

LE CALCUL DES TRESSSES

Patrick Dehornoy



Laboratoire de Mathématiques

Nicolas Oresme, Caen

- D'autres objets que **nombres, fonctions, points et droites** ont une structure mathématique,
par exemple : les **tresses** (et beaucoup d'autres...)

- D'autres objets que **nombres, fonctions, points et droites** ont une structure mathématique,
par exemple : les **tresses** (et beaucoup d'autres...)
- Pourquoi «**calcul des tresses**» ?

- D'autres objets que **nombres**, **fonctions**, **points** et **droites** ont une structure mathématique,
par exemple : les **tresses** (et beaucoup d'autres...)
- Pourquoi «**calcul** des tresses» ?
 - ↔ les tresses généralisent (en un sens) les entiers,
en particulier, il existe des **algorithmes** de tresses,

- D'autres objets que **nombres**, **fonctions**, **points** et **droites** ont une structure mathématique,
par exemple : les **tresses** (et beaucoup d'autres...)
- Pourquoi «**calcul** des tresses» ?
 - ↔ les tresses généralisent (en un sens) les entiers, en particulier, il existe des **algorithmes** de tresses,
 - ↔ applications en **cryptographie**.

- **Qu'est-ce qu'une tresse ?**

- Qu'est-ce qu'une tresse ?

... des brins qui **se croisent** :



↪ (en oubliant la nature des brins) : suite de **croisements**

↔ (en oubliant la nature des brins) : suite de **croisements**

- Une tresse à **1** brin :



↔ (en oubliant la nature des brins) : suite de **croisements**

- Une tresse à **1** brin :



- Une tresse à **2** brins :



↔ (en oubliant la nature des brins) : suite de **croisements**

- Une tresse à **1** brin :



- Une tresse à **2** brins :



- Une tresse à **3** brins :



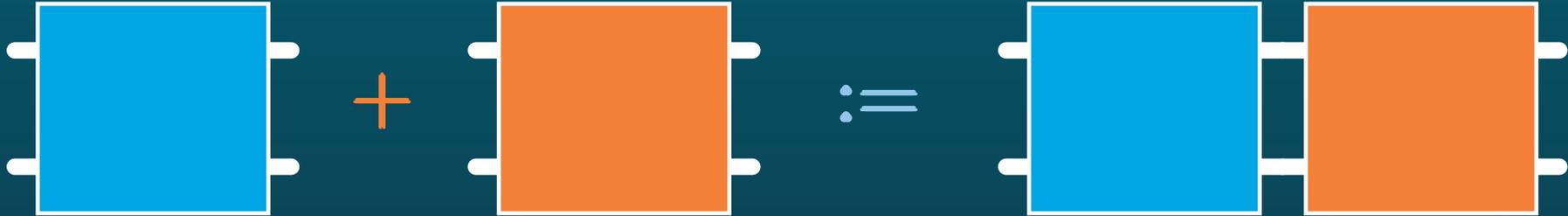
etc...

- **Calculer** avec les tresses à deux brins :

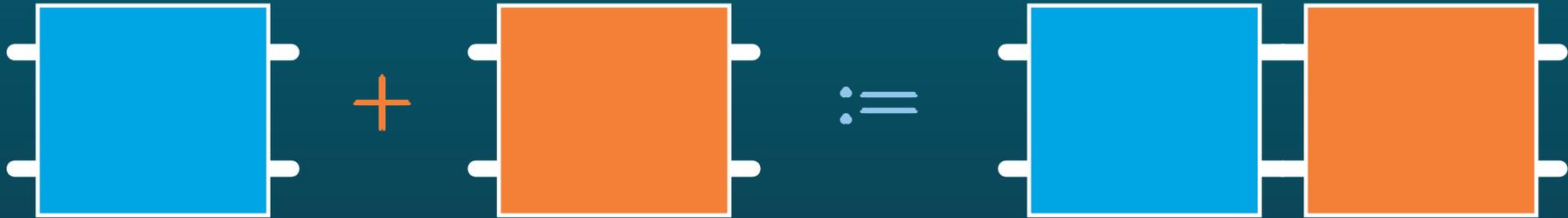
- **Calculer** avec les tresses à deux brins :



- **Calculer** avec les tresses à deux brins :

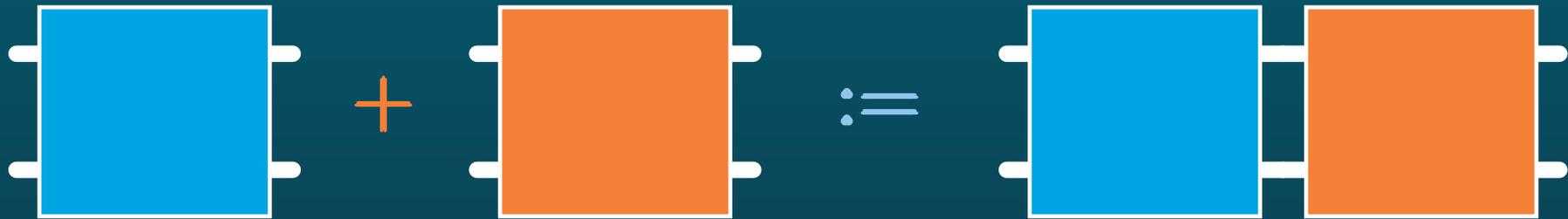


- **Calculer** avec les tresses à deux brins :



↔ Structure obtenue?

- **Calculer** avec les tresses à deux brins :

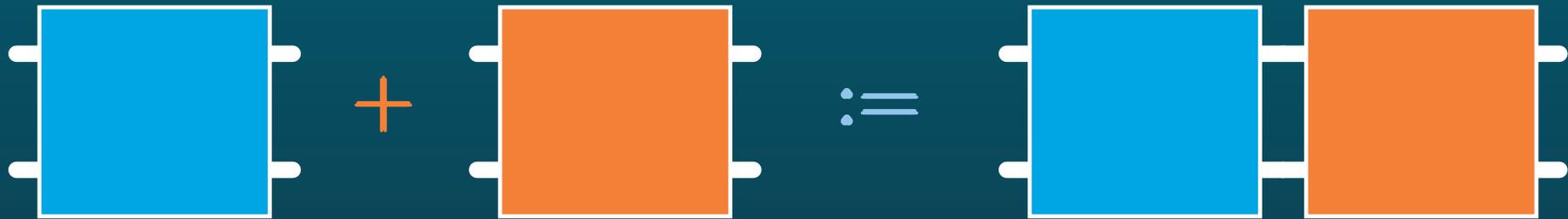


↔ Structure obtenue?

- Posons $0 :=$ , $1 :=$ . Alors :

$$1 + 0 =$$


- **Calculer** avec les tresses à deux brins :



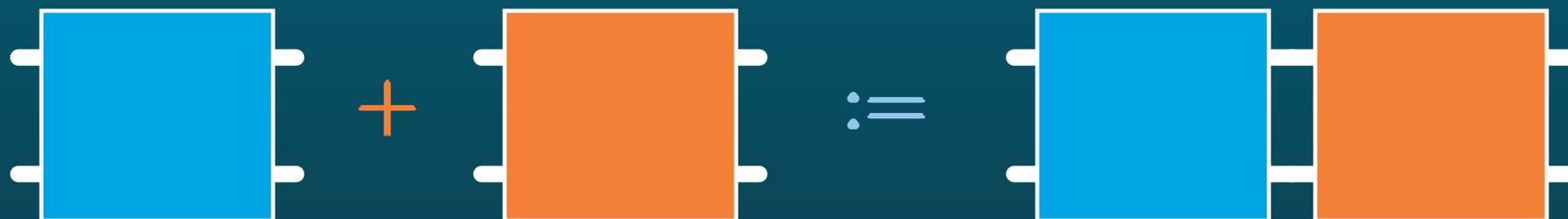
↔ Structure obtenue?

- Posons $0 :=$ , $1 :=$ . Alors :

$$1 + 0 = \text{crossing} + \text{parallel strands} = \text{crossing with parallel strands}$$

The equation shows that the sum of a crossing (1) and two parallel strands (0) is equal to a crossing where the strands continue as parallel lines.

- **Calculer** avec les tresses à deux brins :



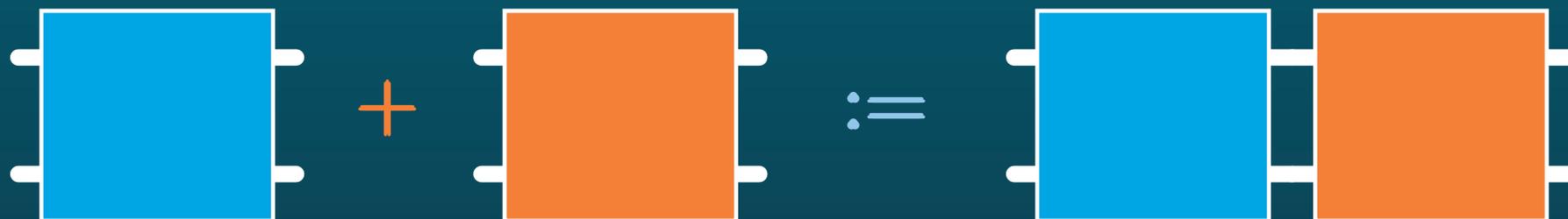
↔ Structure obtenue?

- Posons $0 :=$ , $1 :=$ . Alors :



se déforme en (?)

- **Calculer** avec les tresses à deux brins :



↔ Structure obtenue?

- Posons $0 :=$ , $1 :=$ . Alors :

$$1 + 0 = \text{crossing} + \text{parallel} = \text{crossing with a loop} \approx \text{crossing} = 1.$$

↑ se déforme en (?)

Et aussi :

• $0 + 1 =$  $+$  $=$  \approx  $= 1;$

Et aussi :

• $0 + 1 =$  $+$  $=$  \approx  $= 1;$

• $1 + 1 =$  $+$  $=$ 

Et aussi :

• $0 + 1 =$  $+$  $=$  \approx  $= 1;$

• $1 + 1 =$  $+$  $=$  \rightsquigarrow **2;**

Et aussi :

• $0 + 1 =$  $+$  $=$  \approx  $= 1;$

• $1 + 1 =$  $+$  $=$  $\rightsquigarrow 2;$

• $1 + 2 =$  $+$  $=$  $\rightsquigarrow 3$

Et aussi :

• $0 + 1 =$  $+$  $=$  \approx  $= 1;$

• $1 + 1 =$  $+$  $=$  $\rightsquigarrow 2;$

• $1 + 2 =$  $+$  $=$  $\rightsquigarrow 3$
 $=$  $+$  $= 2 + 1.$

↔ **Soustraction ?**

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) = \text{} + \text{} =$$

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) = \text{} + \text{} = \text{$$

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) = \text{crossing} + \text{crossing} = \text{canceling} \approx \text{canceling}$$

The diagram shows the addition of two crossings. The first crossing is the standard crossing (top-right to bottom-left). The second crossing is the opposite crossing (top-left to bottom-right). Their sum is represented by two crossings placed side-by-side, which are then shown to be equivalent to a single crossing where the two strands are parallel, with the crossing itself highlighted in orange.

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) = \text{crossing} + \text{crossing} = \text{two crossings} \approx \text{canceling crossing}$$

The diagram shows the equation $1 + (-1) =$ followed by a crossing with top-right to bottom-left diagonal, a plus sign, a crossing with top-left to bottom-right diagonal, an equals sign, two crossings (one top-right to bottom-left, one top-left to bottom-right), a tilde symbol, and finally a crossing with top-left to bottom-right diagonal where the crossing itself is highlighted in orange.

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) = \text{} + \text{} = \text{} \approx \text{$$

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) =$$
 $+$  $=$  \approx 

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) = \text{crossing} + \text{crossing} = \text{canceling} \approx \text{dot}$$

The diagram shows the addition of two crossings. The first crossing is the standard crossing (top-left to bottom-right). The second crossing is the opposite crossing (top-right to bottom-left). Their sum is represented by a diagram where the two crossings are placed side-by-side, and they cancel each other out, leaving a single dot. This is indicated by an approximation symbol \approx .

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) =$$
 $+$  $=$  \approx 

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) =$$
 $+$  $=$  \approx 

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) =$$
 $+$  $=$  \approx 

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) =$$
 $+$  $=$  \approx 

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) = \text{crossing} + \text{crossing} = \text{canceling crossings} \approx \text{two parallel lines}$$

The diagram shows the equation $1 + (-1) =$ followed by a crossing with top-right to bottom-left diagonal, a plus sign, a crossing with top-left to bottom-right diagonal, an equals sign, two crossings (one top-right to bottom-left, one top-left to bottom-right) that cancel each other out, an approximation symbol \approx , and finally two parallel horizontal lines, the top one colored orange and the bottom one white.

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) =$$
 $+$  $=$  \approx 

↪ Soustraction ?

• Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) = \text{} + \text{} = \text{} \approx \text{} = 0.$$

↪ Soustraction ?

- Posons $-1 :=$  ($\neq 1 =$  : orientation). Alors

$$1 + (-1) = \text{crossing} + \text{crossing} = \text{canceling} \approx \text{two parallel lines (top orange, bottom white)} = 0.$$

- De même

$$(-1) + 1 = \text{crossing} + \text{crossing} = \text{canceling} \approx \text{two parallel lines (top white, bottom white)} = 0.$$

- Un exemple:

$$1 + (-2) + 3 = \text{diagram 1} + \text{diagram 2} + \text{diagram 3}$$

The equation shows the sum of three diagrams representing the terms 1, -2, and 3. The first diagram is a single crossing. The second diagram is a crossing followed by a loop. The third diagram is a crossing followed by two loops.

● Un exemple:

$$1 + (-2) + 3 = \text{diagram 1} + \text{diagram 2} + \text{diagram 3}$$
$$= \text{diagram 4}$$

The diagrams are composed of white line segments on a dark teal background. Diagram 1 is a single crossing. Diagram 2 is a crossing with a loop. Diagram 3 is a crossing with two loops. Diagram 4 is a single crossing with four loops.

● Un exemple:

$$1 + (-2) + 3 = \text{diagram 1} + \text{diagram 2} + \text{diagram 3}$$
$$= \text{diagram 4}$$
$$\approx \text{diagram 5}$$

The diagrams are composed of white line segments on a dark teal background. Diagram 1 is a single crossing. Diagram 2 is a crossing with a loop. Diagram 3 is a crossing with two loops. Diagram 4 is a sequence of four crossings. Diagram 5 is a sequence of two parallel horizontal lines followed by three crossings.

● Un exemple:

$$\begin{aligned}
 1 + (-2) + 3 &= \text{Diagram 1} + \text{Diagram 2} + \text{Diagram 3} \\
 &= \text{Diagram 4} \\
 &\approx \text{Diagram 5} \\
 &\approx \text{Diagram 6}
 \end{aligned}$$

The diagrams are composed of white lines on a dark teal background. Diagram 1 shows a crossing of two lines. Diagram 2 shows a crossing with a loop. Diagram 3 shows two crossings. Diagram 4 is a single continuous zigzag line. Diagram 5 has a horizontal segment on the left. Diagram 6 has two horizontal segments on the left.

● Un exemple:

$$\begin{aligned}
 1 + (-2) + 3 &= \text{Diagram 1} + \text{Diagram 2} + \text{Diagram 3} \\
 &= \text{Diagram 4} \\
 &\approx \text{Diagram 5} \\
 &\approx \text{Diagram 6} \\
 &\approx \text{Diagram 7}
 \end{aligned}$$

The diagrams are composed of white lines on a dark teal background. Diagram 1 is a crossing. Diagram 2 is a crossing with a loop. Diagram 3 is a crossing with two loops. Diagram 4 is a sequence of four crossings. Diagram 5 has horizontal segments. Diagram 6 has a long horizontal segment. Diagram 7 is a crossing with a loop.

• Un exemple:

$$\begin{aligned}
 1 + (-2) + 3 &= \text{diagram 1} + \text{diagram 2} + \text{diagram 3} \\
 &= \text{diagram 4} \\
 &\approx \text{diagram 5} \\
 &\approx \text{diagram 6} \\
 &\approx \text{diagram 7} = 2.
 \end{aligned}$$

↔ Conclusion : Les tresses à 2 brins forment un **groupe** isomorphe à $(\mathbb{Z}, +)$.

à démontrer



↔ **Conclusion** : Les tresses à 2 brins forment un **groupe** isomorphe à $(\mathbb{Z}, +)$.

à démontrer

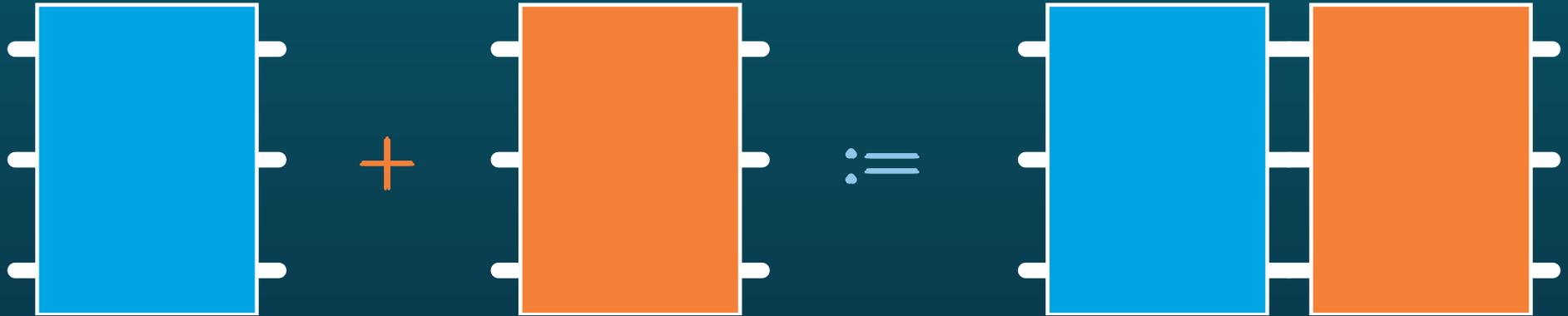


↪ Conclusion : Les tresses à 2 brins forment un **groupe** isomorphe à $(\mathbb{Z}, +)$.

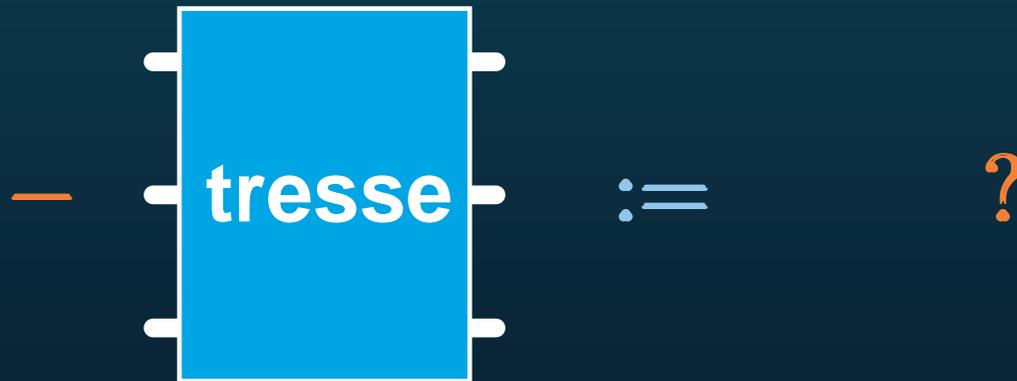
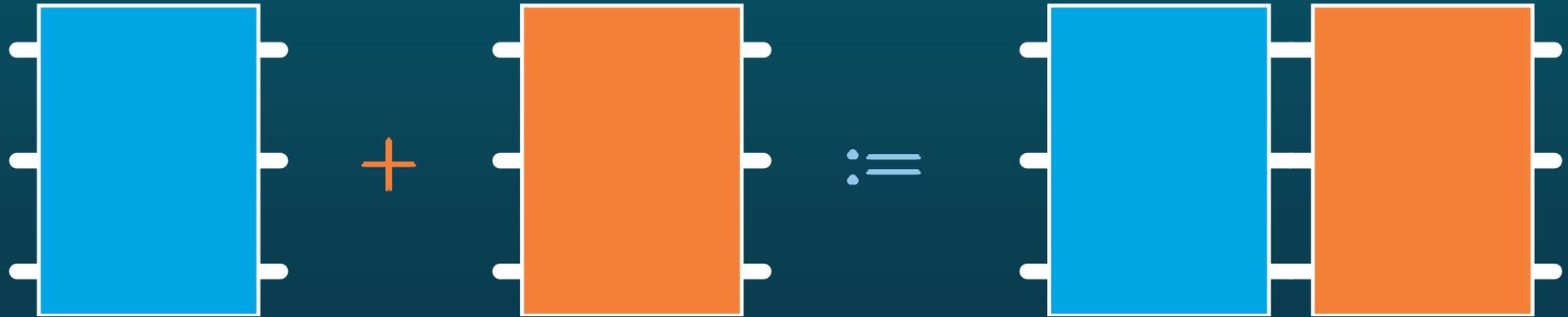
↪ pas une surprise : compter les demi-tours

↔ Même chose pour les tresses à n brins ($n \geq 3$) :

