LE CALCUL DES TRESSES

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 :

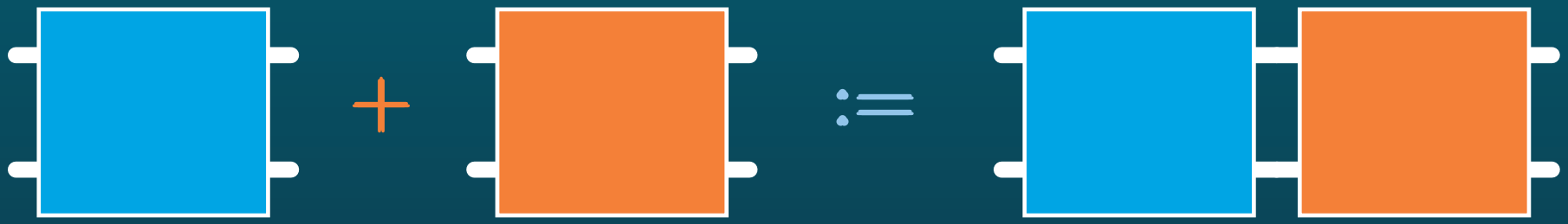
- Une tresse à 3 brins :

etc...

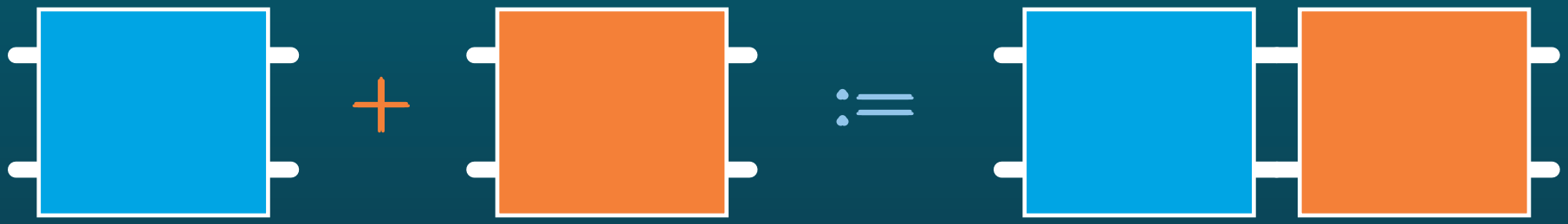
- **Calculer avec les tresses à deux brins :**

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

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

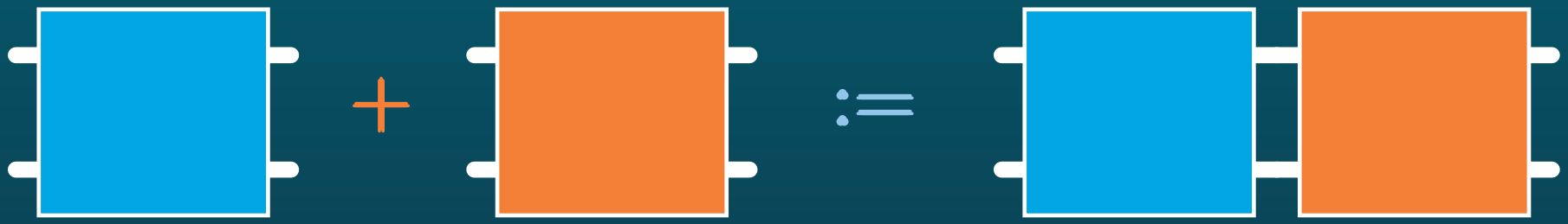


- **Calculer** avec les tresses à deux brins :



⇝ Structure obtenue?

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

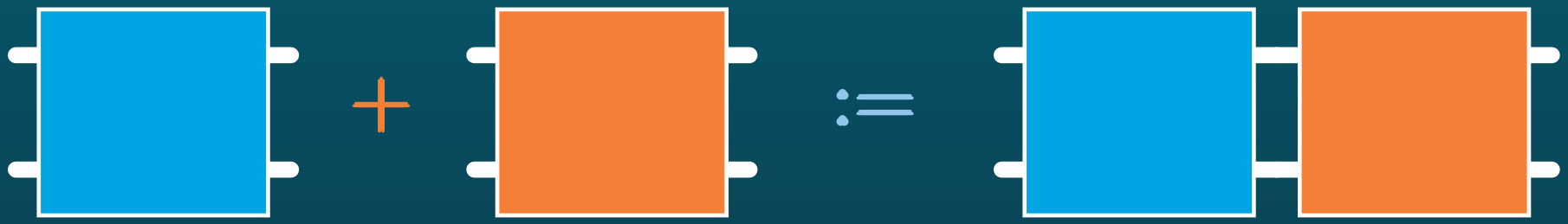


⇝ **Structure obtenue?**

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

$1+0 =$ 

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

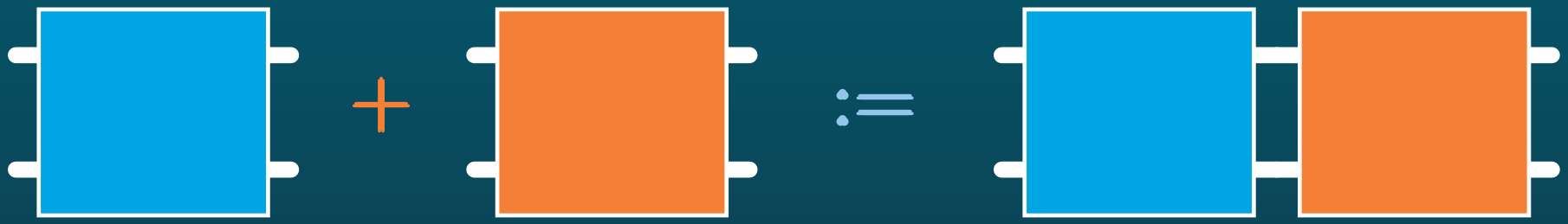


⇝ **Structure obtenue?**

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

$1+0 =$ 

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



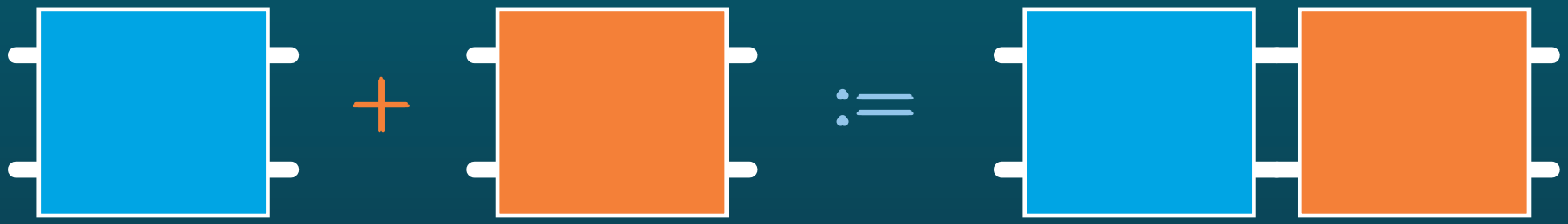
⇝ **Structure obtenue?**

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

$1+0=$ 

se déforme en (?)

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



⇝ **Structure obtenue?**

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

$1+0 =$  $= 1.$

se déforme en (?)

Et aussi :

- $0+1=$  $=1$;

Et aussi :

- $0+1=$  $=1$;
- $1+1=$ 

Et aussi :

- $0+1=$  $=1$;
- $1+1=$  $\rightsquigarrow 2$;

Et aussi :

- $0+1=$  $=1$;
- $1+1=$  $\rightsquigarrow 2$;
- $1+2=$  $\rightsquigarrow 3$

Et aussi :

- $0+1=$  $=1$;
- $1+1=$  $\rightsquigarrow 2$;
- $1+2=$ 

  $=$  $=2+1$.

Soustraction ?

⇝ Soustraction ?

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

$1 + (-1) =$  $+$  $=$

⇝ Soustraction ?

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

$1 + (-1) =$  $+$  $=$ 

⇝ 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

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

⇝ 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

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

⇝ Soustraction ?

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

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

⇝ Soustraction ?

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

$1 + (-1) =$  $+$  $=$  $\approx$  $= 0.$

⇝ Soustraction ?

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

$1 + (-1) =$  $\approx$  $= 0.$

- De même

$(-1) + 1 =$  $\approx$  $= 0.$

- **Un exemple:**

$$1 + (-2) + 3 =$$ 

- **Un exemple:**

$$1 + (-2) + 3 =$$ 

$$=$$ 

- **Un exemple:**

$$1 + (-2) + 3 =$$

$$=$$

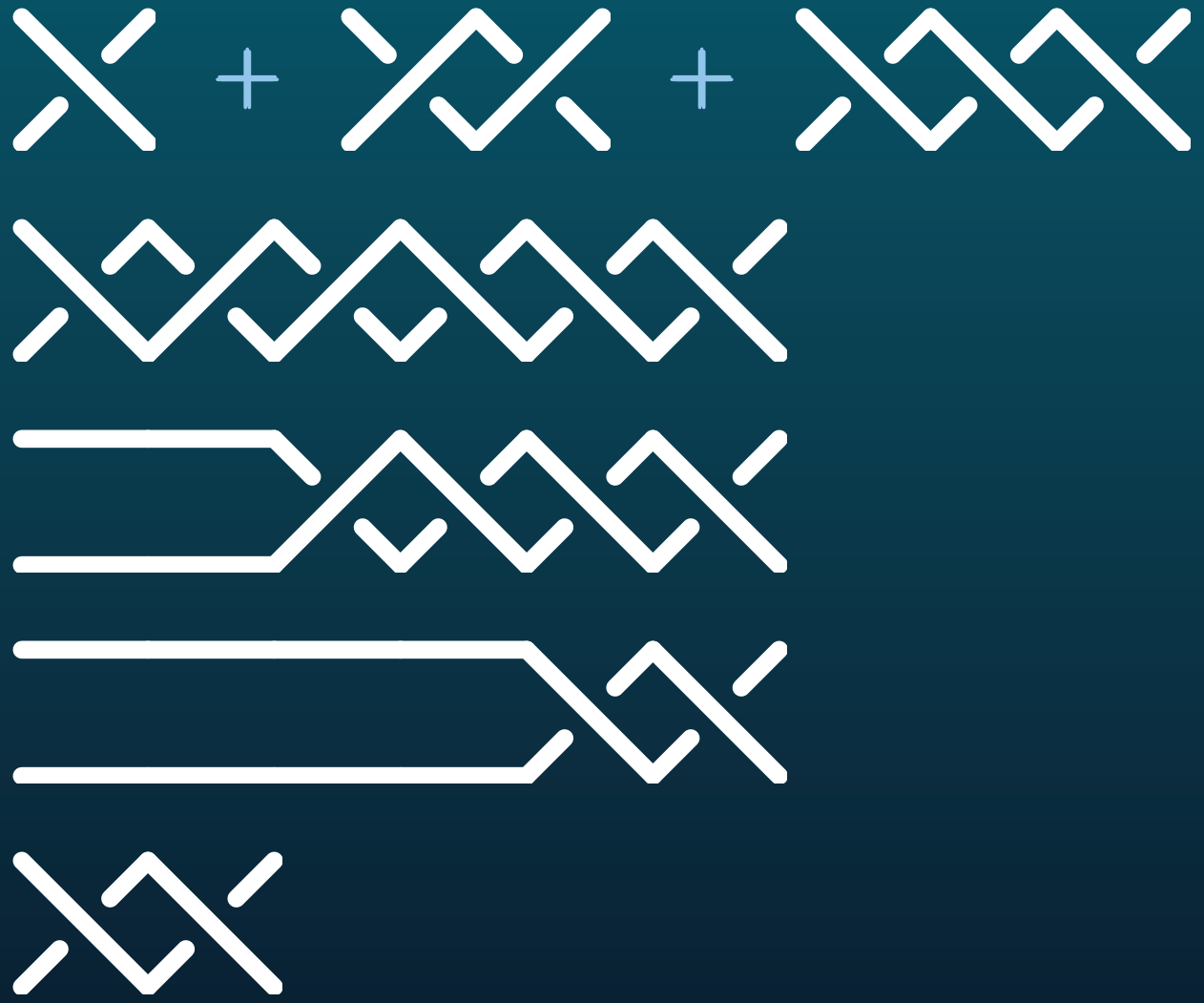
$$\approx$$

- **Un exemple:**

$1 + (-2) + 3 =$  $=$  $\approx$  $\approx$ 

- **Un exemple:**

$1 + (-2) + 3 =$



- **Un exemple:**

$1 + (-2) + 3 =$ 

$=$ 

$\approx$ 

$\approx$ 

$\approx$  $= 2.$

⇝ **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

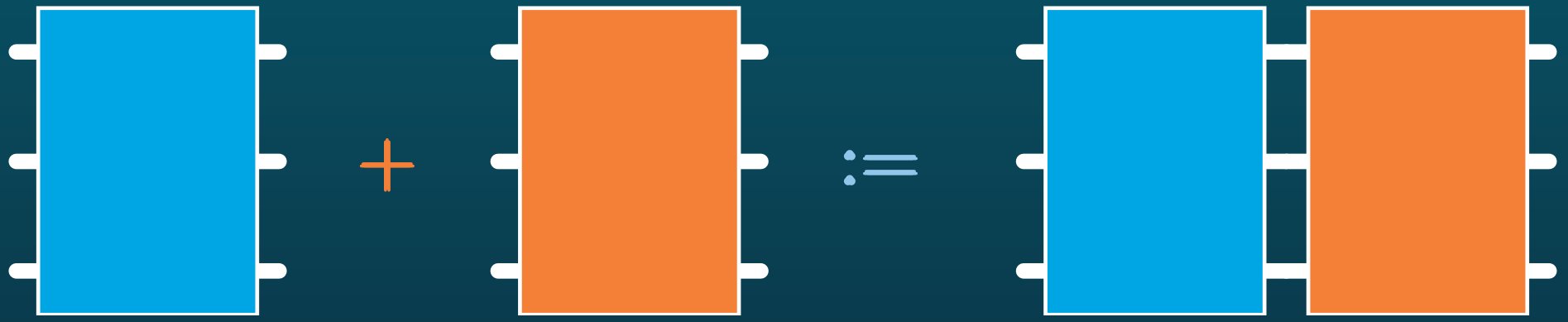
↓

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

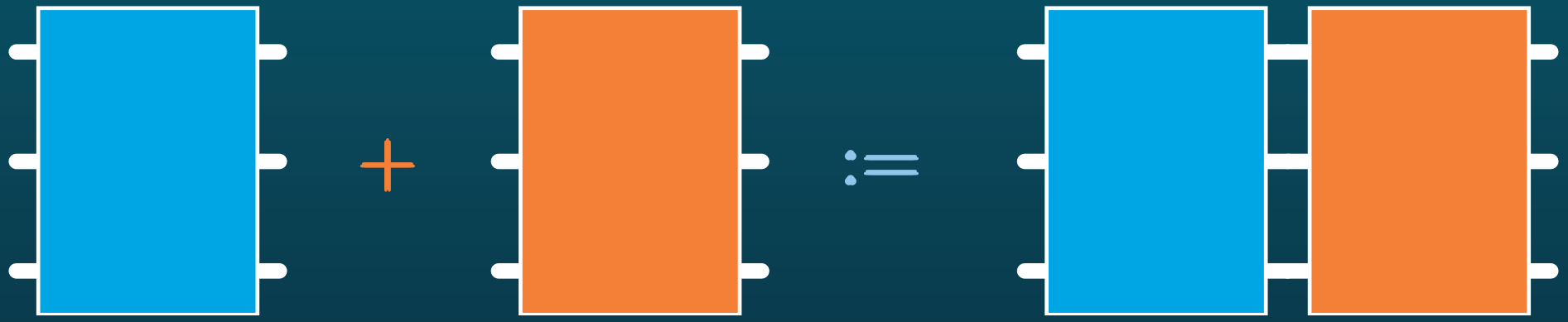
⇝ pas une surprise : compter les demi-tours

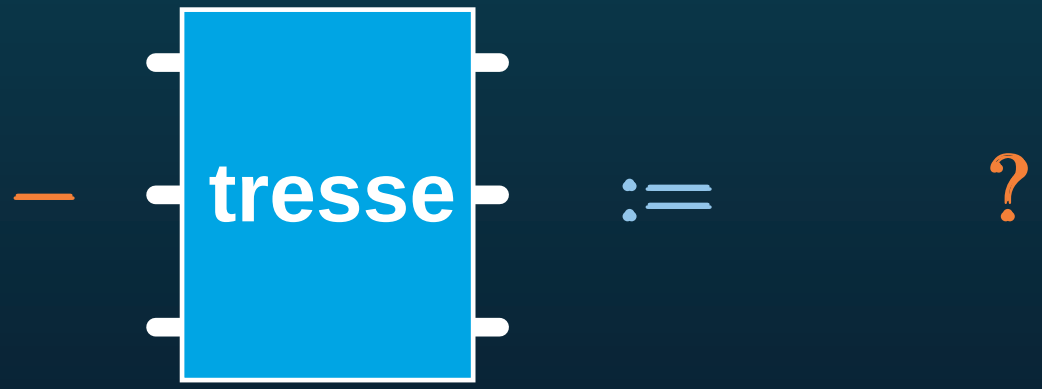
⇝ **Même chose pour les tresses à n brins ($n \geqslant 3$) :**

