gg 0$;

(ii) The set of words drawn in Cayley(Δ_n^d) is closed under handle reduction.

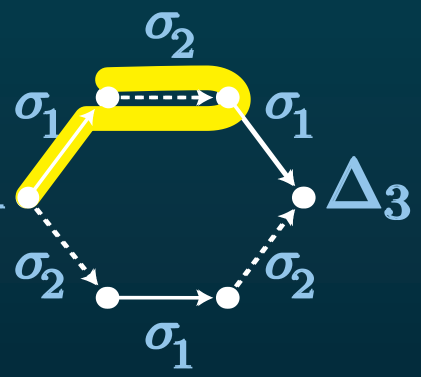
⇝ a boundedness result:

CONVERGENCE OF HANDLE REDUCTION (1)

- Cayley graph of B_n: vertices = braids; edges: $x \bullet \xrightarrow{\sigma_i} \bullet y$ for $y = x\sigma_i$.

⇝ Cayley(Δ_n^d) := restriction of the Cayley graph of B_n to divisors of Δ_n^d (in the sense of the monoid B_n^+)

Example: Cayley(Δ_3) =



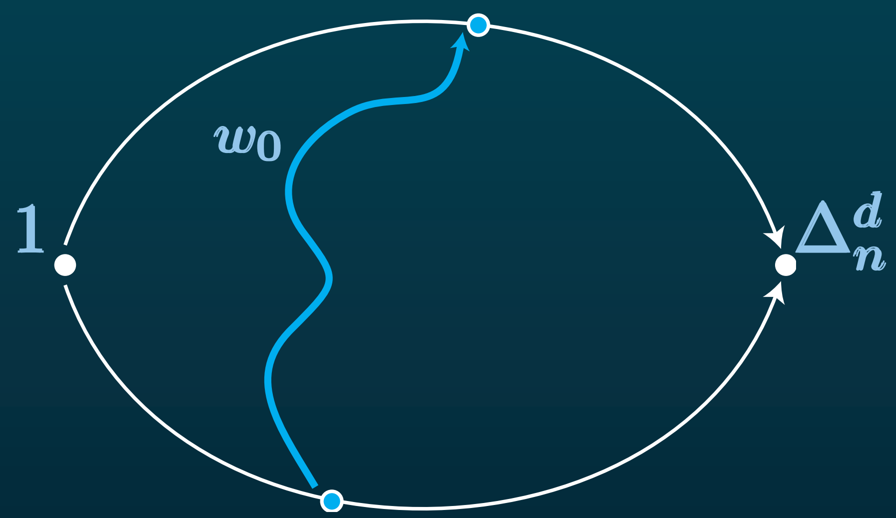
- Word drawn from x in Cayley(Δ_n^d): ***e.g.***, $\begin{cases} \sigma_1\sigma_2\sigma_2^{-1} \text{ drawn from } 1, \\ \sigma_1^2 \text{ not drawn from } 1. \end{cases}$

- Lemma: (i) Every n strand braid word is drawn in Cayley(Δ_n^d) for $d \gg 0$;
(ii) The set of words drawn in Cayley(Δ_n^d) is closed under handle reduction.

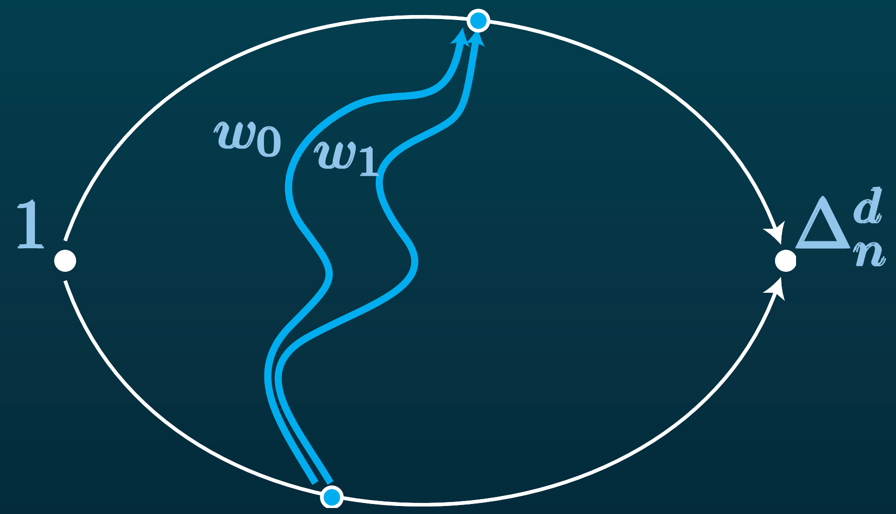
⇝ a boundedness result: when reduction is performed,
all words are drawn in some fixed finite subgraph of the Cayley graph.

- Let $w_0, w_1, \dots$ be a reduction sequence, all words drawn in $\mathsf{Cayley}(\Delta_n^d)$;

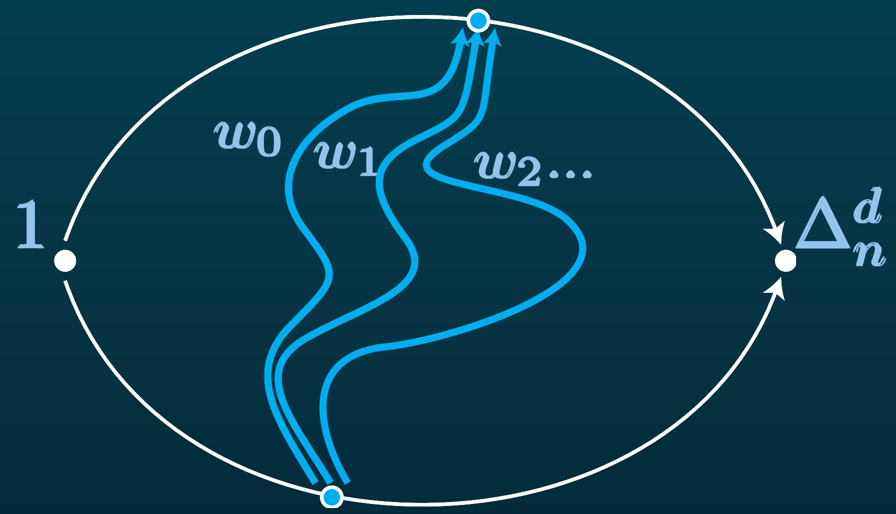
- Let $w_0, w_1, ...$ be a reduction sequence, all words drawn in $\mathsf{Cayley}(\Delta_n^d)$;



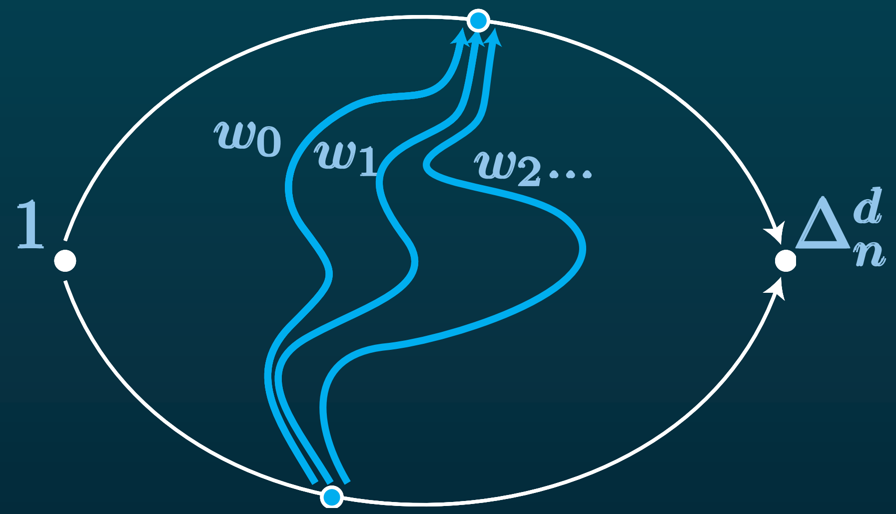
- Let $w_0, w_1, \ldots$ be a reduction sequence, all words drawn in $\mathsf{Cayley}(\Delta_n^d)$;



- Let $w_0, w_1, \ldots$ be a reduction sequence, all words drawn in $\mathrm{Cayley}(\Delta_n^d)$;

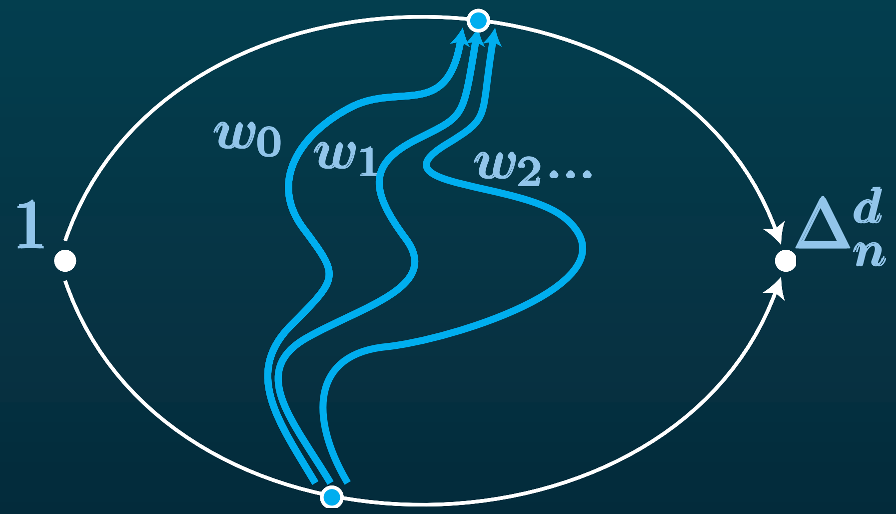


- Let $w_0, w_1, \ldots$ be a reduction sequence, all words drawn in Cayley(Δ_n^d);



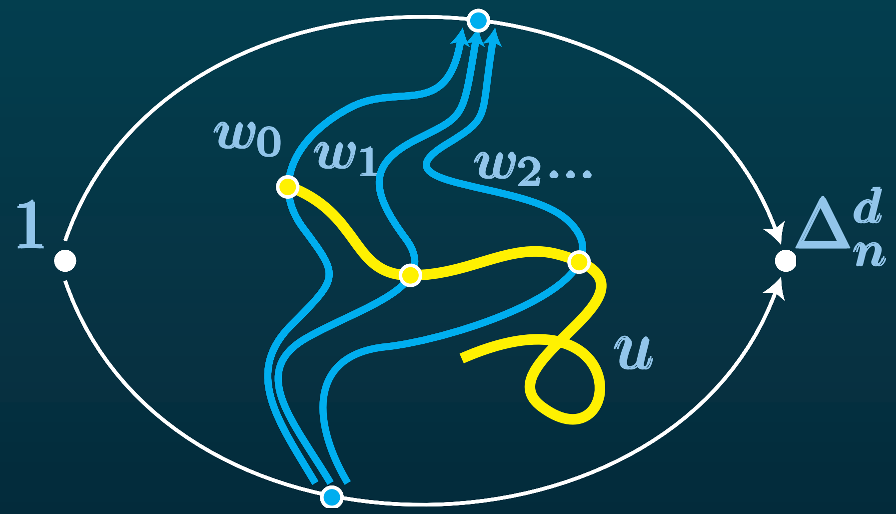
- For an induction on n, enough to prove: $\#\sigma_1$ -handles eventually ↘.

- Let $w_0, w_1, \ldots$ be a reduction sequence, all words drawn in $\mathrm{Cayley}(\Delta_n^d)$;



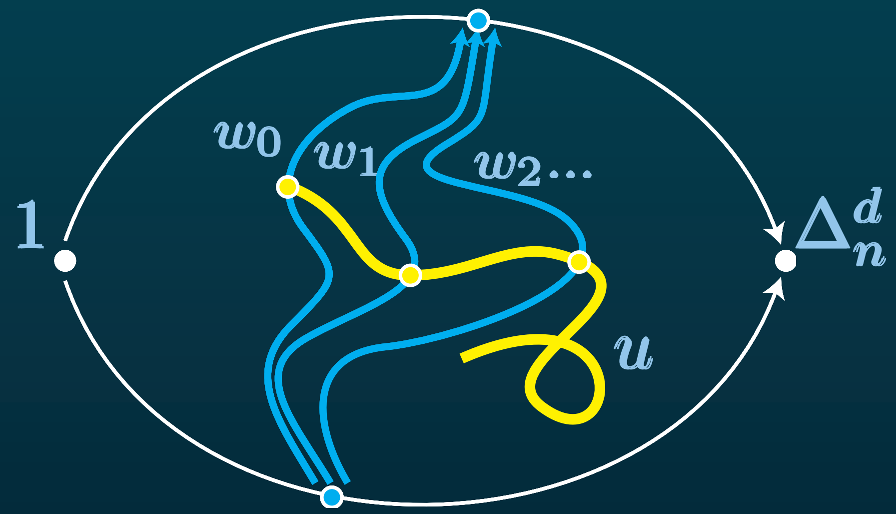
- For an induction on n, enough to prove: $\#\sigma_1$-handles eventually ↘.
- When a σ_1-handle is reduced, new handles may appear, but

- Let $w_0, w_1, \ldots$ be a reduction sequence, all words drawn in $\mathrm{Cayley}(\Delta_n^d)$;



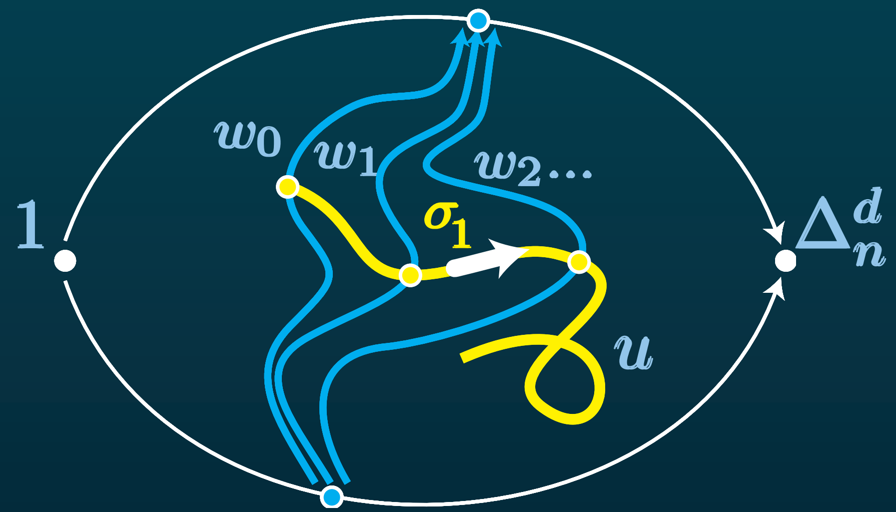
- For an induction on n, enough to prove: $\#\sigma_1$-handles eventually $\searrow$.
- When a σ_1-handle is reduced, new handles may appear, but
  - at most 1 new σ_1-handle ($\rightsquigarrow$ $\#\sigma_1$-handles non-increasing)

- Let $w_0, w_1, \ldots$ be a reduction sequence, all words drawn in $\mathrm{Cayley}(\Delta_n^d)$;



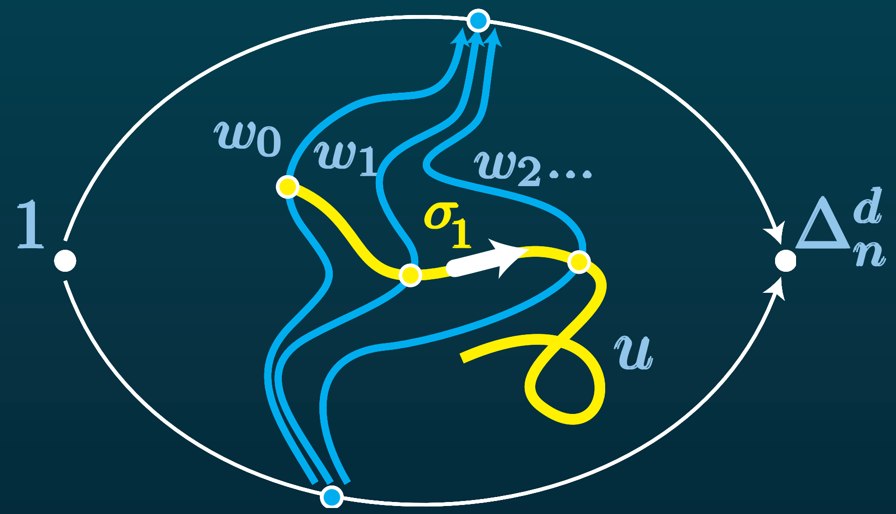
- For an induction on n, enough to prove: $\#\sigma_1$-handles eventually ↘.
- When a σ_1-handle is reduced, new handles may appear, but
  - at most 1 new σ_1-handle (⇝ $\#\sigma_1$-handles non-increasing)
  - there exists a “transversal witness word” u drawn in $\mathrm{Cayley}(\Delta_n^d)$