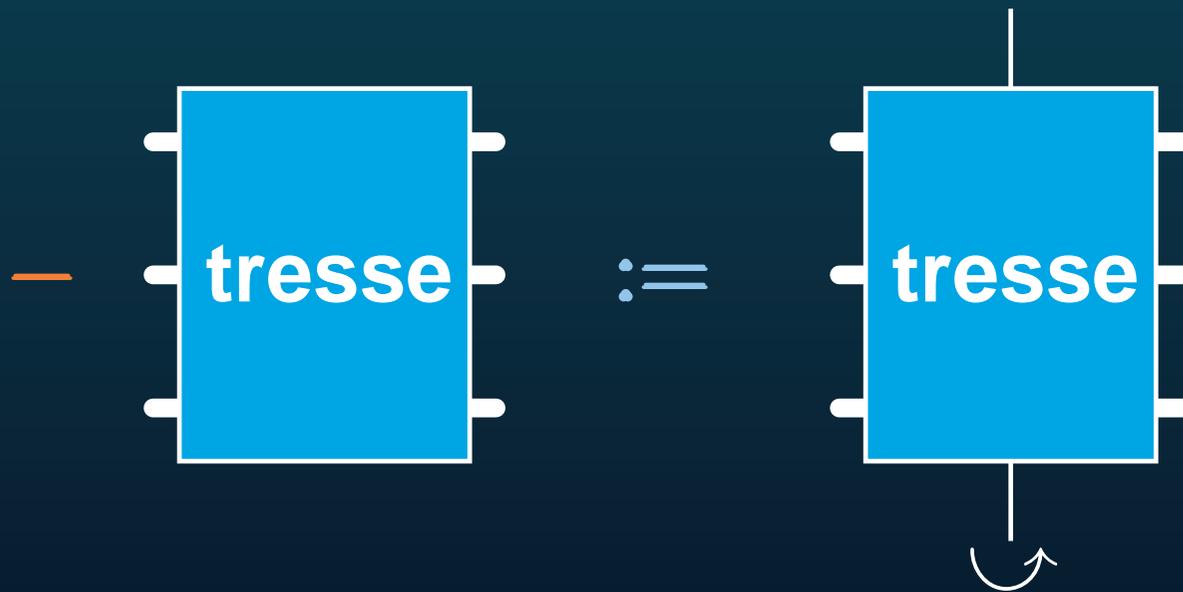
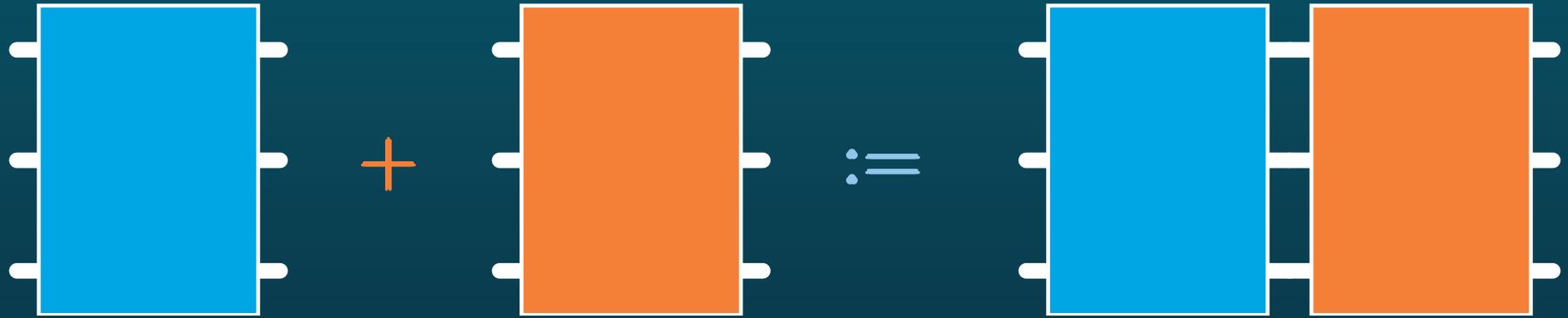
↔ Même chose pour les tresses à n brins ($n \geq 3$) :



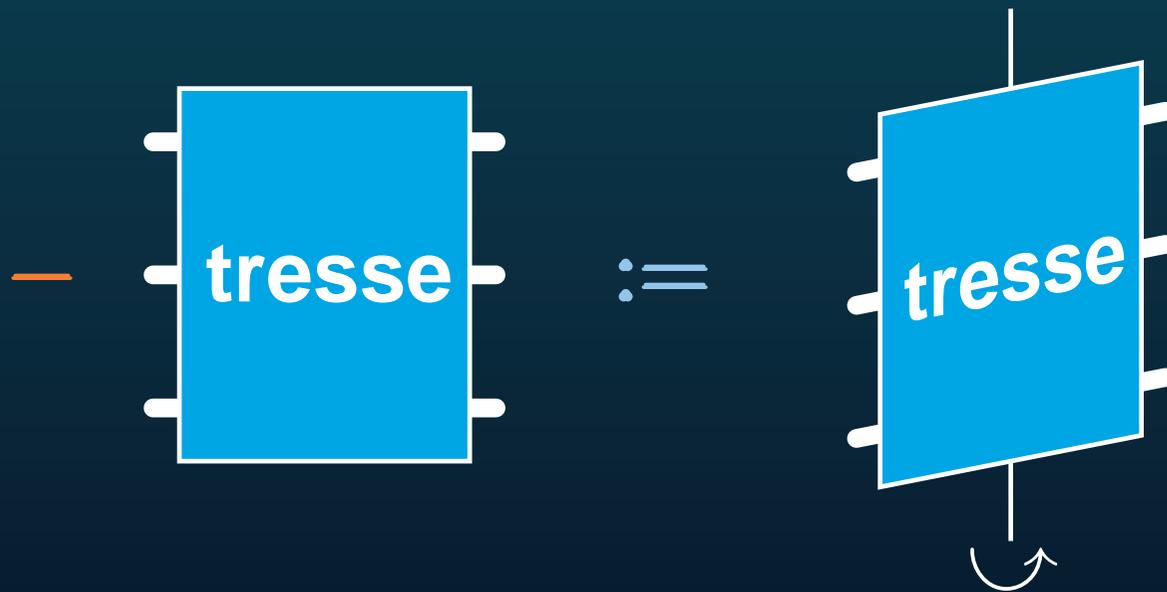
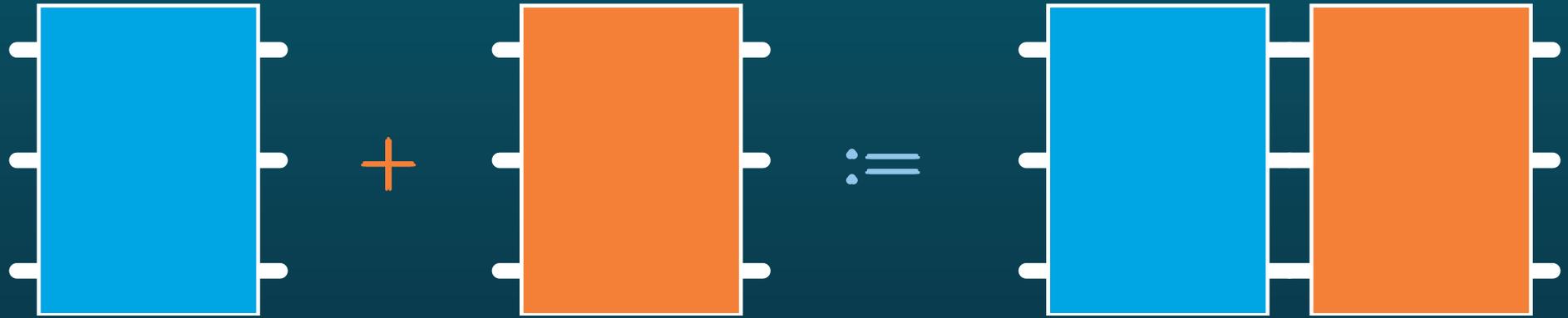
↔ Même chose pour les tresses à n brins ($n \geq 3$) :



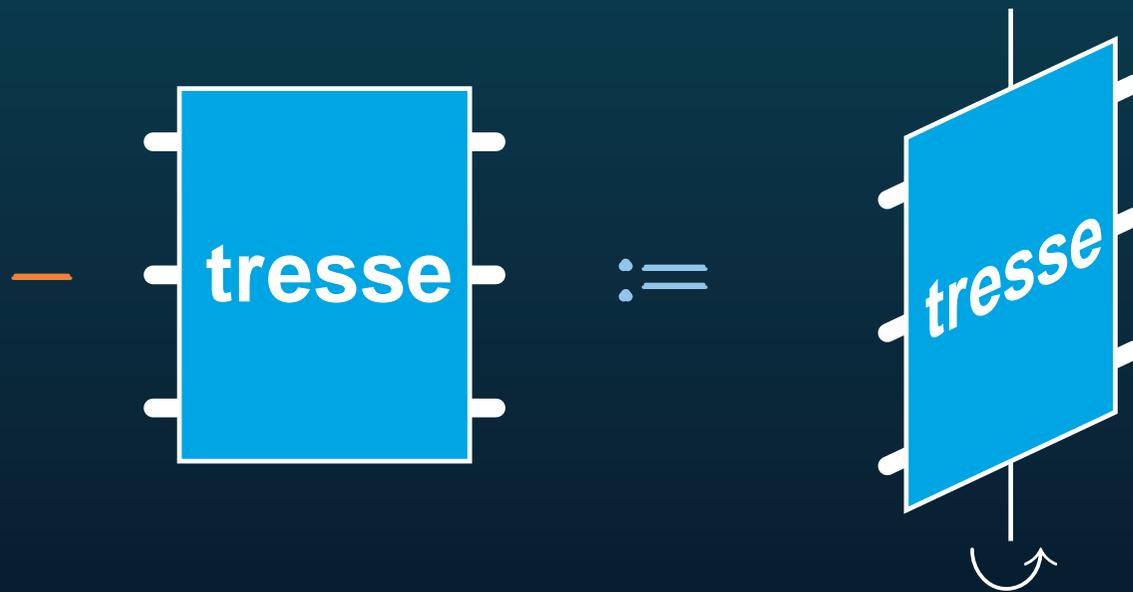
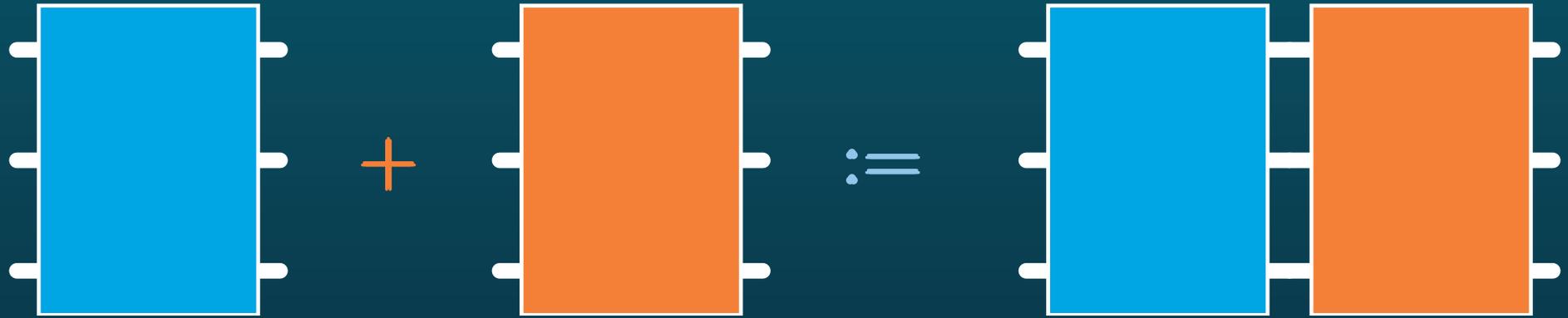
↔ Même chose pour les tresses à n brins ($n \geq 3$) :



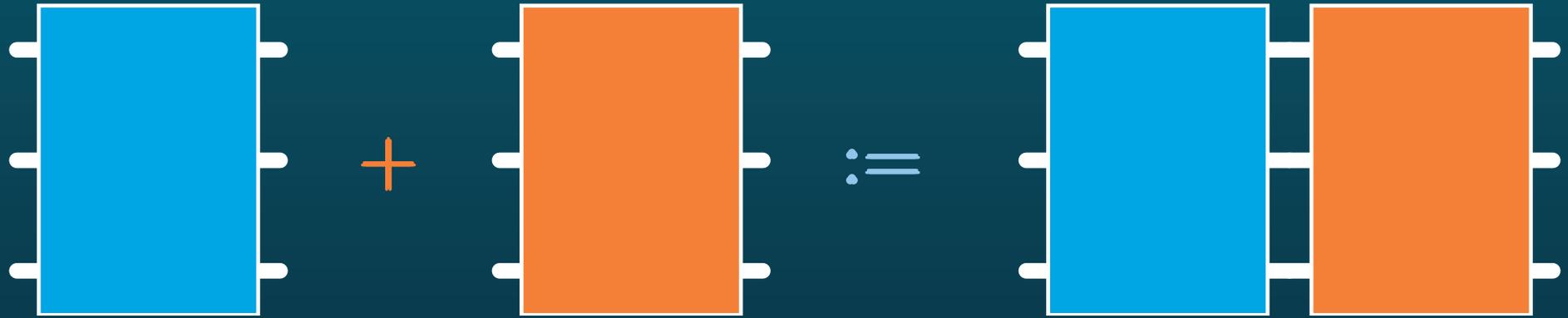
↔ Même chose pour les tresses à n brins ($n \geq 3$) :



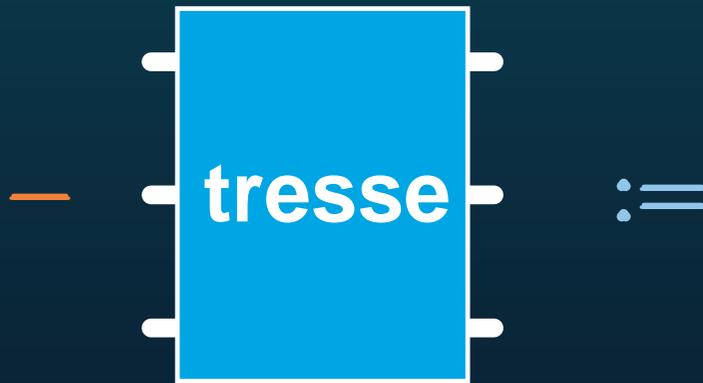
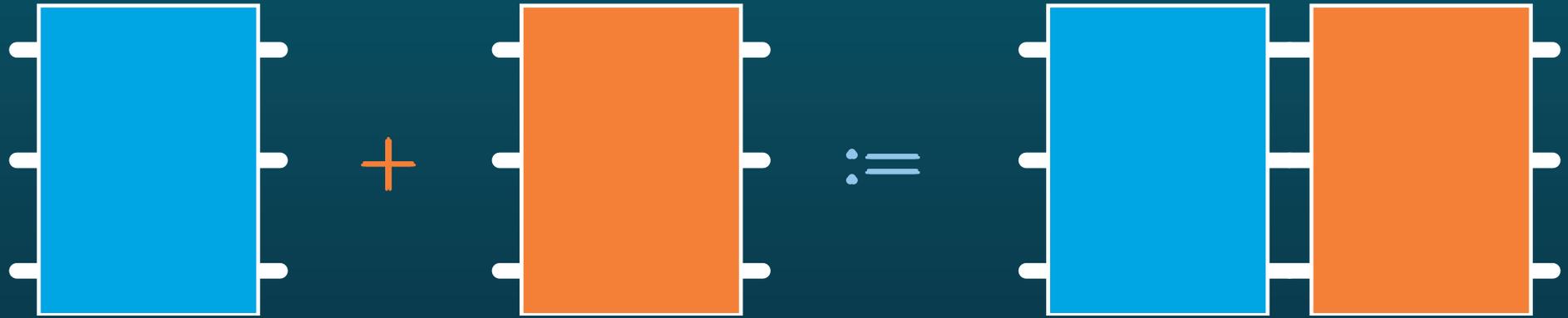
↔ Même chose pour les tresses à n brins ($n \geq 3$) :



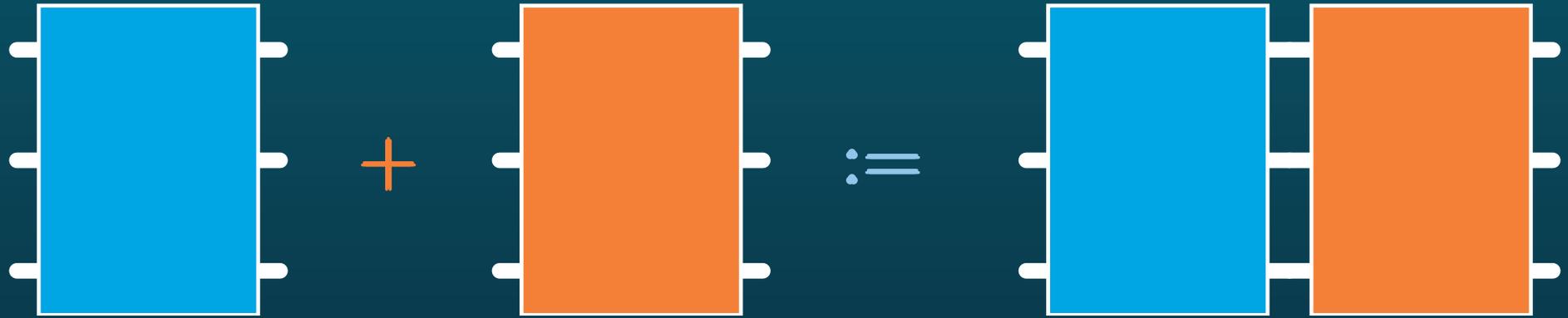
↔ Même chose pour les tresses à n brins ($n \geq 3$) :



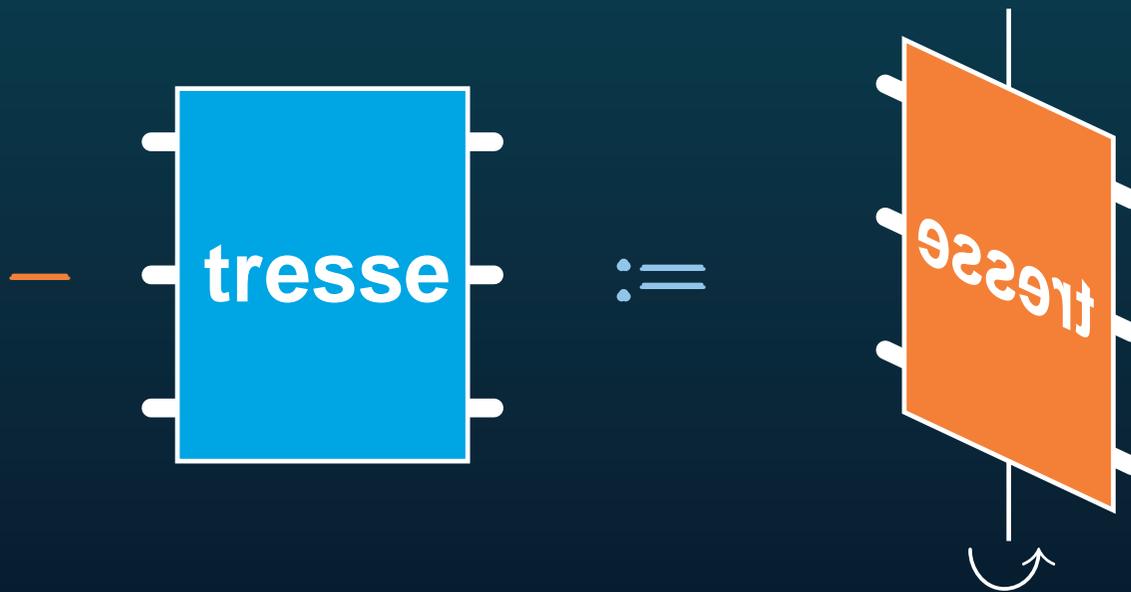
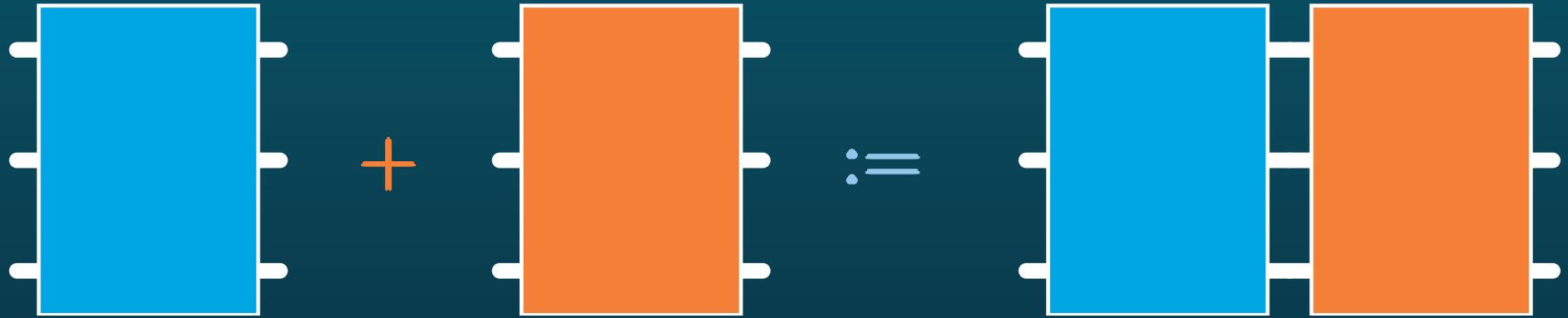
↔ Même chose pour les tresses à n brins ($n \geq 3$) :