⇝ **Même chose pour les tresses à n brins ($n \geqslant 3$) :**



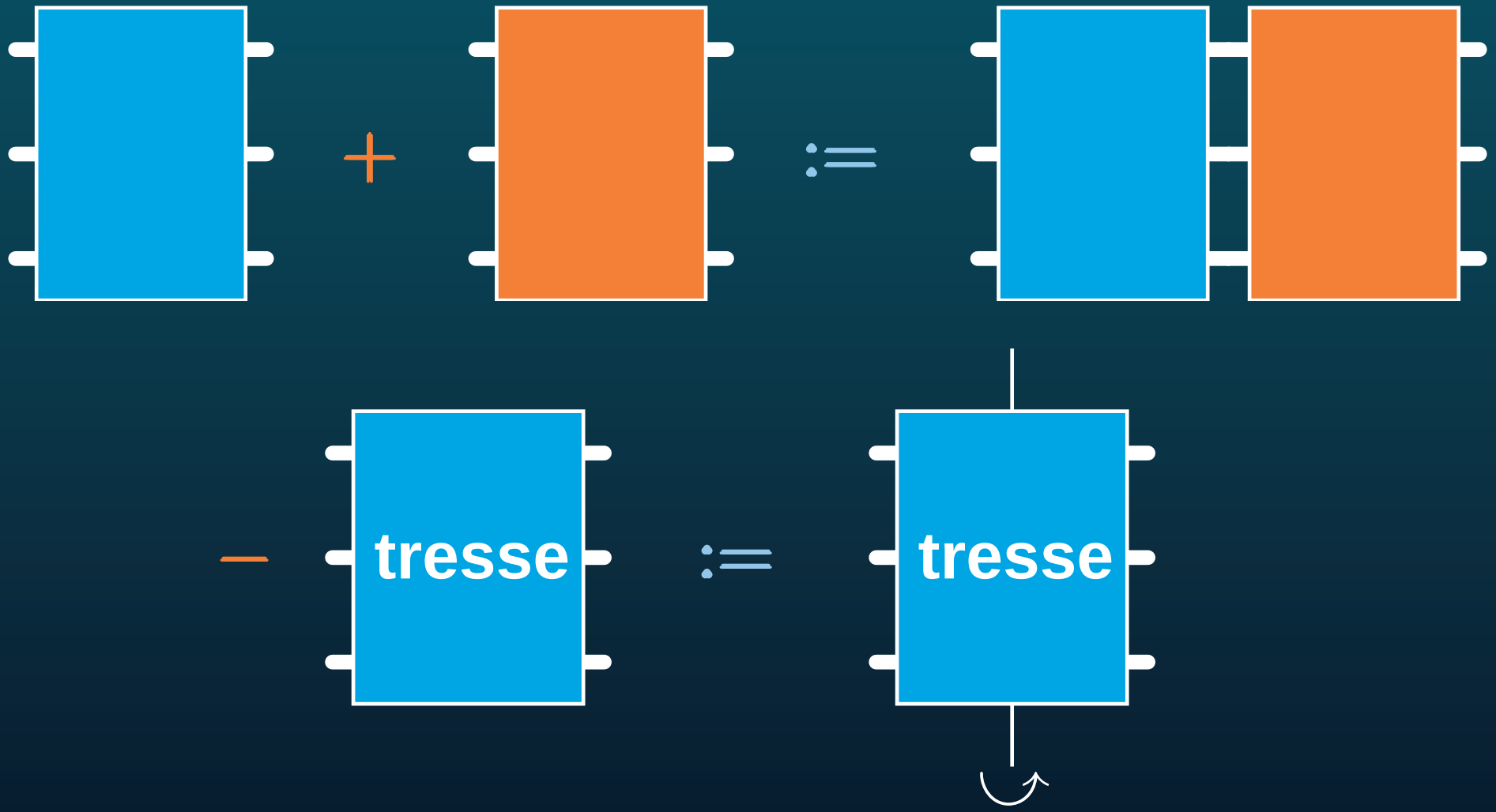
⇝ **Même chose pour les tresses à n brins ($n \geqslant 3$) :**



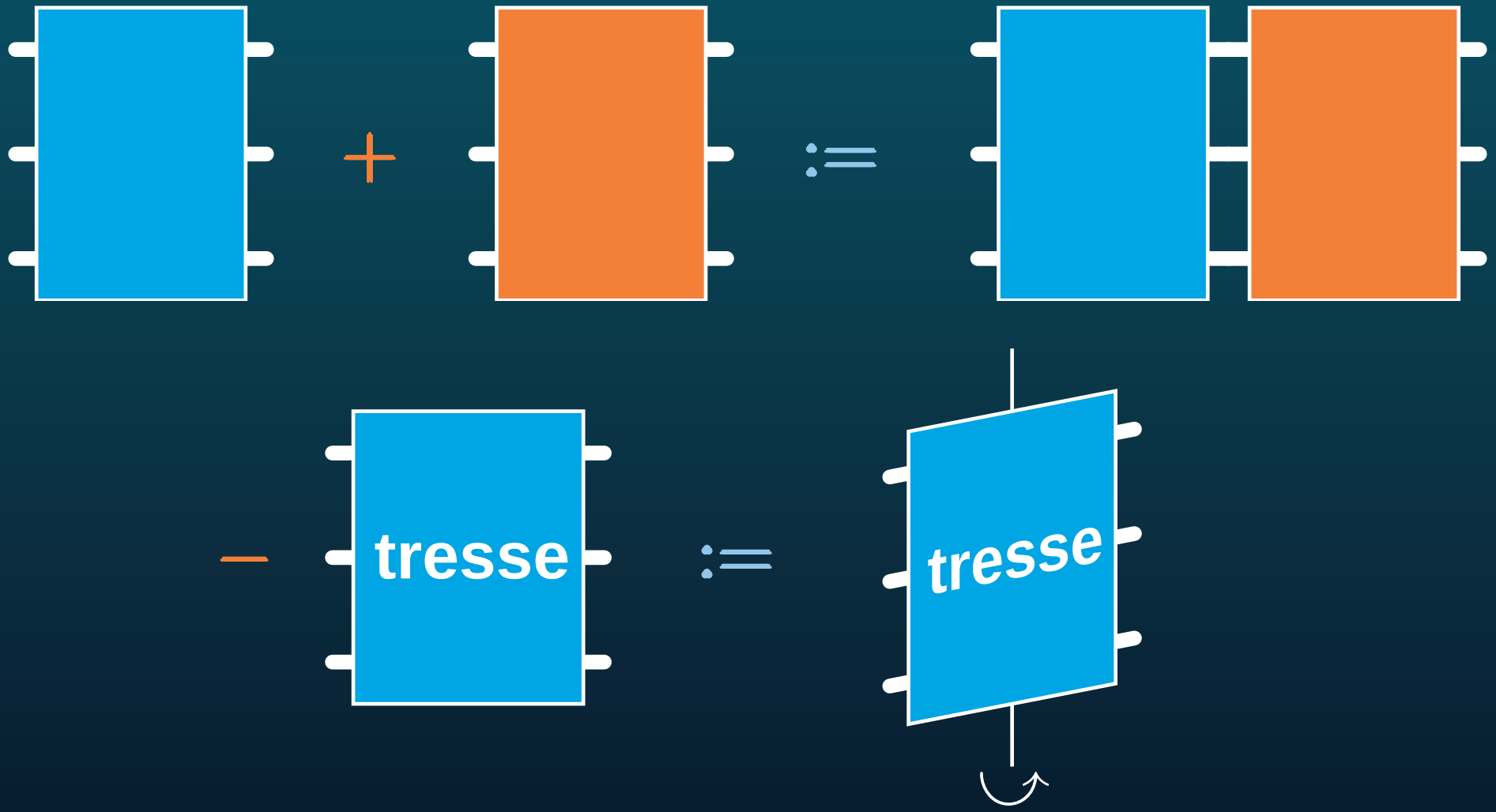


$-$ $:=$?

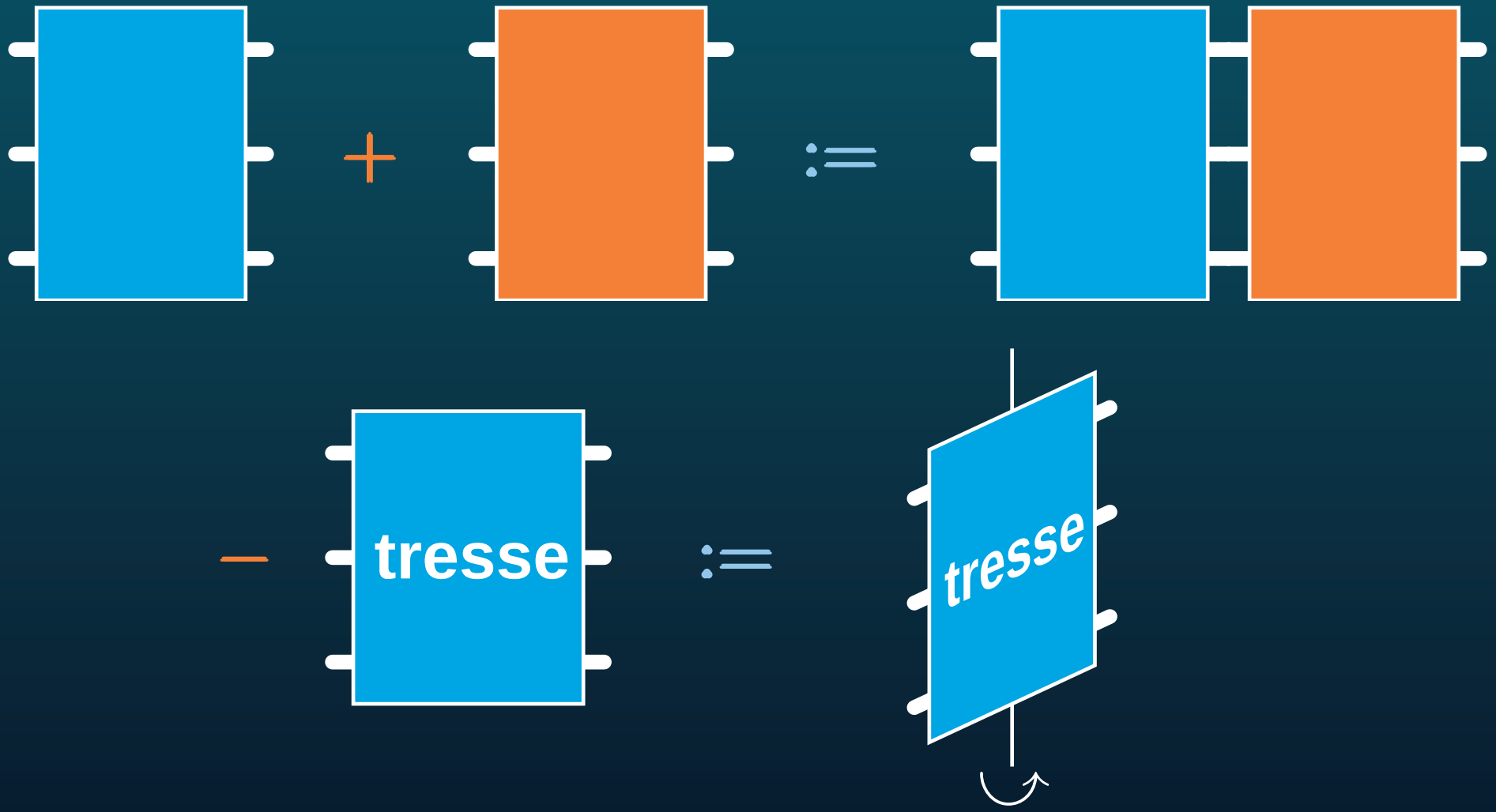
⇝ **Même chose pour les tresses à n brins ($n \geqslant 3$) :**



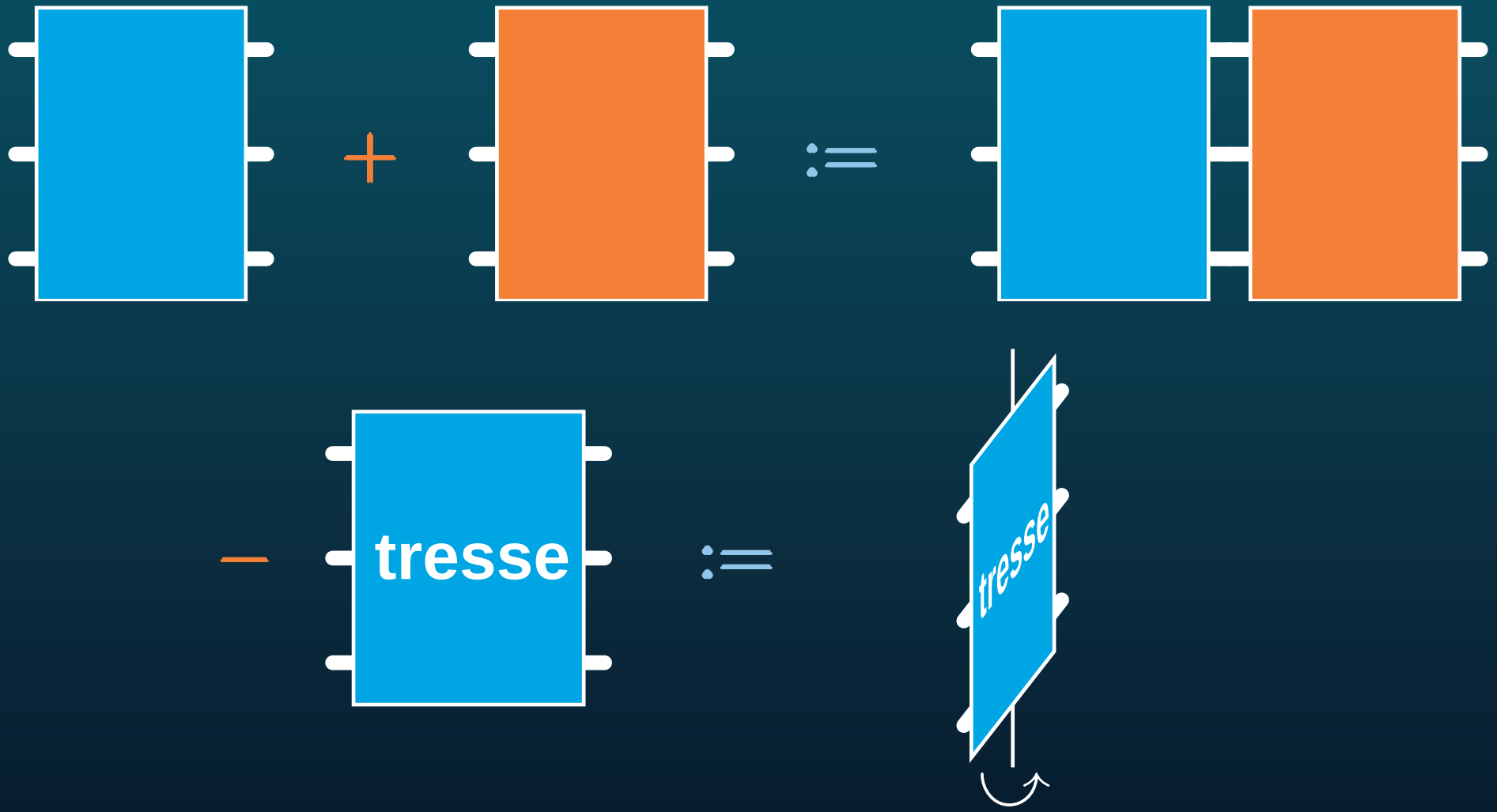
⇝ **Même chose pour les tresses à n brins ($n \geqslant 3$) :**



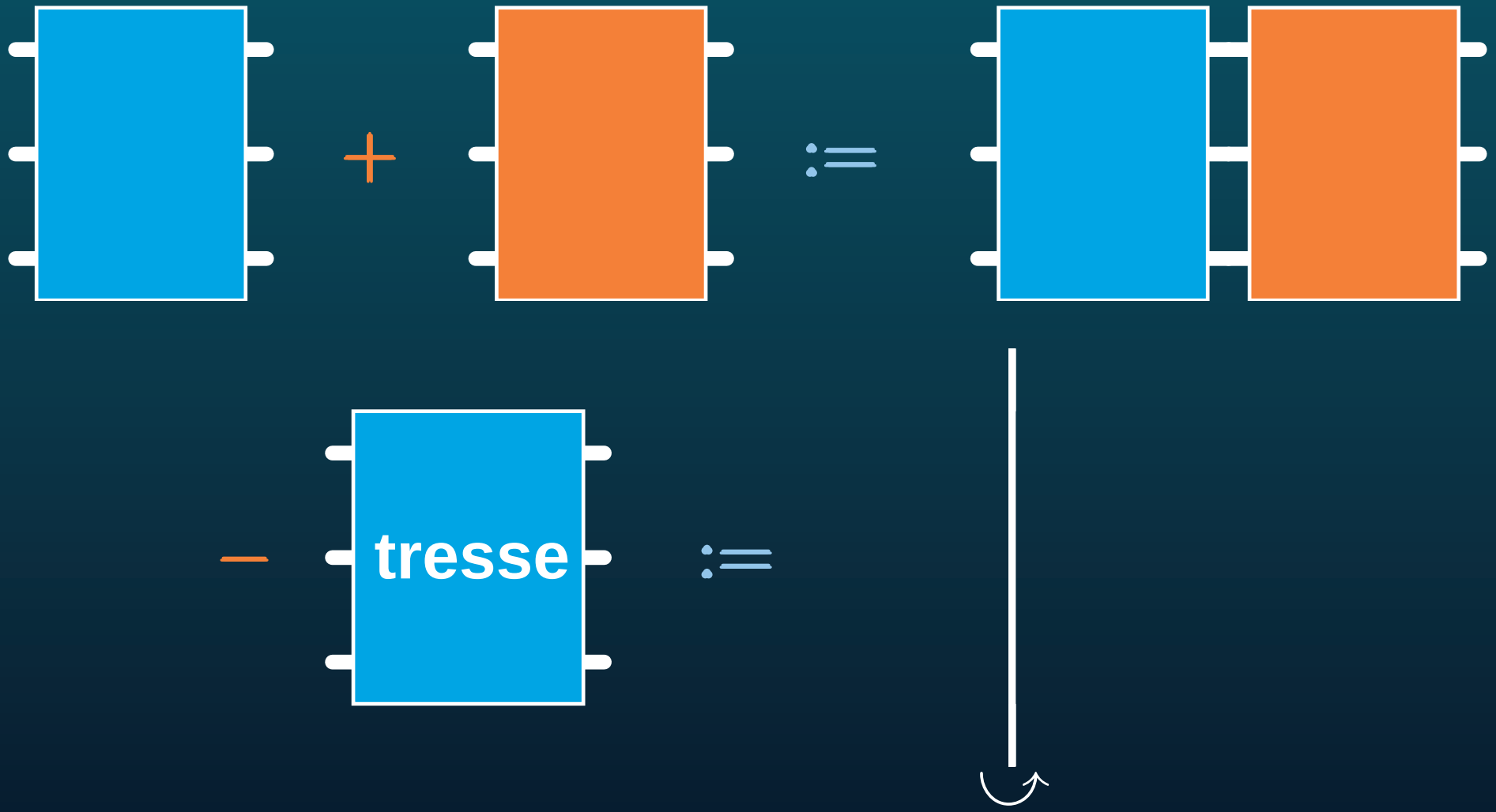
⇝ Même chose pour les tresses à n brins ($n \geqslant 3$) :