- Let $w_0, w_1, \ldots$ be a reduction sequence, all words drawn in $\mathrm{Cayley}(\Delta_n^d)$;



- For an induction on n, enough to prove: $\#\sigma_1$-handles eventually ↘.
- When a σ_1-handle is reduced, new handles may appear, but
  - at most 1 new σ_1-handle (⇝ $\#\sigma_1$-handles non-increasing)
  - there exists a “transversal witness word” u drawn in $\mathrm{Cayley}(\Delta_n^d)$ containing no letter σ_1^{-1}, and exactly N ($\leqslant \infty$) letters σ_1's, where N = number of reductions of the (first) σ_1-handle in $w_0, w_1, \ldots$

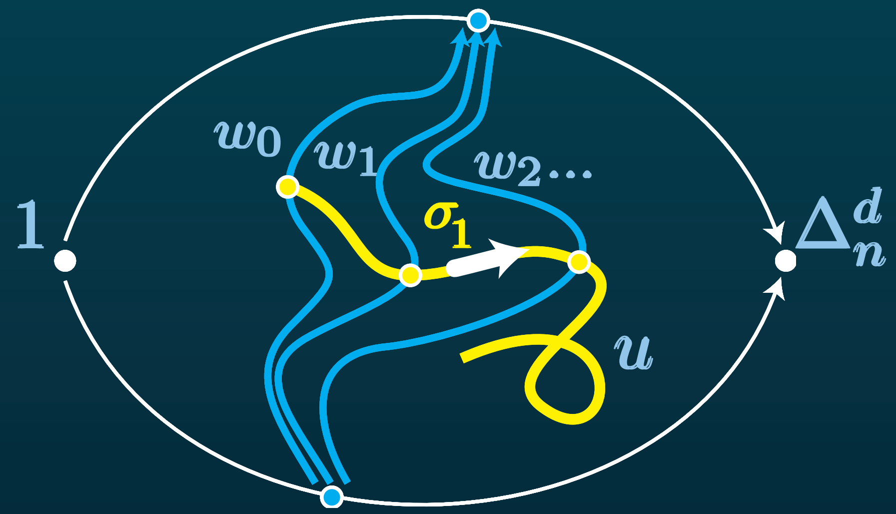
- Let $w_0, w_1, ...$ be a reduction sequence, all words drawn in $\mathsf{Cayley}(\Delta_n^d)$;



- For an induction on n, enough to prove: $\#\sigma_1$-handles eventually ↘.
- When a σ_1-handle is reduced, new handles may appear, but
  - at most 1 new σ_1-handle (⇝ $\#\sigma_1$-handles non-increasing)
  - there exists a "transversal witness word" u drawn in $\mathsf{Cayley}(\Delta_n^d)$ containing no letter σ_1^{-1}, and exactly N ($\leqslant \infty$) letters σ_1's, where N = number of reductions of the (first) σ_1-handle in $w_0, w_1, ...$

- Theorem ("Property A"). A path with no σ_1^{-1} crosses each σ_1 at most once.

- Let $w_0, w_1, \ldots$ be a reduction sequence, all words drawn in $\mathrm{Cayley}(\Delta_n^d)$;



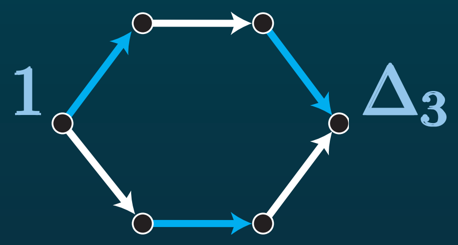
- For an induction on n, enough to prove: $\#\sigma_1$-handles eventually ↘.
- When a σ_1-handle is reduced, new handles may appear, but
  - at most 1 new σ_1-handle (⇝ $\#\sigma_1$-handles non-increasing)
  - there exists a "transversal witness word" u drawn in $\mathrm{Cayley}(\Delta_n^d)$ containing no letter σ_1^{-1}, and exactly N ($\leqslant \infty$) letters σ_1's, where N = number of reductions of the (first) σ_1-handle in $w_0, w_1, \ldots$

- Theorem ("Property A"). A path with no σ_1^{-1} crosses each σ_1 at most once.

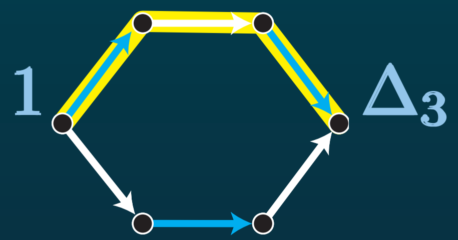
⇝ convergence of reduction □

- $a_{n,d}$:= maximal # of σ_1's in a word with no σ_1^{-1} drawn in $\mathrm{Cayley}(\Delta_n^d)$.

- $a_{n,d}$:= maximal # of σ_1's in a word with no σ_1^{-1} drawn in Cayley(Δ_n^d).

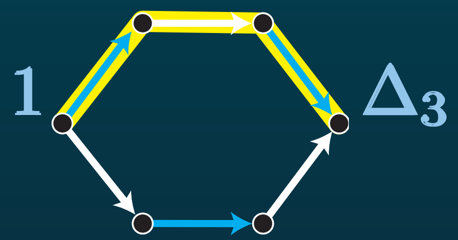


- $a_{n,d}$:= maximal # of σ_1's in a word with no σ_1^{-1} drawn in $\mathrm{Cayley}(\Delta_n^d)$.

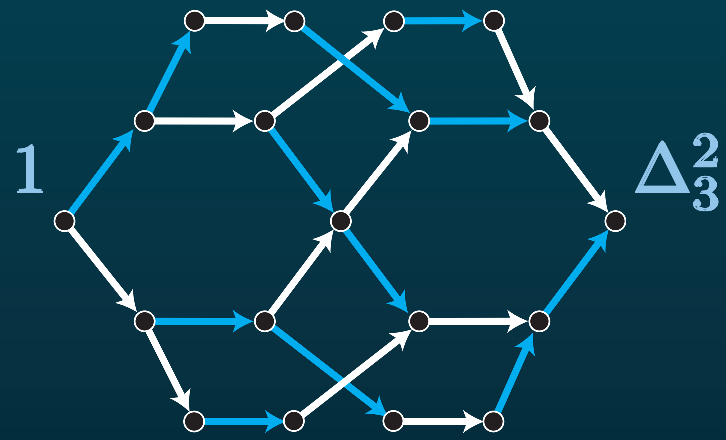


$\rightsquigarrow a_{3,1} \geqslant 2$

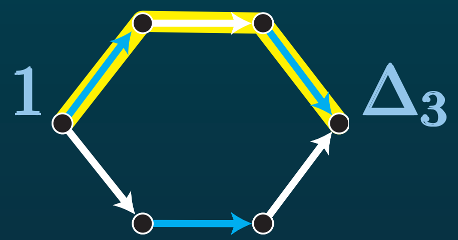
- $a_{n,d}$:= maximal # of σ_1's in a word with no σ_1^{-1} drawn in $\mathrm{Cayley}(\Delta_n^d)$.



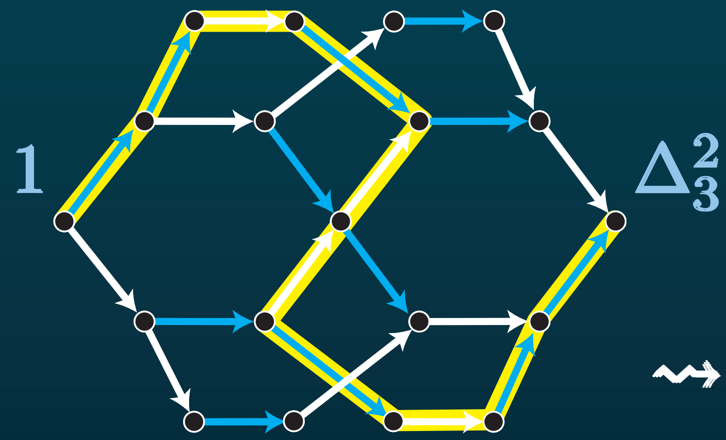
$\rightsquigarrow a_{3,1} \geqslant 2$



- $a_{n,d}$:= maximal # of σ_1's in a word with no σ_1^{-1} drawn in Cayley(Δ_n^d).

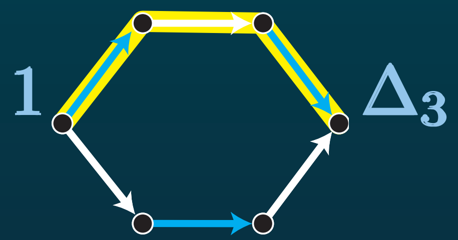


$\rightsquigarrow a_{3,1} \geqslant 2$

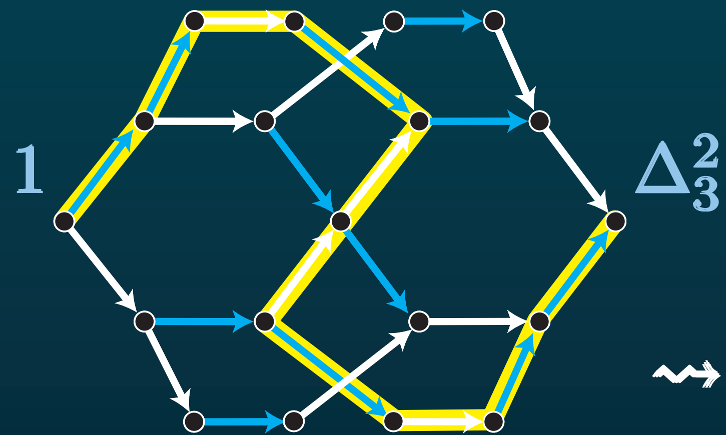


$\rightsquigarrow a_{3,2} \geqslant 6$

- $a_{n,d}$:= maximal # of σ_1's in a word with no σ_1^{-1} drawn in $\mathrm{Cayley}(\Delta_n^d)$.



$\rightsquigarrow a_{3,1} \geqslant 2$

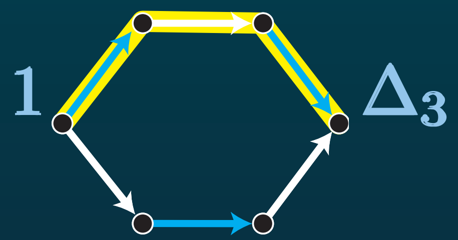


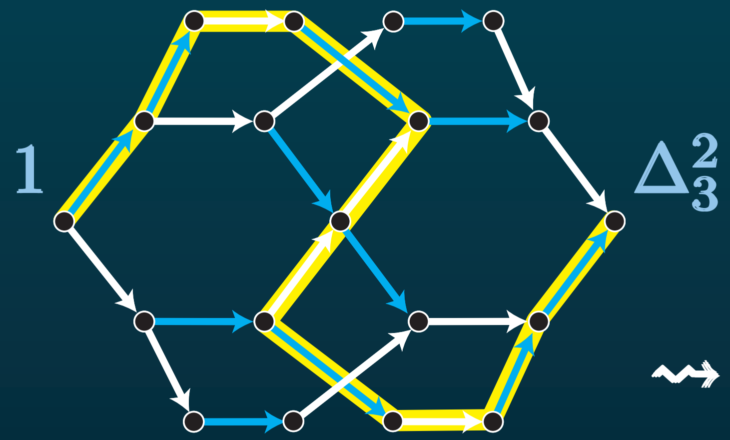
$\rightsquigarrow a_{3,2} \geqslant 6$

- Theorem: $a_{n,d}$ = # normal sequences of the form $(x_1, ..., x_{d-1}, \Delta_{n-1})$.

(in the sense of Solution 2: greedy normal form)

- $a_{n,d}$:= maximal # of σ_1's in a word with no σ_1^{-1} drawn in Cayley(Δ_n^d).



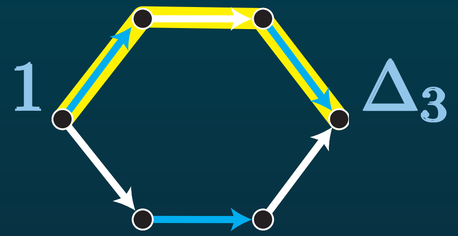


- Theorem: $a_{n,d}$ = # normal sequences of the form $(x_1, ..., x_{d-1}, \Delta_{n-1})$.

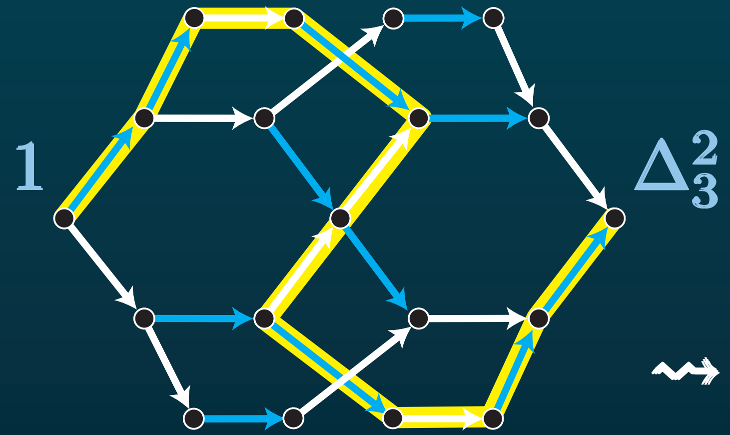
  (in the sense of Solution 2: greedy normal form)

- Now: divisors of Δ_n $\leftrightsquigarrow$ permutations of $\{1, ..., n\}$,

- $a_{n,d}$:= maximal # of σ_1's in a word with no σ_1^{-1} drawn in Cayley(Δ_n^d).



$\rightsquigarrow a_{3,1} \geqslant 2$

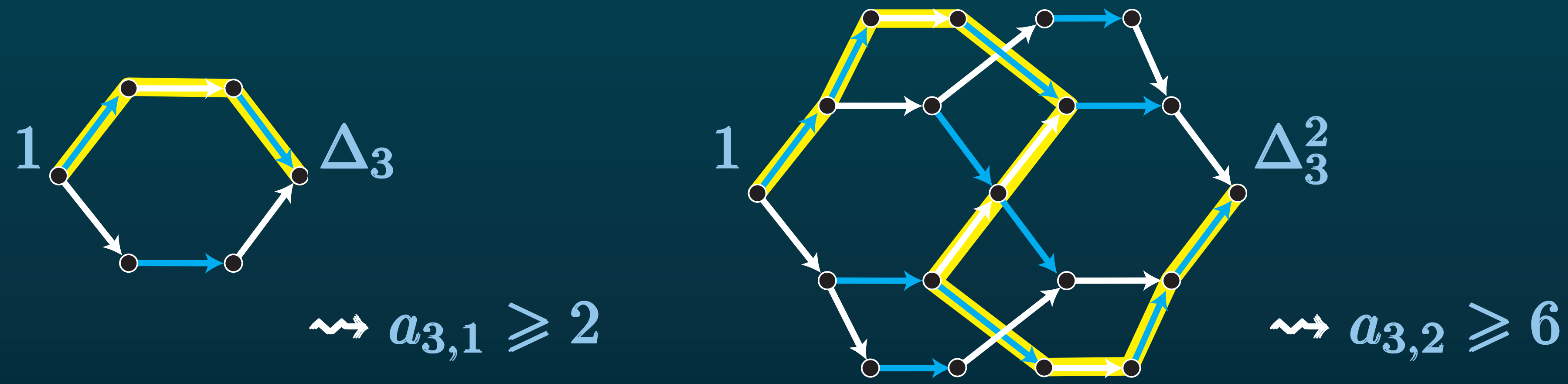


$\rightsquigarrow a_{3,2} \geqslant 6$

- Theorem: $a_{n,d}$ = # normal sequences of the form $(x_1, ..., x_{d-1}, \Delta_{n-1})$.
  (in the sense of Solution 2: greedy normal form)

- Now: divisors of $\Delta_n \leftrightsquigarrow$ permutations of $\{1, ..., n\}$,
  and (f, g) normal $\leftrightsquigarrow$ $\{$ descents of $f^{-1}\} \supseteq \{$ descents of $g\}$,
  i.e., $f^{-1}(i+1) < f^{-1}(i) \Rightarrow g(i+1) < g(i)$

- $a_{n,d}$:= maximal # of σ_1's in a word with no σ_1^{-1} drawn in $\mathrm{Cayley}(\Delta_n^d)$.