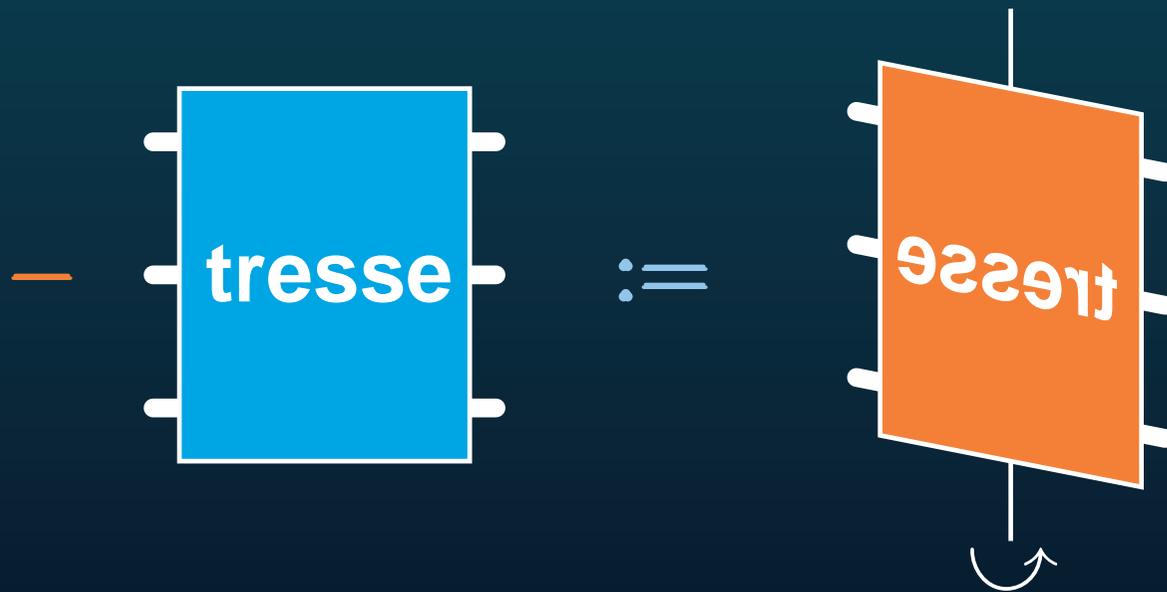
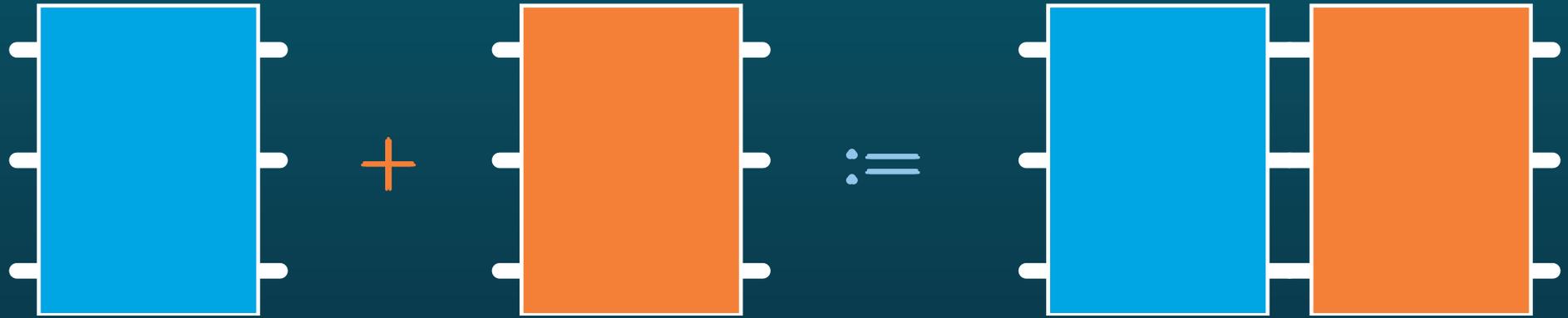
↔ Même chose pour les tresses à n brins ($n \geq 3$) :



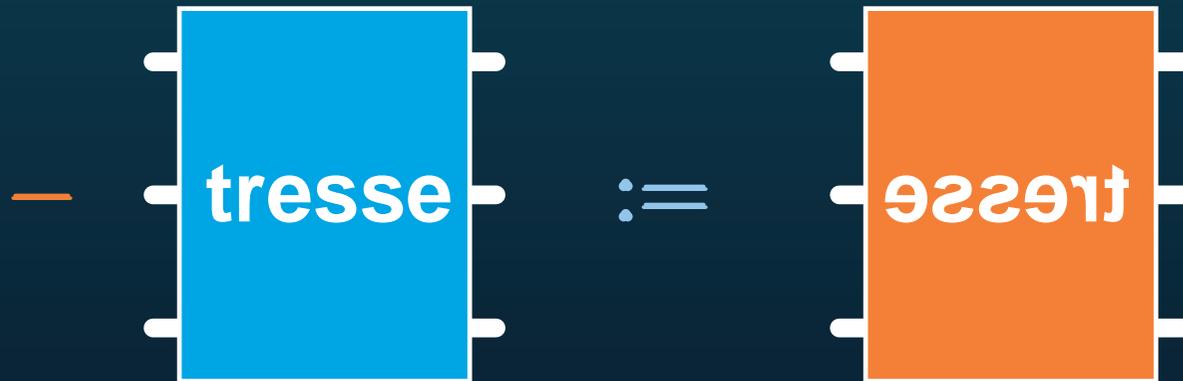
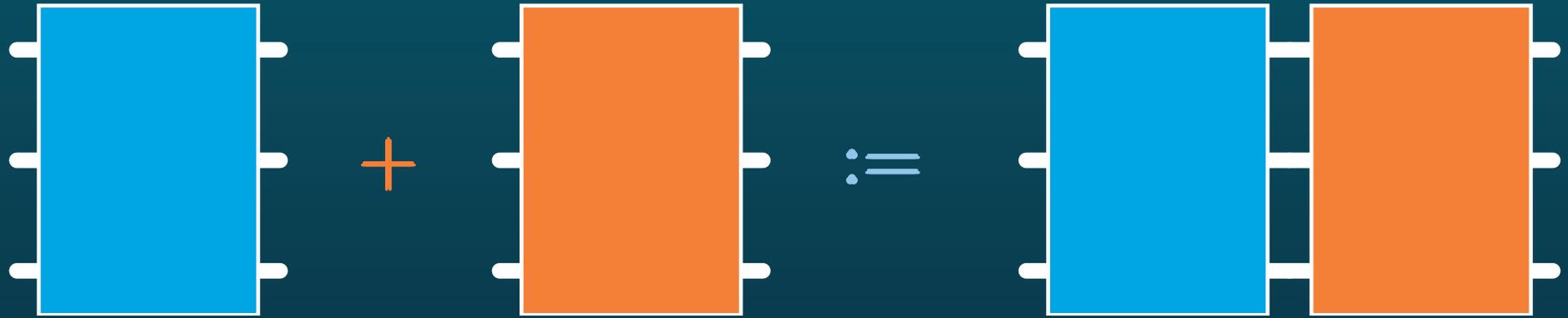
↔ Même chose pour les tresses à n brins ($n \geq 3$) :



↔ Même chose pour les tresses à n brins ($n \geq 3$) :



↔ Même chose pour les tresses à n brins ($n \geq 3$) :



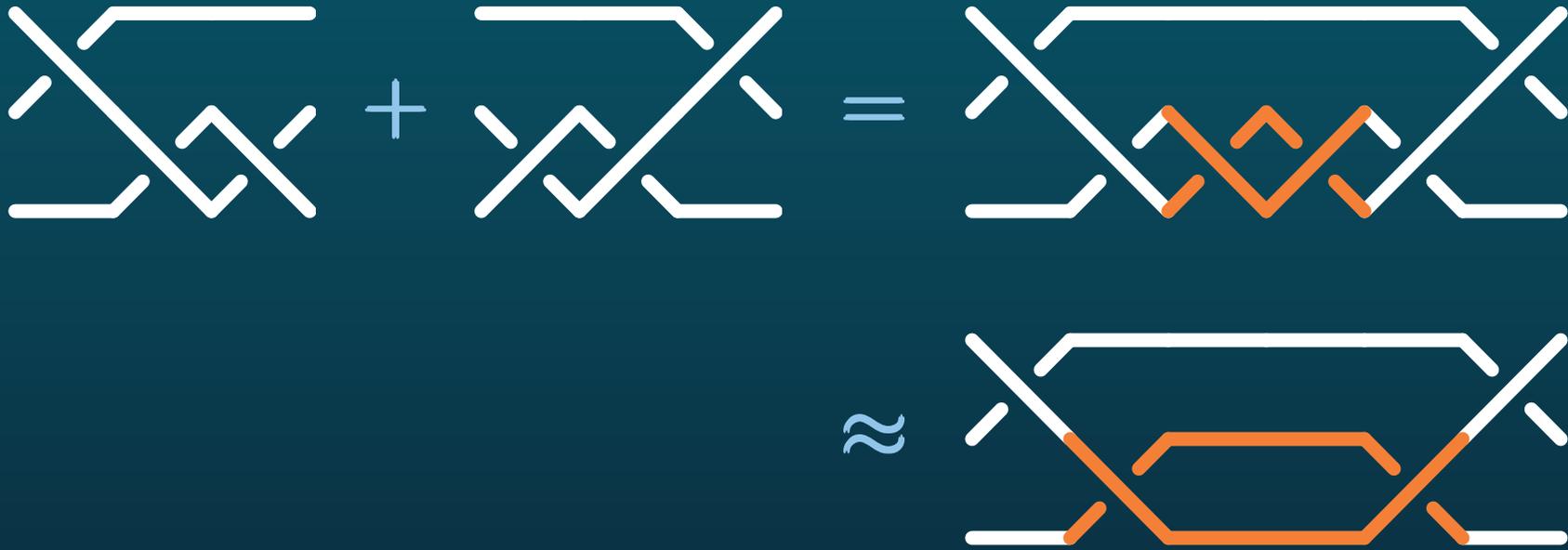
- **Exemple:**



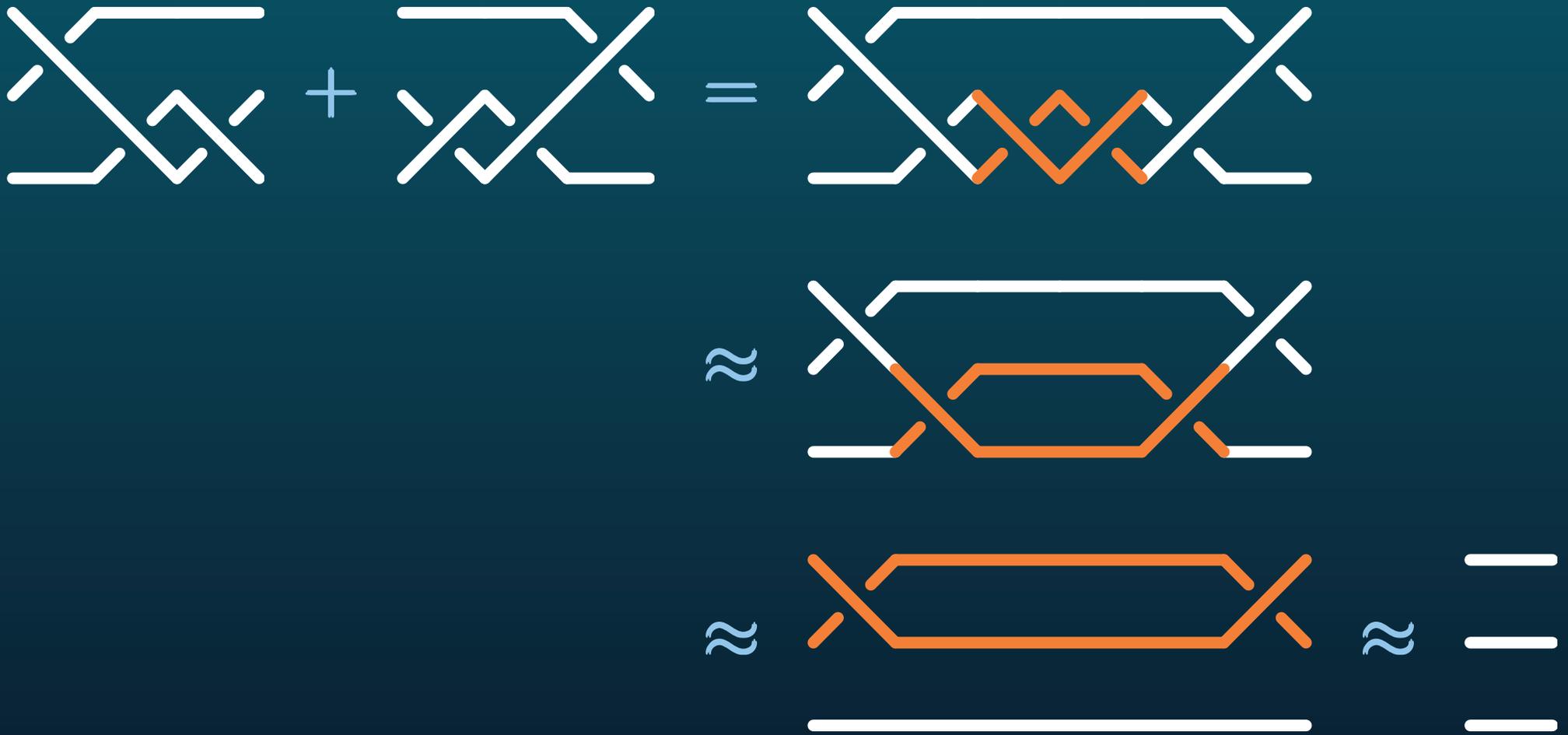
● Example:



● Example:



● Example:



↪ Pour chaque n , un groupe des **tresses à n brins** :
le groupe B_n («braid group»)

↔ Pour chaque n , un groupe des **tresses à n brins** :
le groupe B_n (« braid group »)

- considéré implicitement par C.F. Gauss et A. Hurwitz,
- étudié explicitement par E. Artin vers 1925.

↔ Pour chaque n , un groupe des **tresses à n brins** :
le groupe B_n (« braid group »)

- considéré implicitement par C.F. Gauss et A. Hurwitz,
- étudié explicitement par E. Artin vers 1925.

● Remarque: B_n est **non** commutatif pour $n \geq 3$:

↔ Pour chaque n , un groupe des **tresses à n brins** :
le groupe B_n (« braid group »)

- considéré implicitement par C.F. Gauss et A. Hurwitz,
- étudié explicitement par E. Artin vers 1925.

• Remarque: B_n est **non** commutatif pour $n \geq 3$:



↔ Pour chaque n , un groupe des **tresses à n brins** :
le groupe B_n (« braid group »)

- considéré implicitement par C.F. Gauss et A. Hurwitz,
- étudié explicitement par E. Artin vers 1925.

• Remarque: B_n est **non** commutatif pour $n \geq 3$:



↔ Pour chaque n , un groupe des **tresses à n brins** :
 le groupe B_n (« braid group »)

- considéré implicitement par C.F. Gauss et A. Hurwitz,
- étudié explicitement par E. Artin vers 1925.

• Remarque: B_n est **non** commutatif pour $n \geq 3$:



↔ addition, zéro, opposé ↔ multiplication, unité, inverse

↔ addition, zéro, opposé ↔ multiplication, unité, inverse

↔ codage par des mots plutôt que par des nombres :

↔ addition, zéro, opposé ↔ multiplication, unité, inverse

↔ codage par des mots plutôt que par des nombres :

$$\sigma_i := \begin{array}{c} \vdots \\ \text{---} \quad i+2 \\ \diagdown \quad i+1 \\ \diagup \quad i \\ \text{---} \quad i-1 \\ \vdots \\ \text{---} \quad 1 \end{array},$$

↔ addition, zéro, opposé ↔ multiplication, unité, inverse

↔ codage par des mots plutôt que par des nombres :

$$\sigma_i := \begin{array}{c} \vdots \\ \hline i+2 \\ \times \quad i+1 \\ \quad \quad i \\ \hline i-1 \\ \vdots \\ \hline 1 \end{array}, \quad \sigma_i^{-1} := \begin{array}{c} \vdots \\ \hline i+2 \\ \times \quad i+1 \\ \quad \quad i \\ \hline i-1 \\ \vdots \\ \hline 1 \end{array}.$$

↔ addition, zéro, opposé ↔ multiplication, unité, inverse

↔ codage par des mots plutôt que par des nombres :

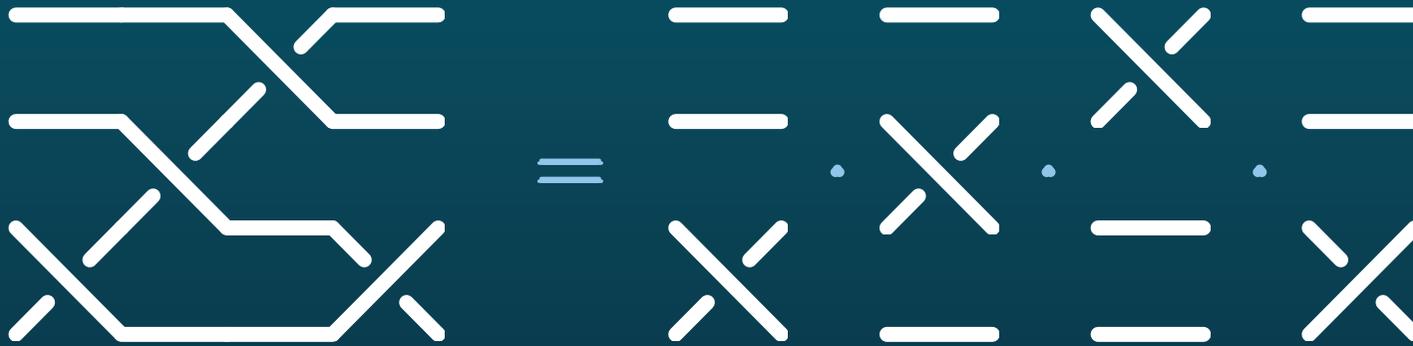
$$\sigma_i := \begin{array}{c} \vdots \\ \text{---} \quad i+2 \\ \times \quad i+1 \\ \quad \quad i \\ \text{---} \quad i-1 \\ \vdots \\ \text{---} \quad 1 \end{array}, \quad \sigma_i^{-1} := \begin{array}{c} \vdots \\ \text{---} \quad i+2 \\ \times \quad i+1 \\ \quad \quad i \\ \text{---} \quad i-1 \\ \vdots \\ \text{---} \quad 1 \end{array}.$$

(aussi **a** pour σ_1 , **A** pour σ_1^{-1} , **b** pour σ_2 , **B** pour σ_2^{-1} , etc.)

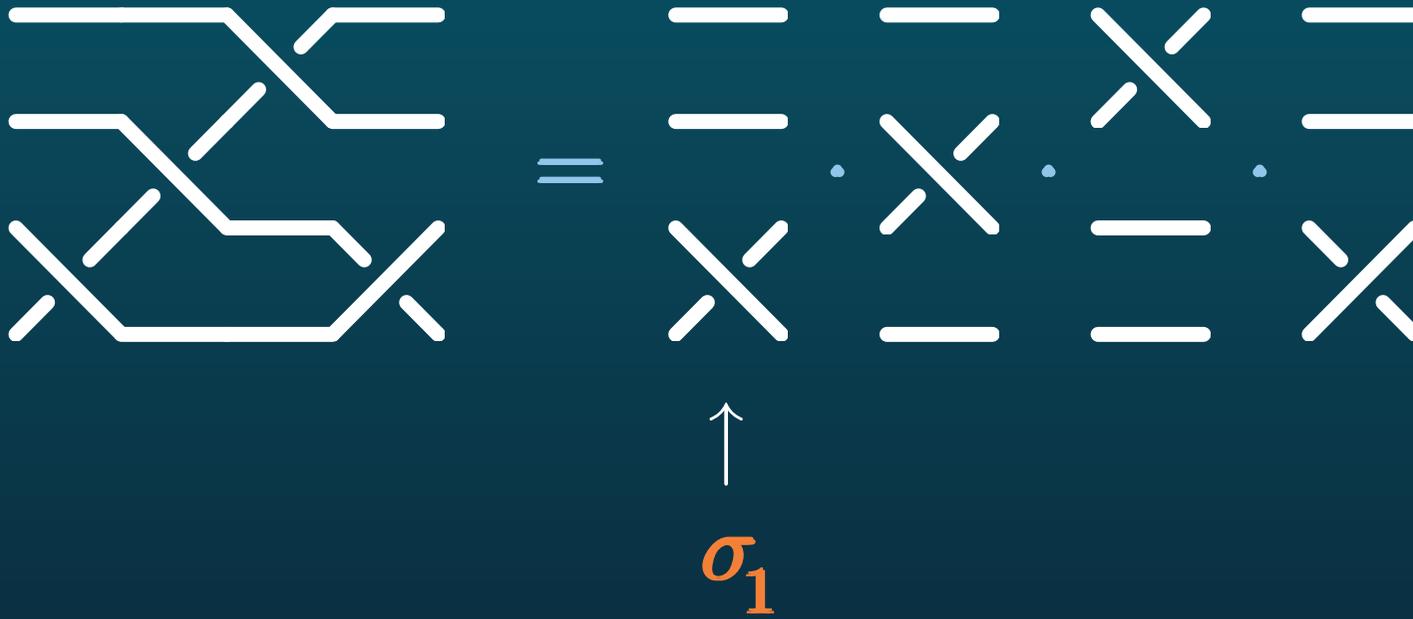
- Un exemple de codage:



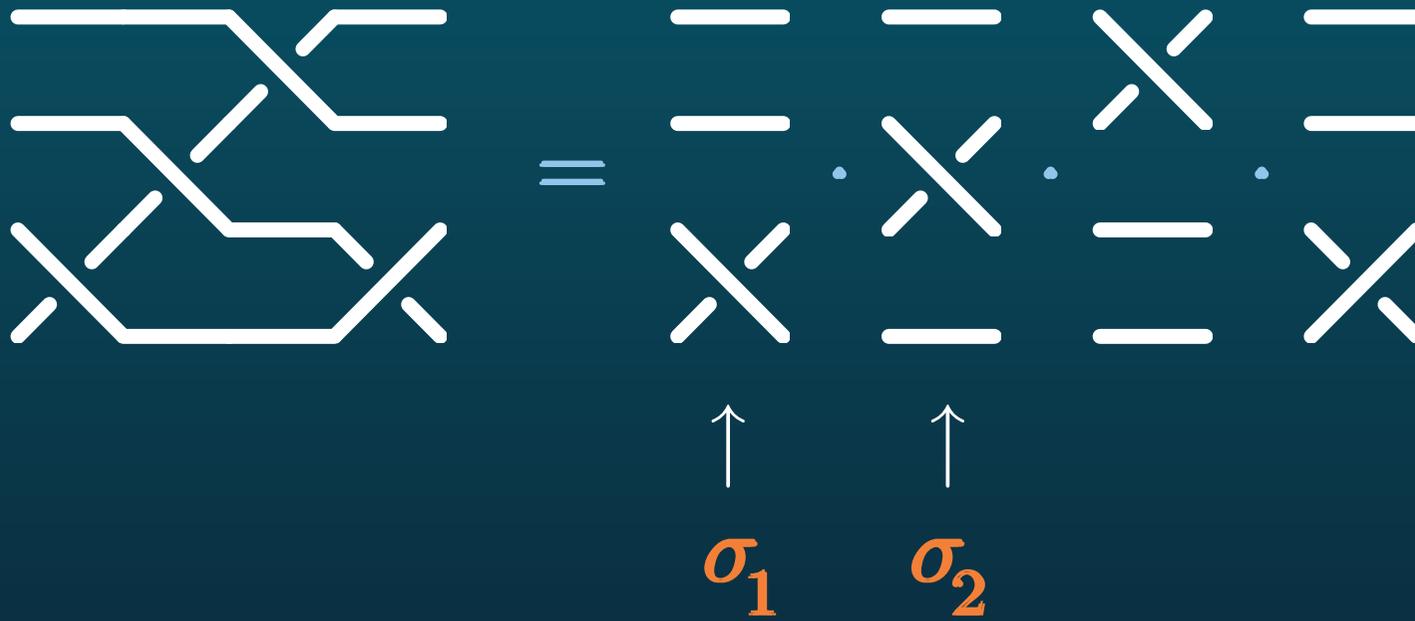
- Un exemple de codage:



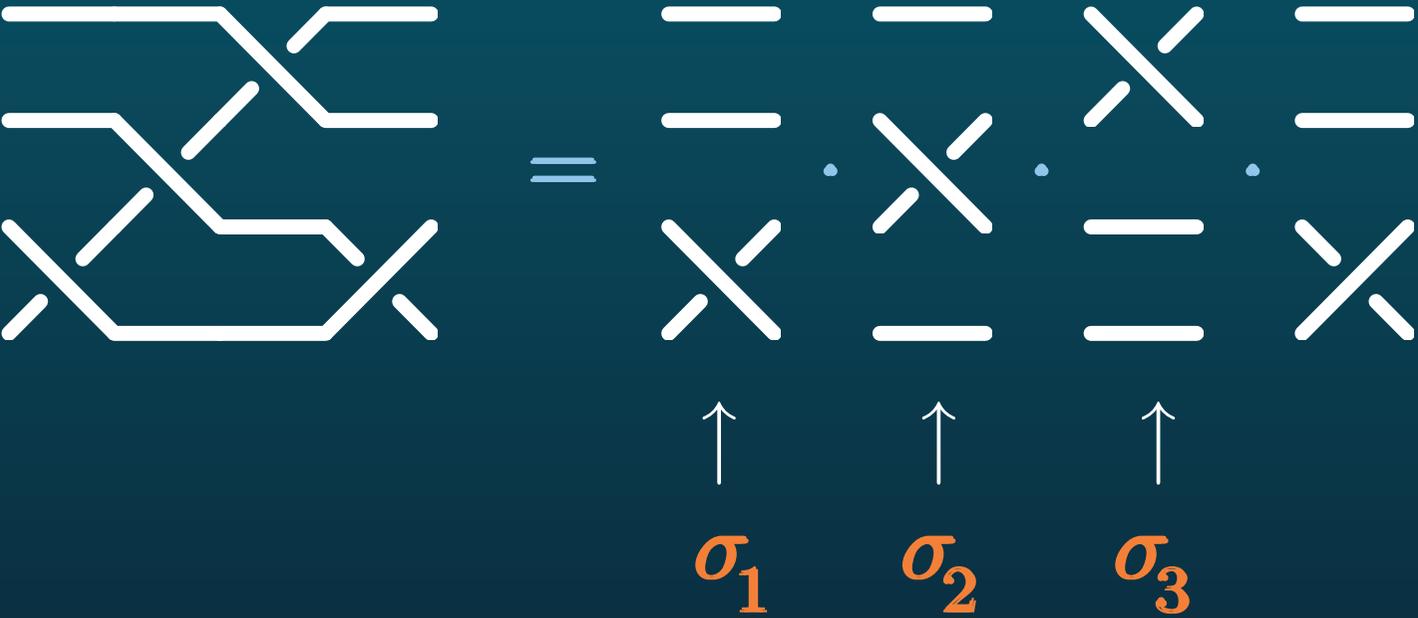
- Un exemple de codage:



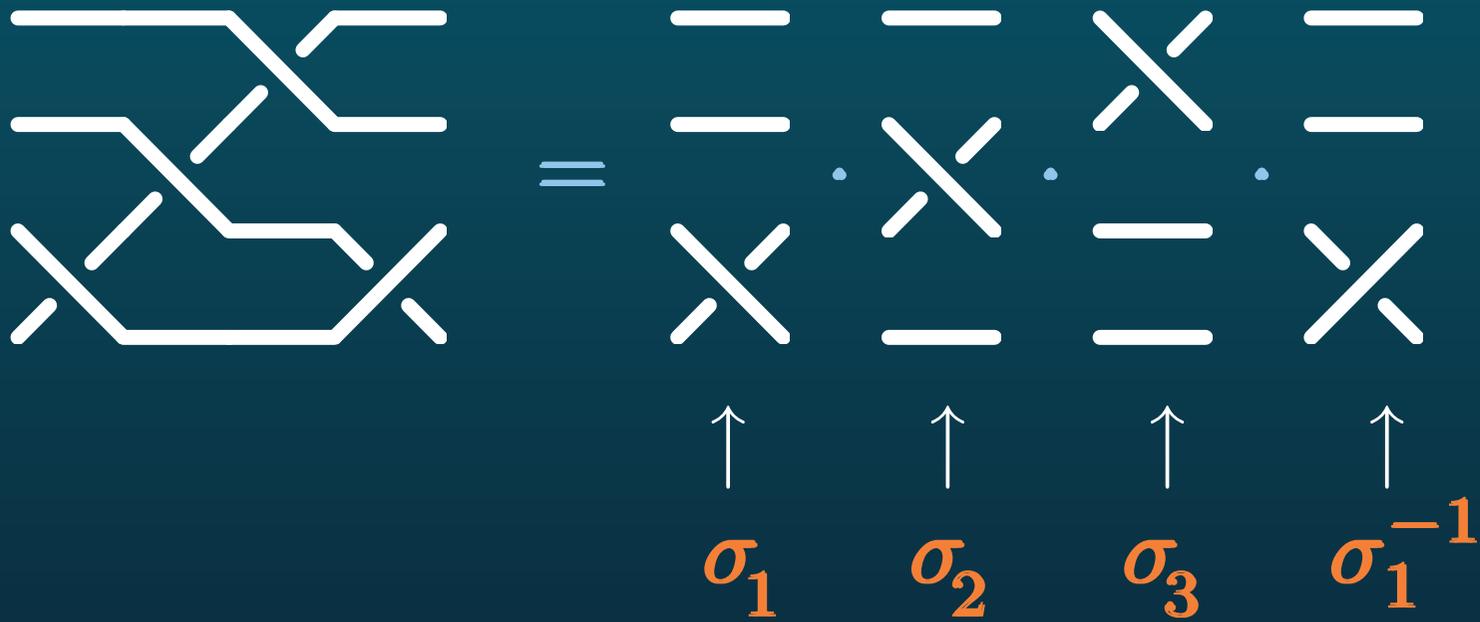
- Un exemple de codage:



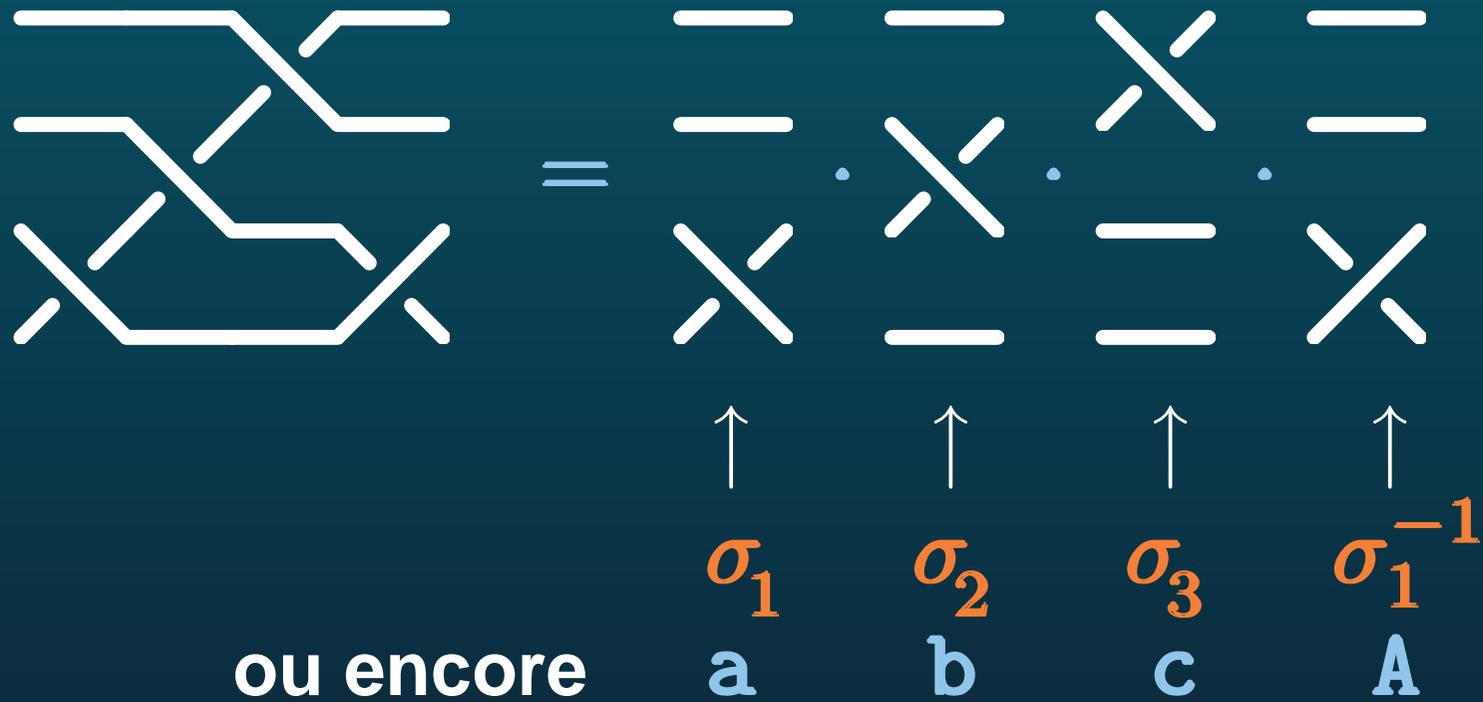
- Un exemple de codage:



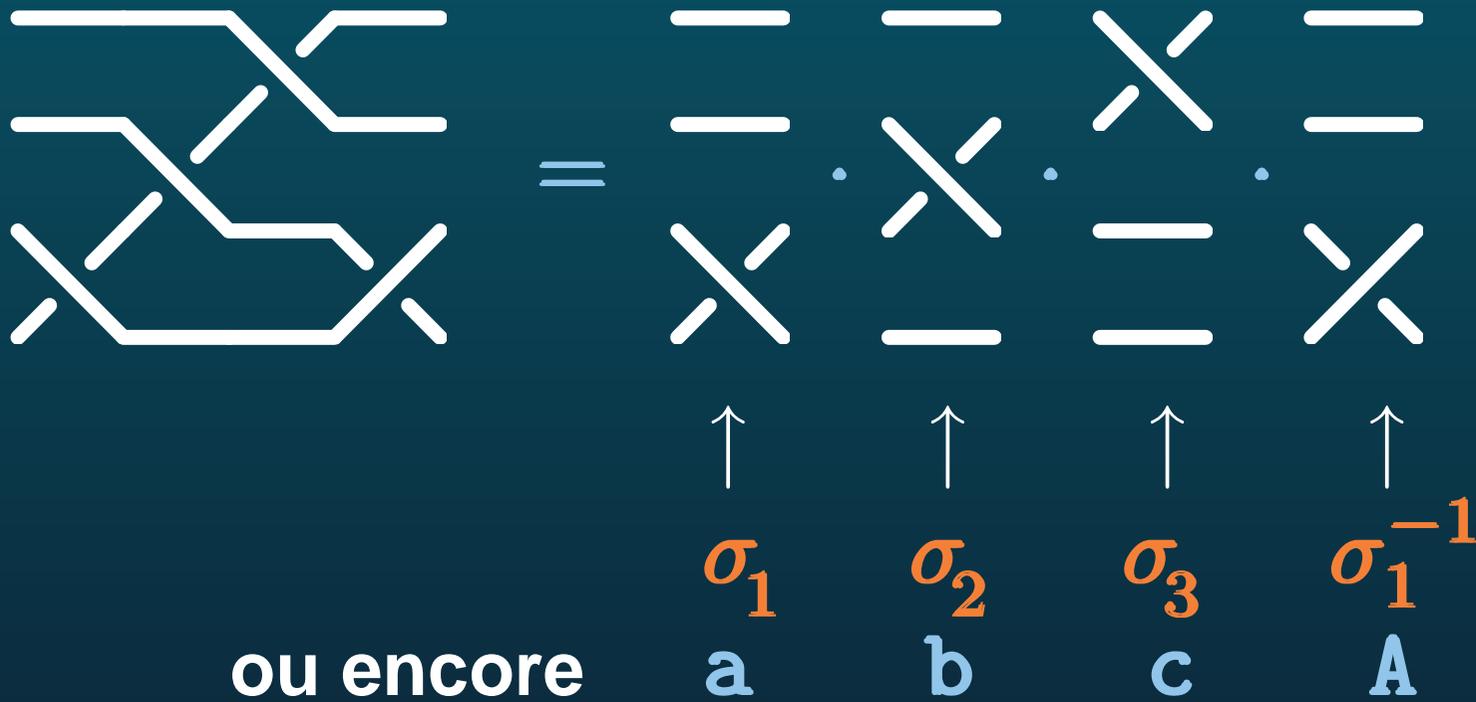
- Un exemple de codage:



- Un exemple de codage:



- Un exemple de codage:



↪ diagramme de tresse $\sigma_1 \sigma_2 \sigma_3 \sigma_1^{-1}$ (ou abcA)

↪ Définition précise de déformation : **isotopie**

↔ Définition précise de déformation : **isotopie**

- diagramme = projection sur \mathbb{R}^2 d'une figure de \mathbb{R}^3



↔ Définition précise de déformation : **isotopie**

- diagramme = projection sur \mathbb{R}^2 d'une figure de \mathbb{R}^3



↔ Définition précise de déformation : **isotopie**

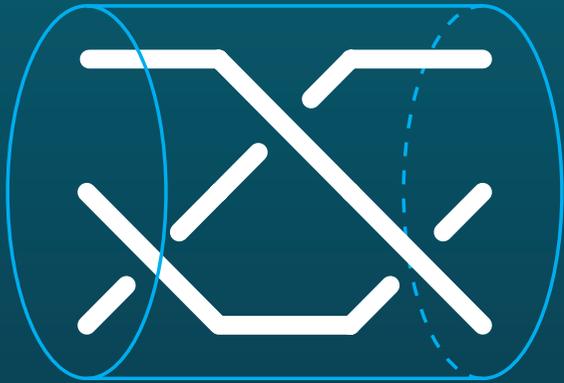
- diagramme = projection sur \mathbb{R}^2 d'une figure de \mathbb{R}^3



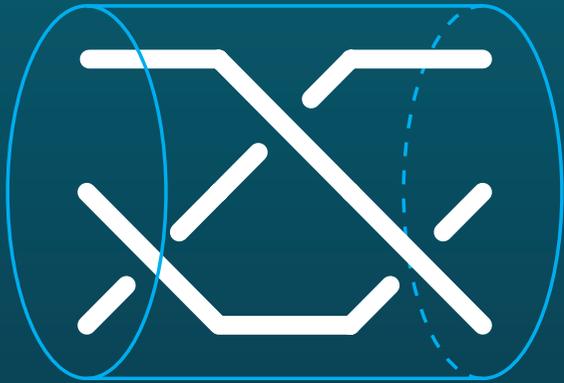
- isotopie = bouger les brins (sur la figure de \mathbb{R}^3)
en laissant les extrémités fixes



$\sigma_1 \sigma_2 \sigma_1 (= aba)$



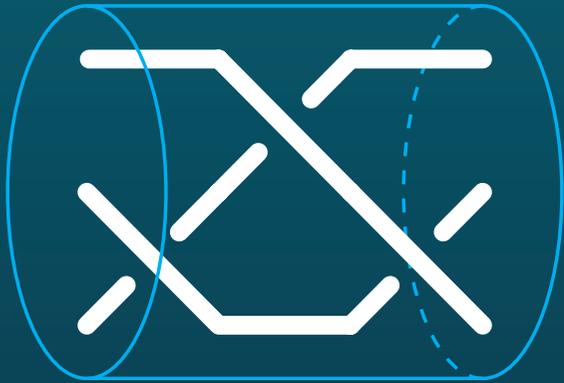
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



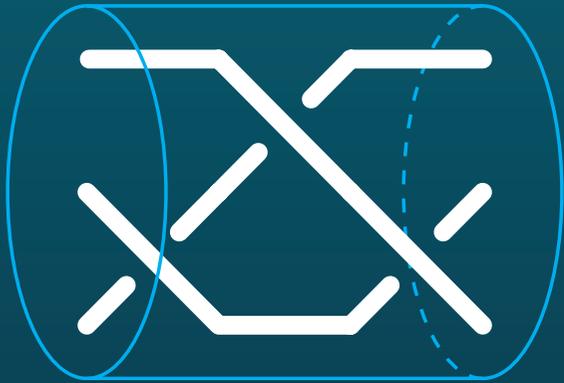
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



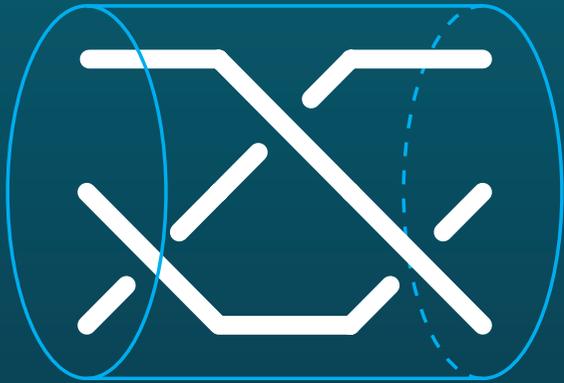
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



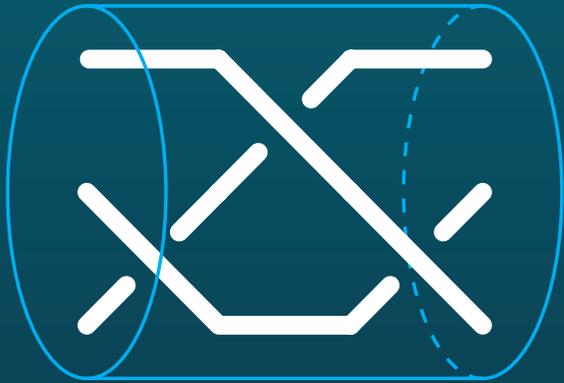
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



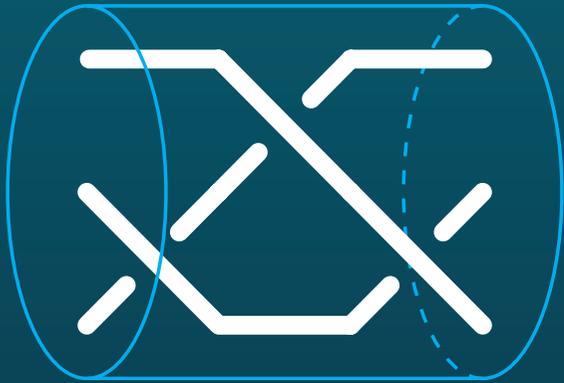
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



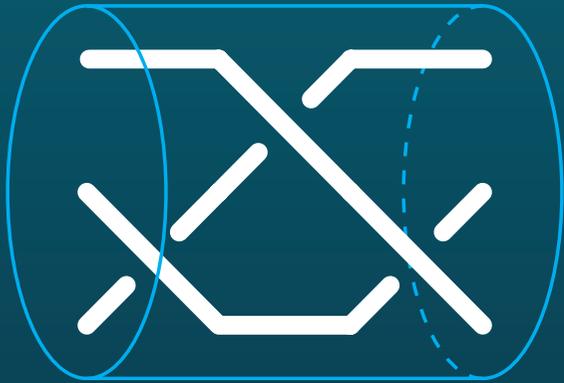
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



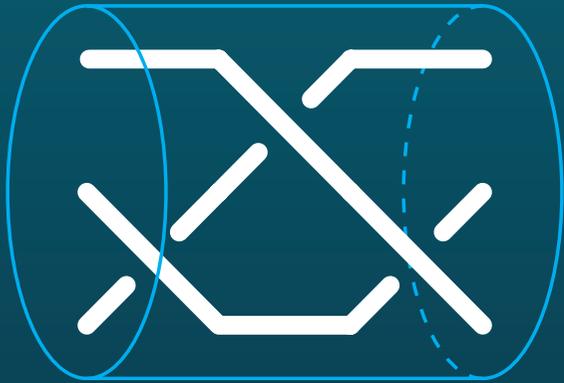
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



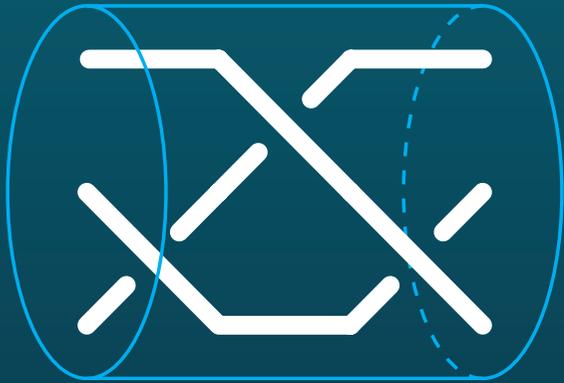
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