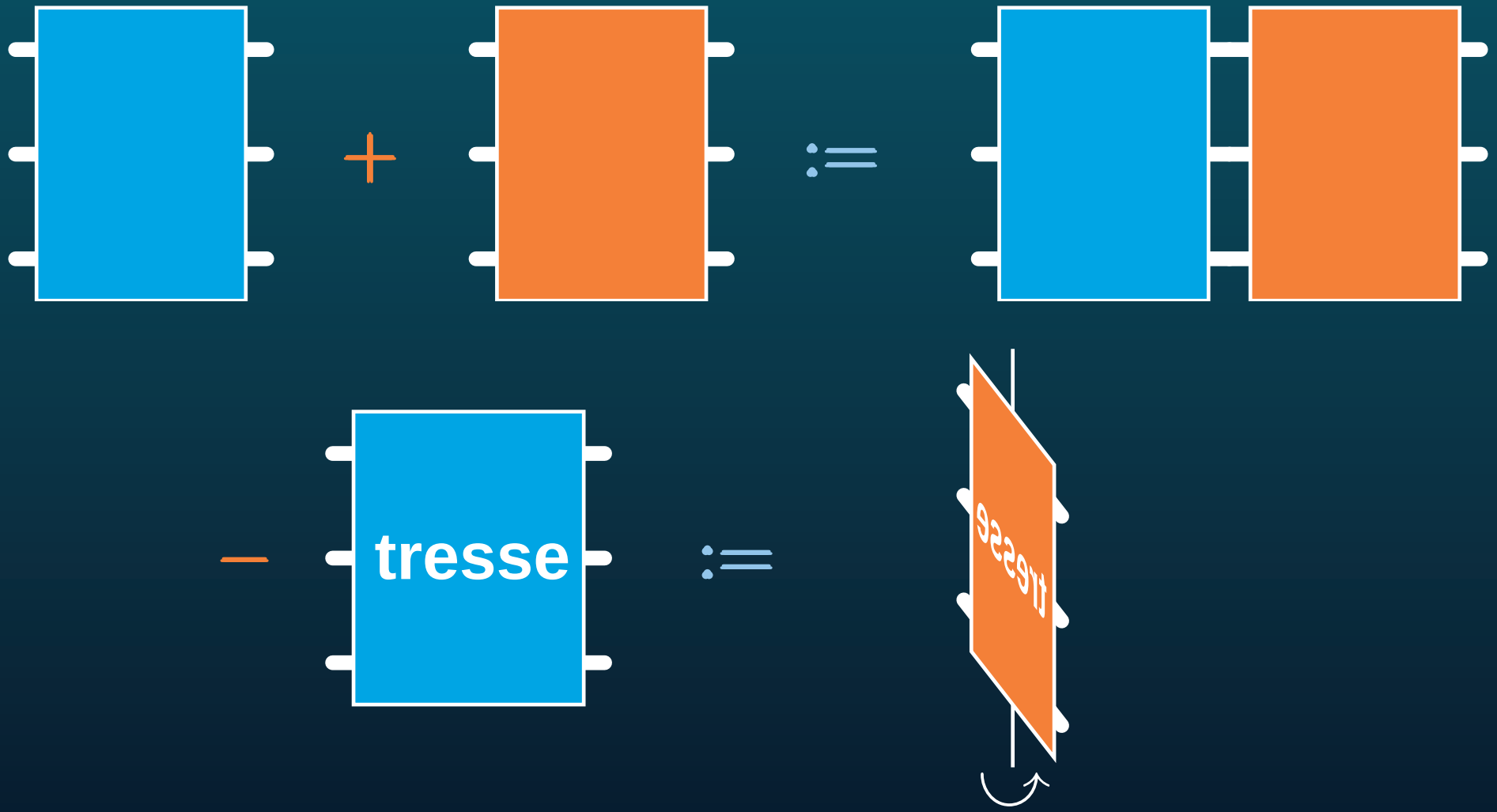
⇝ **Même chose pour les tresses à n brins ($n \geqslant 3$) :**



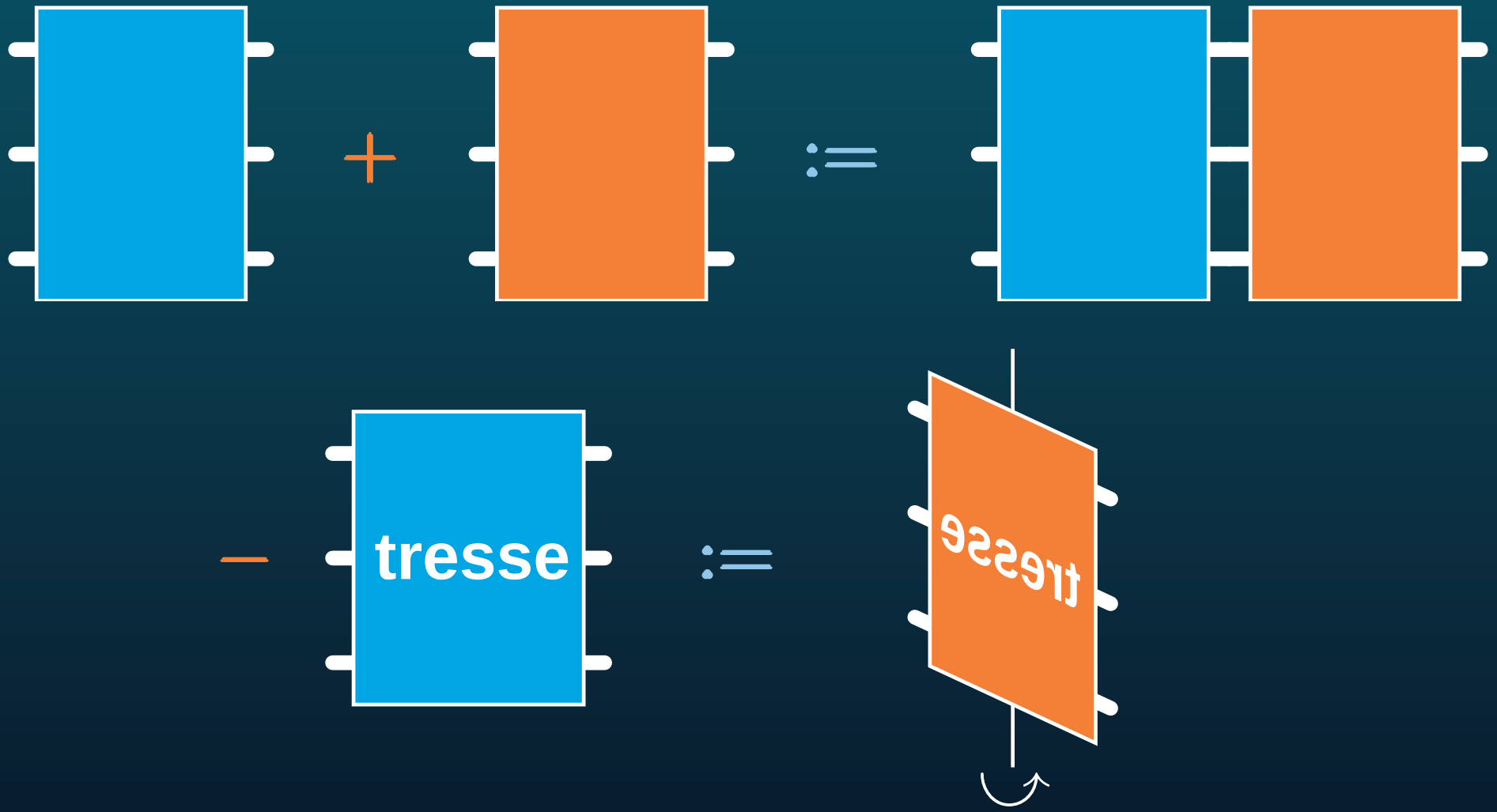
⇝ **Même chose pour les tresses à n brins ($n \geqslant 3$) :**



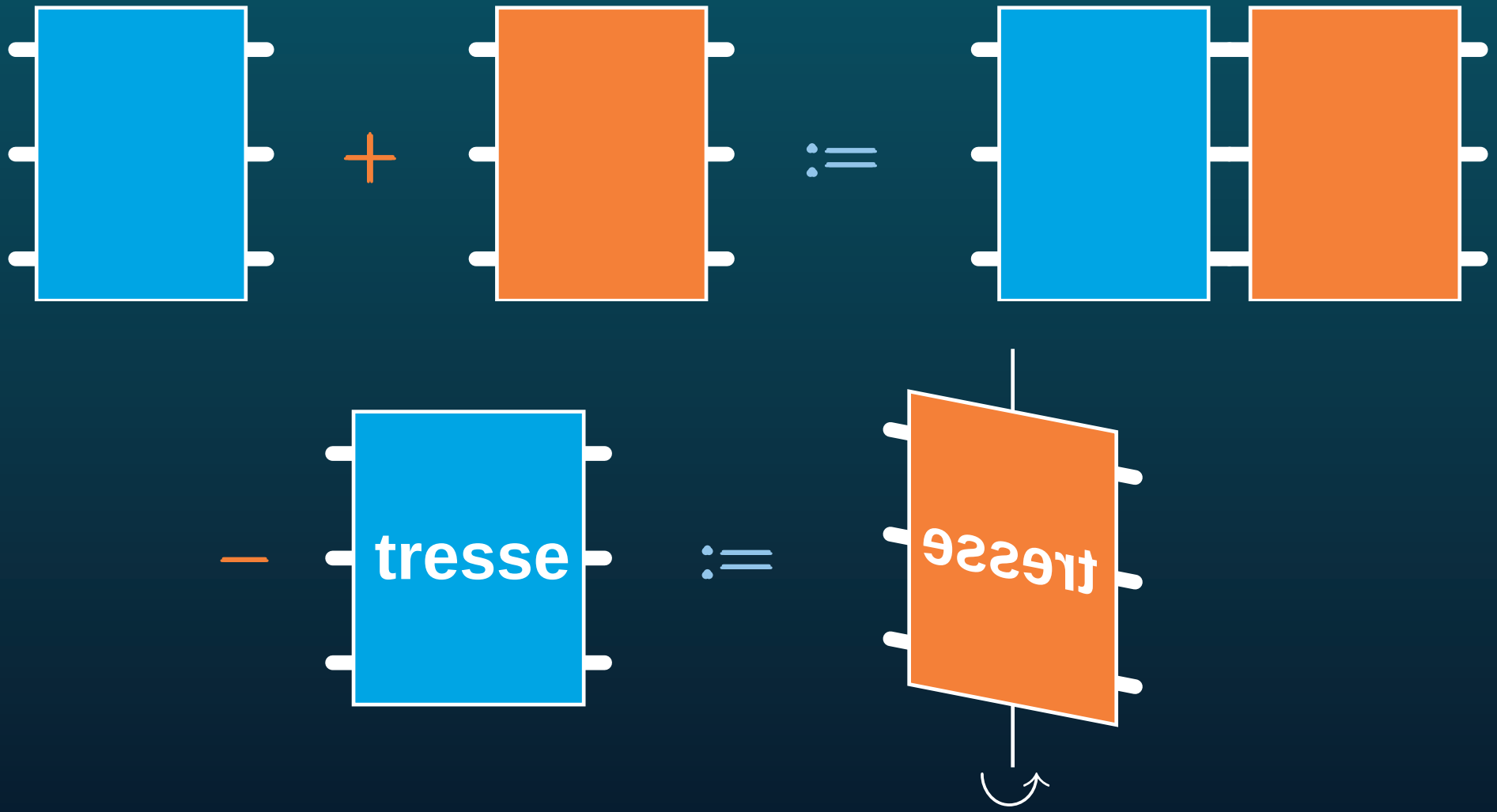
⇝ **Même chose pour les tresses à n brins ($n \geqslant 3$) :**



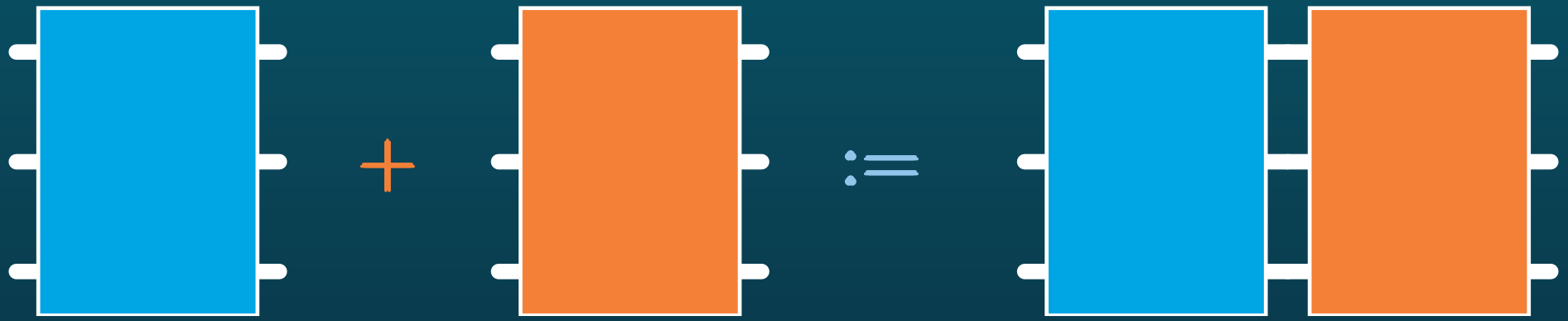
⇝ **Même chose pour les tresses à n brins ($n \geqslant 3$) :**