- Theorem: $a_{n,d}$ = # normal sequences of the form $(x_1, ..., x_{d-1}, \Delta_{n-1})$.

  (in the sense of Solution 2: greedy normal form)

- Now: divisors of Δ_n $\leftrightsquigarrow$ permutations of $\{1, ..., n\}$,

  and (f, g) normal $\leftrightsquigarrow$ $\{$ descents of $f^{-1}\} \supseteq \{$ descents of $g\}$,

  i.e., $f^{-1}(i+1) < f^{-1}(i) \Rightarrow g(i+1) < g(i)$

- Corollary: $a_{n,d}$ expressed from M_n^d, where M_n is the $n! \times n!$ matrix

$$(M_n)_{f,g} := \begin{cases} 1 & \text{if } \{\text{ descents of } f^{-1}\} \supseteq \{\text{ descents of } g\}, \\ 0 & \text{otherwise.} \end{cases}$$

# of partitions of $\{1, ..., n\}$

↓

- Theorem: M_n can be replaced with a $p(n) \times p(n)$ matrix.

# of partitions of $\{1, \ldots, n\}$
$\downarrow$

- Theorem: M_n can be replaced with a $p(n) \times p(n)$ matrix.

(connected with the Solomon descent algebra
and the theory of combinatorial Hopf algebras)

# of partitions of $\{1, \ldots, n\}$
↓

- Theorem: M_n can be replaced with a $p(n) \times p(n)$ matrix.

  (connected with the Solomon descent algebra
  and the theory of combinatorial Hopf algebras)

- Conjecture: $\mathrm{CharPol}(M_n)$ divides $\mathrm{CharPol}(M_{n+1})$.

\# of partitions of $\{1, ..., n\}$ ↓

- Theorem: M_n can be replaced with a $p(n) \times p(n)$ matrix.

(connected with the Solomon descent algebra and the theory of combinatorial Hopf algebras)

- Conjecture: $\mathrm{CharPol}(M_n)$ divides $\mathrm{CharPol}(M_{n+1})$.

$$\mathrm{CharPol}(M_1) = x - 1$$
$$\mathrm{CharPol}(M_2) = \mathrm{CharPol}(M_1) \cdot (x - 1)$$
$$\mathrm{CharPol}(M_3) = \mathrm{CharPol}(M_2) \cdot (x - 2)$$
$$\mathrm{CharPol}(M_4) = \mathrm{CharPol}(M_3) \cdot (x^2 - 6x + 3)$$
$$\mathrm{CharPol}(M_5) = \mathrm{CharPol}(M_4) \cdot (x^2 - 20x + 24)$$
$$\mathrm{CharPol}(M_6) = \mathrm{CharPol}(M_5) \cdot (x^4 - 82x^3 + 359x^2 - 260x + 60) \ \ldots$$

\# of partitions of $\{1, \ldots, n\}$
↓
- Theorem: M_n can be replaced with a $p(n) \times p(n)$ matrix.
(connected with the Solomon descent algebra
and the theory of combinatorial Hopf algebras)

- Conjecture: $\mathrm{CharPol}(M_n)$ divides $\mathrm{CharPol}(M_{n+1})$.

$$\begin{aligned}
\mathrm{CharPol}(M_1) &= x - 1 \\
\mathrm{CharPol}(M_2) &= \mathrm{CharPol}(M_1) \cdot (x - 1) \\
\mathrm{CharPol}(M_3) &= \mathrm{CharPol}(M_2) \cdot (x - 2) \\
\mathrm{CharPol}(M_4) &= \mathrm{CharPol}(M_3) \cdot (x^2 - 6x + 3) \\
\mathrm{CharPol}(M_5) &= \mathrm{CharPol}(M_4) \cdot (x^2 - 20x + 24) \\
\mathrm{CharPol}(M_6) &= \mathrm{CharPol}(M_5) \cdot (x^4 - 82x^3 + 359x^2 - 260x + 60) \ \ldots
\end{aligned}$$

- Question: What is the asymptotic behaviour of $\lambda_{max}(M_n)$?

(Recall) "Property A": A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

$\rightsquigarrow$ Where does this come from?

(Recall) “Property A”: A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

⇝ Where does this come from?

- Braid diagram colourings:
  1. Put colours at input ends of strands
  2. Propagate colours
  3. Look at output strands

(Recall) "Property A": A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

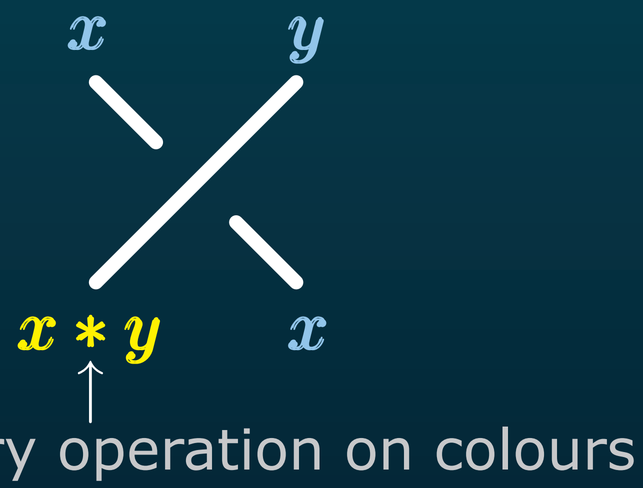
⇝ Where does this come from?

- Braid diagram colourings:
    1. Put colours at input ends of strands
    2. Propagate colours
    3. Look at output strands

(Recall) "Property A": A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

⇝ Where does this come from?

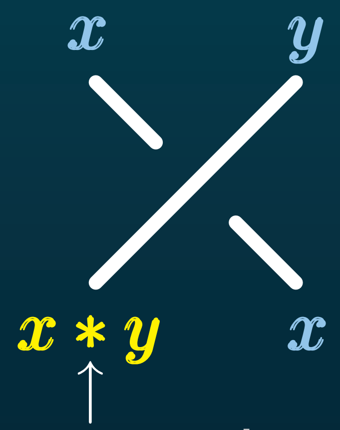
- Braid diagram colourings:
  1. Put colours at input ends of strands
  2. Propagate colours
  3. Look at output strands



(Recall) "Property A": A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

$\rightsquigarrow$ Where does this come from?

- Braid diagram colourings:
  1. Put colours at input ends of strands
  2. Propagate colours
  3. Look at output strands



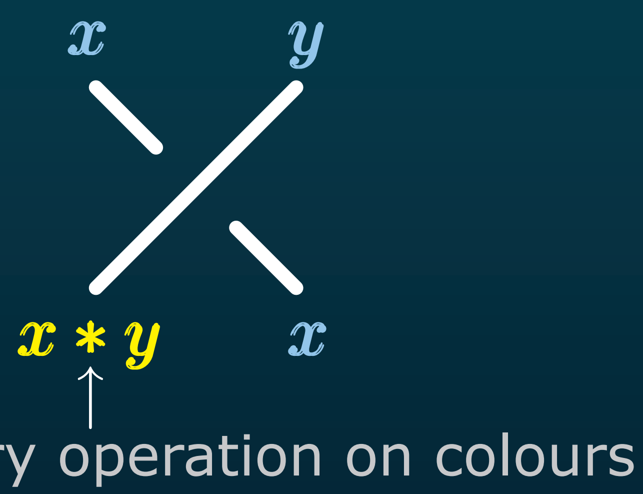
- Now: compatible with braid relations iff $*$ satisfies

(LD): $x * (y * z) = (x * y) * (x * z)$: the (left) self-distributivity law.

(Recall) "Property A": A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

⇝ Where does this come from?

- Braid diagram colourings:
  1. Put colours at input ends of strands
  2. Propagate colours
  3. Look at output strands



- Now: compatible with braid relations iff $*$ satisfies

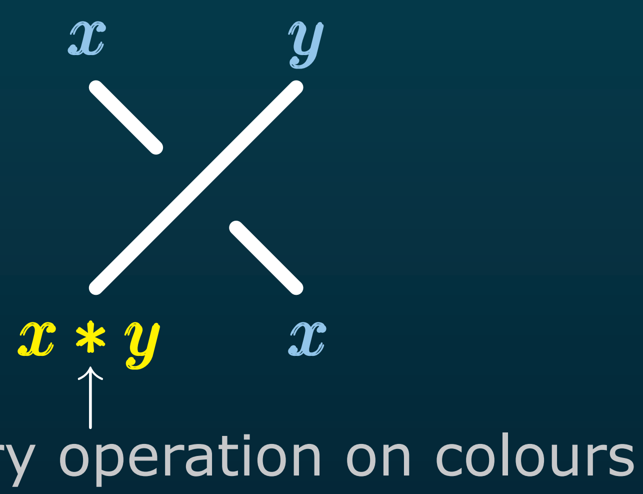
(LD): $x * (y * z) = (x * y) * (x * z)$: the (left) self-distributivity law.

- Classical examples:
  - $x * y = y$ (!!)

(Recall) “Property A”: A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

$\rightsquigarrow$ Where does this come from?

- Braid diagram colourings:
    1. Put colours at input ends of strands
    2. Propagate colours
    3. Look at output strands



- Now: compatible with braid relations iff $*$ satisfies

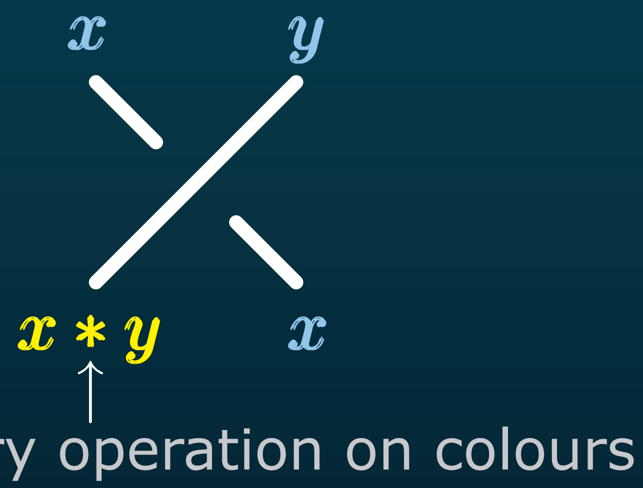
(LD): $x * (y * z) = (x * y) * (x * z)$: the (left) self-distributivity law.

- Classical examples:
    - $x * y = y$ (!!) leads to $B_n \twoheadrightarrow S_n$;

(Recall) "Property A": A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

$\rightsquigarrow$ Where does this come from?

- Braid diagram colourings:
  1. Put colours at input ends of strands
  2. Propagate colours
  3. Look at output strands



- Now: compatible with braid relations iff $*$ satisfies

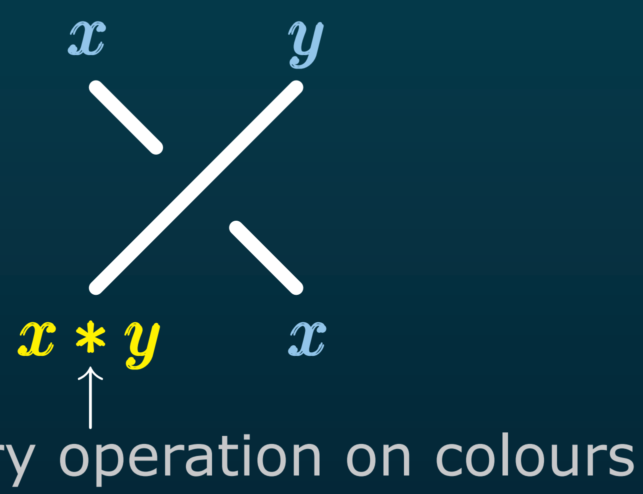
(LD): $x * (y * z) = (x * y) * (x * z)$: the (left) self-distributivity law.

- Classical examples:
  - $x * y = y$ (!!) leads to $B_n \twoheadrightarrow S_n$;
  - $x * y = xyx^{-1}$

(Recall) “Property A”: A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

⇝ Where does this come from?

- Braid diagram colourings:
  1. Put colours at input ends of strands
  2. Propagate colours
  3. Look at output strands



- Now: compatible with braid relations iff $*$ satisfies

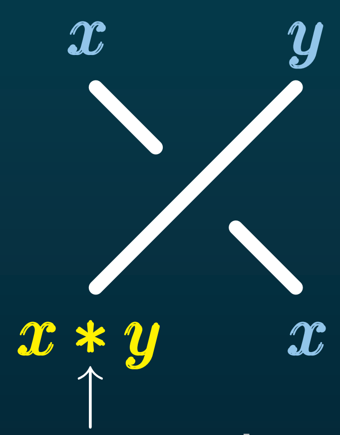
(LD): $x * (y * z) = (x * y) * (x * z)$: the (left) self-distributivity law.

- Classical examples:
  - $x * y = y$ (!!) leads to $B_n \twoheadrightarrow S_n$;
  - $x * y = xyx^{-1}$ leads to $B_n \hookrightarrow \mathtt{Aut}(F_n)$;

(Recall) "Property A": A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

⇝ Where does this come from?

- Braid diagram colourings:
  1. Put colours at input ends of strands
  2. Propagate colours
  3. Look at output strands



- Now: compatible with braid relations iff $*$ satisfies

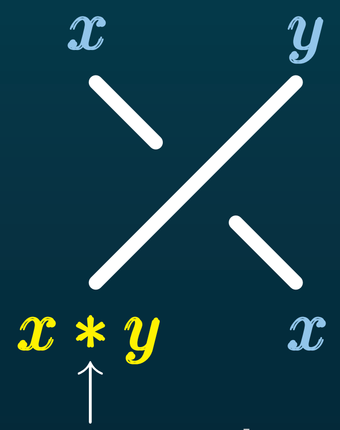
(LD): $x * (y * z) = (x * y) * (x * z)$: the (left) self-distributivity law.

- Classical examples:
  - $x * y = y$ (!!) leads to $B_n \twoheadrightarrow S_n$;
  - $x * y = xyx^{-1}$ leads to $B_n \hookrightarrow \mathtt{Aut}(F_n)$;
  - $x * y = (1-t)x + ty$

(Recall) "Property A": A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

⇝ Where does this come from?

- Braid diagram colourings:
  1. Put colours at input ends of strands
  2. Propagate colours
  3. Look at output strands



- Now: compatible with braid relations iff $*$ satisfies

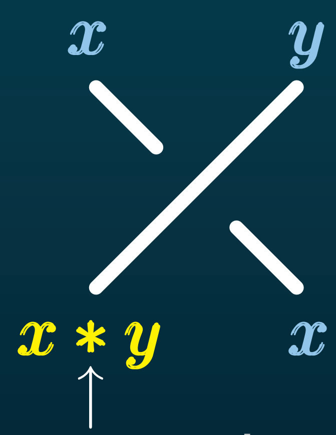
(LD): $x * (y * z) = (x * y) * (x * z)$: the (left) self-distributivity law.

- Classical examples:
  - $x * y = y$ (!!) leads to $B_n \twoheadrightarrow S_n$;
  - $x * y = xyx^{-1}$ leads to $B_n \hookrightarrow \mathtt{Aut}(F_n)$;
  - $x * y = (1-t)x + ty$ leads to $B_n \to GL_n(\mathbb{Z}[t, t^{-1}])$.

(Recall) "Property A": A braid word containing a σ_1 and no σ_1^{-1} is not trivial.

$\rightsquigarrow$ Where does this come from?

- Braid diagram colourings:
    1. Put colours at input ends of strands
    2. Propagate colours
    3. Look at output strands



- Now: compatible with braid relations iff $*$ satisfies

(LD): $x * (y * z) = (x * y) * (x * z)$: the (left) self-distributivity law.

- Classical examples:
    - $x * y = y$ (!!) leads to $B_n \twoheadrightarrow S_n$;
    - $x * y = xyx^{-1}$ leads to $B_n \hookrightarrow \mathtt{Aut}(F_n)$;
    - $x * y = (1-t)x + ty$ leads to $B_n \to GL_n(\mathbb{Z}[t, t^{-1}])$.

(all satisfy $x * x = x$)

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

a set with a binary operation satisfying...

$\downarrow$

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

a set with a binary operation satisfying...
↓

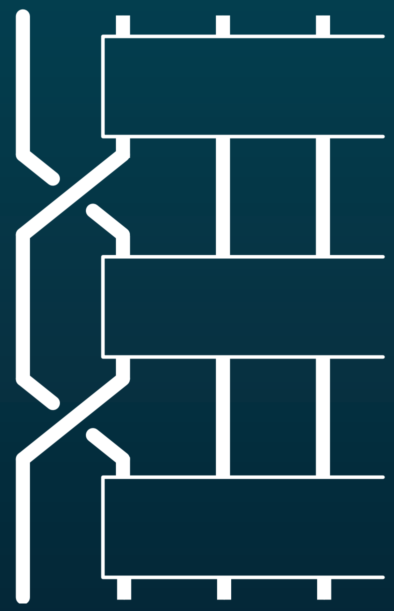
- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

- Proposition: If there exists an orderable LD-system, then Property A is true.

a set with a binary operation satisfying...