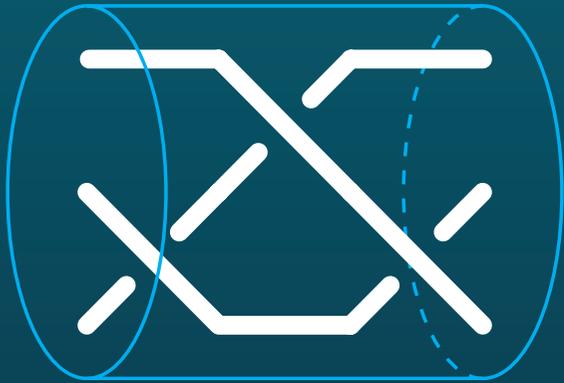
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



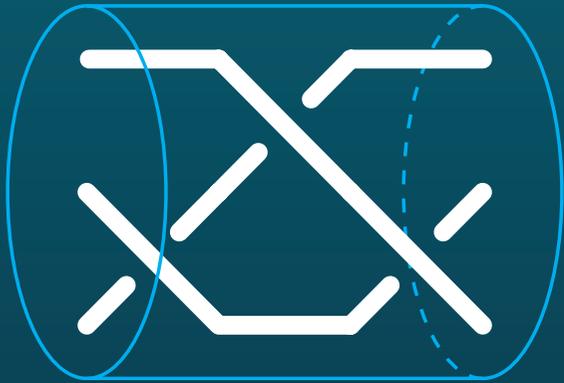
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



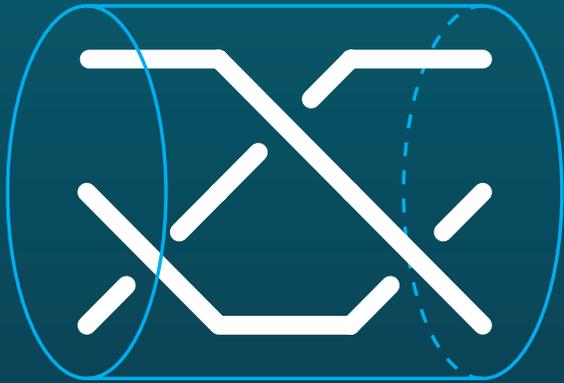
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



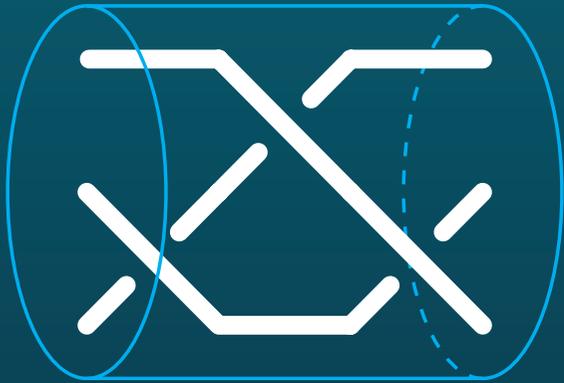
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



$\sigma_1 \sigma_2 \sigma_1 (= aba)$



isotope à



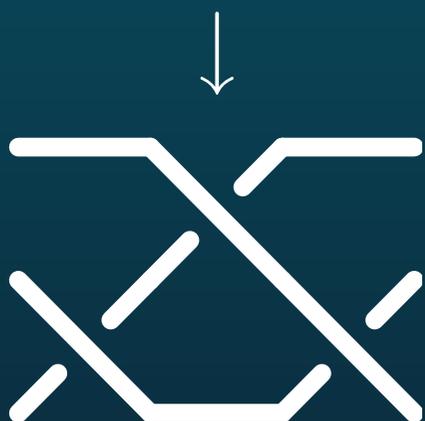
$\sigma_1 \sigma_2 \sigma_1 (= aba)$



$\sigma_2 \sigma_1 \sigma_2 (= bab)$



isotope à



$\sigma_1 \sigma_2 \sigma_1 (= aba)$



$\sigma_2 \sigma_1 \sigma_2 (= bab)$

↔ Relation $\sigma_1 \sigma_2 \sigma_1 = \sigma_2 \sigma_1 \sigma_2$ dans B_n : groupe **non libre**.

● **Théorème (Artin) : Les seules relations entre tresses sont**

- $\sigma_i \sigma_i^{-1} = \sigma_i^{-1} \sigma_i = 1,$

- $\sigma_i \sigma_j = \sigma_j \sigma_i$ pour $|i - j| \geq 2,$

- $\sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j$ pour $|i - j| = 1.$

● **Théorème (Artin) : Les seules relations entre tresses sont**

- $\sigma_i \sigma_i^{-1} = \sigma_i^{-1} \sigma_i = 1,$

- $\sigma_i \sigma_j = \sigma_j \sigma_i$ pour $|i - j| \geq 2,$

- $\sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j$ pour $|i - j| = 1.$



• **Théorème (Artin) : Les seules relations entre tresses sont**

- $\sigma_i \sigma_i^{-1} = \sigma_i^{-1} \sigma_i = 1,$

- $\sigma_i \sigma_j = \sigma_j \sigma_i$ pour $|i - j| \geq 2,$

- $\sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j$ pour $|i - j| = 1.$



• **Théorème (Artin) : Les seules relations entre tresses sont**

- $\sigma_i \sigma_i^{-1} = \sigma_i^{-1} \sigma_i = 1,$

- $\sigma_i \sigma_j = \sigma_j \sigma_i$ pour $|i - j| \geq 2,$

- $\sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j$ pour $|i - j| = 1.$



- Un exemple :



$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$

isotope à



- Un exemple :



$$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$$

isotope à



- Un exemple :



$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$

isotope à



- Un exemple :



$$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$$

isotope à



- Un exemple :



$$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$$

isotope à



- Un exemple :



$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$

isotope à



- Un exemple :



$$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$$

isotope à



- Un exemple :



$$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$$

isotope à



- Un exemple :



$$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$$

isotope à



- Un exemple :



$$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$$

isotope à



$$\sigma_2^{-1} \sigma_1 \sigma_2 (= Bab)$$

- Un exemple :



$$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$$

isotope à



$$\sigma_2^{-1} \sigma_1 \sigma_2 (= Bab)$$

↔ Il **doit** exister une dérivation à partir des relations d'Artin



a b A

- Un exemple :



$$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$$

isotope à



$$\sigma_2^{-1} \sigma_1 \sigma_2 (= Bab)$$

↔ Il **doit** exister une dérivation à partir des relations d'Artin



a b A

≈



B b a b A

- Un exemple :



$$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$$

isotope à



$$\sigma_2^{-1} \sigma_1 \sigma_2 (= Bab)$$

↔ Il **doit** exister une dérivation à partir des relations d'Artin



a b A

≈



B b a b A

≈



B a b a A

- Un exemple :



$\sigma_1 \sigma_2 \sigma_1^{-1} (= abA)$

isotope à



$\sigma_2^{-1} \sigma_1 \sigma_2 (= Bab)$

↔ Il **doit** exister une dérivation à partir des relations d'Artin



a b A

≈



B b a b A

≈



B a b a A

≈



B a b

... ne donne pas d'**algorithme** pour trouver la dérivation

... ne donne pas d'**algorithme** pour trouver la dérivation

↔ problème de base pour **utiliser** le groupe B_n :

Reconnaître quand deux mots codent des tresses isotopes
(= représentent le même élément de B_n)

... ne donne pas d'**algorithme** pour trouver la dérivation

↔ problème de base pour **utiliser** le groupe B_n :

Reconnaître quand deux mots codent des tresses isotopes

(= représentent le même élément de B_n)

cf. représentation des entiers avec des chiffres

... ne donne pas d'**algorithme** pour trouver la dérivation

↔ problème de base pour **utiliser** le groupe B_n :

Reconnaître quand deux mots codent des tresses isotopes

(= représentent le même élément de B_n)

cf. représentation des entiers avec des chiffres

- Cas de **2** brins : compter les demi-tours...

... ne donne pas d'**algorithme** pour trouver la dérivation

↔ problème de base pour **utiliser** le groupe B_n :

Reconnaître quand deux mots codent des tresses isotopes

(= représentent le même élément de B_n)

cf. représentation des entiers avec des chiffres

- Cas de **2** brins : compter les demi-tours...
- Cas de **3** brins et plus : - **E. Artin (1925)**,

... ne donne pas d'**algorithme** pour trouver la dérivation

↔ problème de base pour **utiliser** le groupe B_n :

Reconnaître quand deux mots codent des tresses isotopes

(= représentent le même élément de B_n)

cf. représentation des entiers avec des chiffres

- Cas de **2** brins : compter les demi-tours...
- Cas de **3** brins et plus : - E. Artin (1925),
- F. Garside (1967),

... ne donne pas d'**algorithme** pour trouver la dérivation

↔ problème de base pour **utiliser** le groupe B_n :

Reconnaître quand deux mots codent des tresses isotopes

(= représentent le même élément de B_n)

cf. représentation des entiers avec des chiffres

- Cas de **2** brins : compter les demi-tours...
- Cas de **3** brins et plus : - E. Artin (1925),
 - F. Garside (1967),
 - P. Deligne (1972),

... ne donne pas d'**algorithme** pour trouver la dérivation

↔ problème de base pour **utiliser** le groupe B_n :

Reconnaître quand deux mots codent des tresses isotopes

(= représentent le même élément de B_n)

cf. représentation des entiers avec des chiffres

- Cas de **2** brins : compter les demi-tours...
- Cas de **3** brins et plus : - E. Artin (1925),
 - F. Garside (1967),
 - P. Deligne (1972),
 - W. Thurston (1988),

... ne donne pas d'**algorithme** pour trouver la dérivation

↔ problème de base pour **utiliser** le groupe B_n :

Reconnaître quand deux mots codent des tresses isotopes

(= représentent le même élément de B_n)

cf. représentation des entiers avec des chiffres

- Cas de **2** brins : compter les demi-tours...
- Cas de **3** brins et plus : - E. Artin (1925),
 - F. Garside (1967),
 - P. Deligne (1972),
 - W. Thurston (1988),
 - I. Dynnikov (1999) ...

- Remarque préliminaire :

$$t' \approx t \text{ équivaut à } t^{-1} \cdot t' \approx 1.$$

- Remarque préliminaire :

$$t' \approx t \text{ équivaut à } t^{-1} \cdot t' \approx 1.$$

↔ Si on sait reconnaître $t \approx 1$, on sait reconnaître $t' \approx t$.

- Remarque préliminaire :

$$t' \approx t \text{ équivaut à } t^{-1} \cdot t' \approx 1.$$

↪ Si on sait reconnaître $t \approx 1$, on sait reconnaître $t' \approx t$.

↪ Le vrai problème («**problème de mot pour B_n** ») :
reconnaitre si un mot de tresse représente **1**
⇔ reconnaître si une tresse est vraiment tressée.

- Remarque préliminaire :

$$t' \approx t \text{ équivaut à } t^{-1} \cdot t' \approx 1.$$

↗ Si on sait reconnaître $t \approx 1$, on sait reconnaître $t' \approx t$.

↗ Le vrai problème («**problème de mot pour B_n** ») :
reconnaitre si un mot de tresse représente **1**

⇔ reconnaître si une tresse est vraiment tressée.

↗ algorithme de **démêlage** des tresses ?

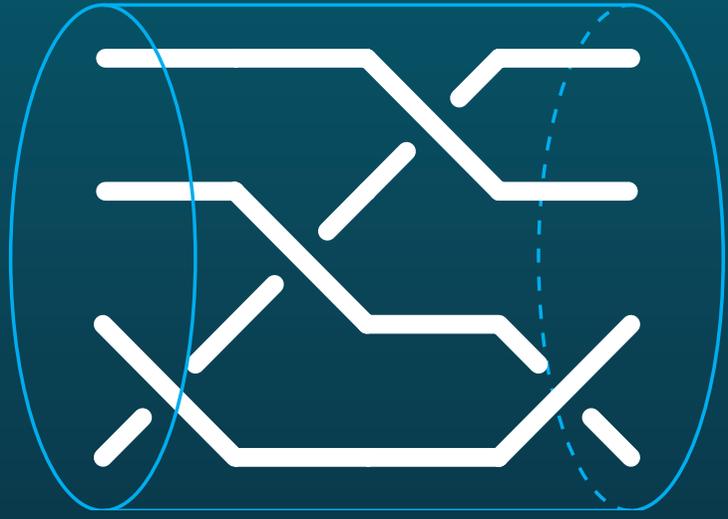
- Un **théorème** : Un mot de tresse qui contient au moins un σ_1 et pas de σ_1^{-1} est non trivial (= ne représente pas 1).

- Un **théorème** : Un mot de tresse qui contient au moins un σ_1 et pas de σ_1^{-1} est non trivial (= ne représente pas 1).

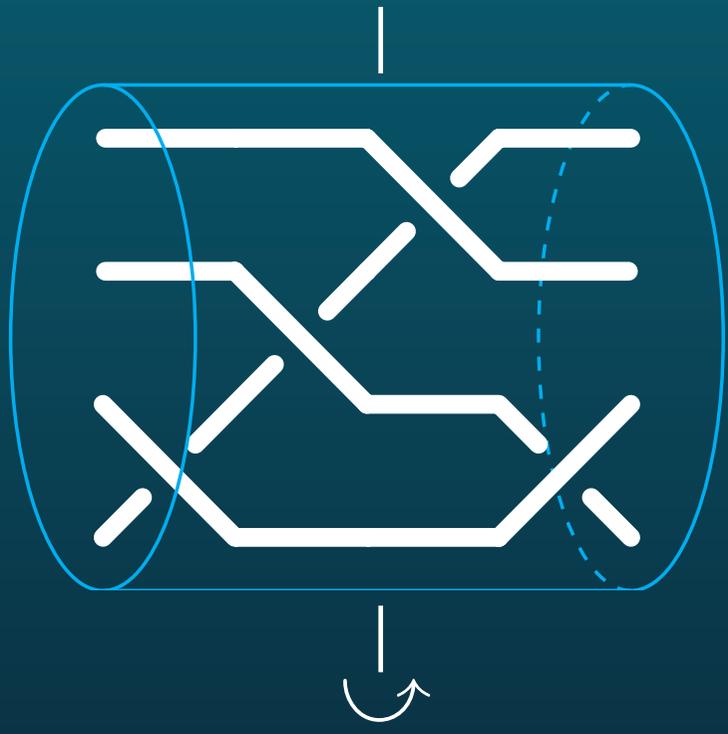


↔ Démonstration?

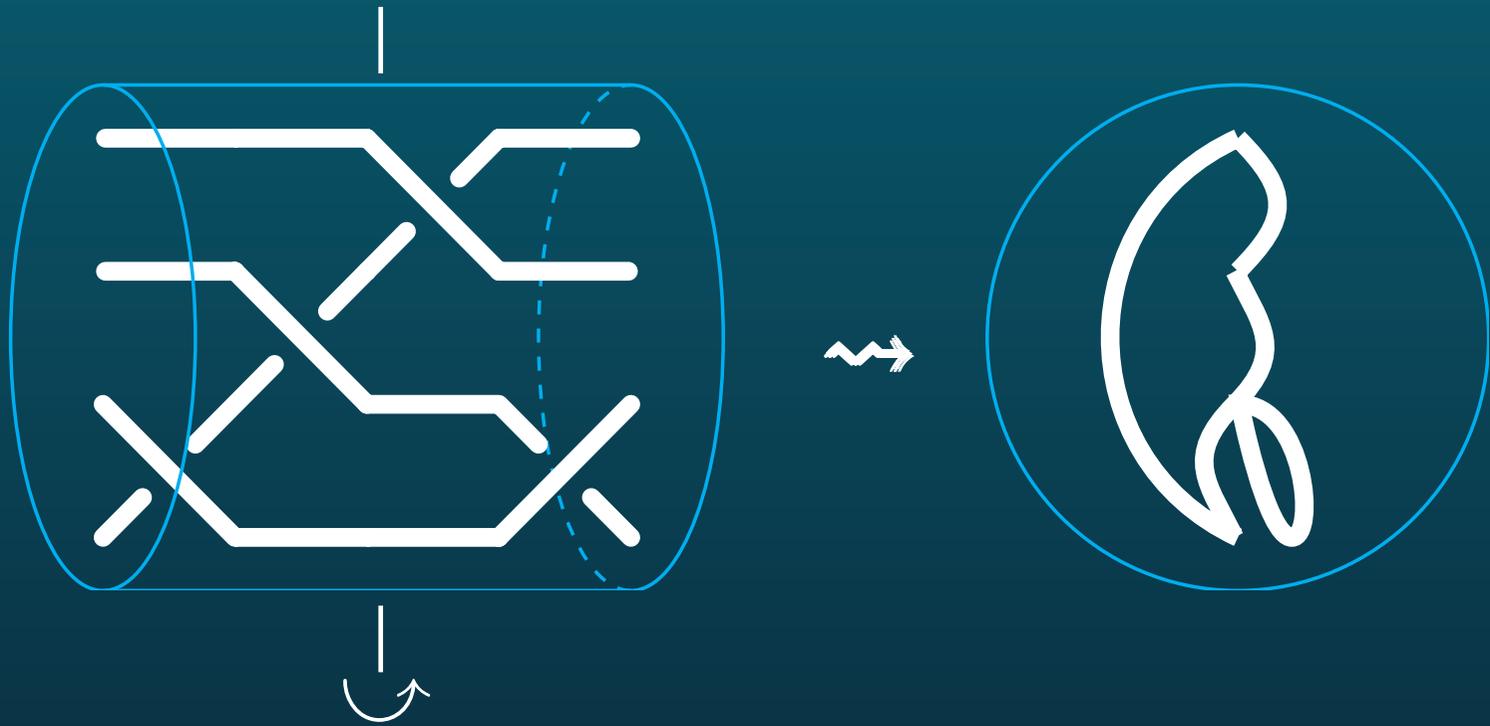
↔ Démonstration (Dyannikov)



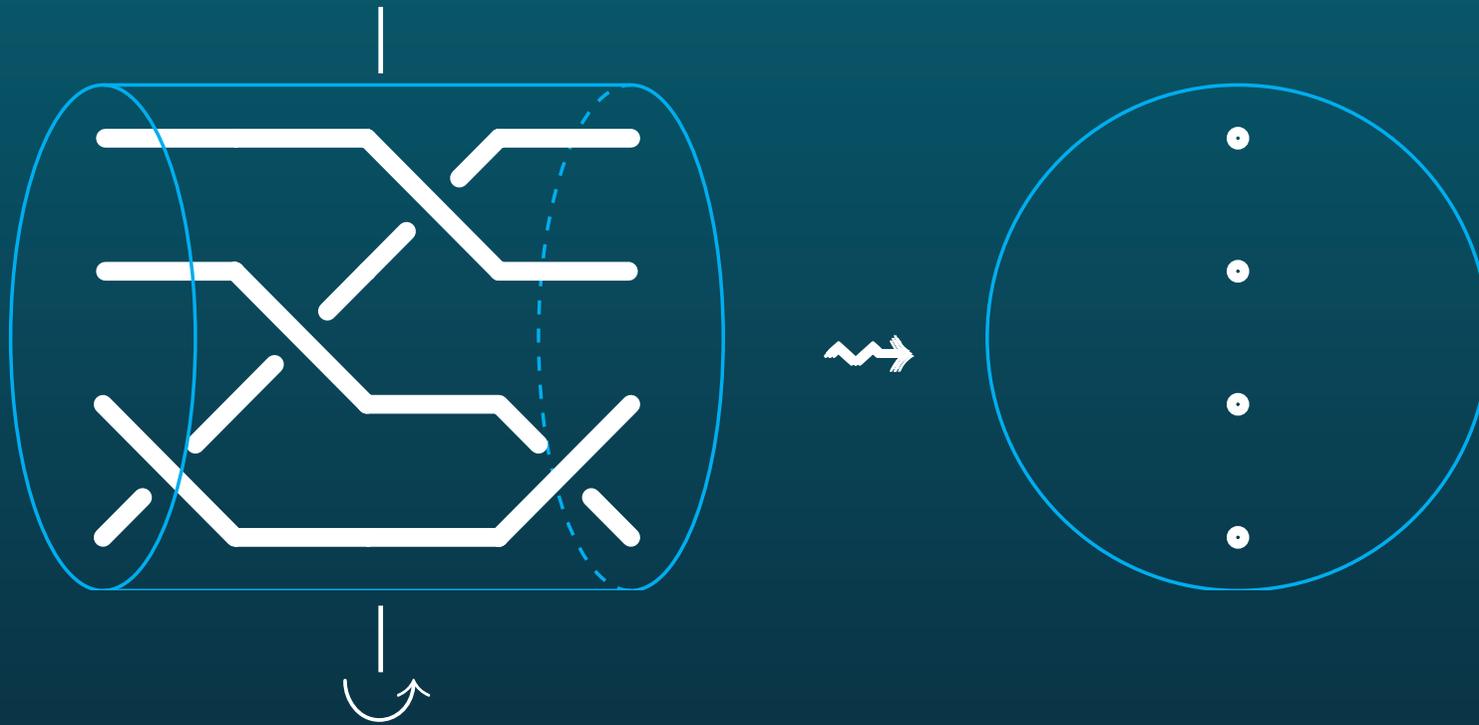
↔ Démonstration (Dyannikov)



↔ Démonstration (Dyannikov)