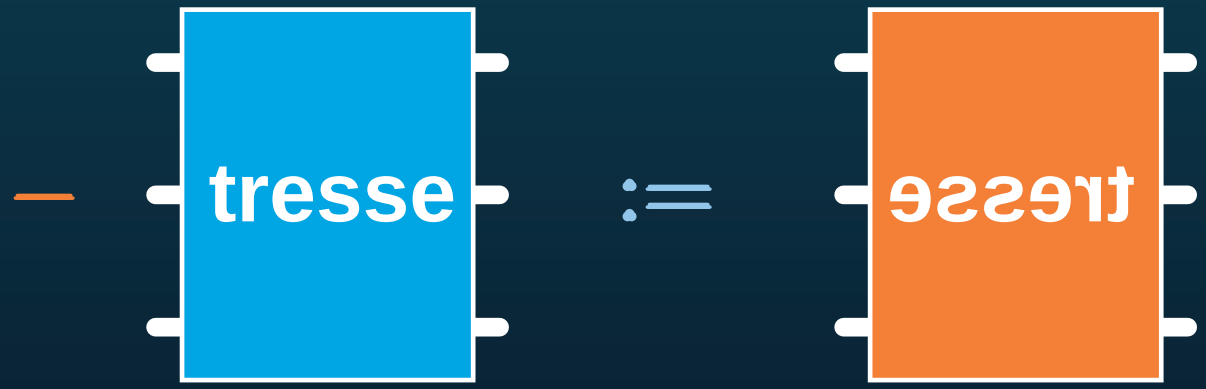


⇝ **Même chose pour les tresses à n brins ($n \geqslant 3$) :**

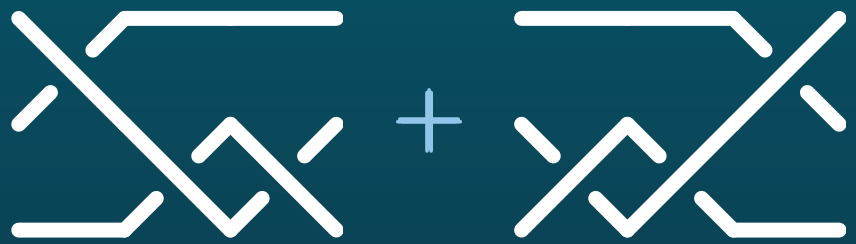


⇝ **Même chose pour les tresses à n brins ($n \geqslant 3$) :**



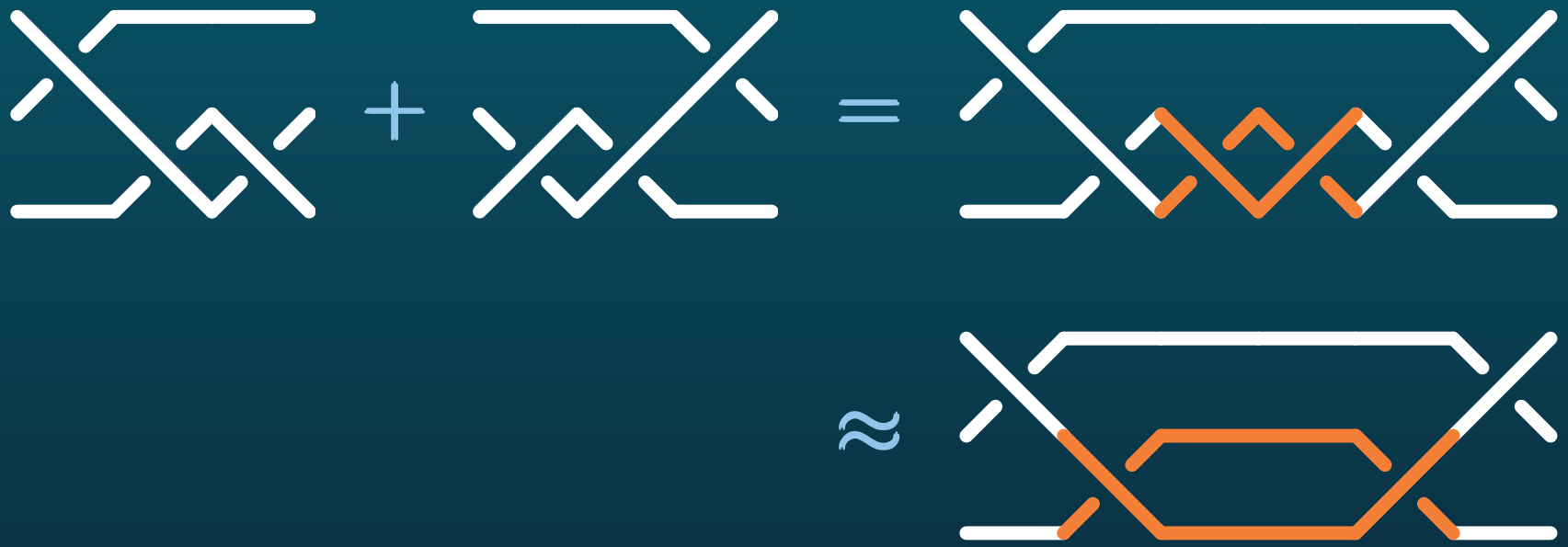


- **Exemple:**

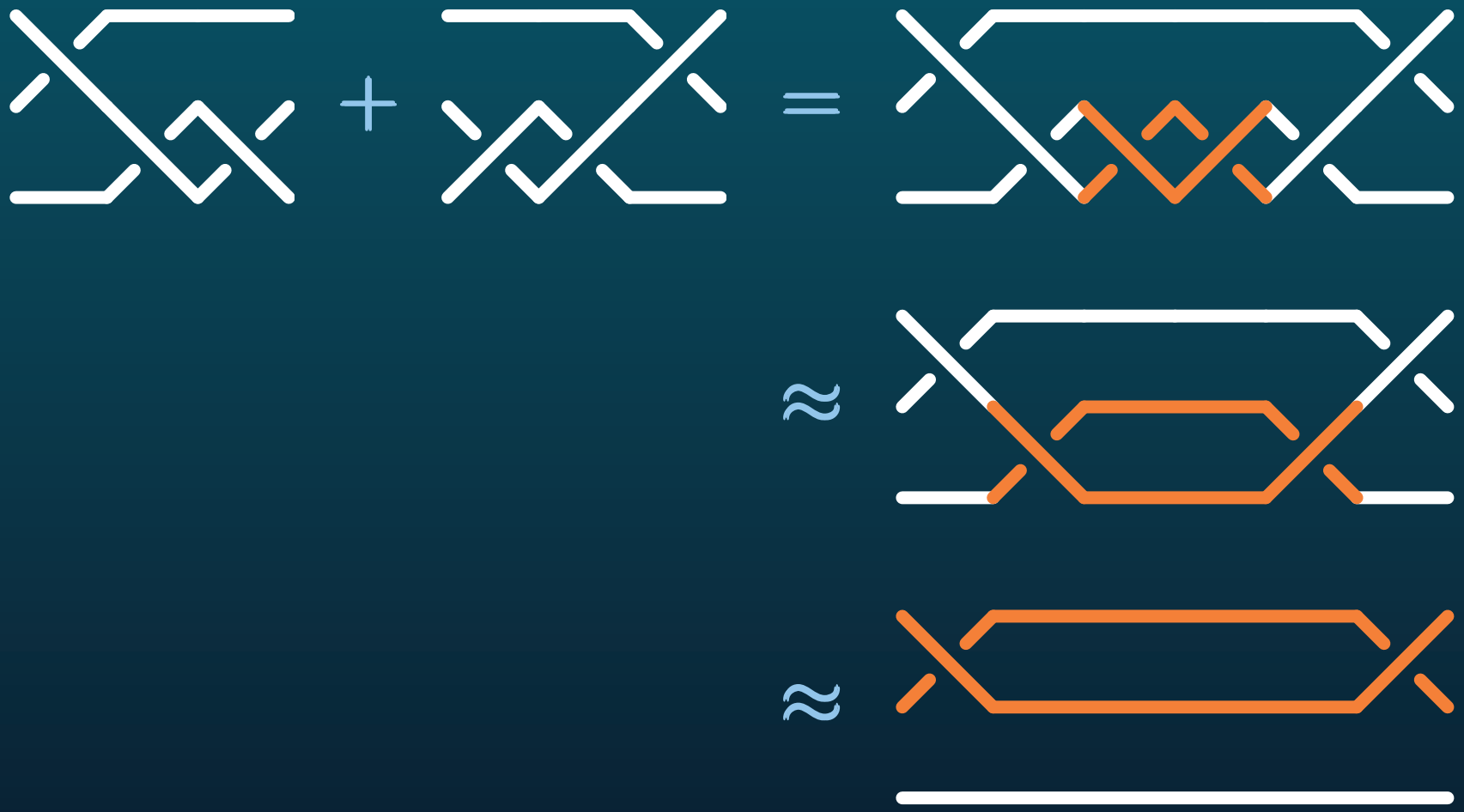


- **Exemple:**

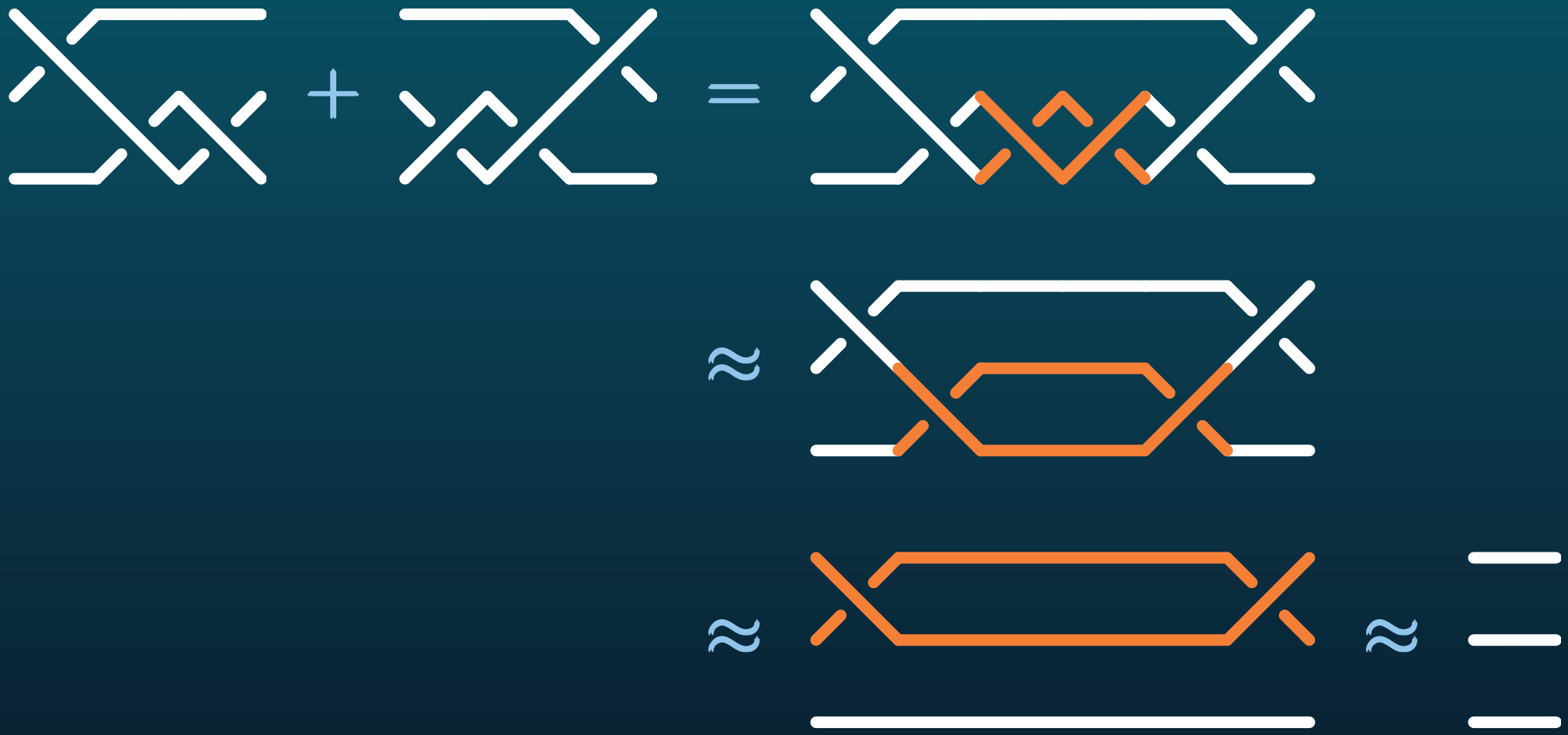
- **Exemple:**



- **Exemple:**



- **Exemple:**



⇝ **Pour chaque n, un groupe des tresses à n brins :**

le groupe B_n ($\ll$ braid group $\gg$)

⇝ **Pour chaque n, un groupe des tresses à n brins :**

le groupe B_n ($\ll$ braid group $\gg$)

- **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 ($\ll$ braid group $\gg$)

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

- **Remarque: B_n est non commutatif pour $n \geqslant 3$:**

⇝ **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 $n \geqslant 3$:**

⇝ **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 $n \geqslant 3$:**

⇝ **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 $n \geqslant 3$:**

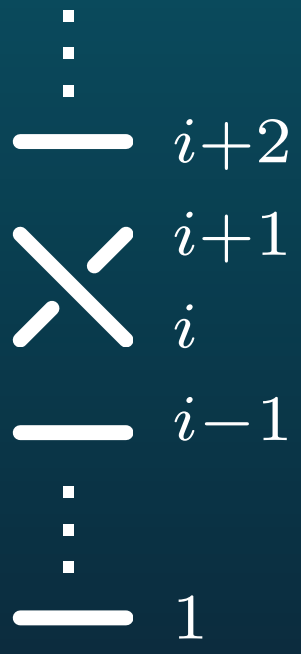
⇝ addition, zéro, opposé ⇝ multiplication, unité, inverse

⇝ addition, zéro, opposé ⇝ multiplication, unité, inverse

⇝ codage par des mots plutôt que par des nombres :

⇝ addition, zéro, opposé ⇝ multiplication, unité, inverse

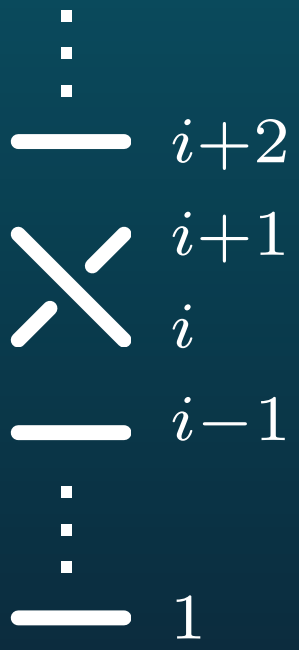
⇝ codage par des mots plutôt que par des nombres :

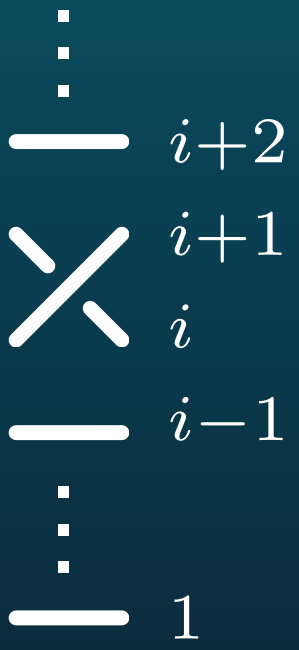
$\sigma_i :=$ 

,

⇝ addition, zéro, opposé ⇝ multiplication, unité, inverse

⇝ codage par des mots plutôt que par des nombres :

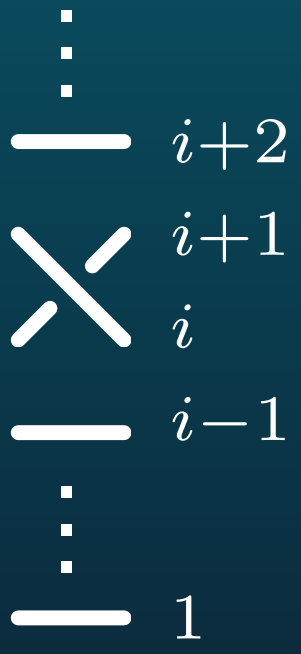
$\sigma_i :=$ 

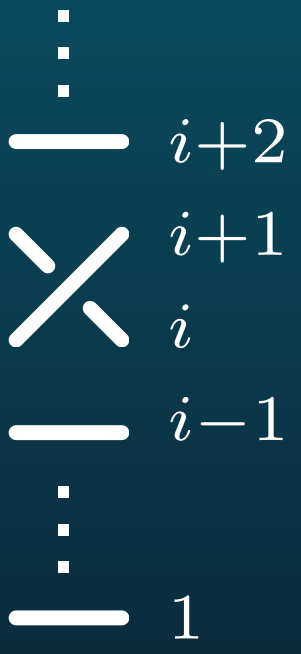
, $\sigma_i^{-1} :=$ 

.

⇝ addition, zéro, opposé ⇝ multiplication, unité, inverse

⇝ codage par des mots plutôt que par des nombres :

$\sigma_i :=$ 

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

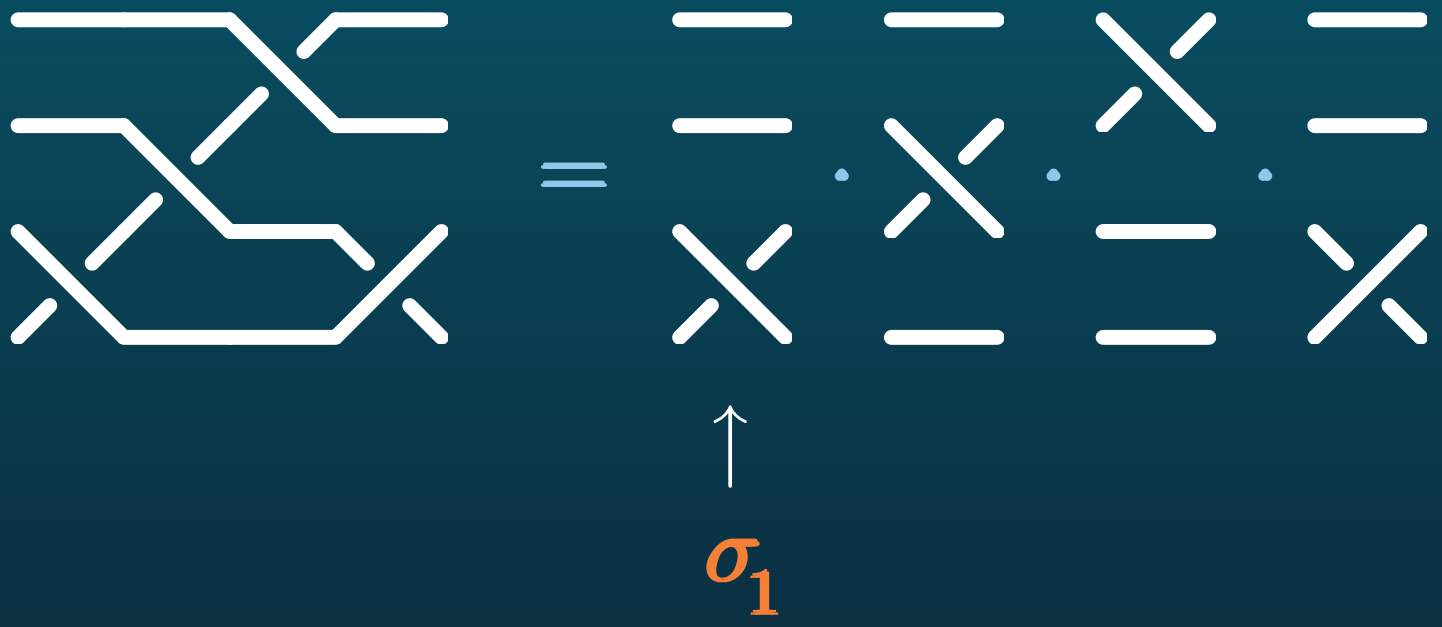
.