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

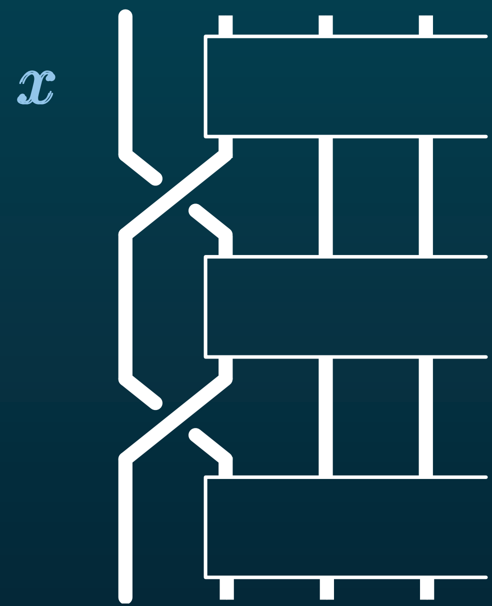
- Proposition: If there exists an orderable LD-system, then Property A is true.



a set with a binary operation satisfying...

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

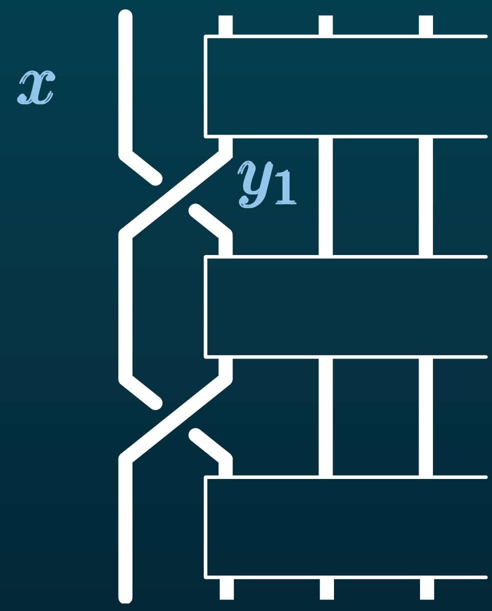
- Proposition: If there exists an orderable LD-system, then Property A is true.



a set with a binary operation satisfying...

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

- Proposition: If there exists an orderable LD-system, then Property A is true.

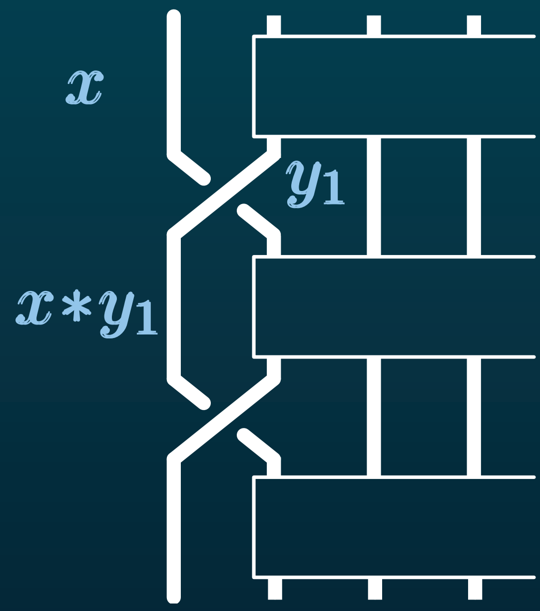


a set with a binary operation satisfying...

↓

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

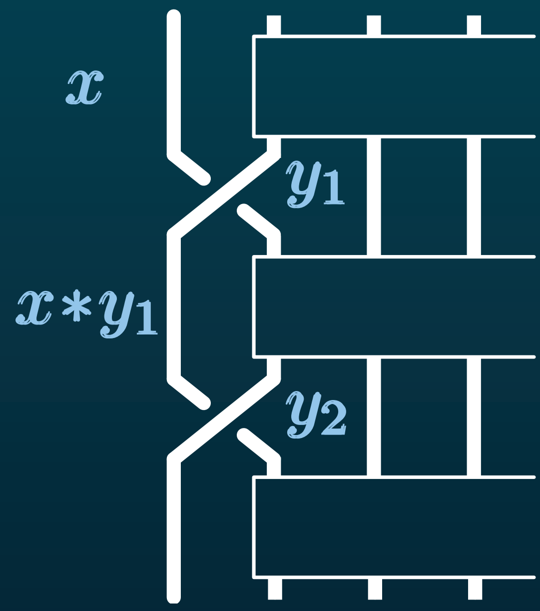
- Proposition: If there exists an orderable LD-system, then Property A is true.



a set with a binary operation satisfying...

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

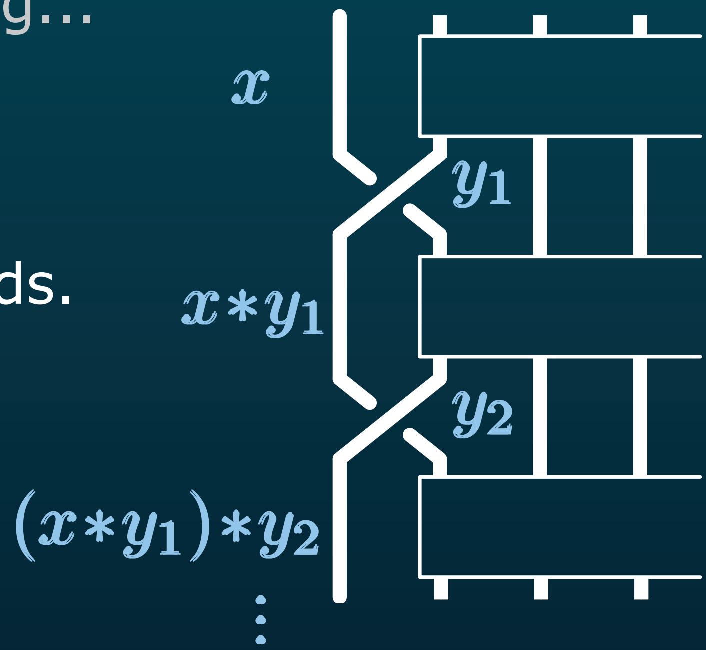
- Proposition: If there exists an orderable LD-system, then Property A is true.



a set with a binary operation satisfying...

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

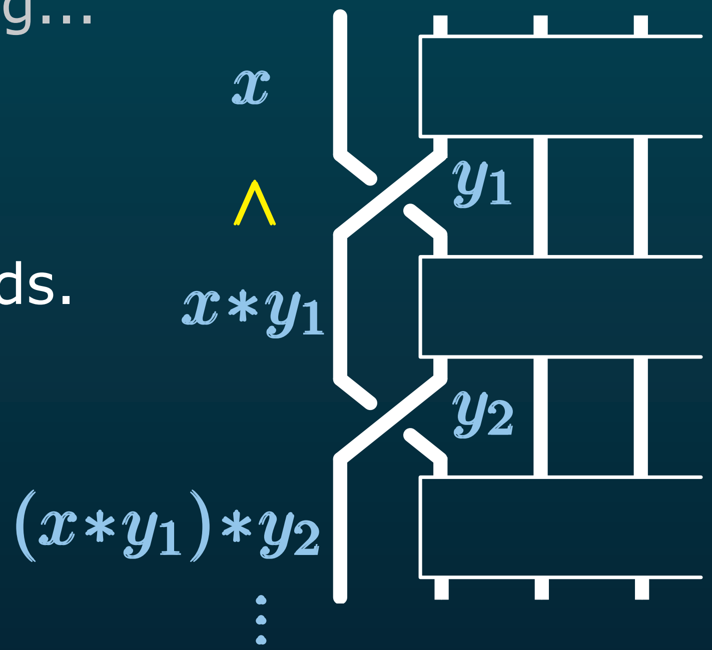
- Proposition: If there exists an orderable LD-system, then Property A is true.



a set with a binary operation satisfying...

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

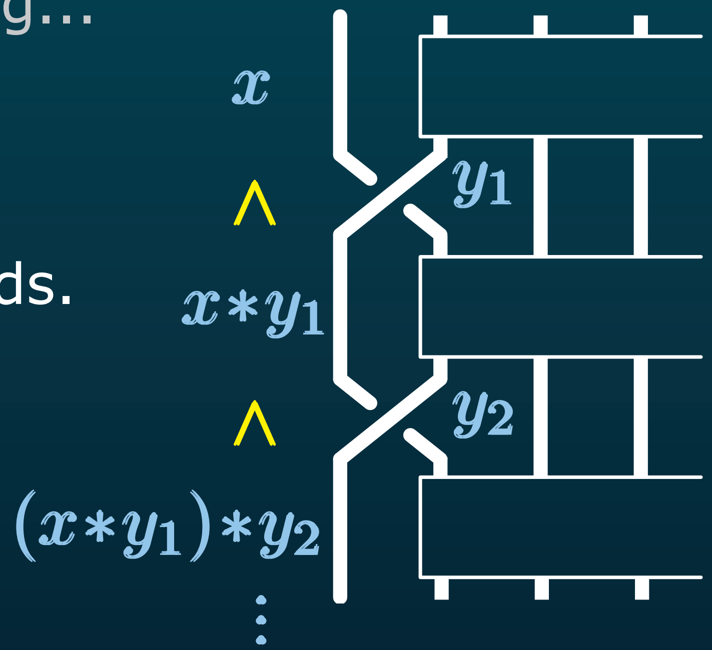
- Proposition: If there exists an orderable LD-system, then Property A is true.



a set with a binary operation satisfying...
↓

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

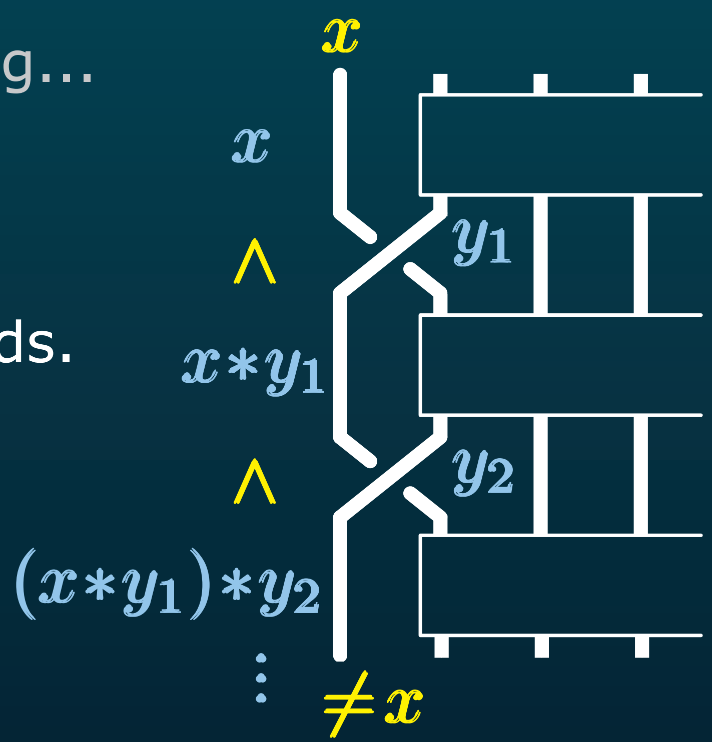
- Proposition: If there exists an orderable LD-system, then Property A is true.



a set with a binary operation satisfying...

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

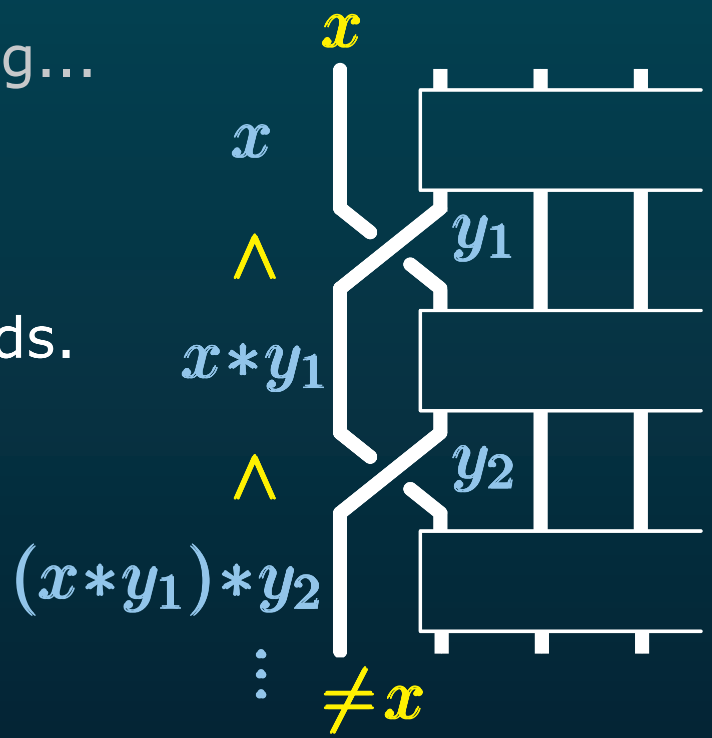
- Proposition: If there exists an orderable LD-system, then Property A is true.



a set with a binary operation satisfying...

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

- Proposition: If there exists an orderable LD-system, then Property A is true.

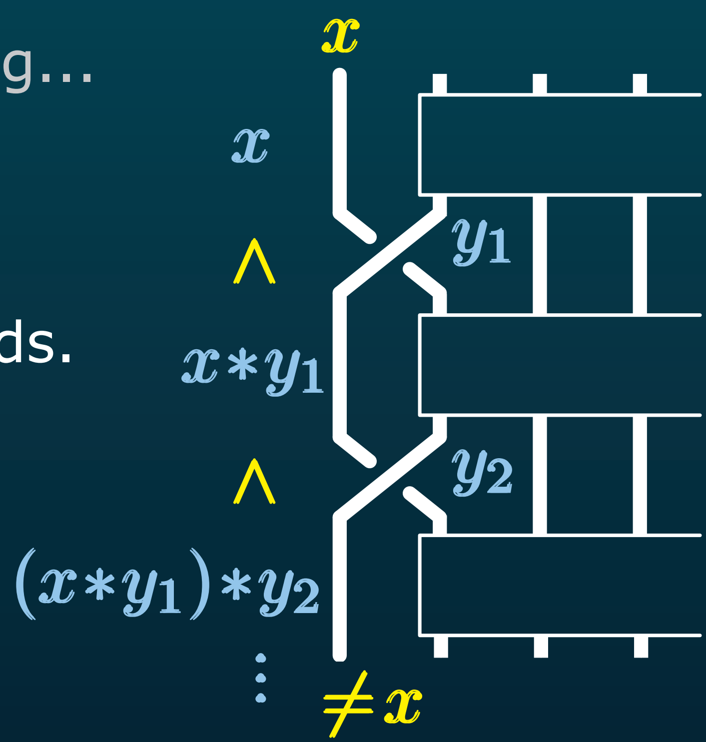


- Theorem (Laver, 1989) If there exists a self-similar rank, then there exists an orderable LD-system.

a set with a binary operation satisfying...

↓

- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.



- Proposition: If there exists an orderable LD-system, then Property A is true.

by Gödel's incompleteness thrm, an unprovable logical assumption

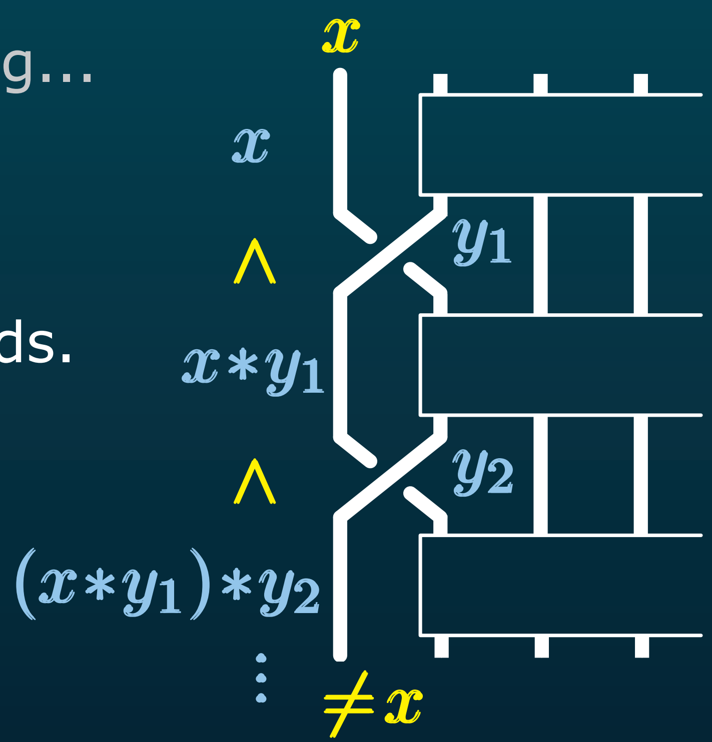
↓

- Theorem (Laver, 1989) If there exists a self-similar rank, then there exists an orderable LD-system.

a set with a binary operation satisfying...