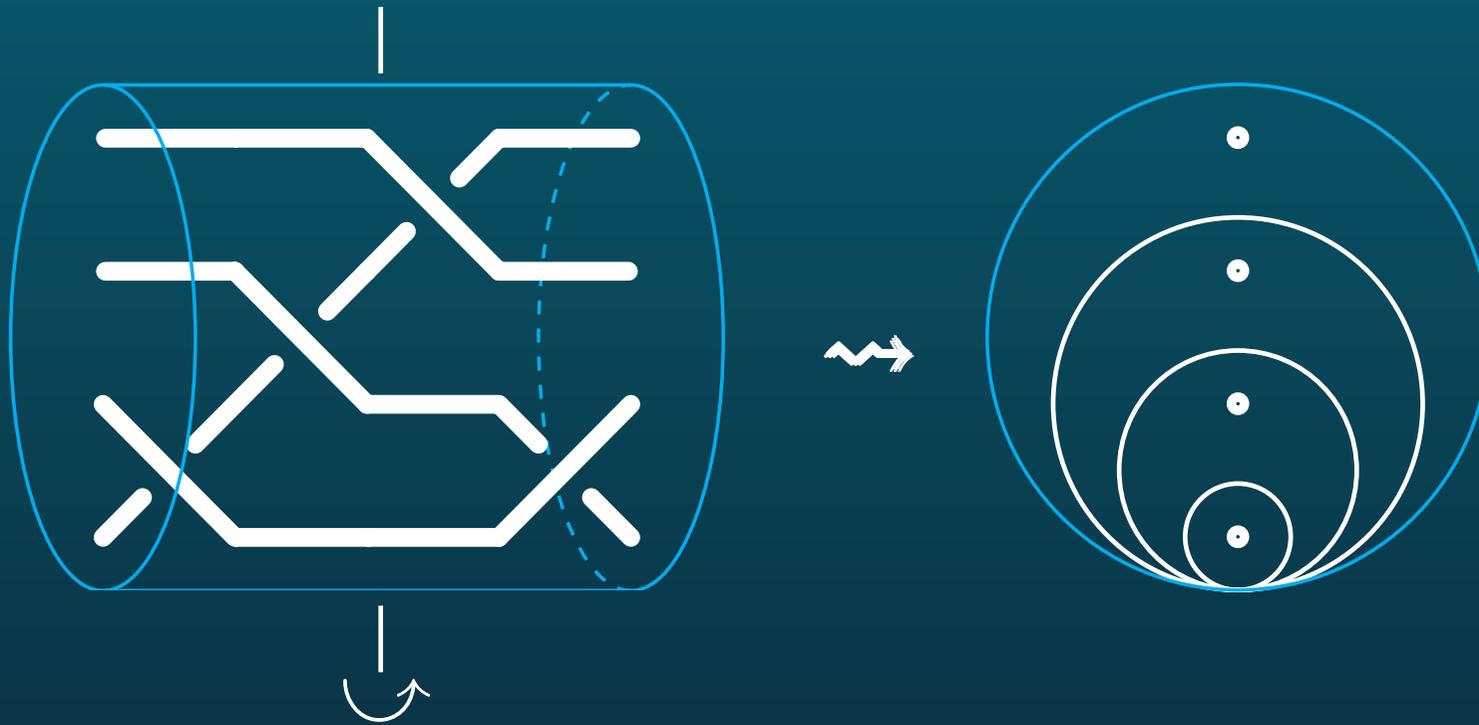


↔ Démonstration (Dyannikov)



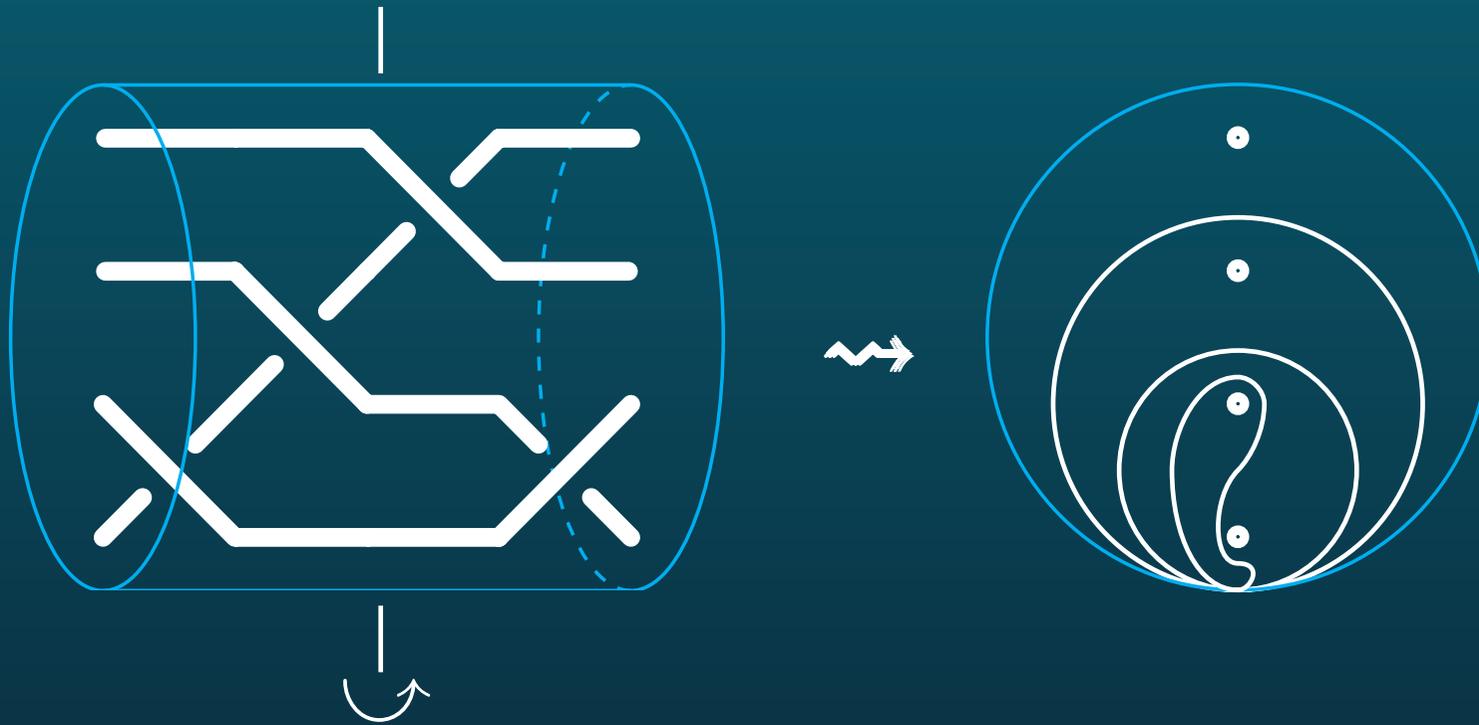
- tresse = **déformation** d'un disque troué ;

↔ Démonstration (Dyannikov)



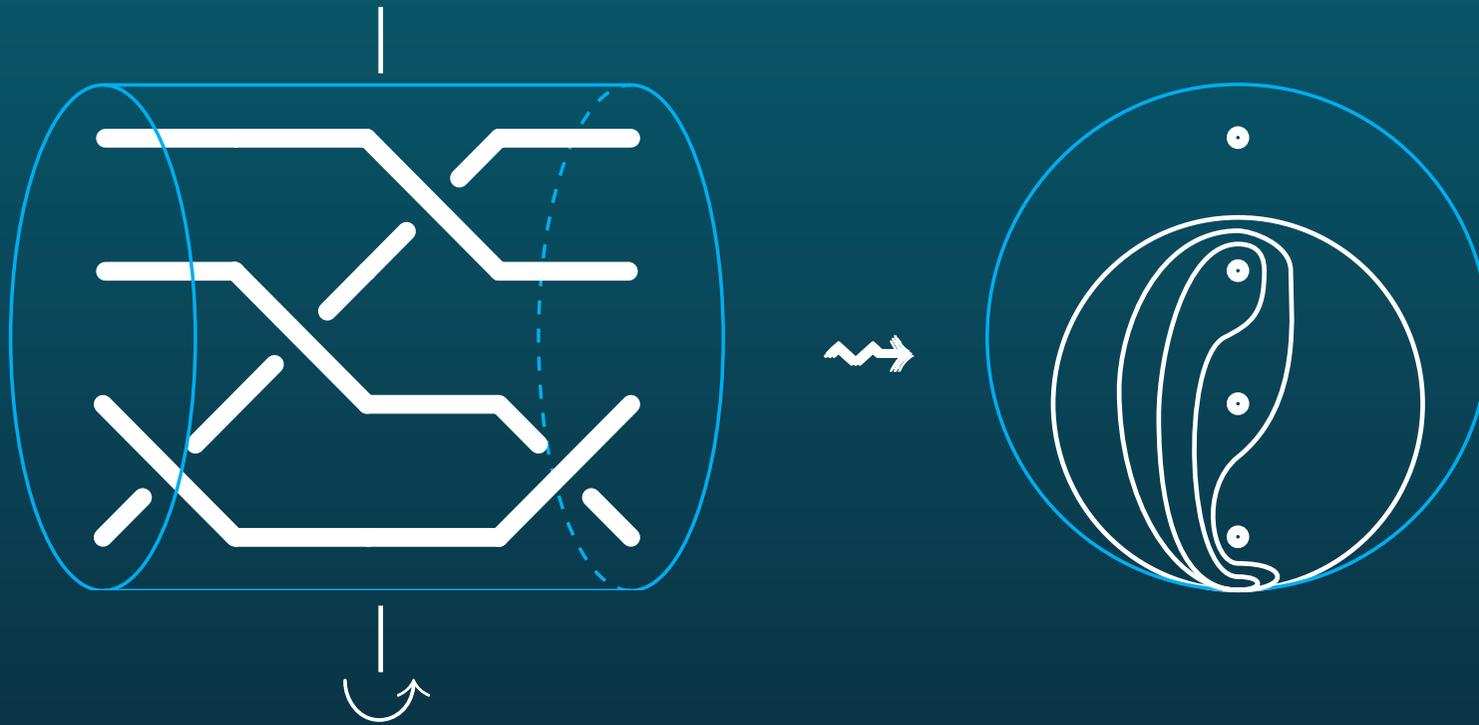
- tresse = **déformation** d'un disque troué ;
- faire agir sur une **lamination** du disque troué ;

↔ Démonstration (Dyannikov)



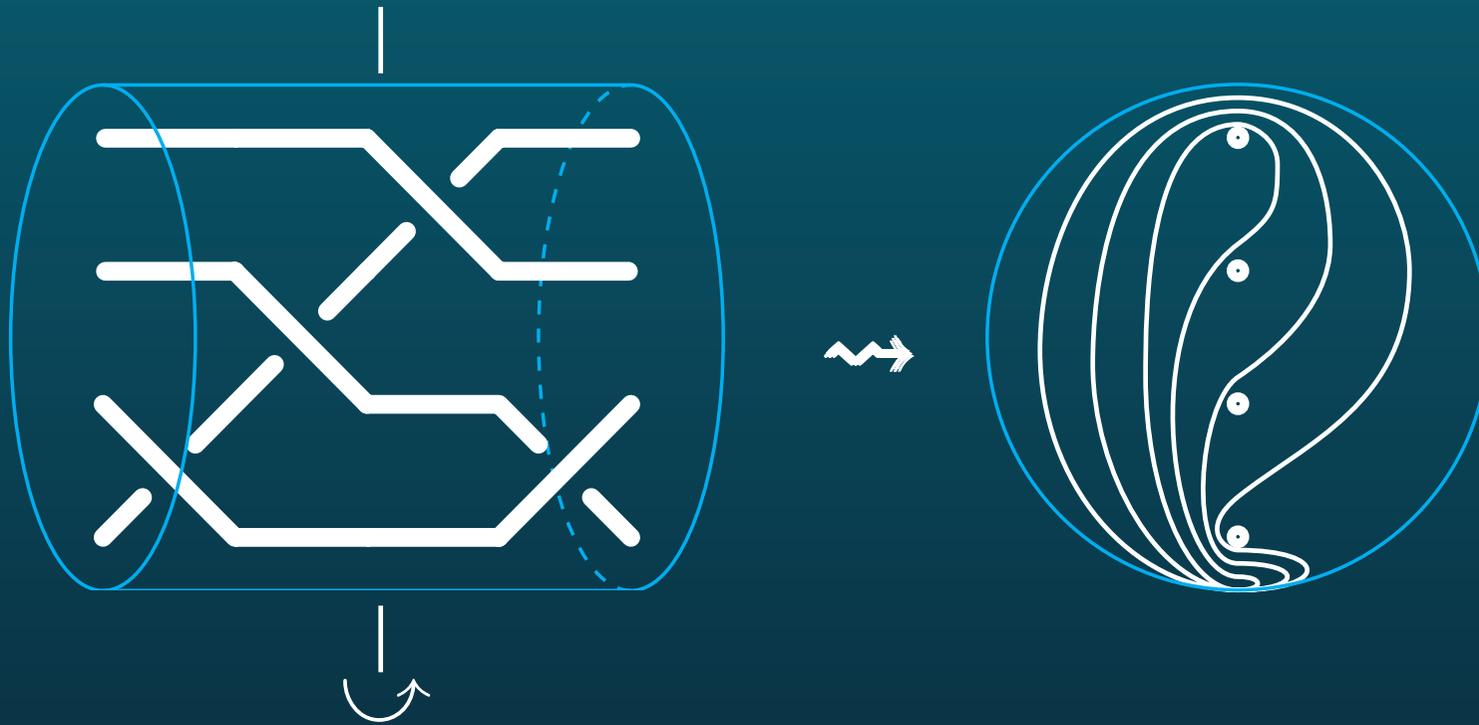
- tresse = **déformation** d'un disque troué ;
- faire agir sur une **lamination** du disque troué ;

↔ Démonstration (Dyannikov)



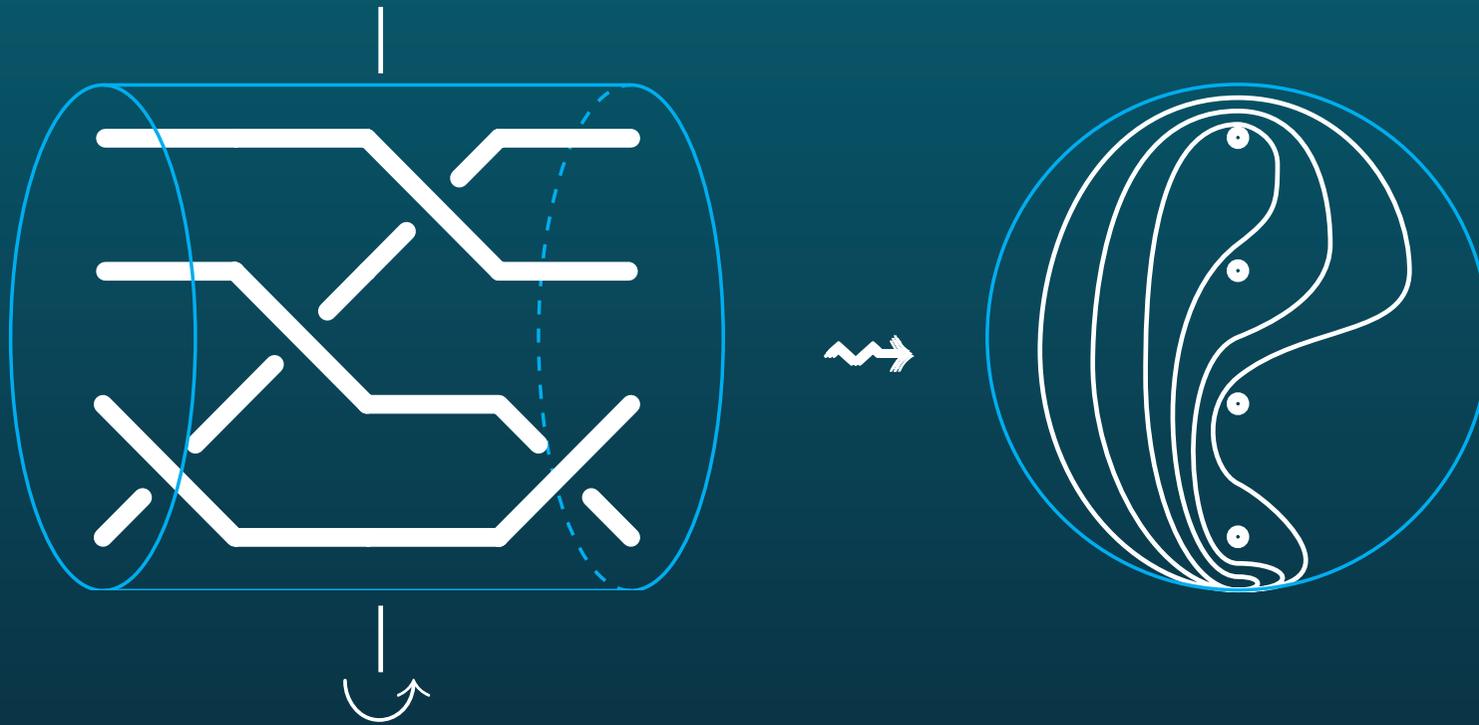
- tresse = **déformation** d'un disque troué ;
- faire agir sur une **lamination** du disque troué ;

↔ Démonstration (Dyannikov)



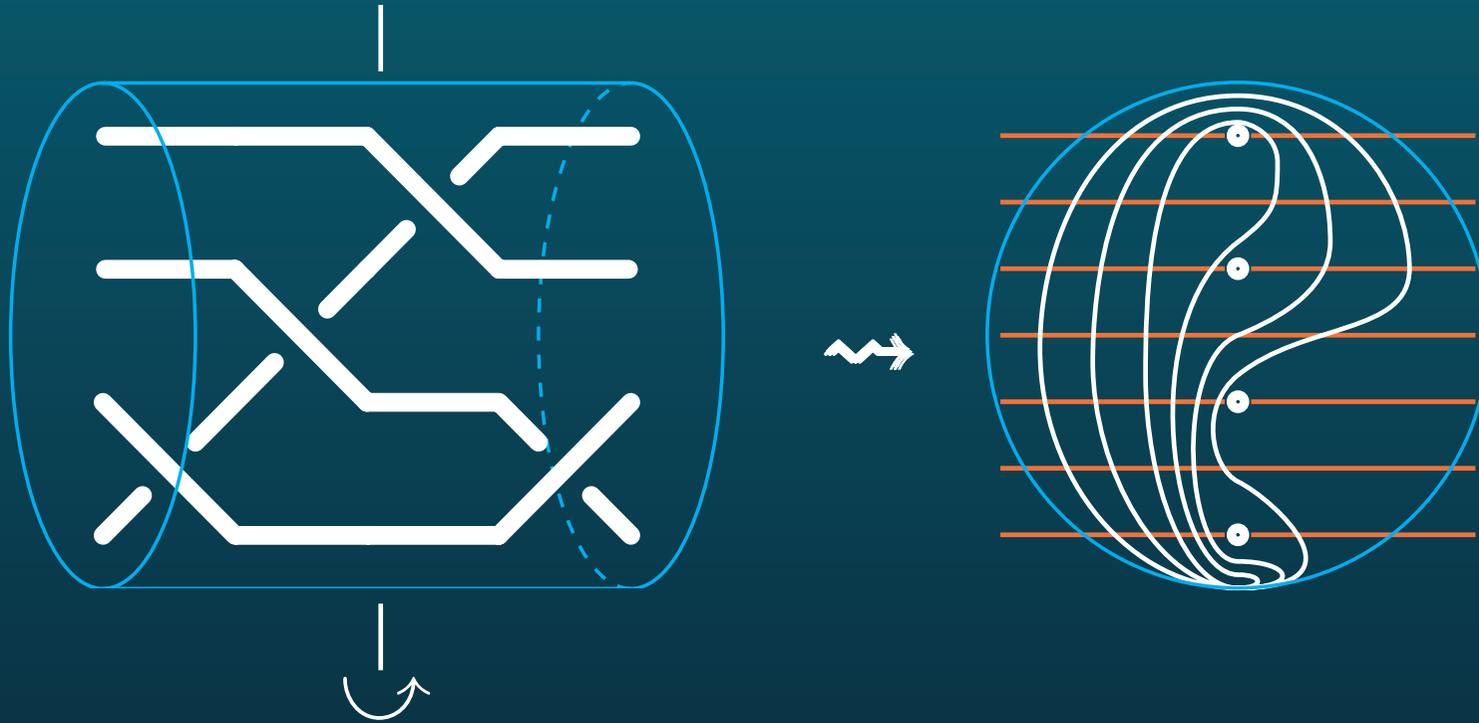
- tresse = **déformation** d'un disque troué ;
- faire agir sur une **lamination** du disque troué ;

↔ Démonstration (Dyannikov)



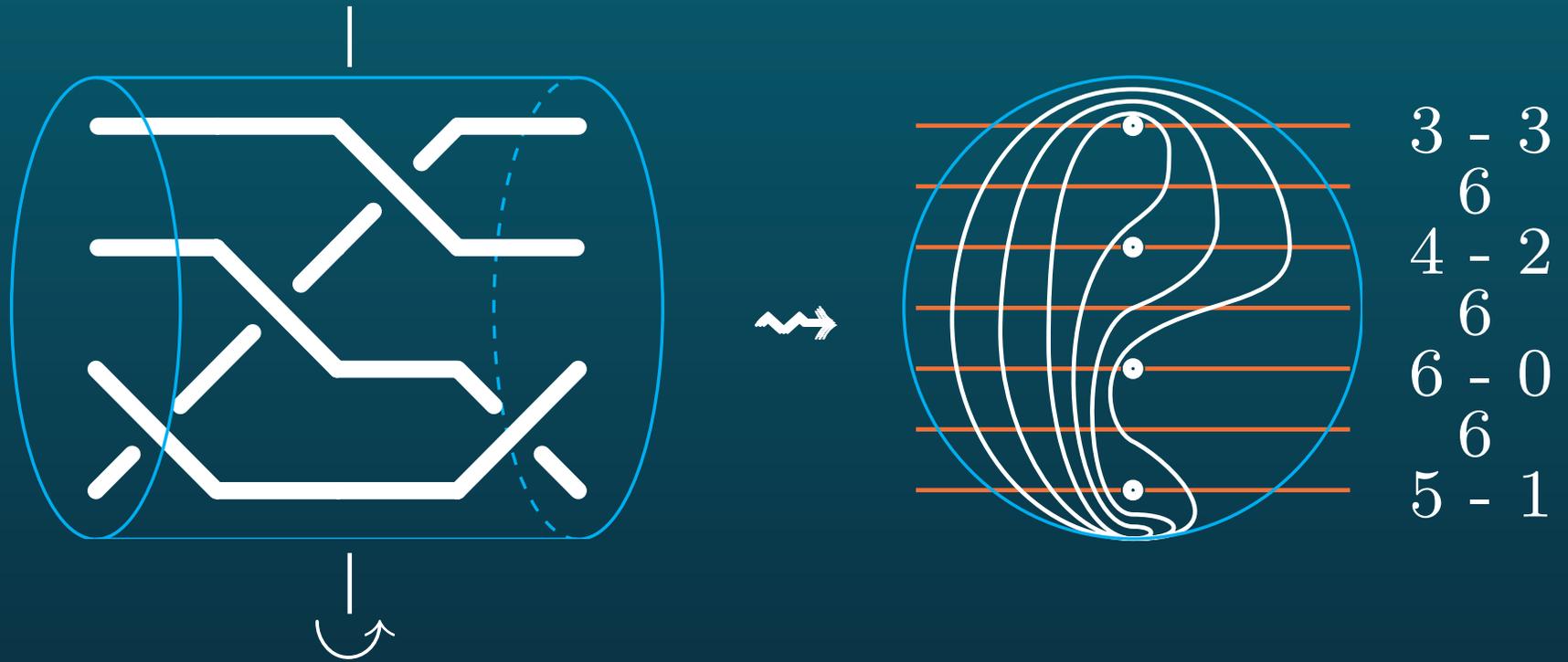
- tresse = **déformation** d'un disque troué ;
- faire agir sur une **lamination** du disque troué ;