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

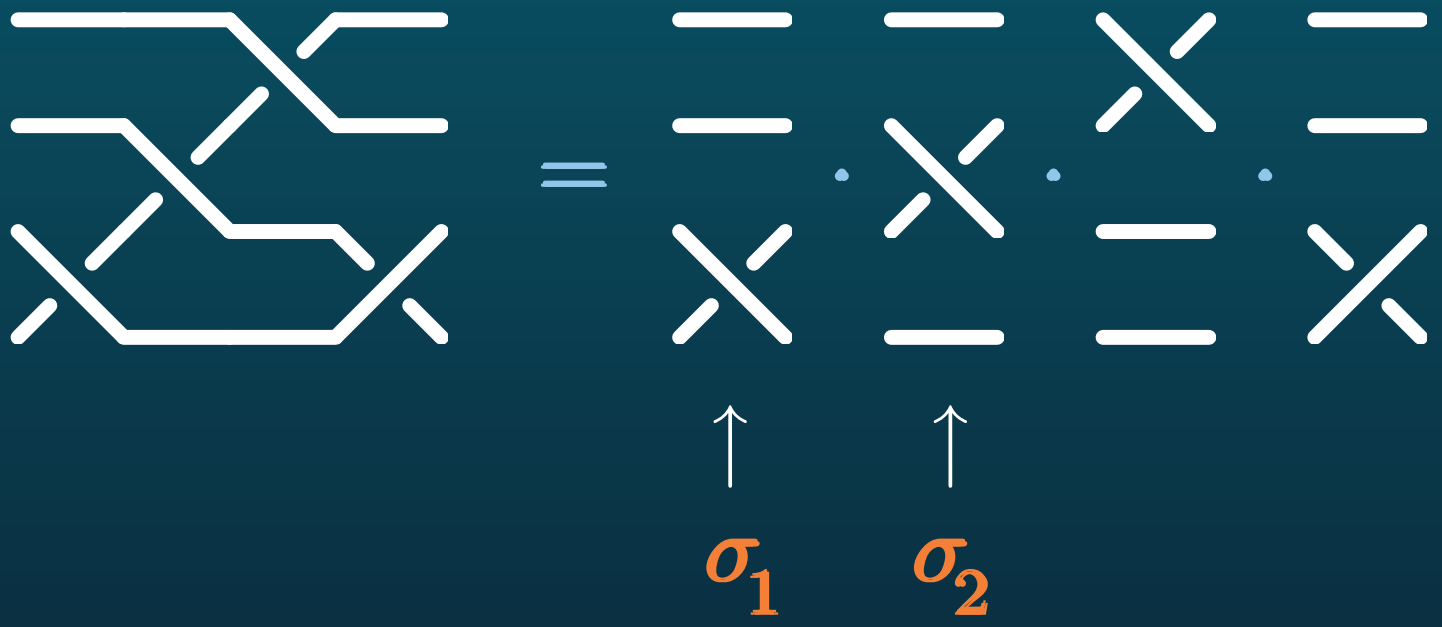
- **Un exemple de codage:**

- **Un exemple de codage:**

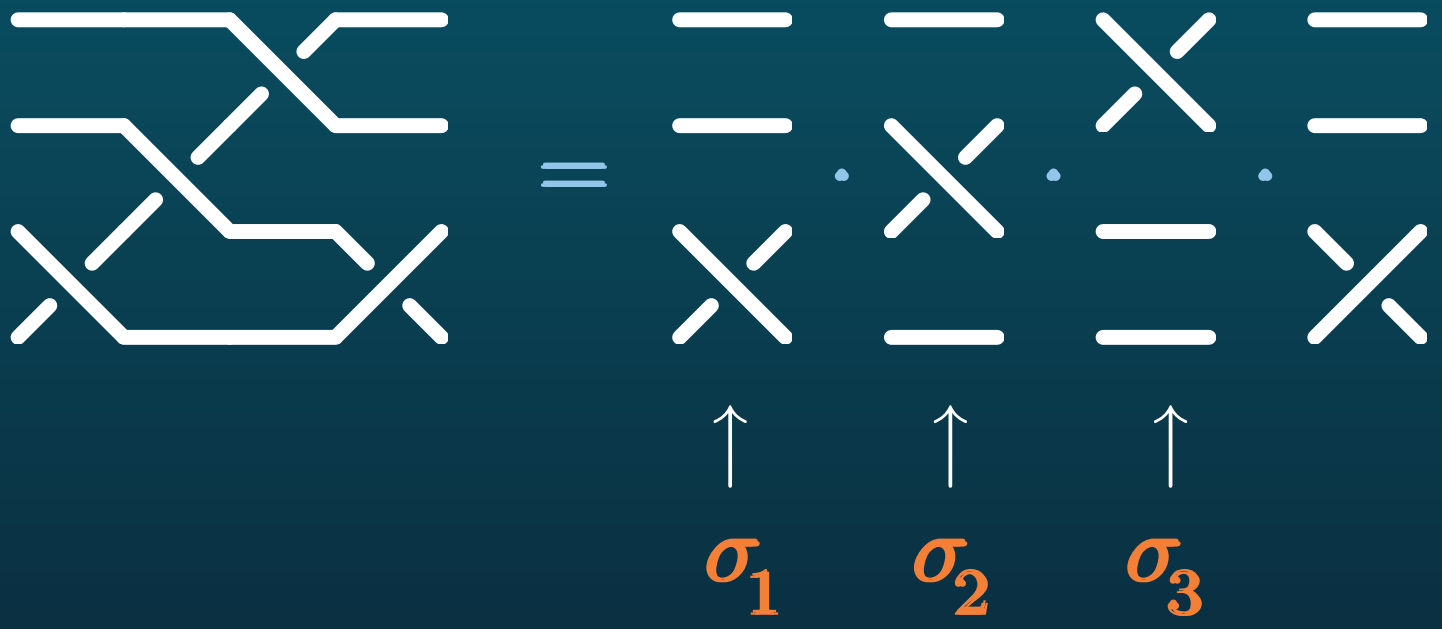
- **Un exemple de codage:**



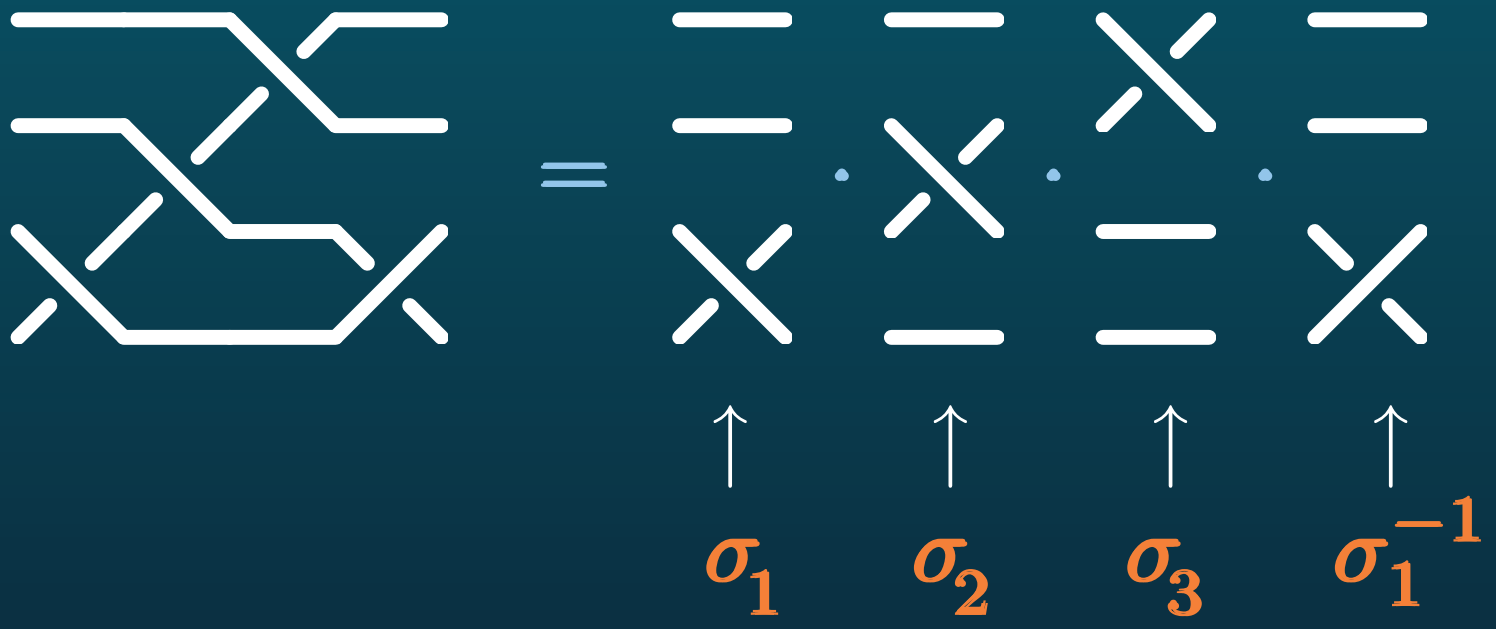
- **Un exemple de codage:**



- **Un exemple de codage:**



- **Un exemple de codage:**



- **Un exemple de codage:**

ou encore a b c A

- **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

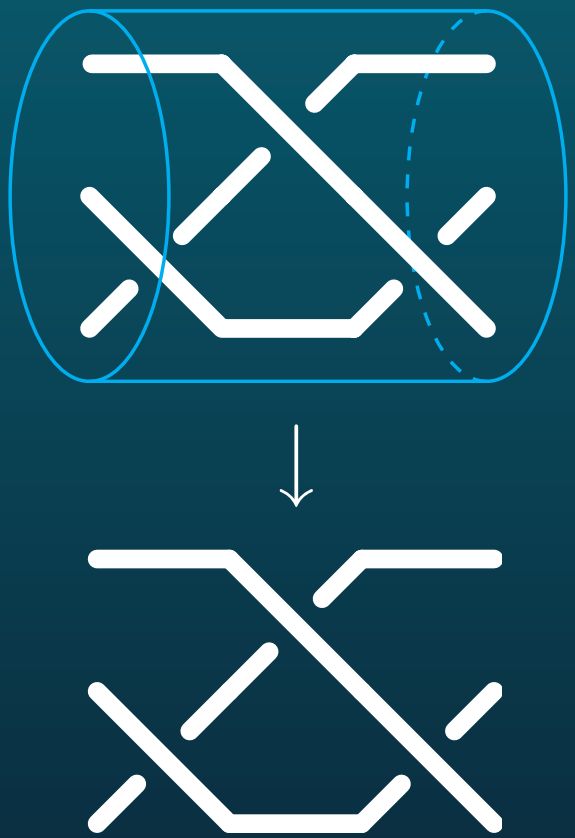
- diagramme = projection sur $\mathbb{R}^2$ d'une figure de $\mathbb{R}^3$

⇝ Définition précise de déformation : isotopie

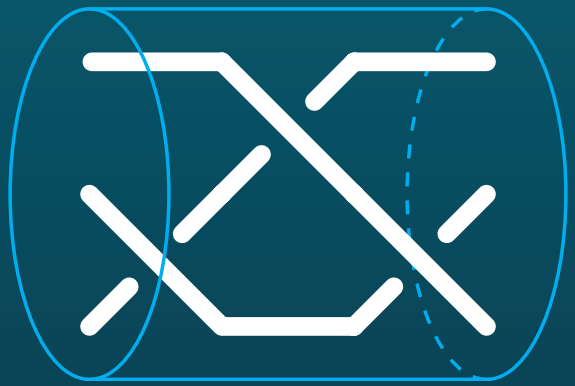
- diagramme = projection sur $\mathbb{R}^2$ d'une figure de $\mathbb{R}^3$

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

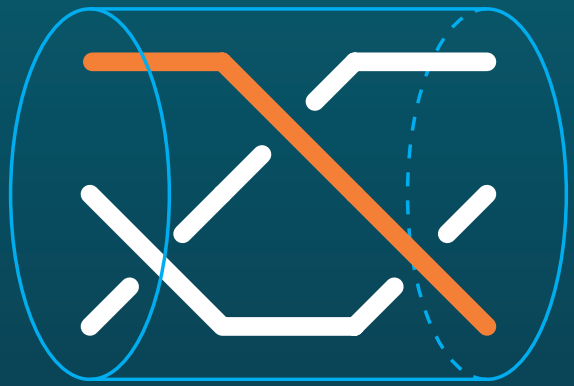
$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)



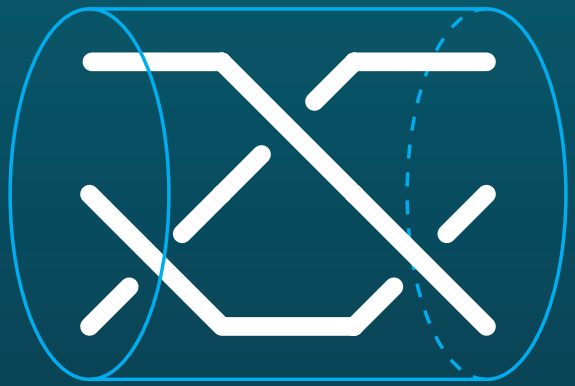
$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)



isotope à

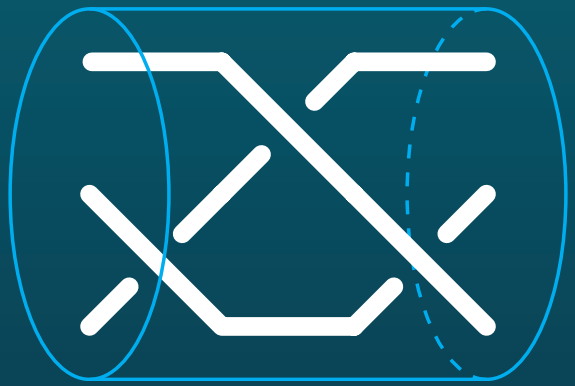


$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)



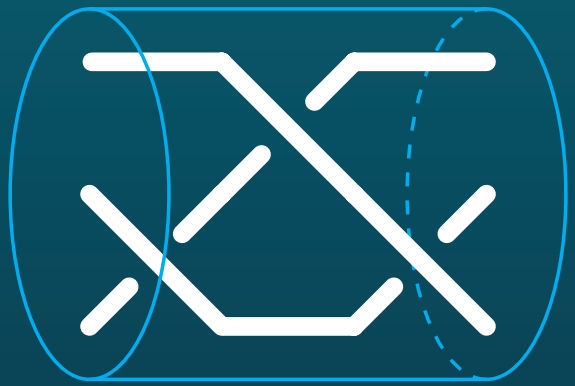
isotope à

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

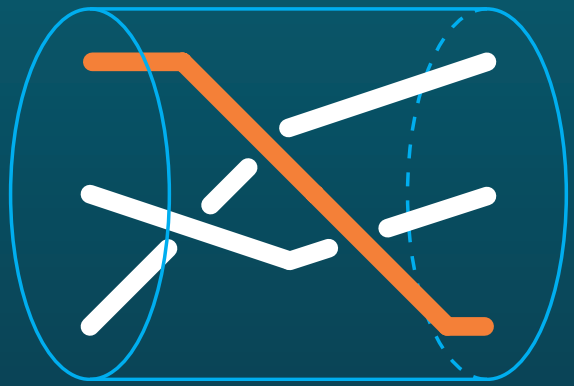


isotope à

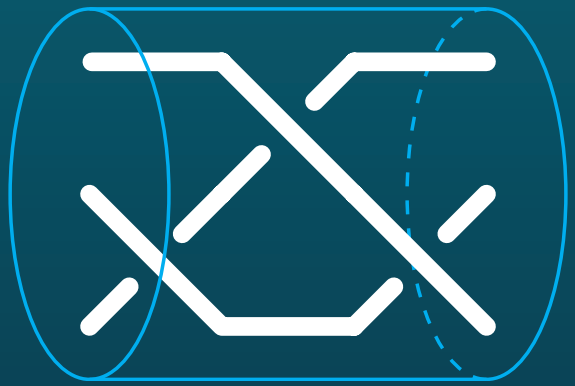
$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)



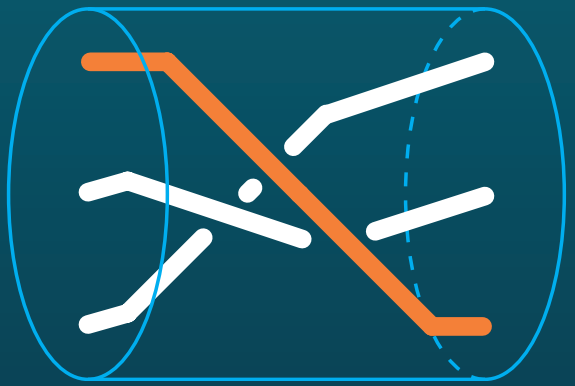
isotope à



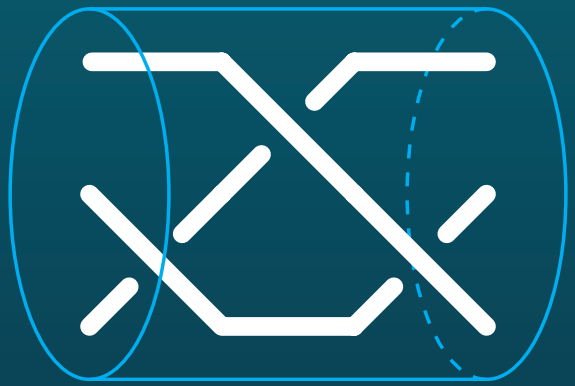
$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)



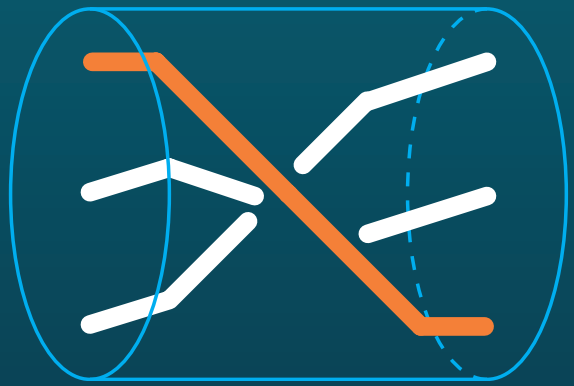
isotope à



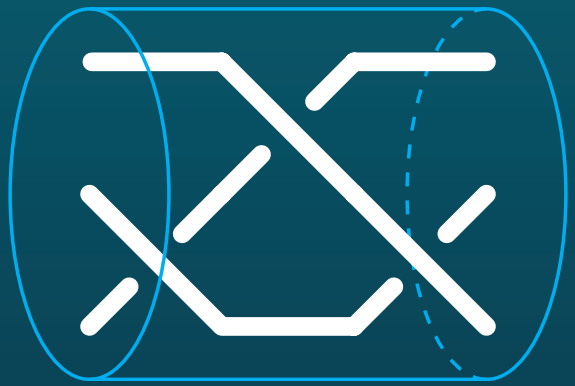
$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)



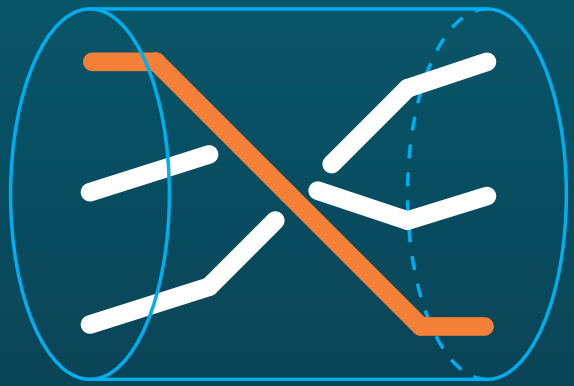
isotope à



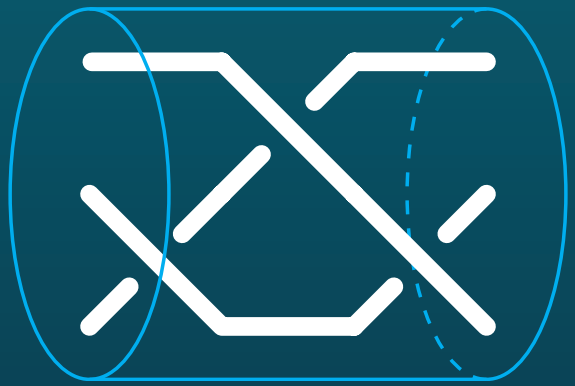
$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)