↓

- Definition: Say that an LD-system $(S,*)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x*y$ always holds.



- Proposition: If there exists an orderable LD-system, then Property A is true.

by Gödel's incompleteness thrm, an unprovable logical assumption

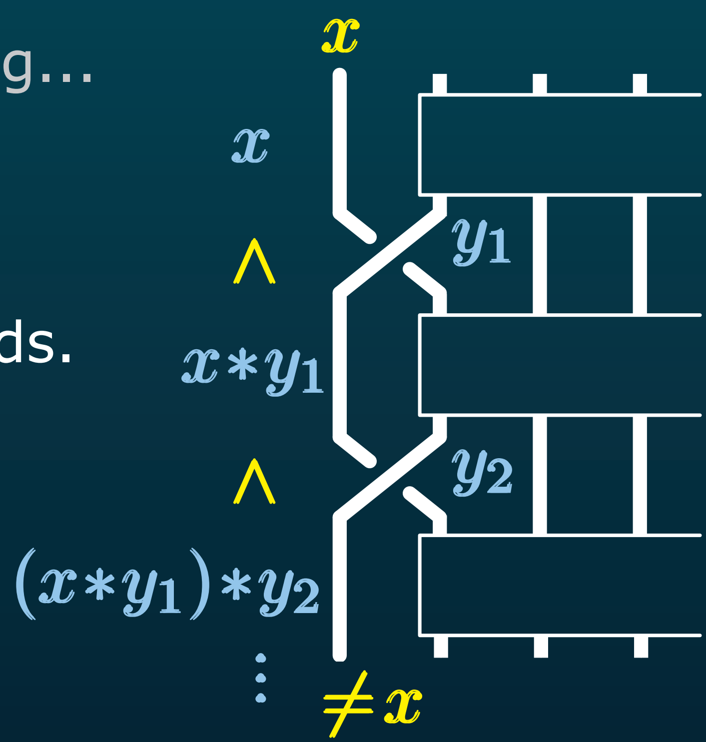
↓

- Theorem (Laver, 1989) If there exists a self-similar rank, then there exists an orderable LD-system.

- Theorem (D., 1992) Free LD-systems are orderable.

a set with a binary operation satisfying...
↓
- Definition: Say that an LD-system $(S, *)$ is orderable if there exists a strict linear ordering $<$ on S s.t. $x < x * y$ always holds.

- Proposition: If there exists an orderable LD-system, then Property A is true.



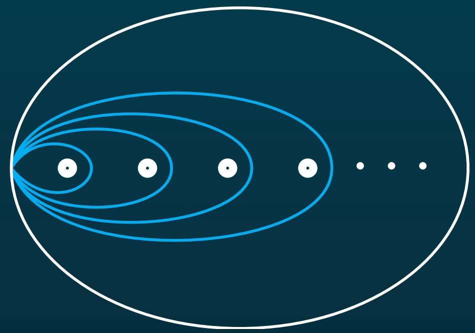
by Gödel's incompleteness thrm, an unprovable logical assumption
↓
- Theorem (Laver, 1989) If there exists a self-similar rank, then there exists an orderable LD-system.

- Theorem (D., 1992) Free LD-systems are orderable.

$\rightsquigarrow$ Handle reduction is an application of Set Theory (?)

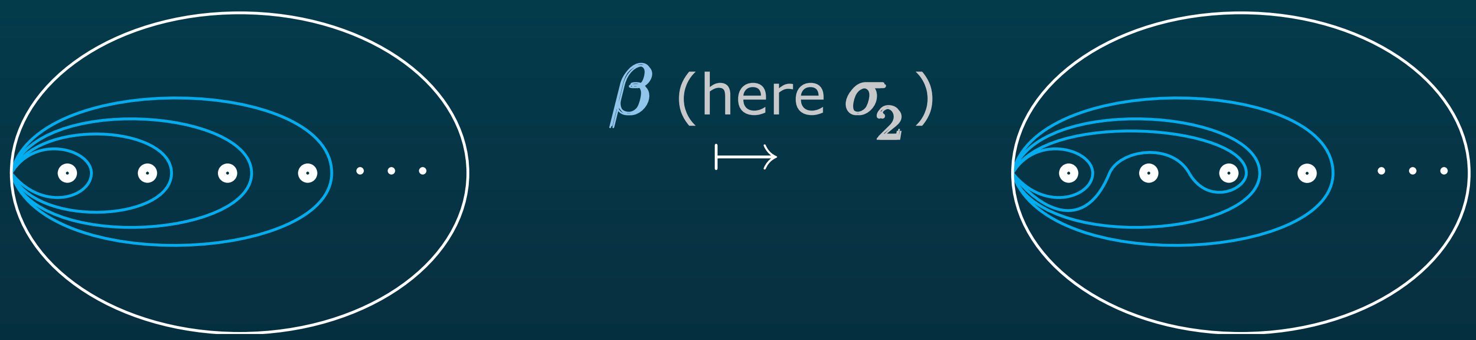
(I. Dynnikov, 1999)

- View B_n as $\mathbf{MCG}(D_n)$, and let the homeo act on a fixed lamination L:



(I. Dynnikov, 1999)

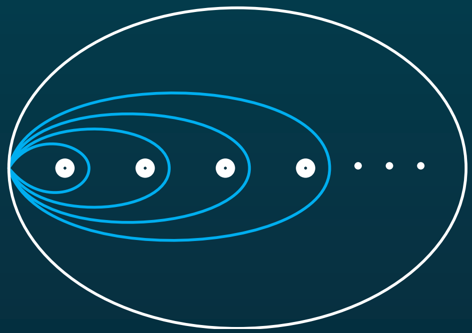
- View B_n as $\mathbf{MCG}(D_n)$, and let the homeo act on a fixed lamination L:



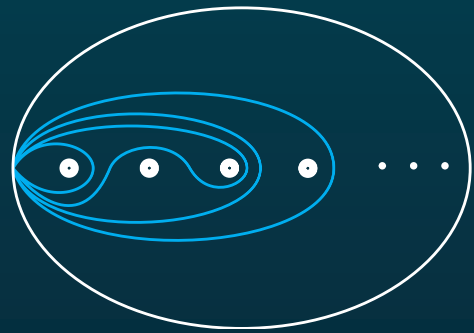
SOLUTION 6: SINGULAR TRIANGULATIONS

(I. Dynnikov, 1999)

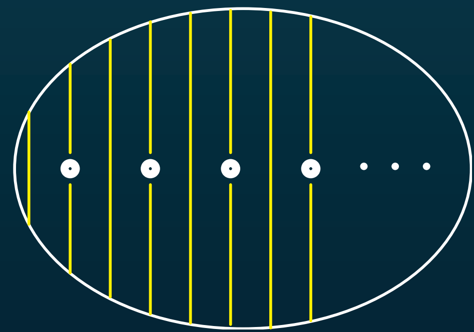
- View B_n as $\mathbf{MCG}(D_n)$, and let the homeo act on a fixed lamination L:



β (here σ_2) $\longmapsto$



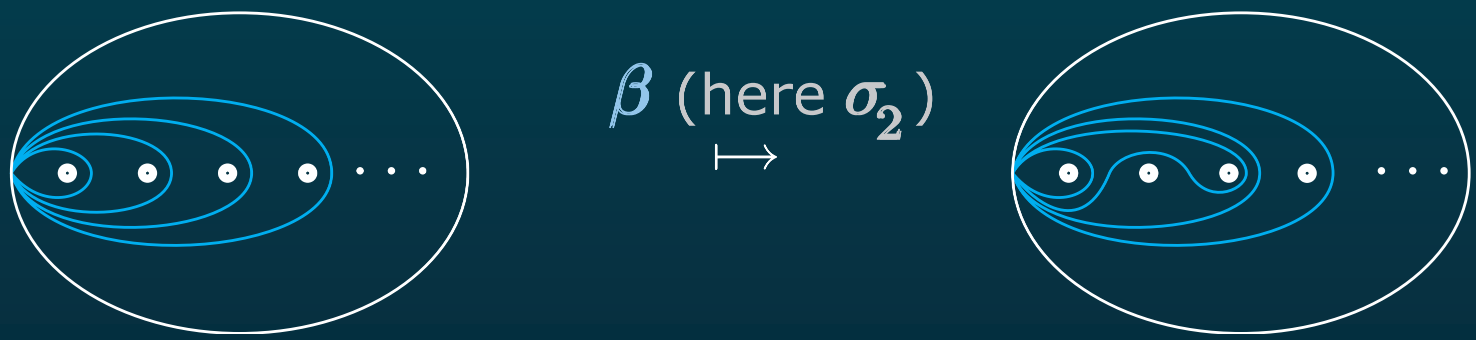
- Count the intersections with some fixed triangulation:

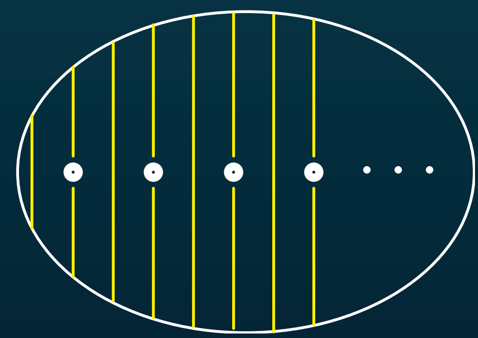


SOLUTION 6: SINGULAR TRIANGULATIONS

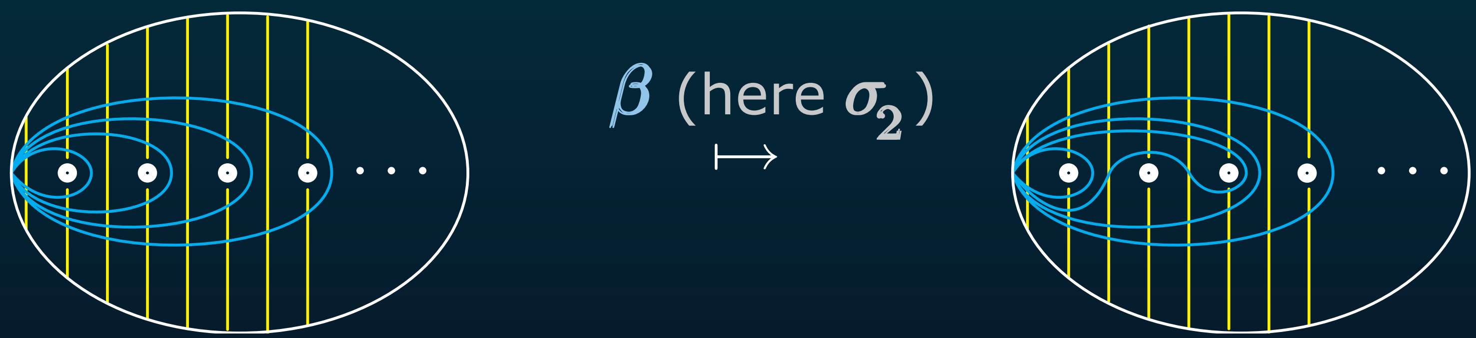
(I. Dynnikov, 1999)

- View B_n as $\mathbf{MCG}(D_n)$, and let the homeo act on a fixed lamination L:





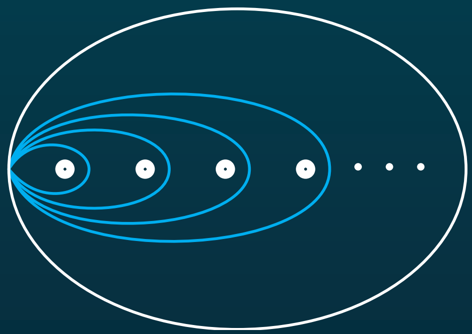
- Count the intersections with some fixed triangulation:



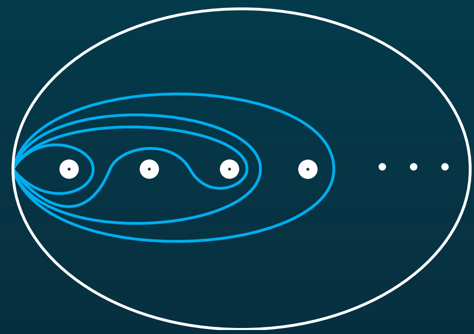
SOLUTION 6: SINGULAR TRIANGULATIONS

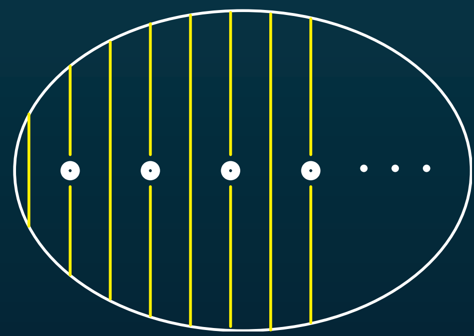
(I. Dynnikov, 1999)

- View B_n as $\mathbf{MCG}(D_n)$, and let the homeo act on a fixed lamination L:

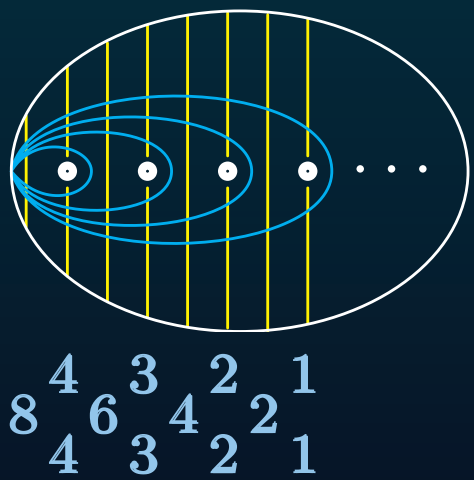


β (here σ_2) $\longmapsto$

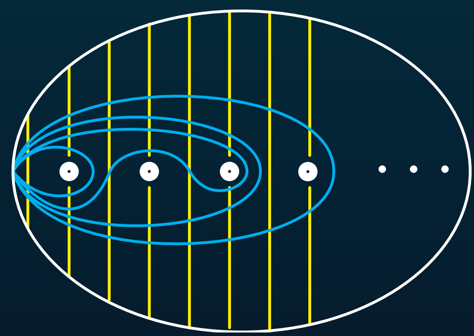




- Count the intersections with some fixed triangulation:

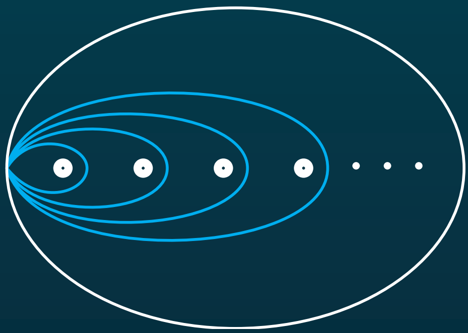


β (here σ_2) $\longmapsto$

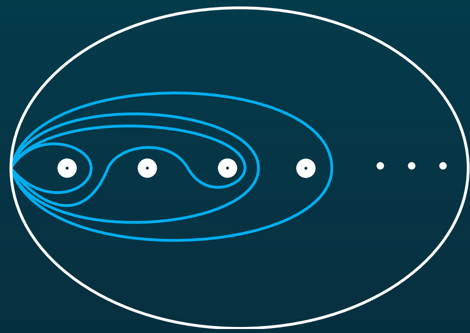


(I. Dynnikov, 1999)

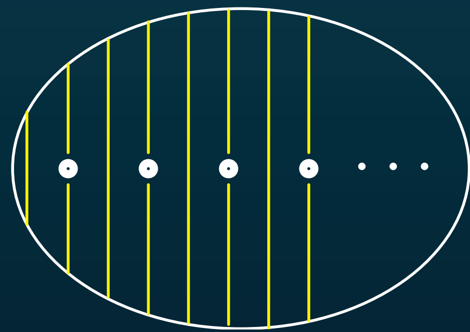
- View B_n as $\mathbf{MCG}(D_n)$, and let the homeo act on a fixed lamination L:

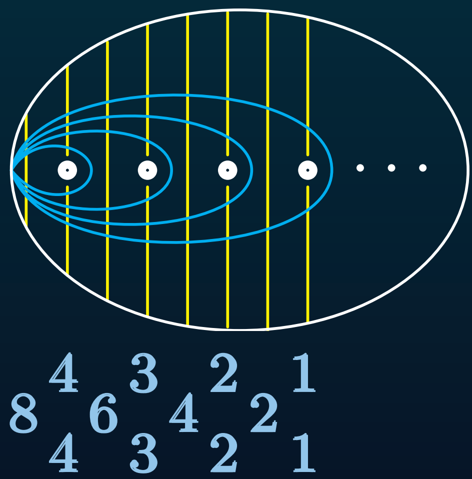


β (here σ_2) $\longmapsto$

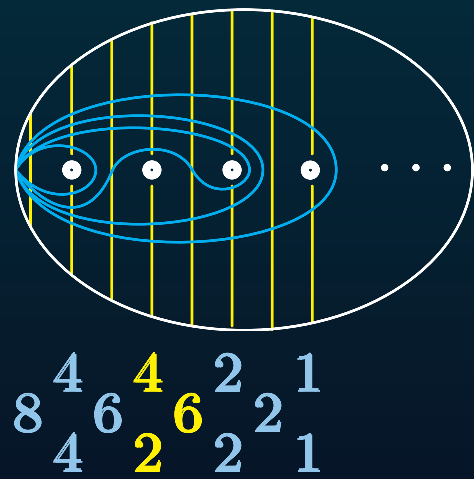


- Count the intersections with some fixed triangulation:



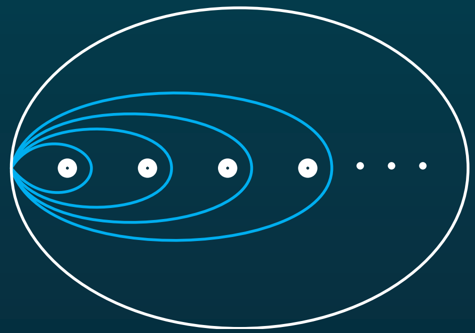


β (here σ_2) $\longmapsto$

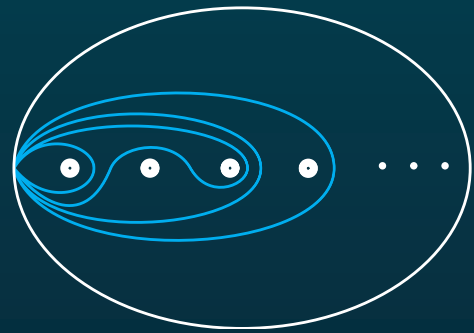


(I. Dynnikov, 1999)

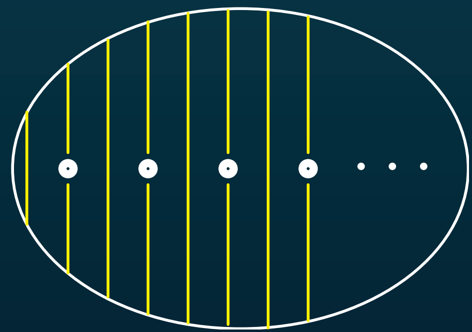
- View B_n as $\mathbf{MCG}(D_n)$, and let the homeo act on a fixed lamination L:

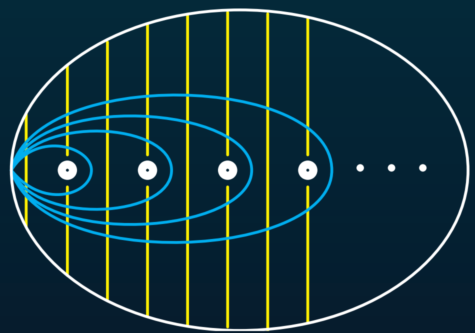


β (here σ_2) $\longmapsto$

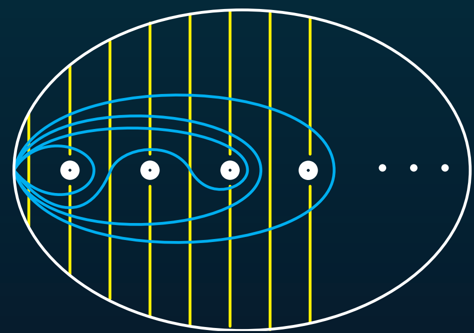


- Count the intersections with some fixed triangulation:





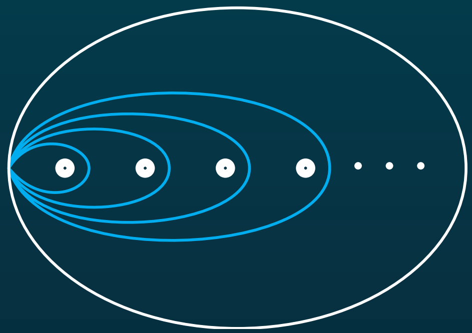
β (here σ_2) $\longmapsto$



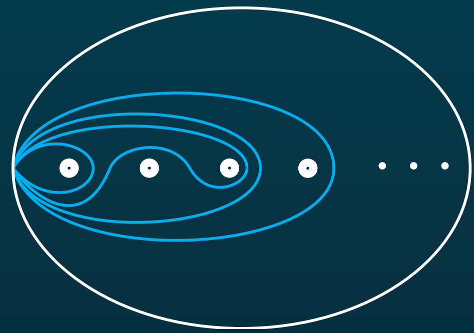
$8\,{}^{4}_{4}\,6\,{}^{3}_{3}\,4\,{}^{2}_{2}\,2\,{}^{1}_{1}\ldots \rightsquigarrow (0,1,0,1,0,1,0,\ldots)$ $\quad$ $8\,{}^{4}_{4}\,6\,{}^{4}_{2}\,6\,{}^{2}_{2}\,2\,{}^{1}_{1}\ldots \rightsquigarrow (0,1,1,0,0,2,0,\ldots)$

(I. Dynnikov, 1999)

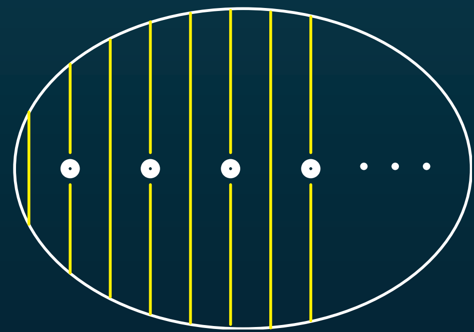
- View B_n as $\mathbf{MCG}(D_n)$, and let the homeo act on a fixed lamination L:

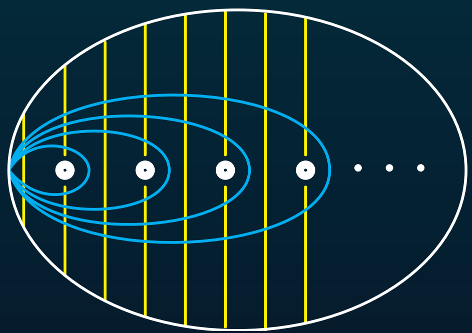


β (here σ_2) $\longmapsto$

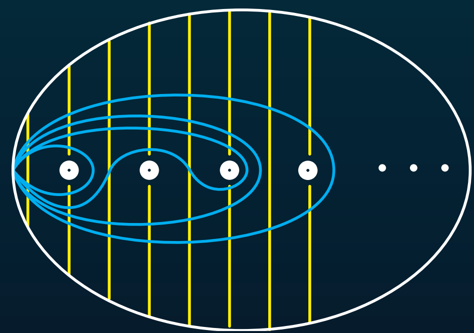


- Count the intersections with some fixed triangulation:





β (here σ_2) $\longmapsto$

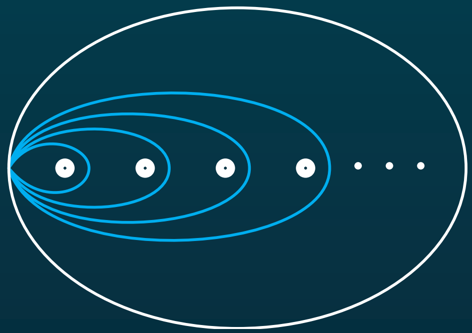


$8\,{}^{4}_{4}\,6\,{}^{3}_{3}\,4\,{}^{2}_{2}\,2\,{}^{1}_{1}\dots \rightsquigarrow (0,1,0,1,0,1,0,\dots)$ $\quad$ $8\,{}^{4}_{4}\,6\,{}^{4}_{2}\,6\,{}^{2}_{2}\,2\,{}^{1}_{1}\dots \rightsquigarrow (0,1,1,0,0,2,0,\dots)$

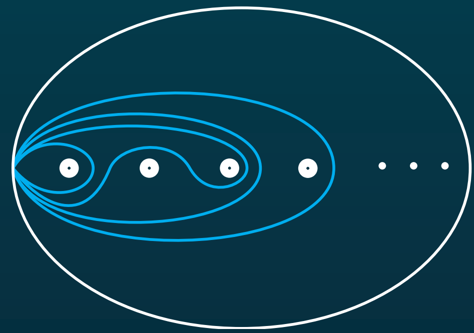
$\rightsquigarrow$ Explicit injection $B_n \hookrightarrow \mathbb{Z}^{2n}$: coordinates for $L \cdot \beta$.

(I. Dynnikov, 1999)

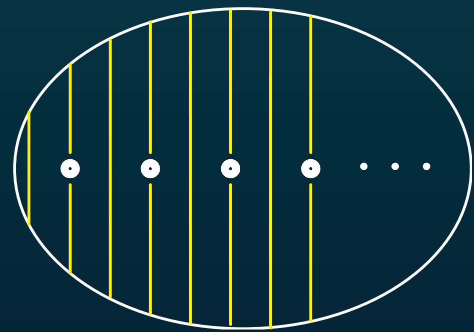
- View B_n as $\mathbf{MCG}(D_n)$, and let the homeo act on a fixed lamination L:

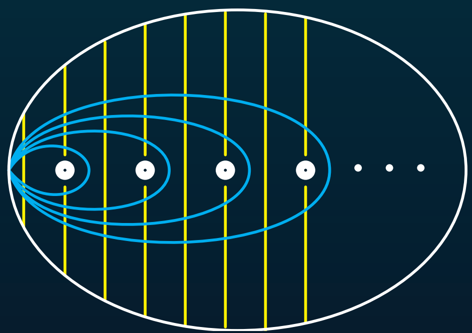


β (here σ_2) $\longmapsto$

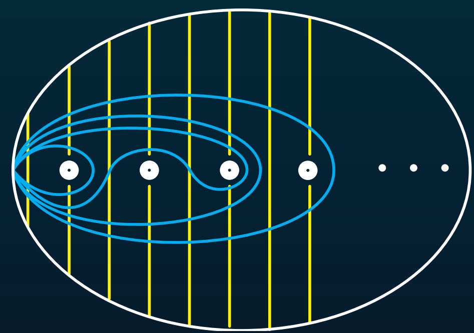


- Count the intersections with some fixed triangulation:





β (here σ_2) $\longmapsto$



$8\,{}^{4}_{4}\,6\,{}^{3}_{3}\,4\,{}^{2}_{2}\,2\,{}^{1}_{1}\ldots \rightsquigarrow (0,1,0,1,0,1,0,\ldots)$ $\quad$ $8\,{}^{4}_{4}\,6\,{}^{4}_{2}\,6\,{}^{2}_{2}\,2\,{}^{1}_{1}\ldots \rightsquigarrow (0,1,1,0,0,2,0,\ldots)$

$\rightsquigarrow$ Explicit injection $B_n \hookrightarrow \mathbb{Z}^{2n}$: coordinates for $L \cdot \beta$.

- Behind: automatic structure for mapping class groups (Mosher)

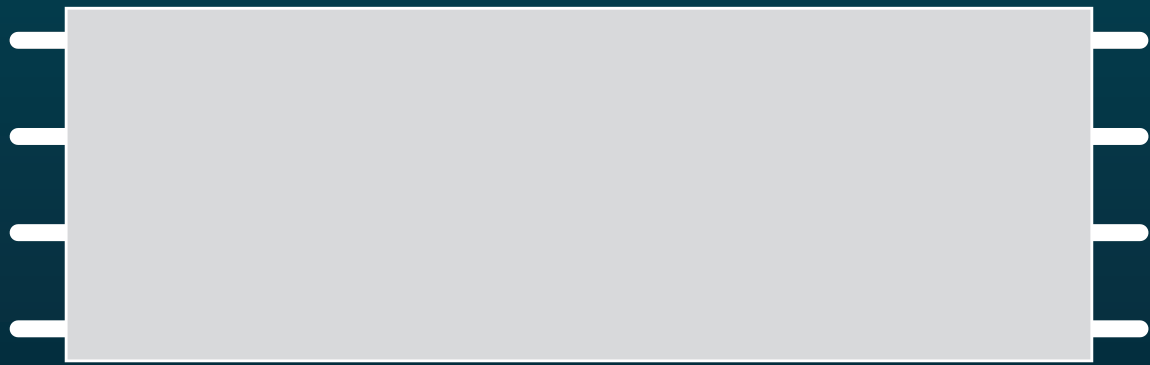
(D., 2007)

- Another unique normal form for (positive) braids.

SOLUTION 7: ALTERNATING DECOMPOSITION

(D., 2007)

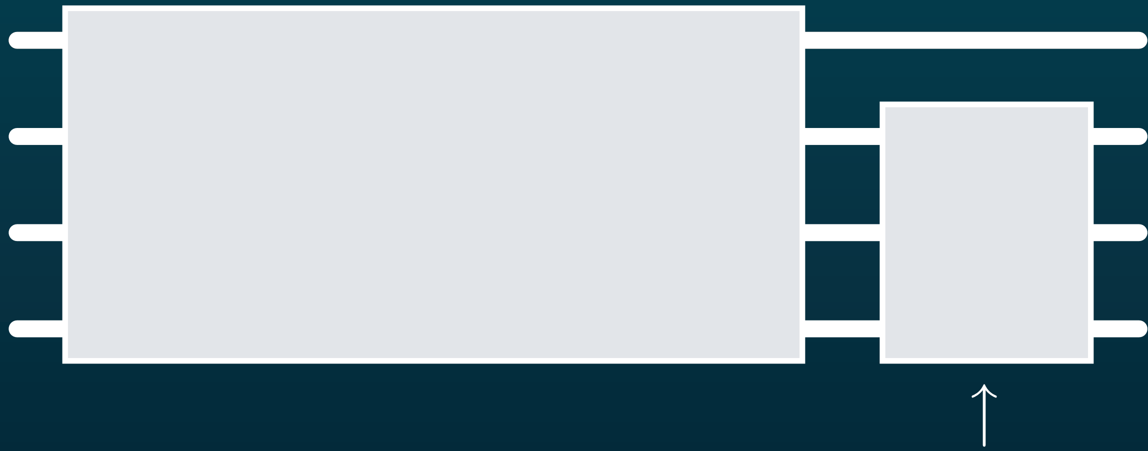
- Another unique normal form for (positive) braids.



SOLUTION 7: ALTERNATING DECOMPOSITION

(D., 2007)

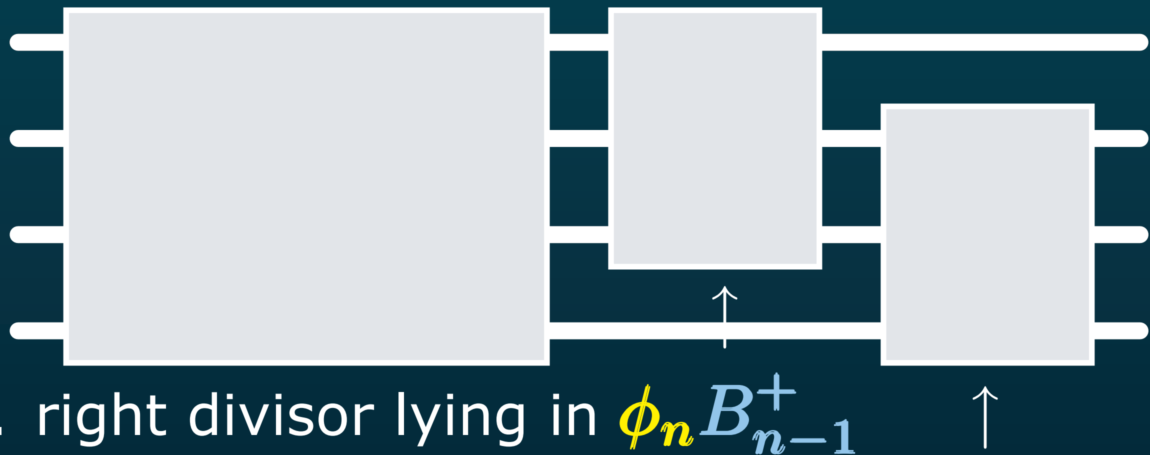
- Another unique normal form for (positive) braids.



max. right divisor lying in B^+_{n-1}

(D., 2007)

- Another unique normal form for (positive) braids.