↔ Démonstration (Dyannikov)



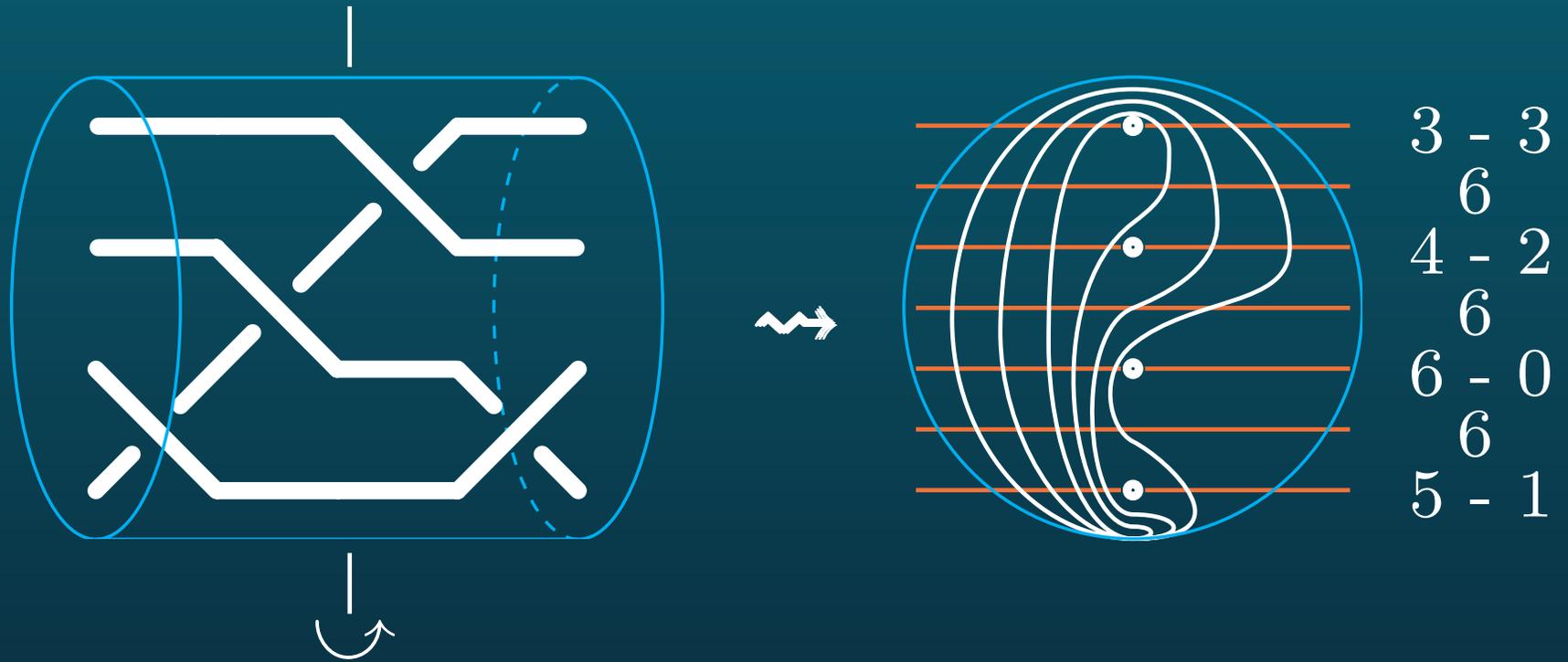
- tresse = **déformation** d'un disque troué ;
 - faire agir sur une **lamination** du disque troué ;
 - **compter** les intersections avec une grille ;

↔ Démonstration (Dyannikov)



- tresse = **déformation** d'un disque troué ;
- faire agir sur une **lamination** du disque troué ;
- **compter** les intersections avec une grille ;

↔ Démonstration (Dyannikov)



- tresse = **déformation** d'un disque troué ;
- faire agir sur une **lamination** du disque troué ;
- **compter** les intersections avec une grille ;

$$a'_1 = a_1 + ((a_2 - a_1)^+ + b_1)^+ \text{ pour } x' = x\sigma_1$$

$$a'_1 = a_1 \text{ pour } x' = x\sigma_i^{\pm 1} \text{ avec } i \geq 2.$$

- Un **théorème** : Un mot de tresse qui contient au moins un σ_1 et pas de σ_1^{-1} est non trivial (= ne représente pas 1).



- Un **théorème** : Un mot de tresse qui contient au moins un σ_1 et pas de σ_1^{-1} est non trivial (= ne représente pas 1).



- *idem* pour mot contenant σ_1^{-1} et pas σ_1 ;

- Un **théorème** : Un mot de tresse qui contient au moins un σ_1 et pas de σ_1^{-1} est non trivial (= ne représente pas 1).



- *idem* pour mot contenant σ_1^{-1} et pas σ_1 ;
 - *idem* avec σ_2 et σ_2^{-1} pour mot sans σ_1 ni σ_1^{-1} ;

- Un **théorème** : Un mot de tresse qui contient au moins un σ_1 et pas de σ_1^{-1} est non trivial (= ne représente pas 1).



- *idem* pour mot contenant σ_1^{-1} et pas σ_1 ;
 - *idem* avec σ_2 et σ_2^{-1} pour mot sans σ_1 ni σ_1^{-1} ;

↔ problème : mots contenant σ_1 **et** σ_1^{-1} .

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↔ une **poignée**

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

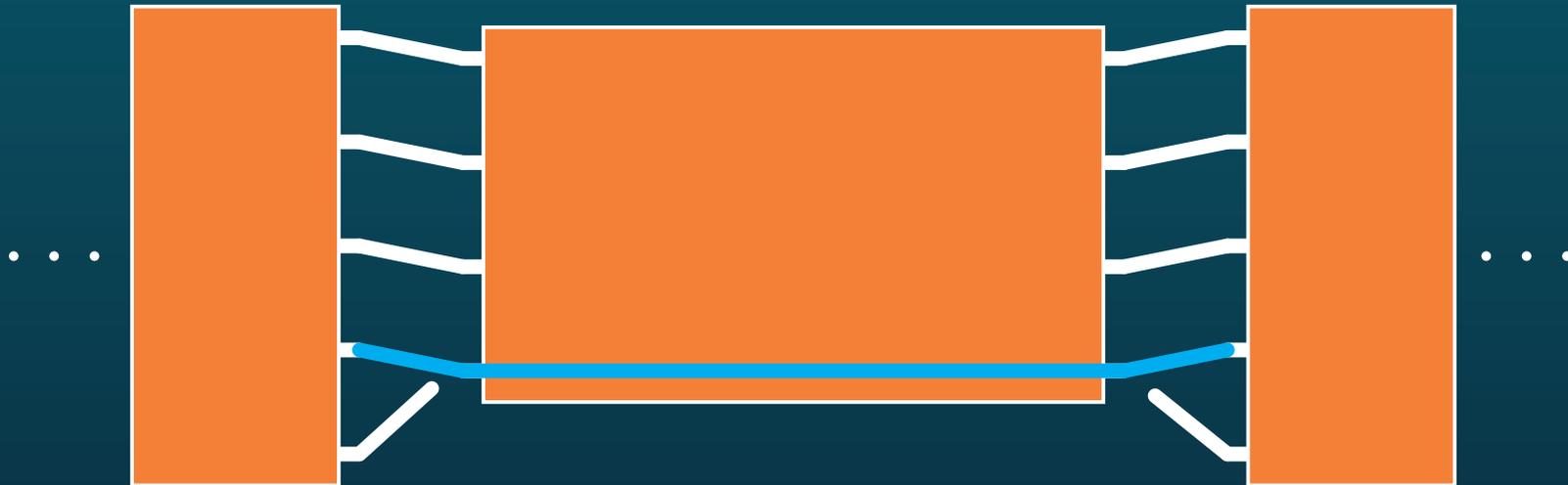
- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

- Le cas $\langle\langle \sigma_1 \text{ et } \sigma_1^{-1} \rangle\rangle$:



↪ une **poignée**

↪ comment s'en débarrasser ?

↪ **itérer** jusqu'à ce que plus de poignée (?)

- Un exemple :



abbAAb

- Un exemple :



abbAAb

- Un exemple :



abbAAb



BaabAb

● Un exemple :



abbAAb



BaabAb



BaBabb

{ deux a ($= \sigma_1$)
zéro A ($= \sigma_1^{-1}$)

● Un exemple :



abbAAb



BaabAb



BaBabb

{ deux a ($= \sigma_1$)
zéro A ($= \sigma_1^{-1}$)

tresse **non** triviale

... on ne sait pas (!)

... on ne sait pas (!)

De toute façon, on peut faire (beaucoup) mieux :

↔ solution **locale**

... on ne sait pas (!)

De toute façon, on peut faire (beaucoup) mieux :

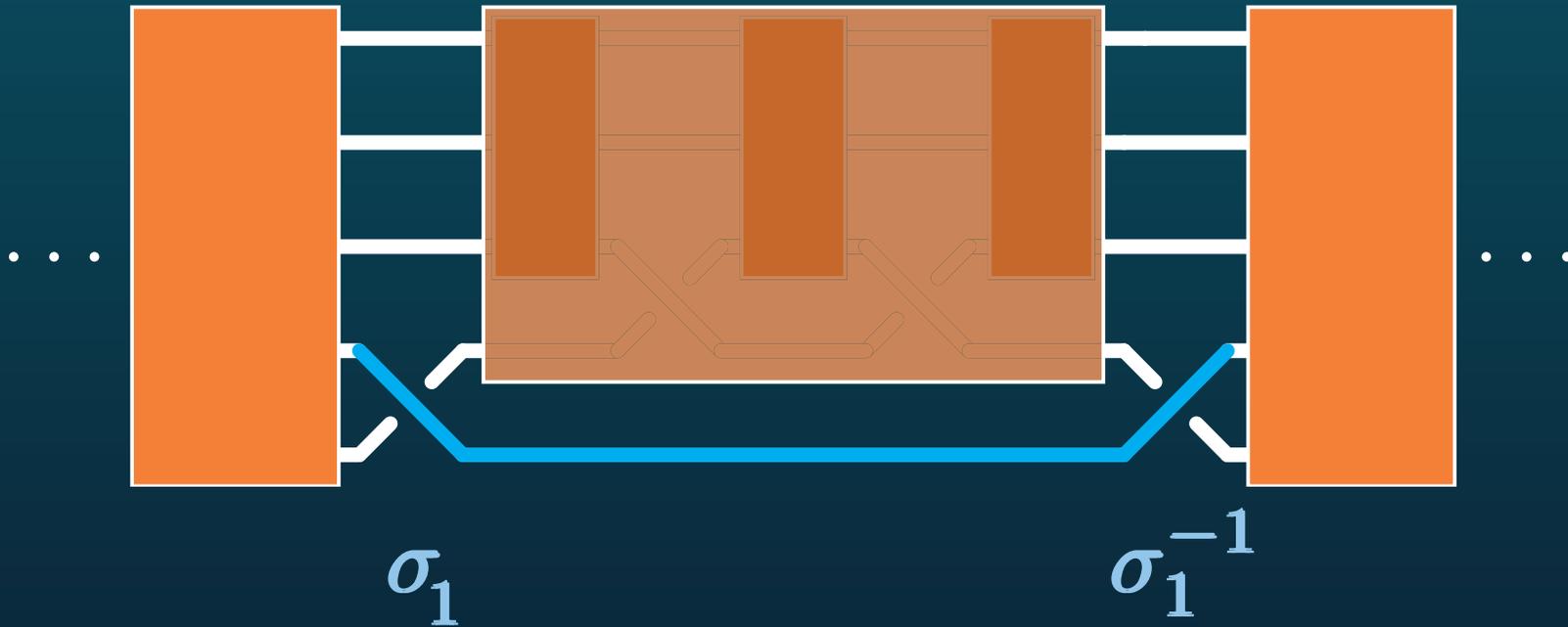
↔ solution **locale**



... on ne sait pas (!)

De toute façon, on peut faire (beaucoup) mieux :

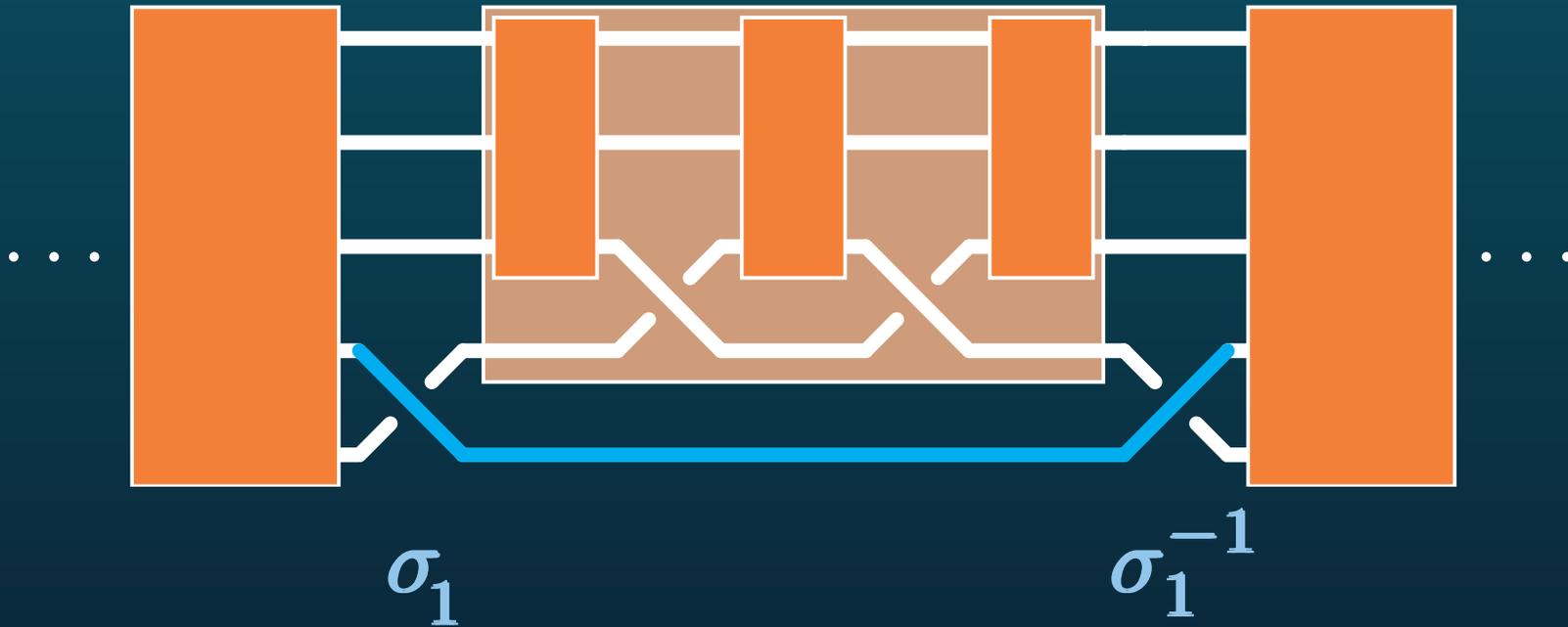
↔ solution **locale**



... on ne sait pas (!)

De toute façon, on peut faire (beaucoup) mieux :

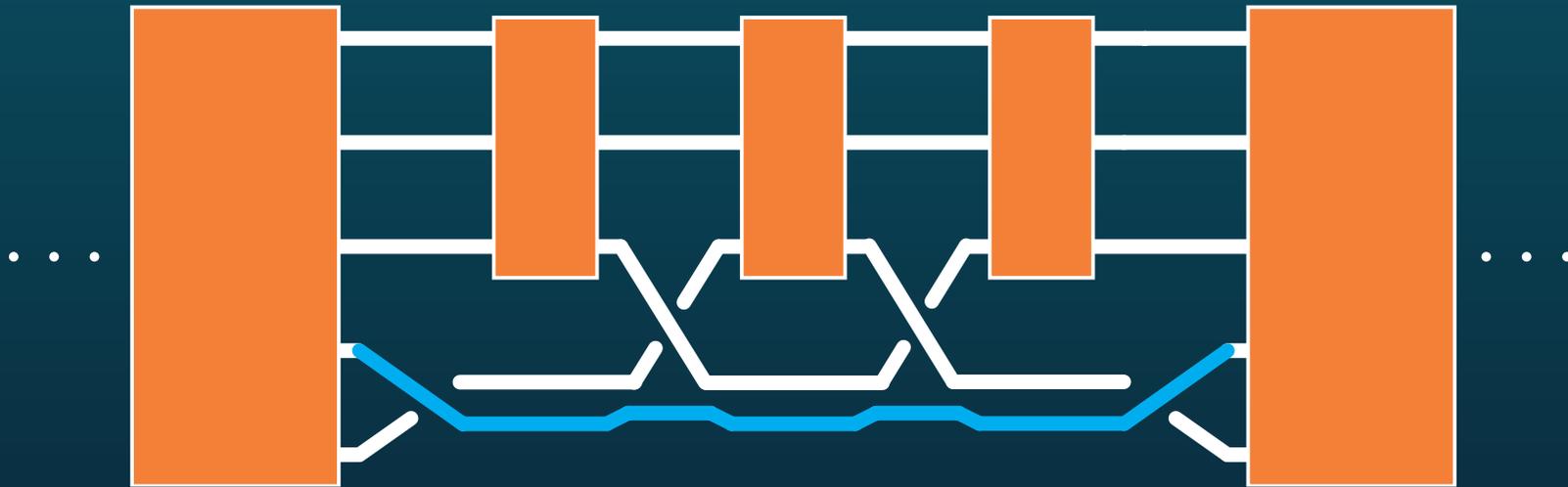
↔ solution **locale**



... on ne sait pas (!)

De toute façon, on peut faire (beaucoup) mieux :

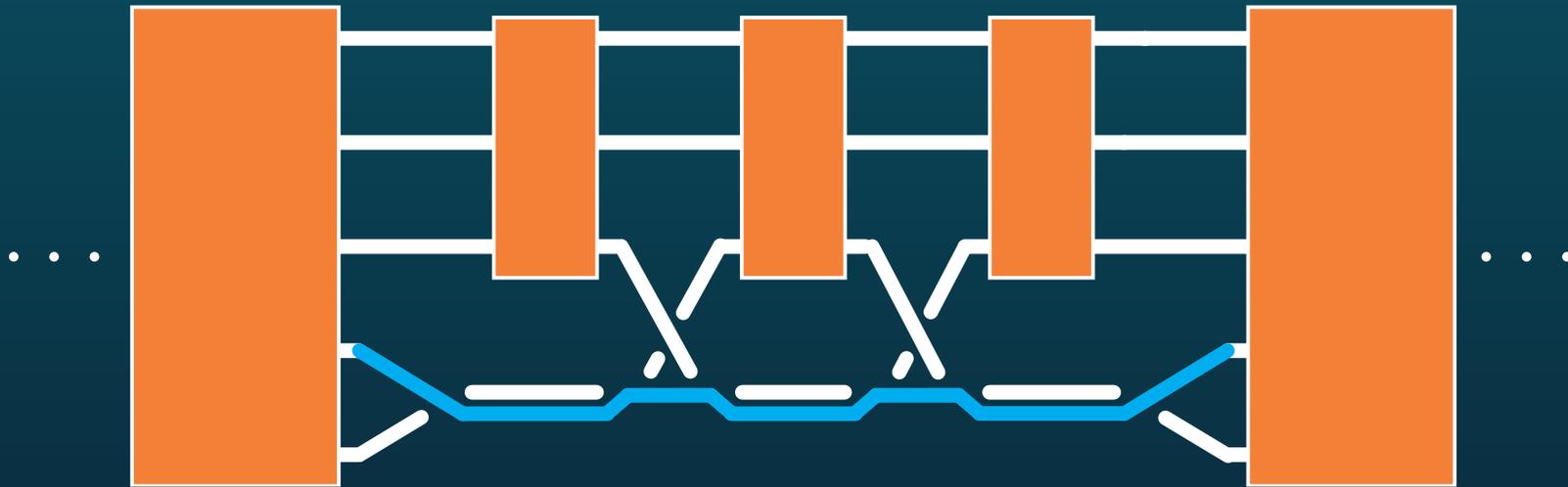
↔ solution **locale**



... on ne sait pas (!)