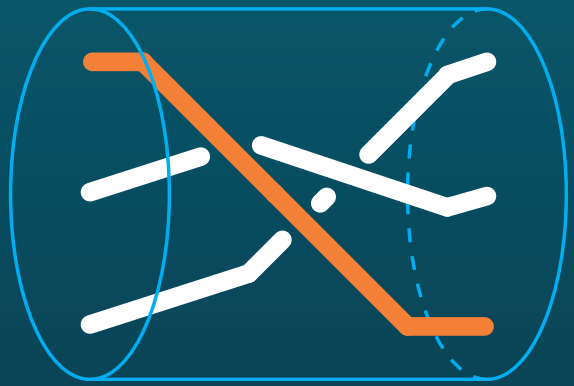
isotope à



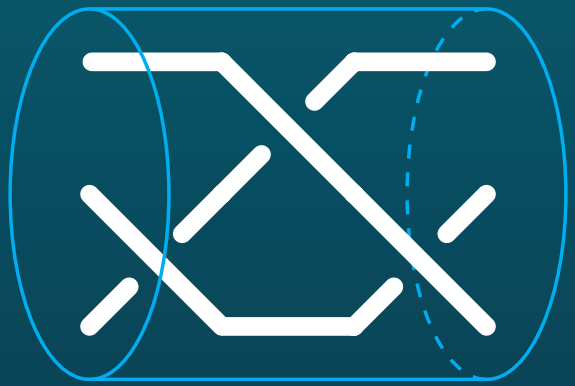
$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)



isotope à

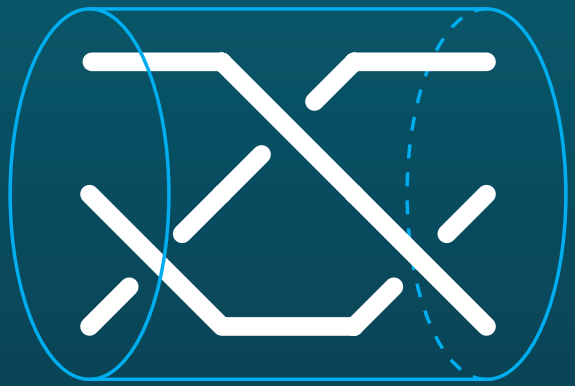


$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)



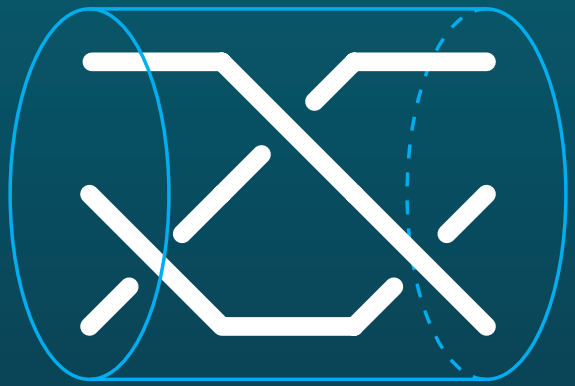
isotope à

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



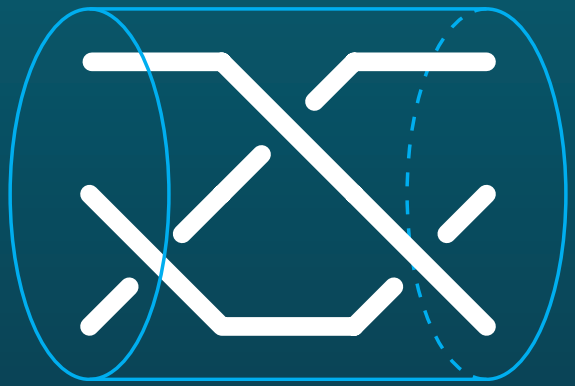
isotope à

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

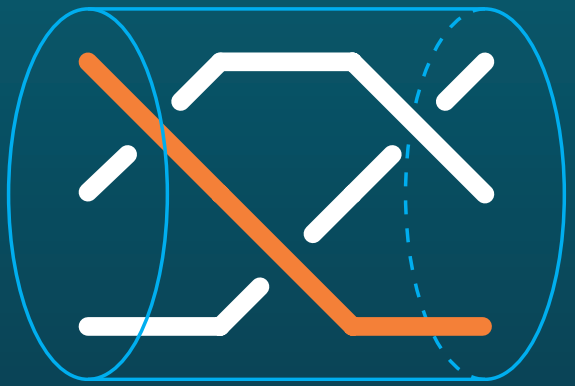


isotope à

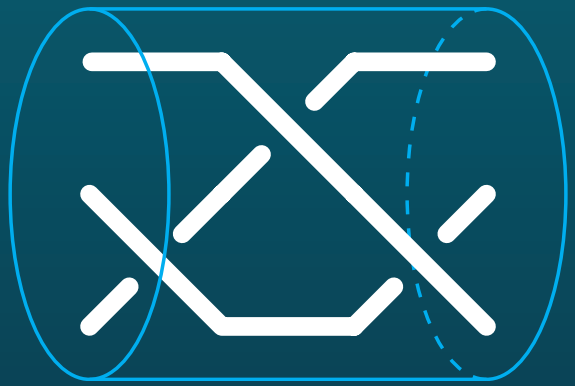
$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)



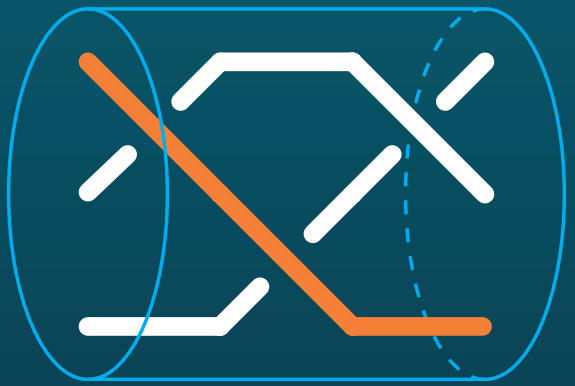
isotope à

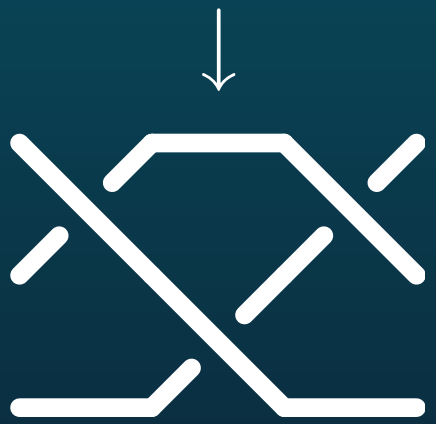


$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)



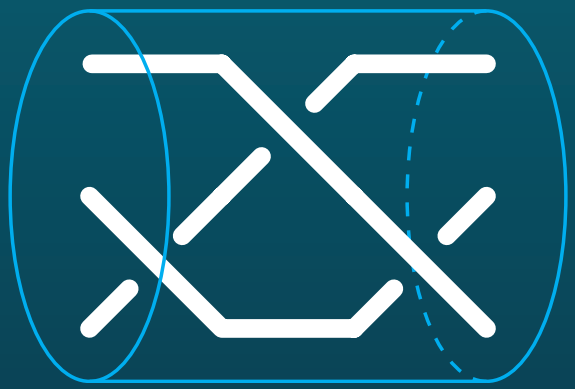
isotope à



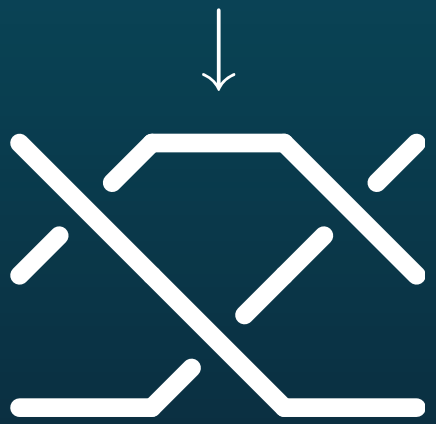


$\sigma_1 \ \sigma_2 \ \sigma_1$ (= aba)

$\sigma_2 \ \sigma_1 \ \sigma_2$ (= 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 $|i - j| \geqslant 2$,
  - $\sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j$ 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** $|i - j| \geqslant 2$,
  - $\sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j$ **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** $|i - j| \geqslant 2$,
- $\sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j$ **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** $|i-j| \geqslant 2$,
  - $\sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j$ **pour** $|i-j| = 1$.

- **Un exemple :**

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

- **Un exemple :**

isotope à

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

- **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 :**

isotope à

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

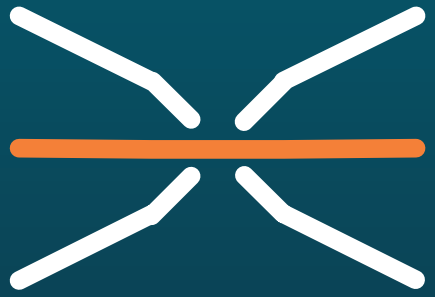
- **Un exemple :**

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

isotope à

- **Un exemple :**

isotope à



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

- **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 :**

isotope à

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

$\sigma_2^{-1} \ \sigma_1 \ \sigma_2$ (= 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 :**

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

- **Remarque préliminaire :**

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

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

- **Remarque préliminaire :**

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

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

⇝ **Le vrai problème (≪problème de mot pour B_n≫):**
reconnaître si un mot de tresse représente 1
⇔ reconnaître si une tresse est vraiment tressée.

- **Remarque préliminaire :**

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

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

⇝ **Le vrai problème (≪problème de mot pour B_n≫):**
reconnaître si un mot de tresse représente 1
⇔ reconnaître si une tresse est vraiment tressée.

⇝ **algorithme de démêlage des tresses ?**

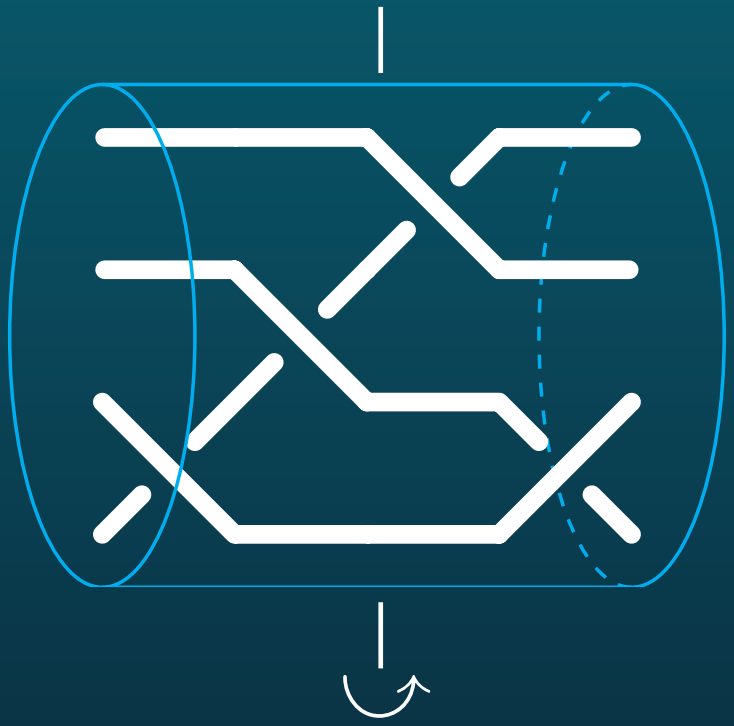
- **Un théorème : Un mot de tresse qui contient au moins un σ_1 et pas de σ_1^{-1} est non trivial (= ne représente pas 1).**

- **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).**

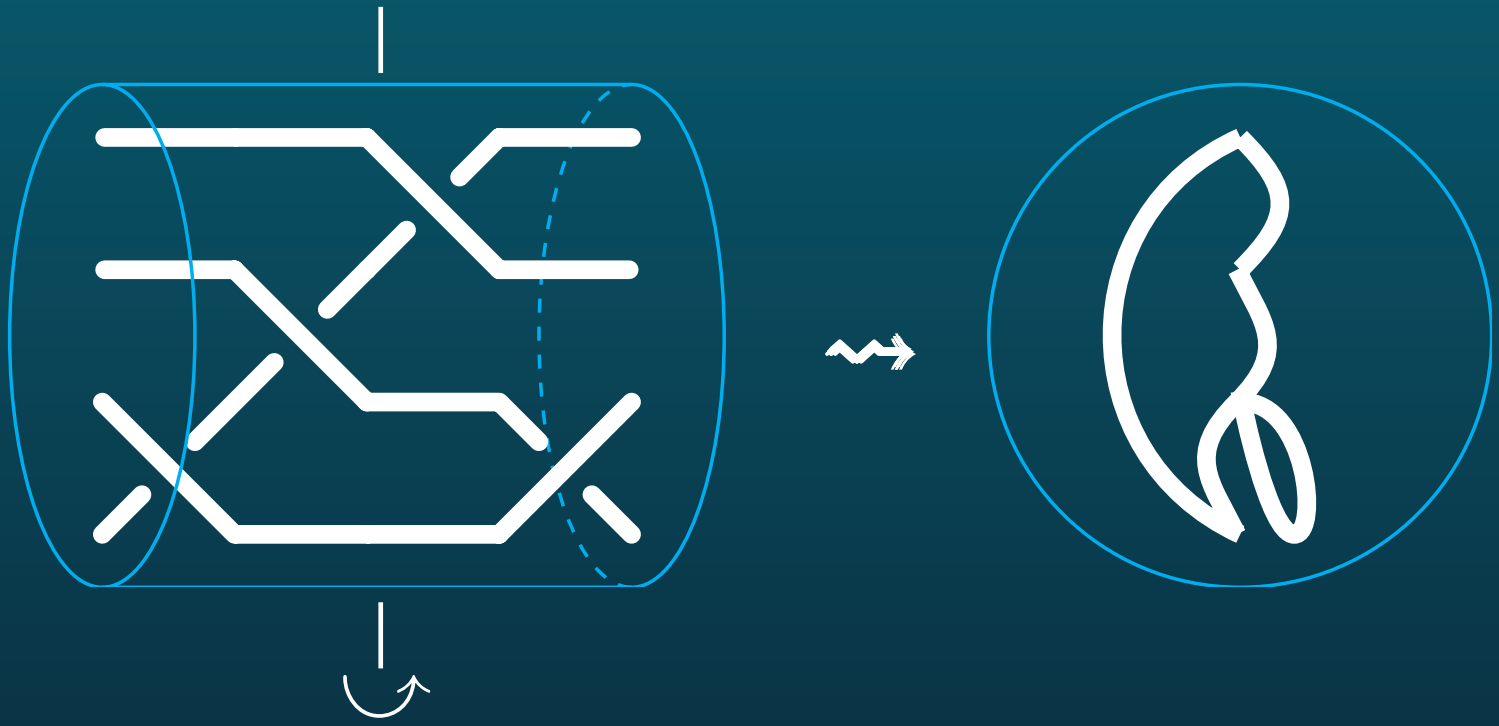
⇝ Démonstration?

⇝ Démonstration (Dynnikov)

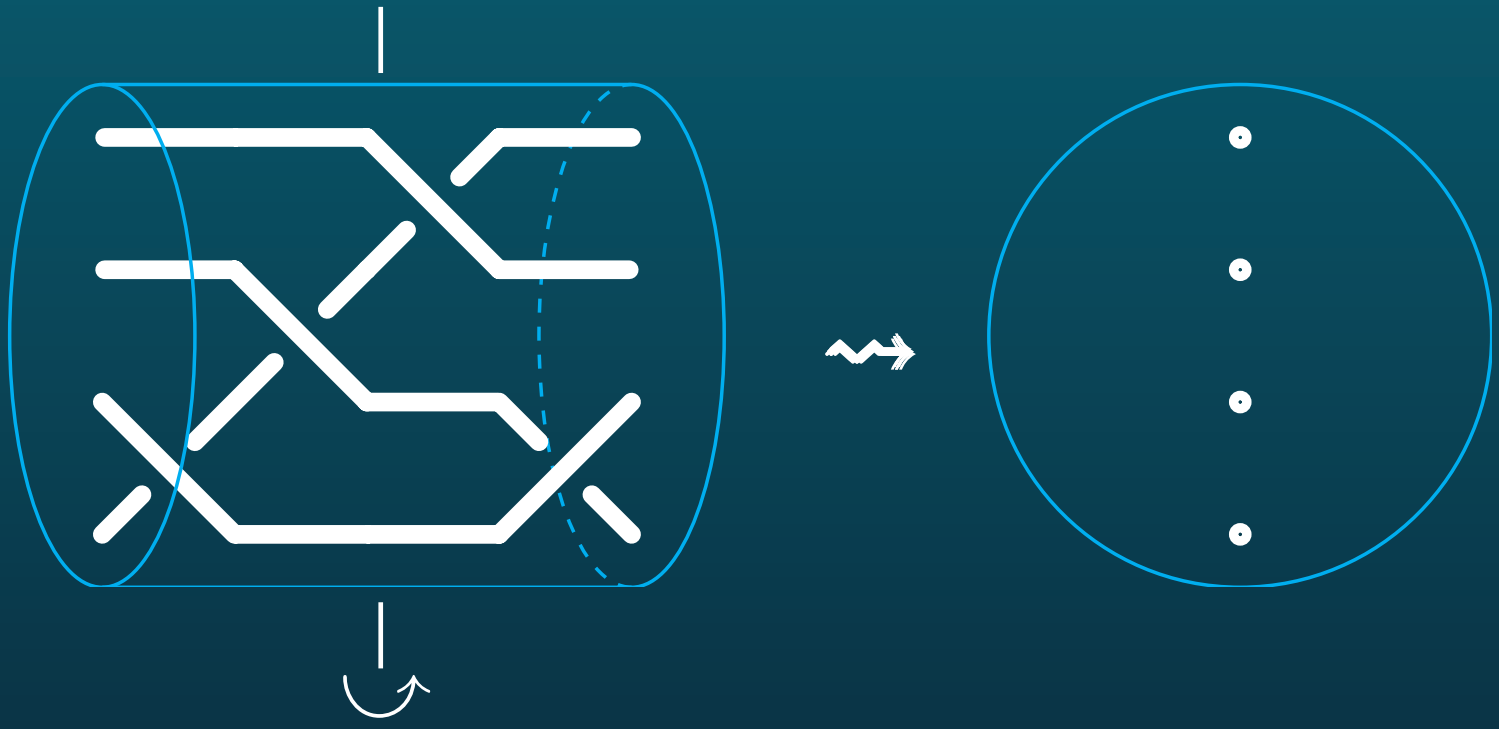
Démonstration (Dynnikov)



⇝ Démonstration (Dynnikov)