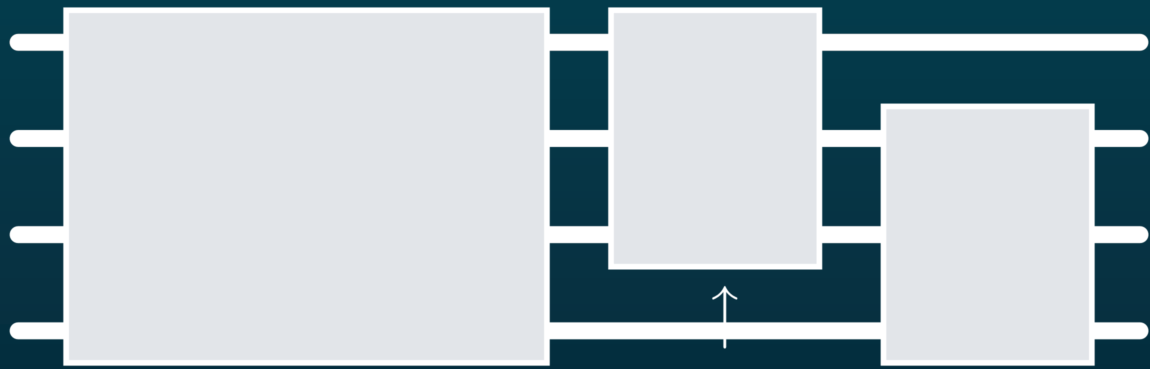
max. right divisor lying in $\phi_n B^+_{n-1}$

max. right divisor lying in B^+_{n-1}

SOLUTION 7: ALTERNATING DECOMPOSITION

(D., 2007)

- Another unique normal form for (positive) braids.



↑ max. right divisor lying in $\phi_n B^+_{n-1}$

↑ max. right divisor lying in B^+_{n-1}

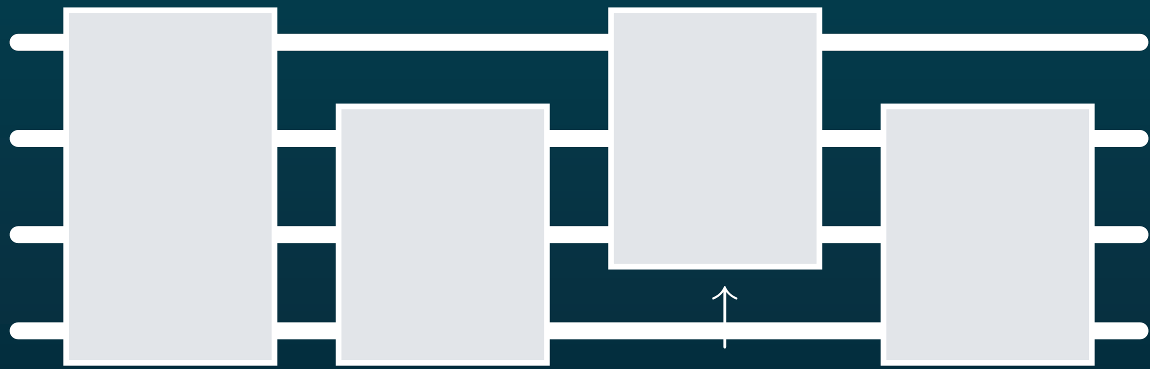
↗ the flip automorphism $\sigma_i \mapsto \sigma_{n-i}$

= conjugation by Garside's element Δ_n

SOLUTION 7: ALTERNATING DECOMPOSITION

(D., 2007)

- Another unique normal form for (positive) braids.



max. right divisor lying in $\phi_n B_{n-1}^+$ ↑

↑ max. right divisor lying in B_{n-1}^+

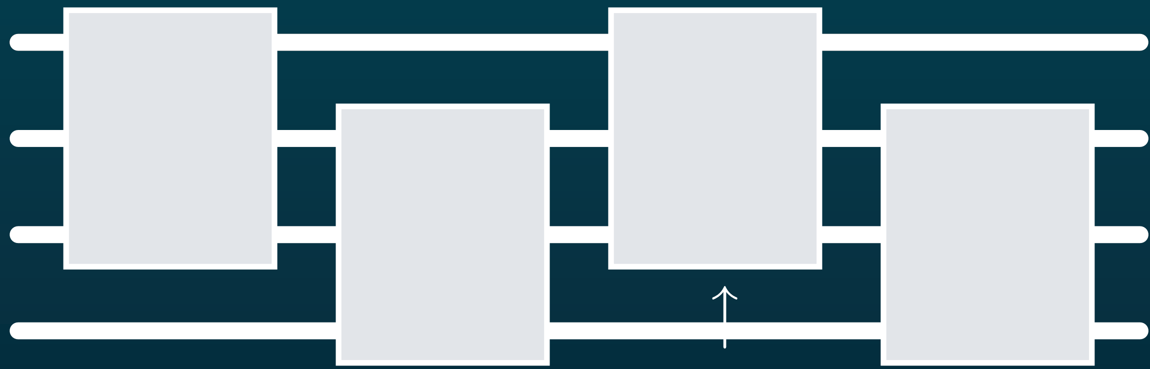
↗ the flip automorphism $\sigma_i \mapsto \sigma_{n-i}$

= conjugation by Garside's element Δ_n

SOLUTION 7: ALTERNATING DECOMPOSITION

(D., 2007)

- Another unique normal form for (positive) braids.



max. right divisor lying in $\phi_n B^+_{n-1}$

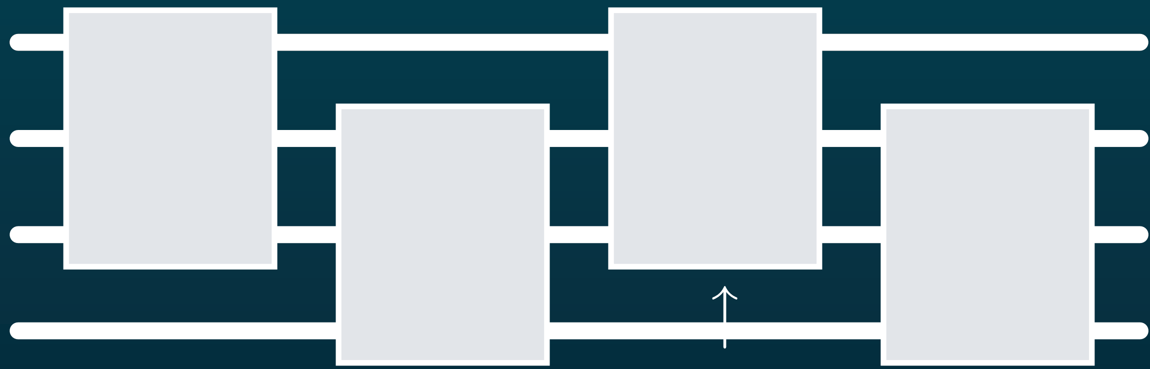
max. right divisor lying in B^+_{n-1}

the flip automorphism $\sigma_i \mapsto \sigma_{n-i}$

= conjugation by Garside's element Δ_n

(D., 2007)

- Another unique normal form for (positive) braids.



max. right divisor lying in $\phi_n B_{n-1}^+$ ↑

↑ max. right divisor lying in B_{n-1}^+

↗ the flip automorphism $\sigma_i \mapsto \sigma_{n-i}$

= conjugation by Garside's element Δ_n

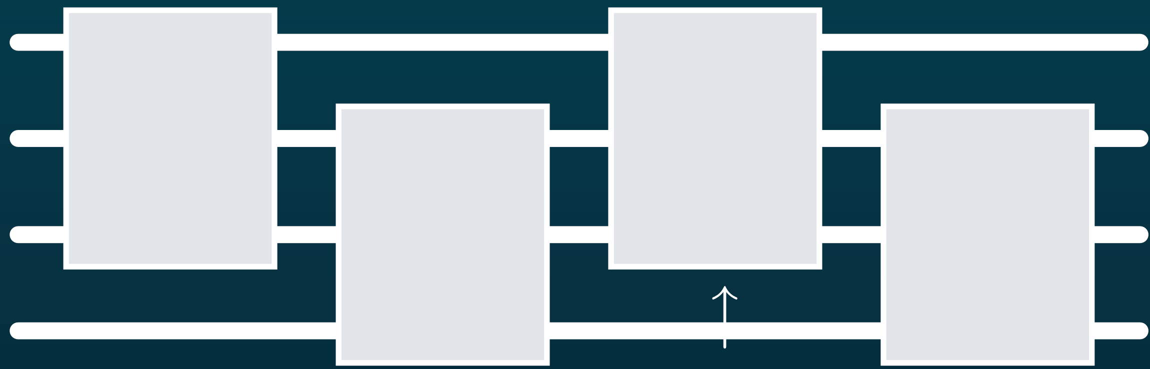
- Then: Every braid in B_n^+ admits a unique decomposition

$$x = \phi_n^{p-1} x_p \cdot \ldots \cdot \phi_n^2 x_3 \cdot \phi_n x_2 \cdot x_1$$

such that $x_p, \ldots, x_1$ lie in B_{n-1}^+

(D., 2007)

- Another unique normal form for (positive) braids.



max. right divisor lying in $\phi_n B_{n-1}^+$

max. right divisor lying in B_{n-1}^+

the flip automorphism $\sigma_i \mapsto \sigma_{n-i}$

= conjugation by Garside's element Δ_n

- Then: Every braid in B_n^+ admits a unique decomposition

$$x = \phi_n^{p-1} x_p \cdot \ldots \cdot \phi_n^2 x_3 \cdot \phi_n x_2 \cdot x_1$$

such that $x_p, \ldots, x_1$ lie in B_{n-1}^+ and the only σ_k that is a right divisor of $\phi_n^{p-k} x_p \cdot \ldots \cdot \phi_n x_{k+1} \cdot x_k$ is σ_1.

SOLUTION 6: FLIP NORMAL FORM VS. ORDER

- By iterating: flip normal form

- By iterating: flip normal form

$\rightsquigarrow$ quadratic time solution to the braid isotopy problem.

- By iterating: flip normal form

$\rightsquigarrow$ quadratic time solution to the braid isotopy problem.

- Behind: Standard braid order (“Dehornoy order"), Burckel's approach

- By iterating: flip normal form

$\rightsquigarrow$ quadratic time solution to the braid isotopy problem.

- Behind: Standard braid order (“Dehornoy order"), Burckel's approach

- Definition: For x, y braids, say that $x < y$ holds if, among all word expressions of $x^{-1}y$, there is at least one where the generator σ_i with higher index occurs only positively (σ_i occurs, σ_i^{-1} does not).

- By iterating: flip normal form

$\rightsquigarrow$ quadratic time solution to the braid isotopy problem.

- Behind: Standard braid order ("Dehornoy order"), Burckel's approach

- Definition: For x, y braids, say that $x < y$ holds if, among all word expressions of $x^{-1}y$, there is at least one where the generator σ_i with higher index occurs only positively (σ_i occurs, σ_i^{-1} does not).

- Assume $x, y \in B_n^+$, and let $(x_p, ..., x_1)$, $(y_q, ..., y_1)$ be the flip-decompositions of x and y. Then $x < y$ holds iff

• By iterating: flip normal form

$\rightsquigarrow$ quadratic time solution to the braid isotopy problem.

• Behind: Standard braid order (“Dehornoy order"), Burckel's approach

• Definition: For x, y braids, say that $x < y$ holds if, among all word expressions of $x^{-1}y$, there is at least one where the generator σ_i with higher index occurs only positively (σ_i occurs, σ_i^{-1} does not).

• Assume $x, y \in B_n^+$, and let $(x_p, ..., x_1)$, $(y_q, ..., y_1)$ be the flip-decompositions of x and y. Then $x < y$ holds iff

-either $p < q$,

- By iterating: flip normal form

$\rightsquigarrow$ quadratic time solution to the braid isotopy problem.

- Behind: Standard braid order (“Dehornoy order"), Burckel's approach

- Definition: For x, y braids, say that $x < y$ holds if, among all word expressions of $x^{-1}y$, there is at least one where the generator σ_i with higher index occurs only positively (σ_i occurs, σ_i^{-1} does not).

- Assume $x, y \in B_n^+$, and let $(x_p, ..., x_1)$, $(y_q, ..., y_1)$ be the flip-decompositions of x and y. Then $x < y$ holds iff
  - -either $p < q$,
  - -or $p = q$ and $(x_p, ..., x_1)$ is lexicographically smaller than $(y_q, ..., y_1)$.

- By iterating: flip normal form

  $\rightsquigarrow$ quadratic time solution to the braid isotopy problem.

- Behind: Standard braid order (“Dehornoy order"), Burckel's approach

- Definition: For x, y braids, say that $x < y$ holds if, among all word expressions of $x^{-1}y$, there is at least one where the generator σ_i with higher index occurs only positively (σ_i occurs, σ_i^{-1} does not).

- Assume $x, y \in B_n^+$, and let $(x_p, ..., x_1)$, $(y_q, ..., y_1)$ be the flip-decompositions of x and y. Then $x < y$ holds iff
  - -either $p < q$,
  - -or $p = q$ and $(x_p, ..., x_1)$ is lexicographically smaller than $(y_q, ..., y_1)$.

  $\rightsquigarrow$ completely defines the order on B_n^+ from the order on B_{n-1}^+

SOLUTION 8: THE DUAL BRAID MONOID

(Birman–Ko–Lee, 1997)

(Birman–Ko–Lee, 1997)

- New (redundant) family of generators: for $1 \leqslant i < j \leqslant n$, put

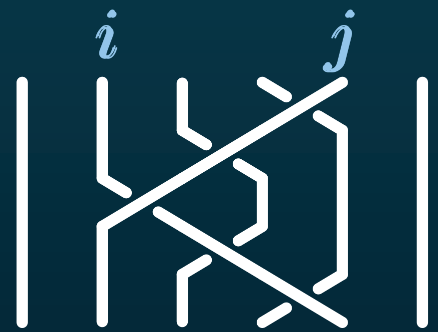
$$a_{i,j} = \sigma_{j-1} ... \sigma_{i+1} \sigma_i \sigma_{i+1}^{-1} ... \sigma_{j-1}^{-1}.$$

(Birman–Ko–Lee, 1997)

- New (redundant) family of generators: for $1 \leqslant i < j \leqslant n$, put

$$a_{i,j} = \sigma_{j-1} \dots \sigma_{i+1} \sigma_i \sigma_{i+1}^{-1} \dots \sigma_{j-1}^{-1}.$$

$a_{i,j} = \cdots$



$\cdots$

(Birman–Ko–Lee, 1997)

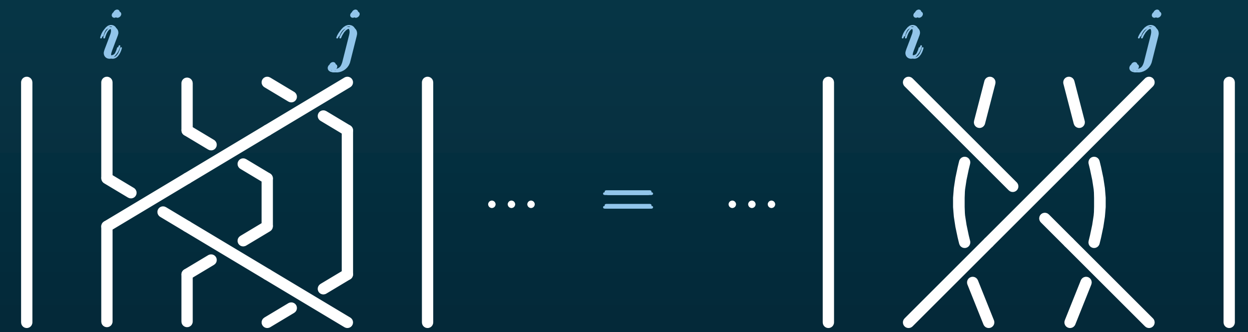
- New (redundant) family of generators: for $1 \leqslant i < j \leqslant n$, put

$$a_{i,j} = \sigma_{j-1} \ldots \sigma_{i+1} \sigma_i \sigma_{i+1}^{-1} \ldots \sigma_{j-1}^{-1}.$$

(Birman–Ko–Lee, 1997)

- New (redundant) family of generators: for $1 \leqslant i < j \leqslant n$, put

$$a_{i,j} = \sigma_{j-1} \ldots \sigma_{i+1} \sigma_i \sigma_{i+1}^{-1} \ldots \sigma_{j-1}^{-1}.$$

$a_{i,j} = \cdots$  $\cdots$

- Def: BKL_n^+ = submonoid of B_n generated by all $a_{i,j}$'s.