De toute façon, on peut faire (beaucoup) mieux :

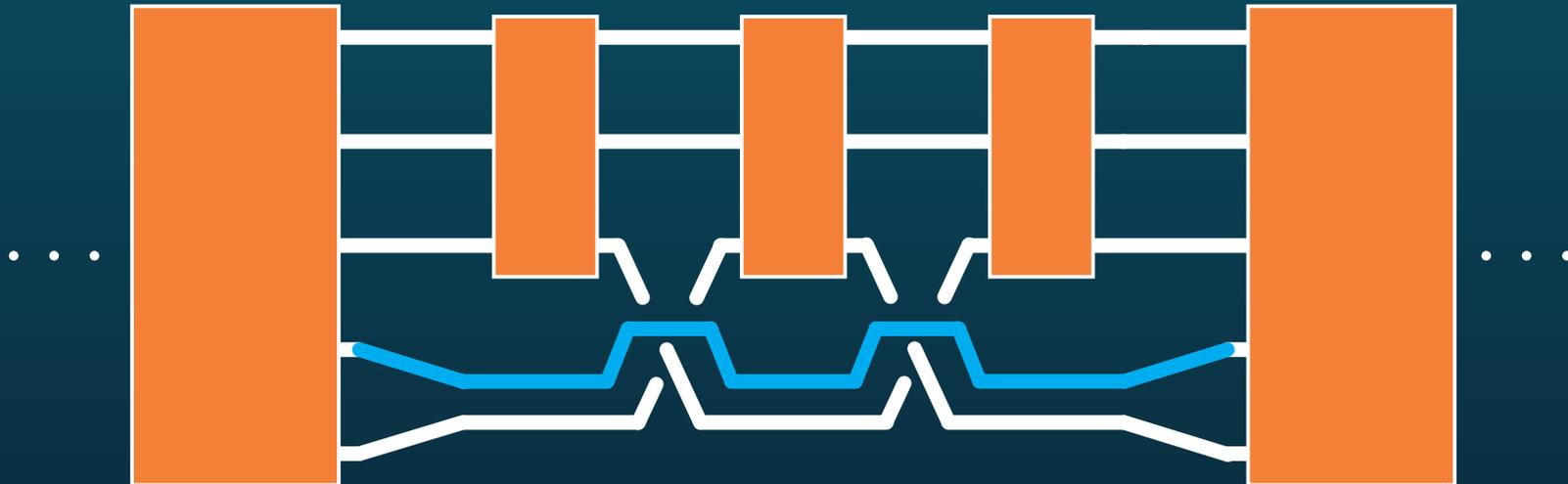
↔ solution **locale**



... on ne sait pas (!)

De toute façon, on peut faire (beaucoup) mieux :

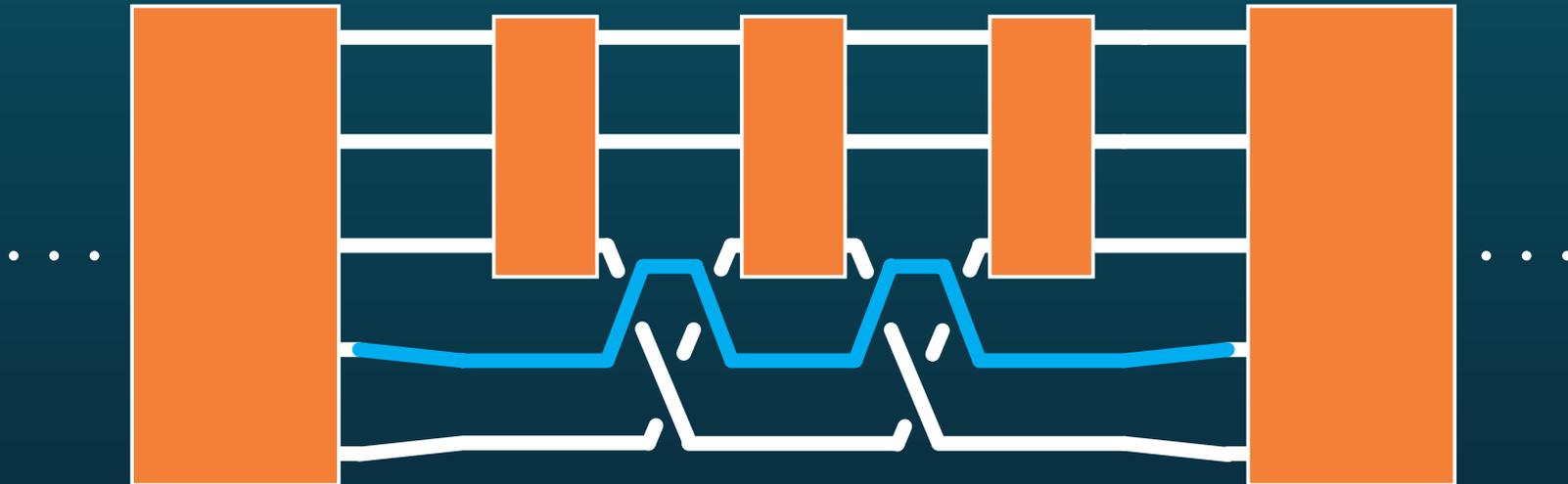
↔ solution **locale**



... on ne sait pas (!)

De toute façon, on peut faire (beaucoup) mieux :

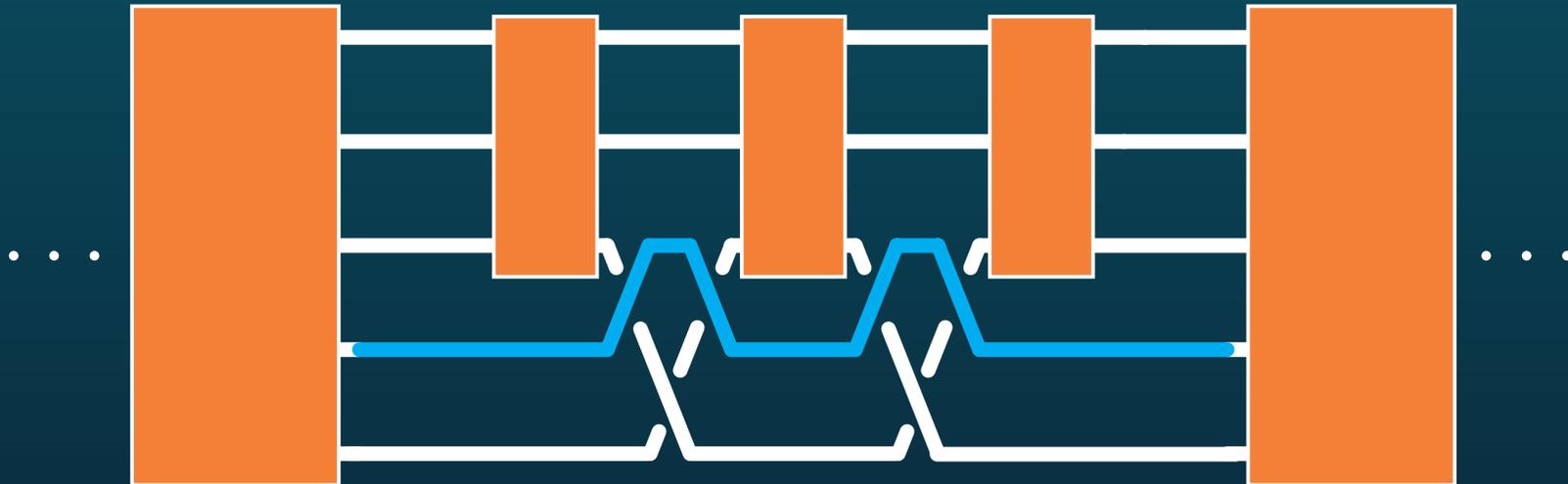
↔ solution **locale**



... on ne sait pas (!)

De toute façon, on peut faire (beaucoup) mieux :

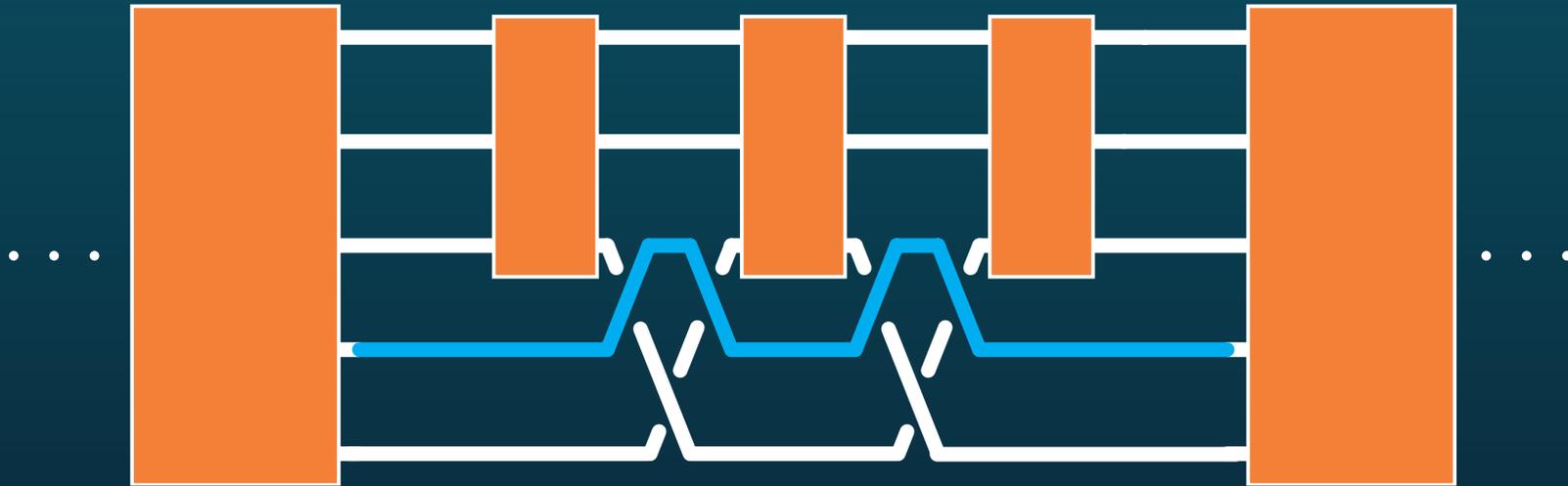
↔ solution **locale**



... on ne sait pas (!)

De toute façon, on peut faire (beaucoup) mieux :

↔ solution **locale**



- Théorème: A condition de réduire les sous-poignées d'abord, cela marche **toujours** (= l'itération se termine).

- **Démonstration?**

- Démonstration?

↪ **Graphe de Cayley** d'un groupe G

relativement à des générateurs s_1, \dots, s_n :

- sommets = éléments de G ,
- arêtes : une arête étiquetée s_i de a à b pour $b = as_i$.

- Démonstration?

↪ **Graphe de Cayley** d'un groupe G

relativement à des générateurs s_1, \dots, s_n :

- sommets = éléments de G ,

- arêtes : une arête étiquetée s_i de a à b pour $b = as_i$.

- Exemple : \mathbb{Z} et 1



- Démonstration?

↪ Graphe de Cayley d'un groupe G

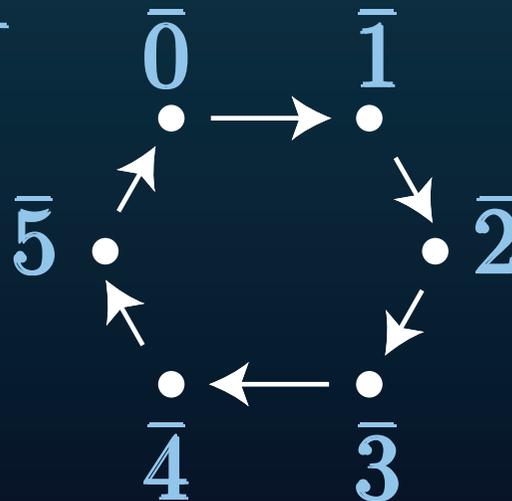
relativement à des générateurs s_1, \dots, s_n :

- sommets = éléments de G ,
- arêtes : une arête étiquetée s_i de a à b pour $b = as_i$.

- Exemple : \mathbb{Z} et 1



- Exemple : $\mathbb{Z}/6\mathbb{Z}$ et $\bar{1}$



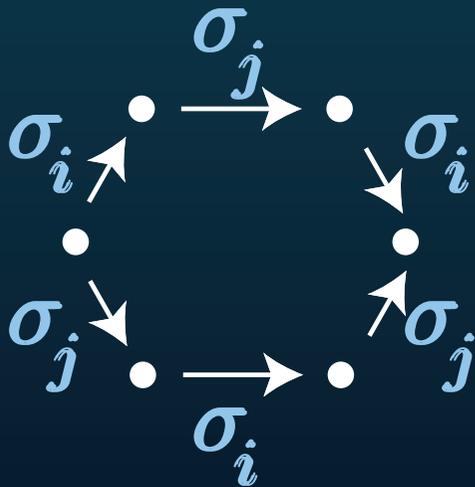
- Exemple : S_4 et (12) , (23) , (34)



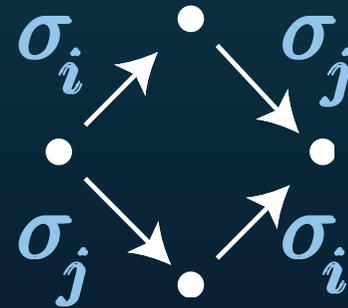
- Exemple : S_4 et $(12), (23), (34)$



- Exemple : B_n et $\sigma_1, \dots, \sigma_{n-1} \rightsquigarrow$ enchevêtrement de



et de



- Une méthode **facile** à implémenter :

- Une méthode **facile** à implémenter :

- poignée = sous-mot **a...A** ou **A...a** sans **a** ni **A** dans ...

- Une méthode **facile** à implémenter :

- poignée = sous-mot **a...A** ou **A...a** sans **a** ni **A** au milieu;
- pas de sous-poignée = pas à la fois **b** et **B** au milieu;

- Une méthode **facile** à implémenter :

- poignée = sous-mot **a...A** ou **A...a** sans **a** ni **A** au milieu;
- pas de sous-poignée = pas à la fois **b** et **B** au milieu;
- réduire = 1. supprimer **a** et **A**;

- Une méthode **facile** à implémenter :

- poignée = sous-mot **a...A** ou **A...a** sans **a** ni **A** au milieu;
- pas de sous-poignée = pas à la fois **b** et **B** au milieu;
- réduire = 1. supprimer **a** et **A**;
2. remplacer **b** par **Bab**

- Une méthode **facile** à implémenter :

- poignée = sous-mot $a...A$ ou $A...a$ sans a ni A au milieu;
- pas de sous-poignée = pas à la fois b et B au milieu;
- réduire = 1. supprimer a et A ;
2. remplacer b par Bab et B par BAb (cas $a...A$),

- Une méthode **facile** à implémenter :

- poignée = sous-mot $a...A$ ou $A...a$ sans a ni A au milieu;
- pas de sous-poignée = pas à la fois b et B au milieu;
- réduire = 1. supprimer a et A ;
2. remplacer b par Bab et B par BAb (cas $a...A$),
ou b par baB et B par bAA (cas $A...a$);

- Une méthode **facile** à implémenter :
 - poignée = sous-mot $a...A$ ou $A...a$ sans a ni A au milieu;
 - pas de sous-poignée = pas à la fois b et B au milieu;
 - réduire = 1. supprimer a et A ;
2. remplacer b par Bab et B par BAb (cas $a...A$),
ou b par baB et B par bAA (cas $A...a$);
- et (très) **efficace**: mot de 10.000 lettres \approx 1sec.

- Une méthode **facile** à implémenter :

- poignée = sous-mot $a...A$ ou $A...a$ sans a ni A au milieu;

- pas de sous-poignée = pas à la fois b et B au milieu;

- réduire = 1. supprimer a et A ;

- 2. remplacer b par Bab et B par BAb (cas $a...A$),

- ou b par baB et B par bAA (cas $A...a$);

- et (très) **efficace**: mot de 10.000 lettres \approx 1sec.

- mais $\left\{ \begin{array}{l} \text{preuve élémentaire de convergence} \\ \text{complexité algorithmique} \end{array} \right\}$ **inconnues ...**

- **Pourquoi étudier les groupes de tresses ?**

- Pourquoi étudier les groupes de tresses ?

- Réponse 1 (**mathématiques**) :

tresse = permutation + son histoire

● Pourquoi étudier les groupes de tresses ?

- Réponse 1 (**mathématiques**) :

tresse = permutation + son histoire



- Pourquoi étudier les groupes de tresses ?

- Réponse 1 (**mathématiques**) :

tresse = permutation + son histoire



↪ groupe de tresses = extension du groupe symétrique

↪ combinatoire, théorie de Coxeter

- Pourquoi étudier les groupes de tresses ?

- Réponse 1 (**mathématiques**) :

tresse = permutation + son histoire



↪ groupe de tresses = extension du groupe symétrique

↪ combinatoire, théorie de Coxeter

F. Garside, P. Deligne, E. Brieskorn, ...

- Réponse 2 (**physique**) :

- Réponse 2 (**physique**) :

groupe des tresses = symétries de l'équation de Yang–Baxter

$$(R \otimes \text{id})(\text{id} \otimes R)(R \otimes \text{id}) = (\text{id} \otimes R)(R \otimes \text{id})(\text{id} \otimes R)$$

$$\text{où } R : V \otimes V \rightarrow V \otimes V.$$

(cf. relation de tresse $\mathbf{aba} = \mathbf{bab}$)

- Réponse 2 (**physique**) :

groupe des tresses = symétries de l'équation de Yang–Baxter

$$(R \otimes \text{id})(\text{id} \otimes R)(R \otimes \text{id}) = (\text{id} \otimes R)(R \otimes \text{id})(\text{id} \otimes R)$$

$$\text{où } R : V \otimes V \rightarrow V \otimes V.$$

(cf. relation de tresse $aba = bab$)

↔ représentations matricielles, cohomologie

- Réponse 2 (**physique**) :

groupe des tresses = symétries de l'équation de Yang–Baxter

$$(R \otimes \text{id})(\text{id} \otimes R)(R \otimes \text{id}) = (\text{id} \otimes R)(R \otimes \text{id})(\text{id} \otimes R)$$

$$\text{où } R : V \otimes V \rightarrow V \otimes V.$$

(cf. relation de tresse $aba = bab$)

↔ représentations matricielles, cohomologie

V. Arnold, D. Krammer, S. Bigelow, ...

- Réponse 3 (**chimie, biologie**) :

tresse = nœud ouvert

- Réponse 3 (**chimie, biologie**) :

tresse = nœud ouvert



- Réponse 3 (**chimie, biologie**) :

tresse = nœud ouvert



- Réponse 3 (**chimie, biologie**) :

tresse = nœud ouvert



- Réponse 3 (**chimie, biologie**) :

tresse = nœud ouvert



↔ classification, invariants, groupes quantiques

- Réponse 3 (**chimie, biologie**) :

tresse = nœud ouvert



↔ classification, invariants, groupes quantiques

V. Jones, V. Turaev, L. Kauffman, ...

- Réponse 4 (**cryptographie**) :

groupe de tresses = groupe simple **et** compliqué

- Réponse 4 (**cryptographie**) :

groupe de tresses = groupe simple **et** compliqué

↑
on peut calculer

- Réponse 4 (**cryptographie**) :

groupe de tresses = groupe simple **et** compliqué

on peut calculer ↑
il y a des problèmes difficiles ↑

- **Echange de clé** : Alice et Bob fixent une clé commune s , mais un intrus observant les échanges ne peut la trouver.

- **Echange de clé** : Alice et Bob fixent une clé commune s , mais un intrus observant les échanges ne peut la trouver.

- On choisit p **publique** dans B_{2n} ;

- **Echange de clé** : Alice et Bob fixent une clé commune s , mais un intrus observant les échanges ne peut la trouver.

- On choisit p **publique** dans B_{2n} ;

- Alice choisit a dans B_n , et envoie $p_A = apa^{-1}$ à Bob;

• **Echange de clé** : Alice et Bob fixent une clé commune s , mais un intrus observant les échanges ne peut la trouver.

- On choisit p **publique** dans B_{2n} ;

- Alice choisit a dans B_n , et envoie $p_A = apa^{-1}$ à Bob;

- Bob choisit b dans $B_{n,2n}$ ←, et envoie $p_B = bpb^{-1}$ à Alice;

tresses sur $\sigma_{n+1} \dots \sigma_{2n-1}$

• **Echange de clé** : Alice et Bob fixent une clé commune s , mais un intrus observant les échanges ne peut la trouver.

- On choisit p **publique** dans B_{2n} ;

- Alice choisit a dans B_n , et envoie $p_A = apa^{-1}$ à Bob;

- Bob choisit b dans $B_{n,2n}$, et envoie $p_B = bpb^{-1}$ à Alice;

- Alice calcule $s = ap_B a^{-1}$; Bob calcule $s = bp_A b^{-1}$.

tresses sur $\sigma_{n+1} \dots \sigma_{2n-1}$

- **Echange de clé** : Alice et Bob fixent une clé commune s , mais un intrus observant les échanges ne peut la trouver.

- On choisit p **publique** dans B_{2n} ;

- Alice choisit a dans B_n , et envoie $p_A = apa^{-1}$ à Bob;

- Bob choisit b dans $B_{n,2n}$, et envoie $p_B = bpb^{-1}$ à Alice;

- Alice calcule $s = ap_B a^{-1}$; Bob calcule $s = bp_A b^{-1}$.

tresses sur $\sigma_{n+1} \dots \sigma_{2n-1}$

- **Justification**: On a $ab = ba$, donc

$$ap_B a^{-1} = abpb^{-1}a^{-1} = bap_a^{-1}b^{-1} = bp_A b^{-1}.$$

- **Echange de clé** : Alice et Bob fixent une clé commune s , mais un intrus observant les échanges ne peut la trouver.

- On choisit p **publique** dans B_{2n} ;

- Alice choisit a dans B_n , et envoie $p_A = apa^{-1}$ à Bob;

- Bob choisit b dans $B_{n,2n}$, et envoie $p_B = bpb^{-1}$ à Alice;

- Alice calcule $s = ap_B a^{-1}$; Bob calcule $s = bp_A b^{-1}$.

tresses sur $\sigma_{n+1} \dots \sigma_{2n-1}$

- **Justification**: On a $ab = ba$, donc

$$ap_B a^{-1} = abpb^{-1}a^{-1} = bap_a^{-1}b^{-1} = bp_A b^{-1}.$$

- **Sécurité**: Tirer a ou b de (p, apa^{-1}) et (p, bpb^{-1}) ?

<http://www.math.unicaen.fr/~dehornoy>

<http://www.math.unicaen.fr/~dehornoy>

<http://www.math.unicaen.fr/lmno>

<http://www.math.unicaen.fr/~dehornoy>

<http://www.math.unicaen.fr/Imno>

Merci !