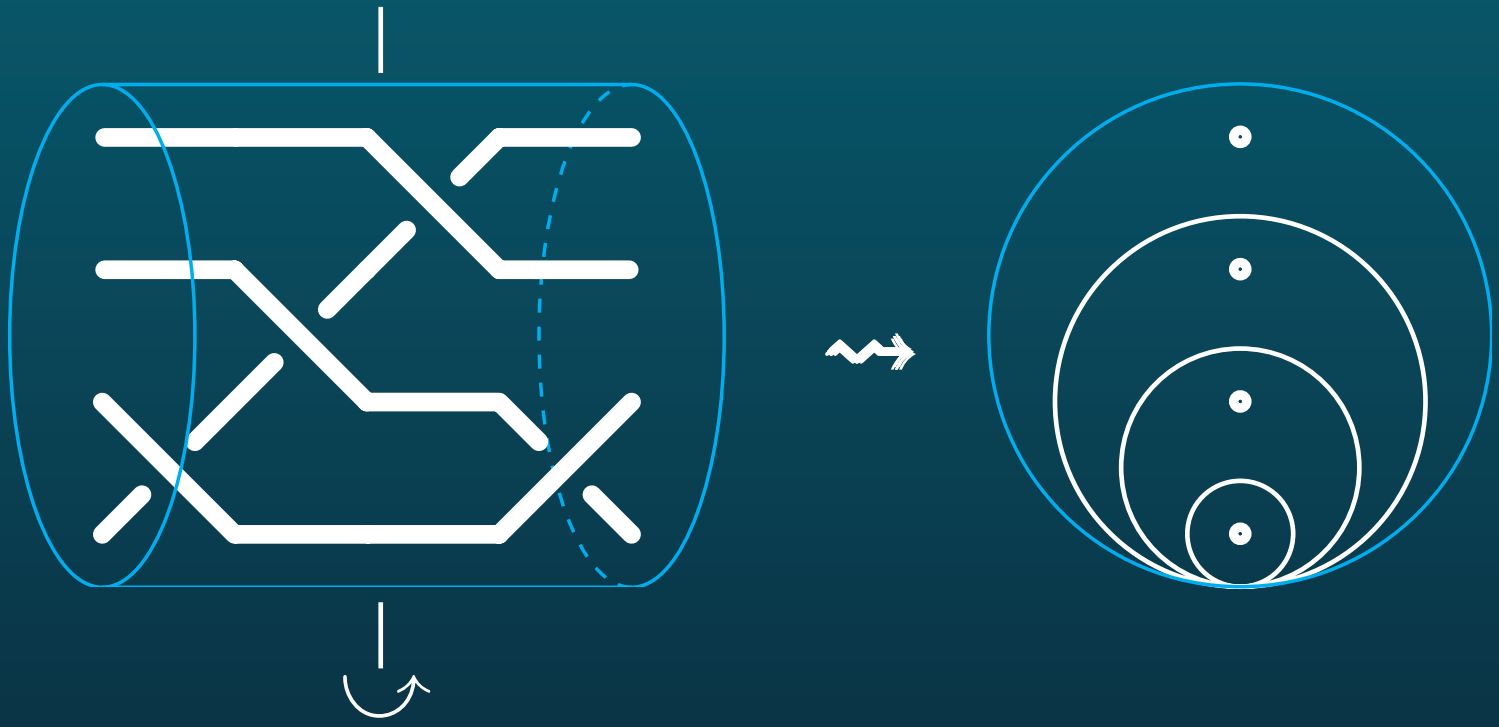


⇝ Démonstration (Dynnikov)



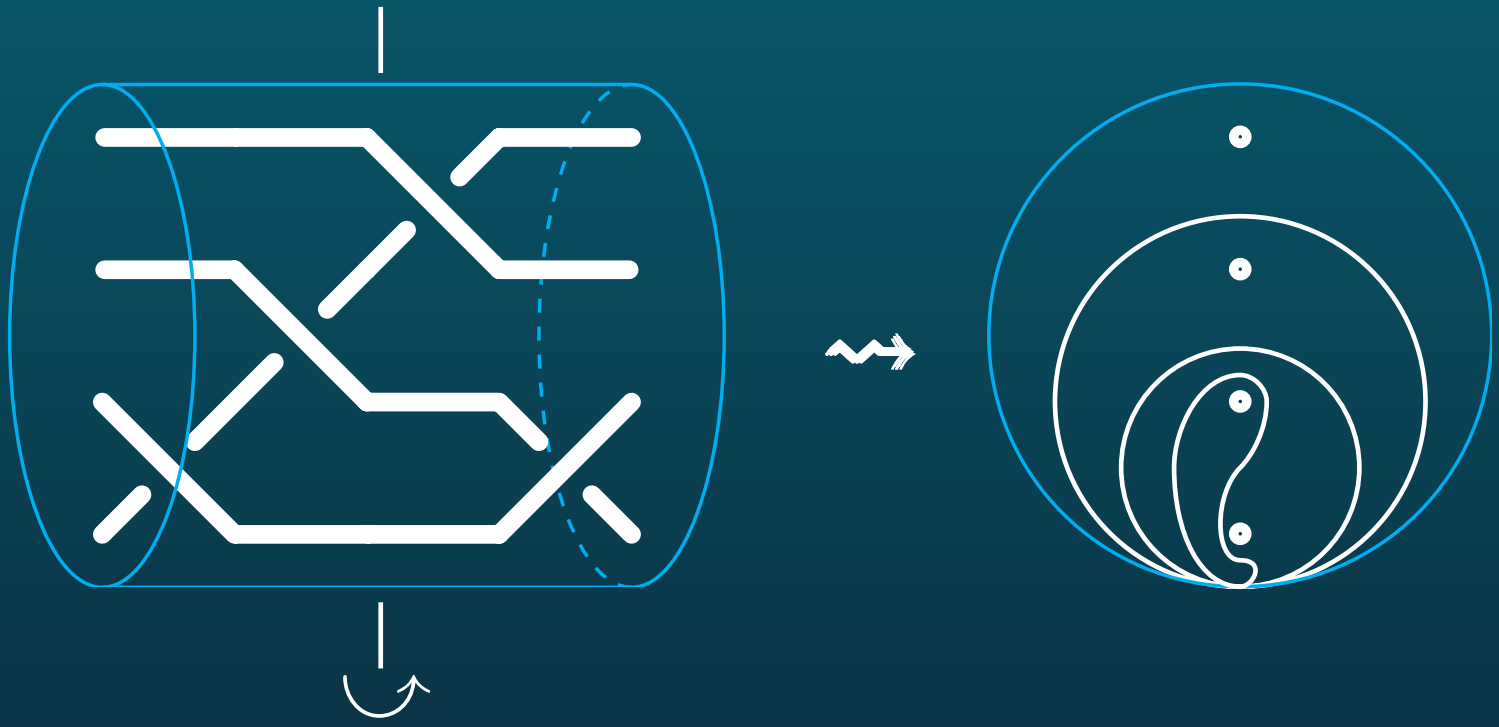
- tresse = déformation d’un disque troué ;

⇝ Démonstration (Dynnikov)



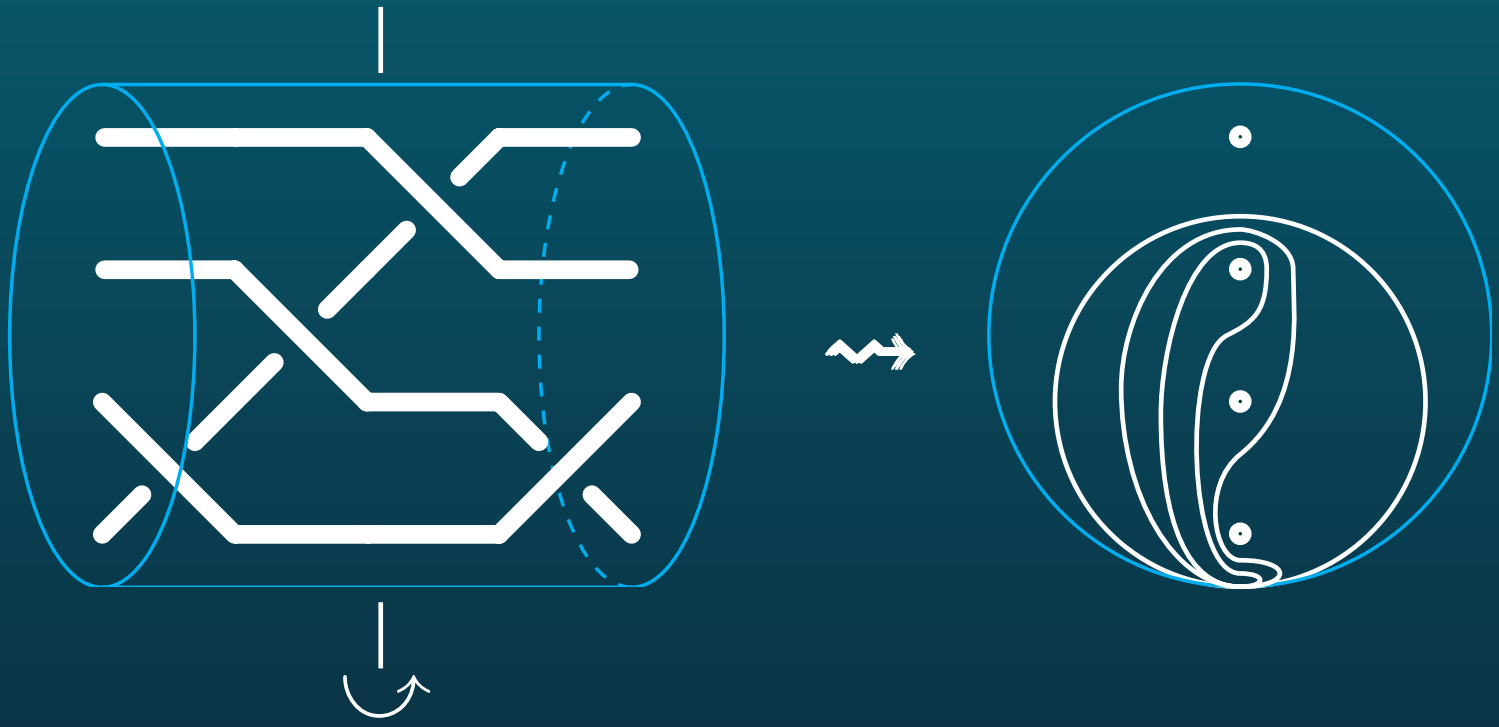
- tresse = déformation d’un disque troué ;
  - faire agir sur une lamination du disque troué ;

⇝ Démonstration (Dynnikov)



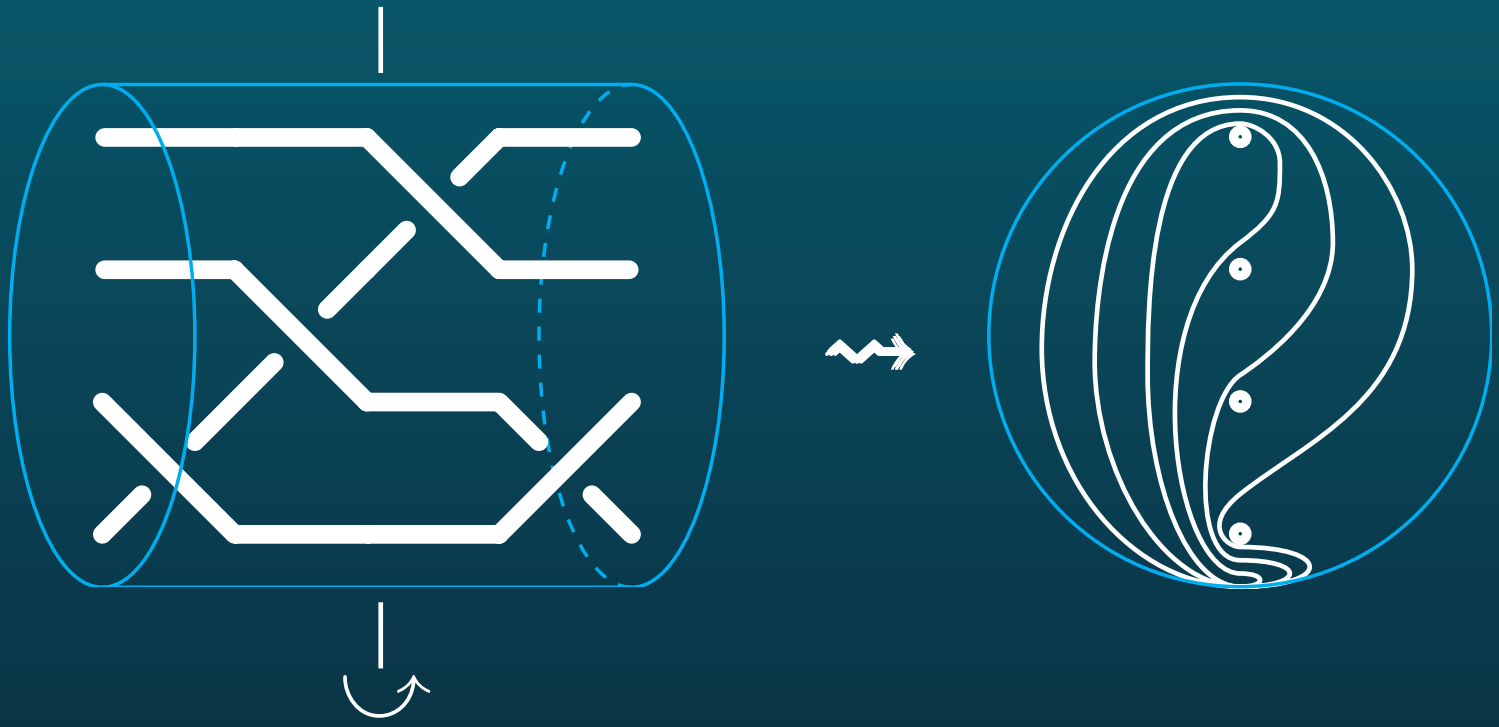
- tresse = déformation d’un disque troué ;
  - faire agir sur une lamination du disque troué ;

⇝ Démonstration (Dynnikov)



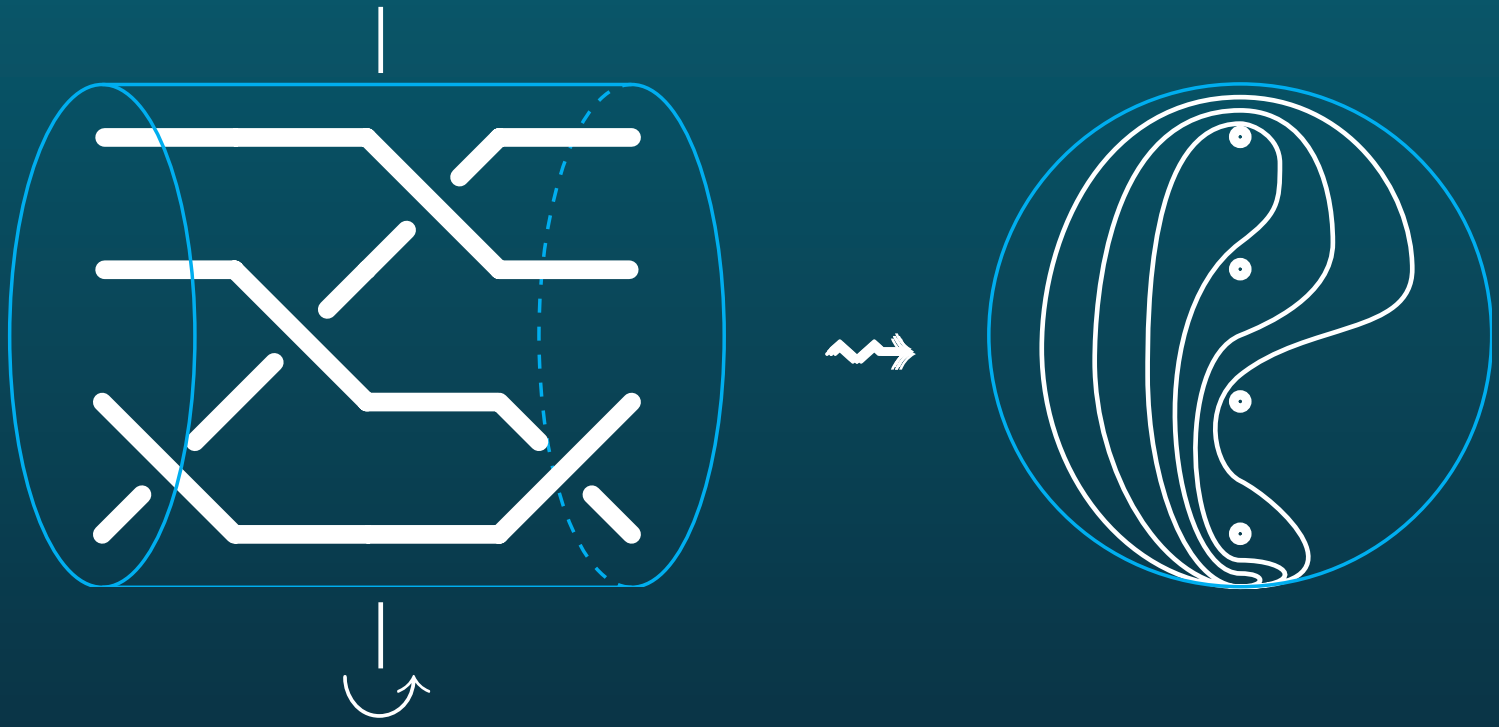
- tresse = déformation d’un disque troué ;
  - faire agir sur une lamination du disque troué ;

⇝ Démonstration (Dynnikov)