  ⇝ Another Garside structure, with Garside element $\delta_n = \sigma_{n-1} \ldots \sigma_2 \sigma_1$:

(Birman–Ko–Lee, 1997)

- New (redundant) family of generators: for $1 \leqslant i < j \leqslant n$, put

$$a_{i,j} = \sigma_{j-1} \dots \sigma_{i+1} \sigma_i \sigma_{i+1}^{-1} \dots \sigma_{j-1}^{-1}.$$

- Def: BKL_n^+ = submonoid of B_n generated by all $a_{i,j}$'s.

  ⇝ Another Garside structure, with Garside element $\delta_n = \sigma_{n-1} \dots \sigma_2 \sigma_1$: the dual Garside structure on B_n (same group, different monoids)

  ⇝ Greedy normal form

(Birman–Ko–Lee, 1997)

- New (redundant) family of generators: for $1 \leqslant i < j \leqslant n$, put

$$a_{i,j} = \sigma_{j-1} \dots \sigma_{i+1} \sigma_i \sigma_{i+1}^{-1} \dots \sigma_{j-1}^{-1}.$$

- Def: BKL_n^+ = submonoid of B_n generated by all $a_{i,j}$'s.

⇝ Another Garside structure, with Garside element $\delta_n = \sigma_{n-1} \dots \sigma_2 \sigma_1$: the dual Garside structure on B_n (same group, different monoids)

⇝ Greedy normal form

⇝ new quadratic solution to the braid isotopy problem

(Birman–Ko–Lee, 1997)

- New (redundant) family of generators: for $1 \leqslant i < j \leqslant n$, put

$$a_{i,j} = \sigma_{j-1} \ldots \sigma_{i+1} \sigma_i \sigma_{i+1}^{-1} \ldots \sigma_{j-1}^{-1}.$$

- Def: BKL_n^+ = submonoid of B_n generated by all $a_{i,j}$'s.
  - ⇝ Another Garside structure, with Garside element $\delta_n = \sigma_{n-1} \ldots \sigma_2 \sigma_1$: the dual Garside structure on B_n (same group, different monoids)
    - ⇝ Greedy normal form
      - ⇝ new quadratic solution to the braid isotopy problem
    - ⇝ Automatic structure, etc.

SOLUTION 9: THE CYCLING NORMAL FORM

(Fromentin, 2007)

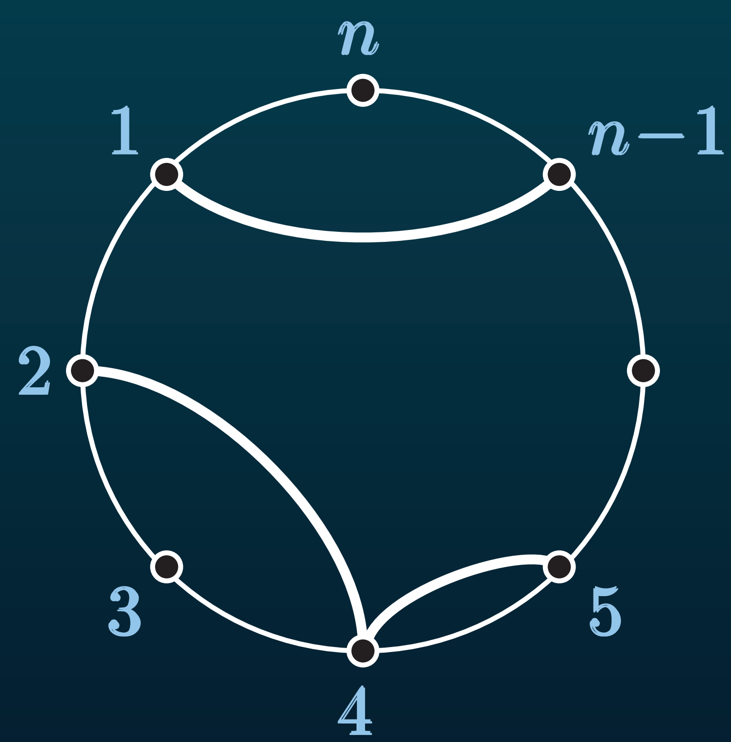
(Fromentin, 2007)

- Mix the ideas of the flip normal form and the Birman-Ko-Lee monoids

(Fromentin, 2007)

- Mix the ideas of the flip normal form and the Birman-Ko-Lee monoids

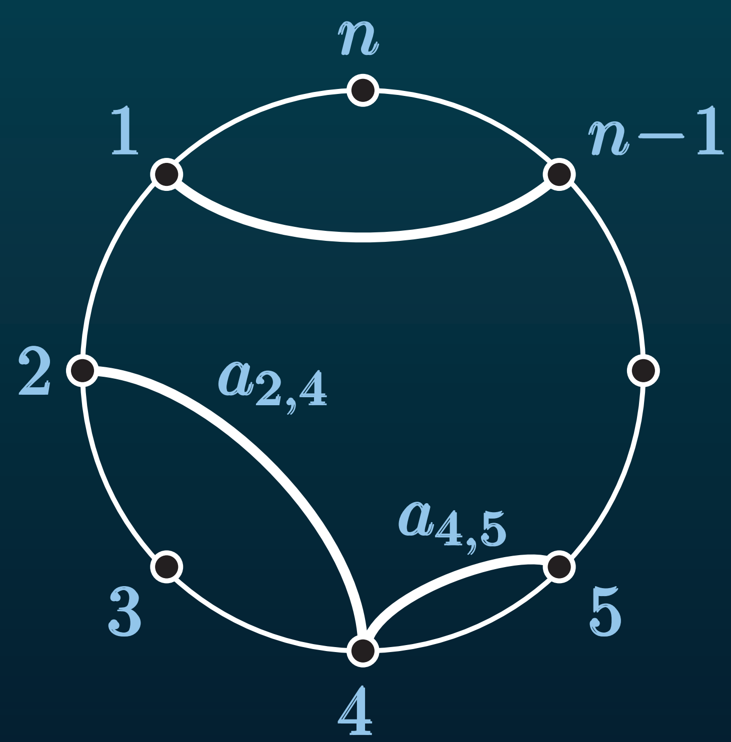
⇝ Generators $a_{i,j}$ of BKL_n^+
= chords of a circle



(Fromentin, 2007)

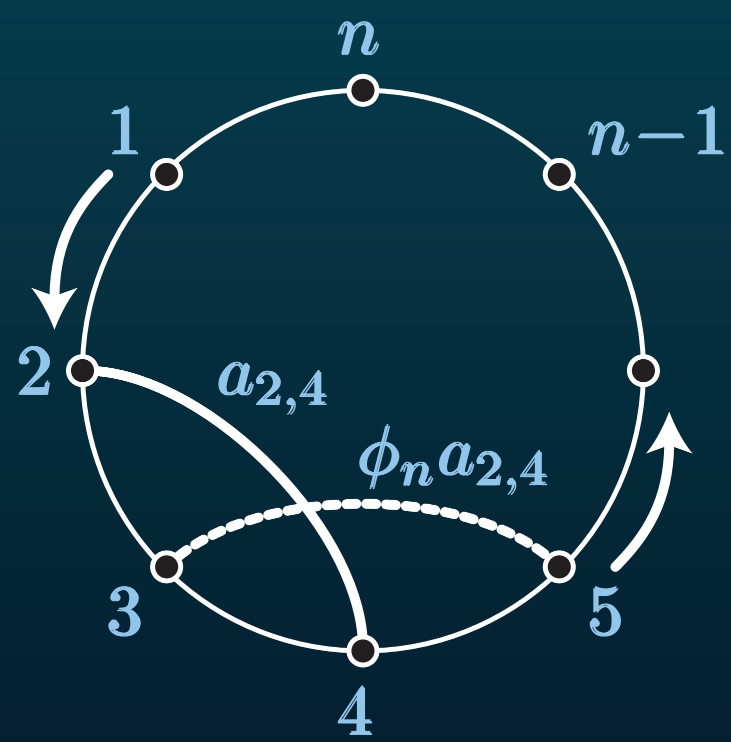
- Mix the ideas of the flip normal form and the Birman-Ko-Lee monoids

⇝ Generators $a_{i,j}$ of BKL_n^+
= chords of a circle



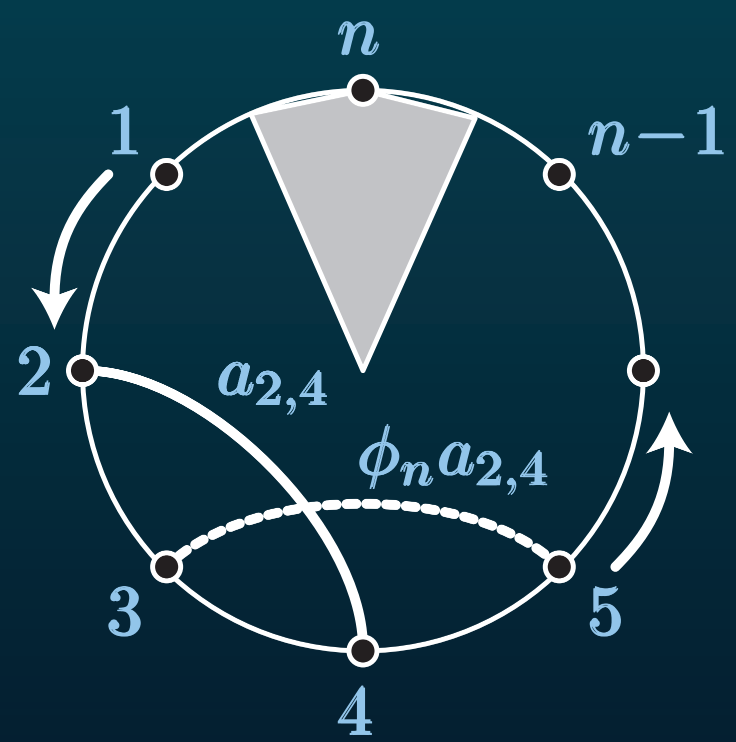
(Fromentin, 2007)

- Mix the ideas of the flip normal form and the Birman-Ko-Lee monoids
  - ⇝ Generators $a_{i,j}$ of BKL_n^+
    = chords of a circle
  - ⇝ Conjugation by δ_n
    = rotation by $2\pi/n$



(Fromentin, 2007)

- Mix the ideas of the flip normal form and the Birman-Ko-Lee monoids
  - ⇝ Generators $a_{i,j}$ of BKL_n^+
    = chords of a circle
  - ⇝ Conjugation by δ_n
    = rotation by $2\pi/n$
  - ⇝ Submonoid BKL_{n-1}^+
    = remove a $2\pi/n$-sector



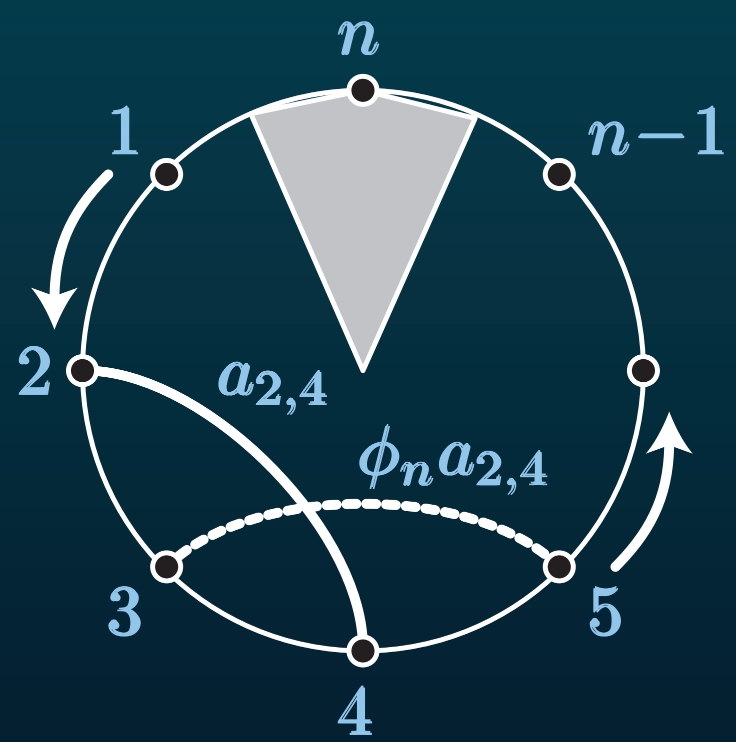
(Fromentin, 2007)

- Mix the ideas of the flip normal form and the Birman-Ko-Lee monoids

⇝ Generators $a_{i,j}$ of BKL_n^+
= chords of a circle

⇝ Conjugation by δ_n
= rotation by $2\pi/n$

⇝ Submonoid BKL_{n-1}^+
= remove a $2\pi/n$-sector



- Then: Every braid in BKL_n^+ admits a unique decomposition

$$x = \phi_n^{p-1} x_p \cdot \ldots \cdot \phi_n^2 x_3 \cdot \phi_n x_2 \cdot x_1$$

s.t. $x_p, \ldots, x_1$ lie in BKL_{n-1}^+ and the only $a_{i,j}$'s that is are right divisors of $\phi_n^{p-k} x_p \cdot \ldots \cdot \phi_n x_{k+1} \cdot x_k$ are $a_{i,n-1}$'s.

SOLUTION 8: CYCLING NORMAL FORM VS. ORDER

⇝ One more quadratic time solution to the braid isotopy problem,

$\rightsquigarrow$ One more quadratic time solution to the braid isotopy problem,
but also, mainly:

• Theorem (Fromentin, 2007) Assume $x, y \in BKL_n^+$, and let $(x_p, ..., x_1)$, $(y_q, ..., y_1)$ be the cycling decompositions of x and y. Then $x < y$ holds iff
- either $p < q$,
- or $p = q$ and $(x_p, ..., x_1)$ is lexicographically smaller than $(y_q, ..., y_1)$.

⇝ One more quadratic time solution to the braid isotopy problem,
but also, mainly:

- Theorem (Fromentin, 2007) Assume $x, y \in BKL_n^+$, and let $(x_p, ..., x_1)$, $(y_q, ..., y_1)$ be the cycling decompositions of x and y. Then $x < y$ holds iff
  - either $p < q$,
  - or $p = q$ and $(x_p, ..., x_1)$ is lexicographically smaller than $(y_q, ..., y_1)$.

⇝ New simple existence proof for the braid order;

$\rightsquigarrow$ One more quadratic time solution to the braid isotopy problem,
but also, mainly:

• Theorem (Fromentin, 2007) Assume $x, y \in BKL_n^+$, and let $(x_p, ..., x_1)$, $(y_q, ..., y_1)$ be the cycling decompositions of x and y. Then $x < y$ holds iff
- either $p < q$,
- or $p = q$ and $(x_p, ..., x_1)$ is lexicographically smaller than $(y_q, ..., y_1)$.

$\rightsquigarrow$ New simple existence proof for the braid order;

$\rightsquigarrow$ The restriction of the braid order to BKL_n^+ is a well-order of ordinal type $\omega^{\omega^{n-2}}$.

⇝ One more quadratic time solution to the braid isotopy problem,
but also, mainly:

- Theorem (Fromentin, 2007) Assume $x, y \in BKL_n^+$, and let $(x_p, ..., x_1)$, $(y_q, ..., y_1)$ be the cycling decompositions of x and y. Then $x < y$ holds iff
  - either $p < q$,
  - or $p = q$ and $(x_p, ..., x_1)$ is lexicographically smaller than $(y_q, ..., y_1)$.

⇝ New simple existence proof for the braid order;

⇝ The restriction of the braid order to BKL_n^+ is
a well-order of ordinal type $\omega^{\omega^{n-2}}$.

⇝ a distinguished element in each nonempty subset,

⇝ One more quadratic time solution to the braid isotopy problem,
but also, mainly:

• Theorem (Fromentin, 2007) Assume $x, y \in BKL_n^+$, and let $(x_p, ..., x_1)$, $(y_q, ..., y_1)$ be the cycling decompositions of x and y. Then $x < y$ holds iff

- either $p < q$,
- or $p = q$ and $(x_p, ..., x_1)$ is lexicographically smaller than $(y_q, ..., y_1)$.

⇝ New simple existence proof for the braid order;

⇝ The restriction of the braid order to BKL_n^+ is
a well-order of ordinal type $\omega^{\omega^{n-2}}$.

⇝ a distinguished element in each nonempty subset,
typically in each conjugacy class.