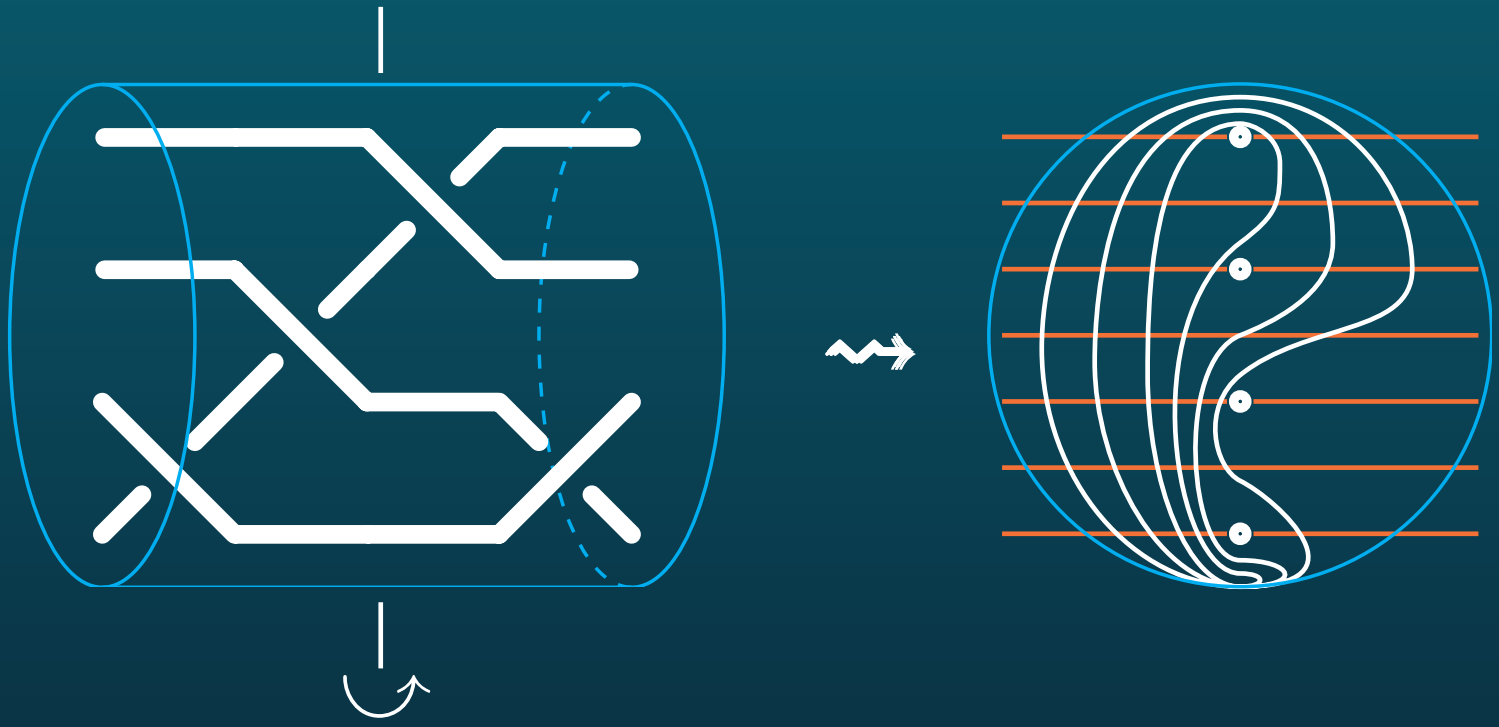
- tresse = déformation d’un disque troué ;
  - faire agir sur une lamination du disque troué ;

⇝ Démonstration (Dynnikov)



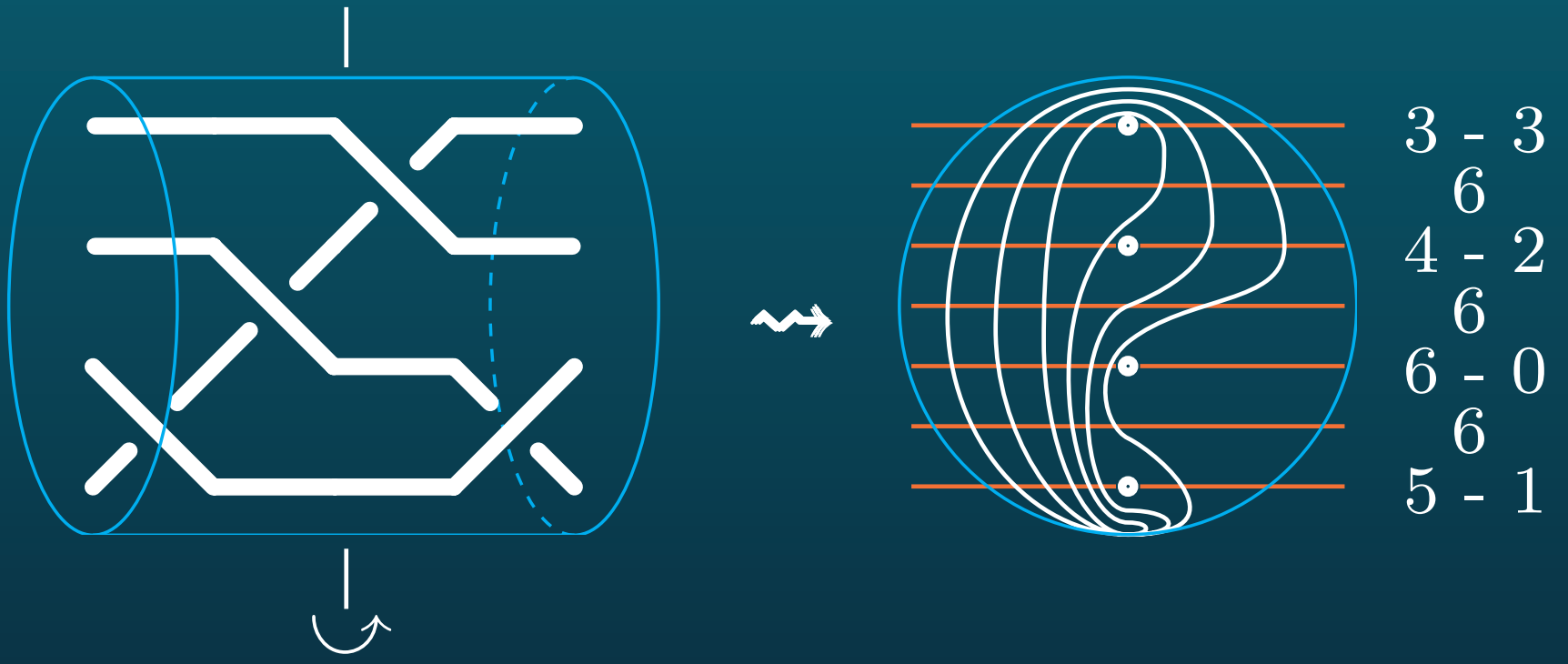
- tresse = déformation d’un disque troué ;
  - faire agir sur une lamination du disque troué ;

⇝ Démonstration (Dynnikov)



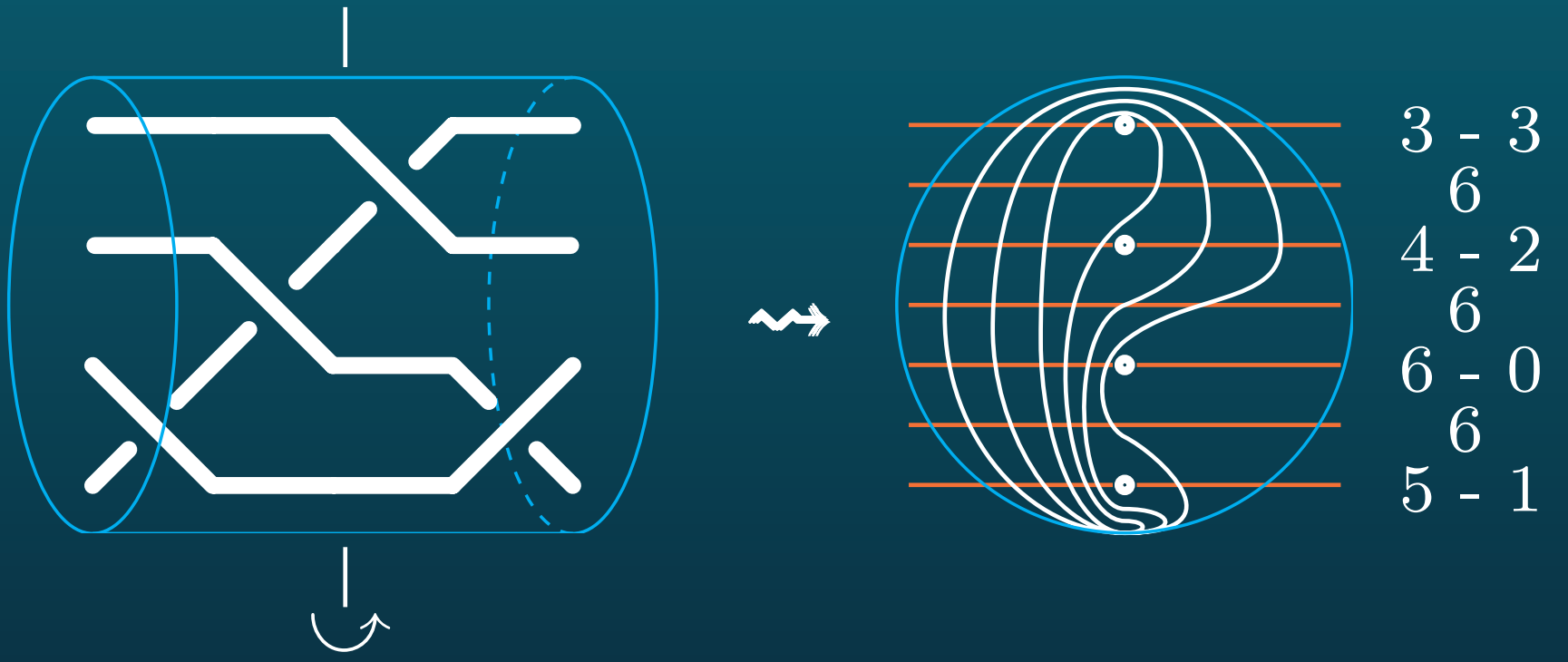
- tresse = déformation d'un disque troué ;
  - faire agir sur une lamination du disque troué ;
    - compter les intersections avec une grille ;

⇝ Démonstration (Dynnikov)



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

⇝ Démonstration (Dynnikov)



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

$$a'_1 = a_1 + ((a_2 - a_1)^+ + b_1)^+ \text{ pour } x' = x\sigma_1$$
$$a'_1 = a_1 \text{ pour } x' = x\sigma_i^{\pm 1} \text{ avec } i \geqslant 2.$$

- **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 $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:**

- **Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:**

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

$\rightsquigarrow$ une poignée

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

⇝ une poignée

⇝ comment s'en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

⇝ une poignée

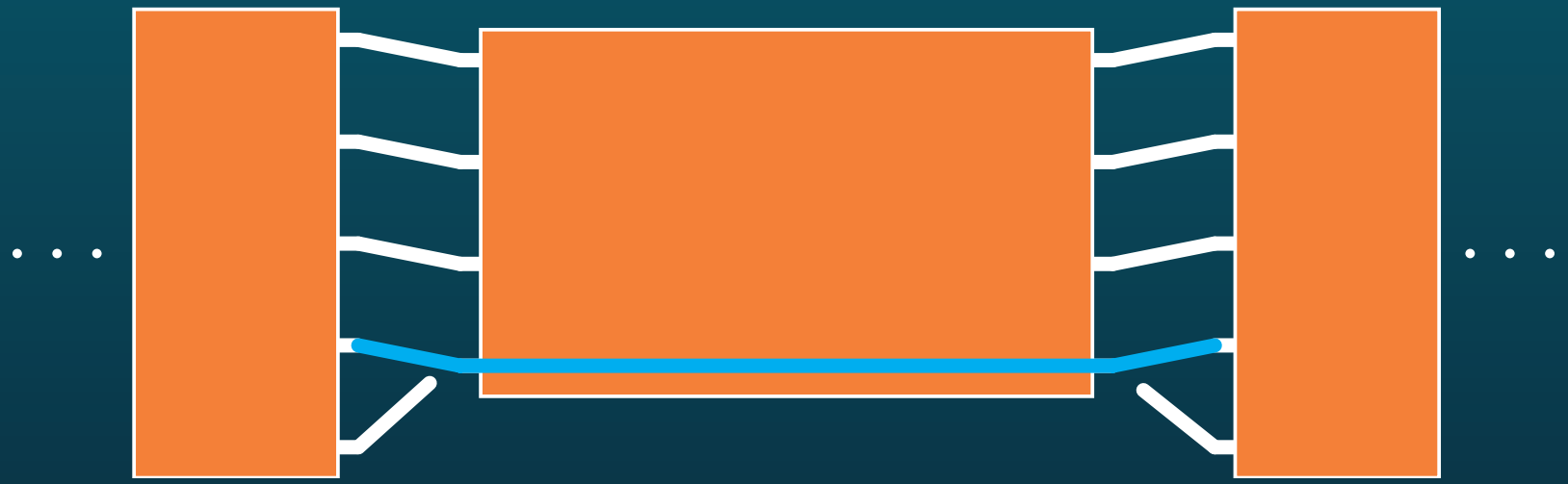
⇝ comment s’en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

⇝ une poignée

⇝ comment s’en débarrasser ?

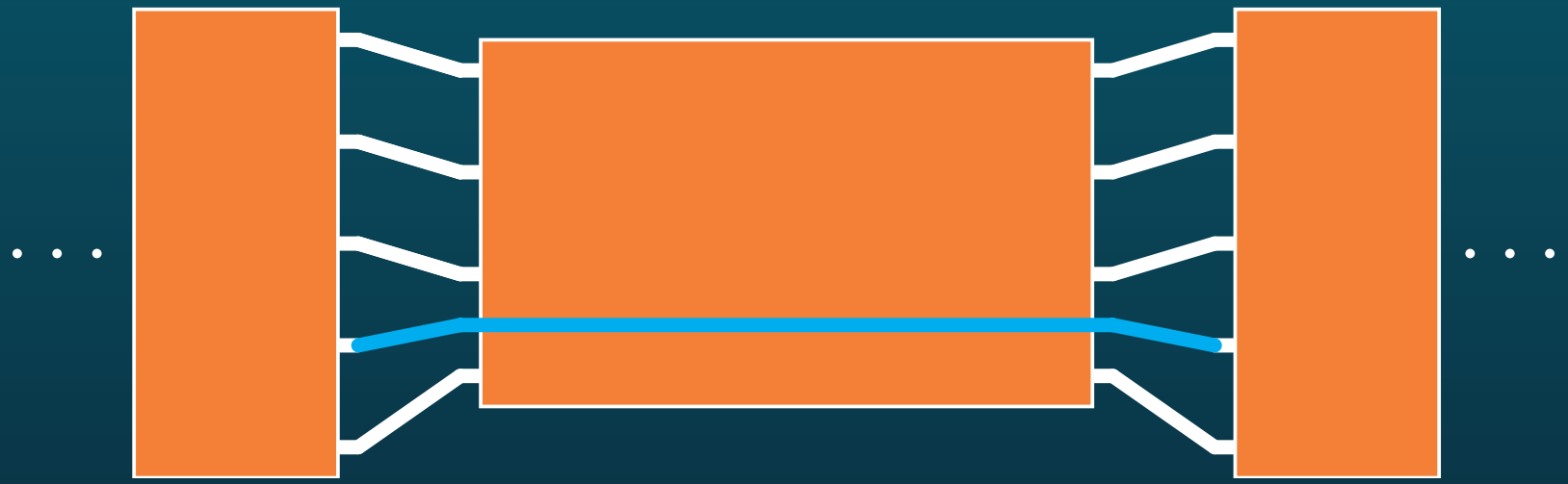
- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:



⇝ une poignée

⇝ comment s’en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:



⇝ une poignée

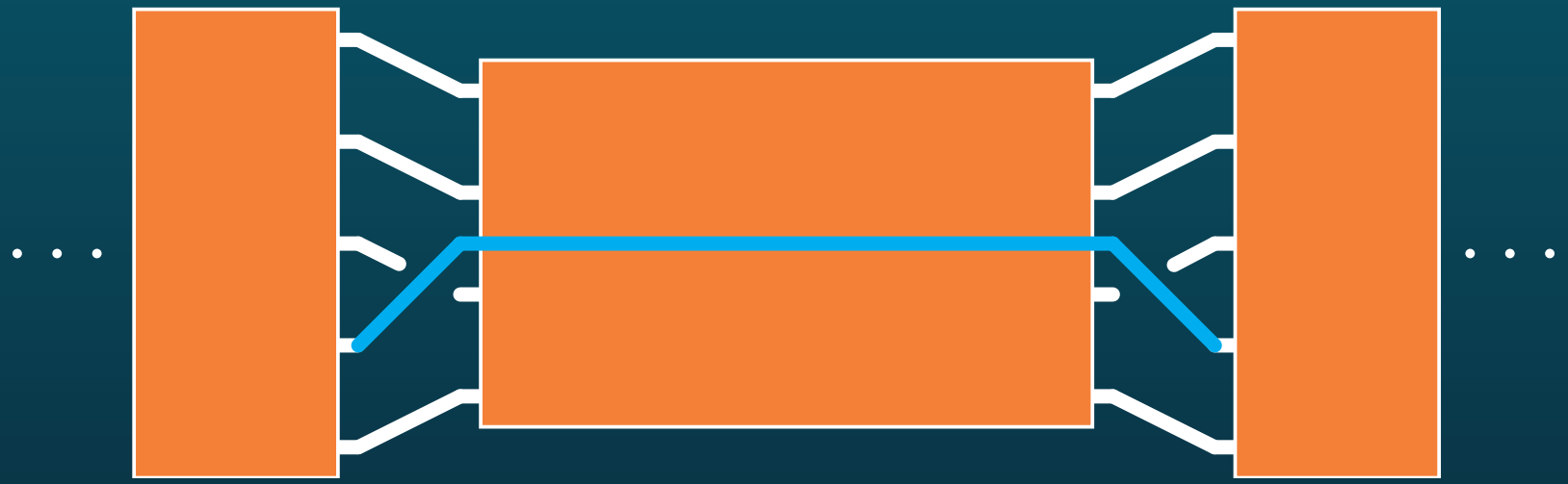
⇝ comment s’en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

⇝ une poignée

⇝ comment s’en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:



⇝ une poignée

⇝ comment s’en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

⇝ une poignée

⇝ comment s’en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

⇝ une poignée

⇝ comment s’en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

⇝ une poignée

⇝ comment s’en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

⇝ une poignée

⇝ comment s’en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

⇝ une poignée

⇝ comment s’en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

⇝ une poignée

⇝ comment s’en débarrasser ?

- Le cas $\ll \sigma_1$ et $\sigma_1^{-1} \gg$:

⇝ une poignée

⇝ comment s'en débarrasser ?

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

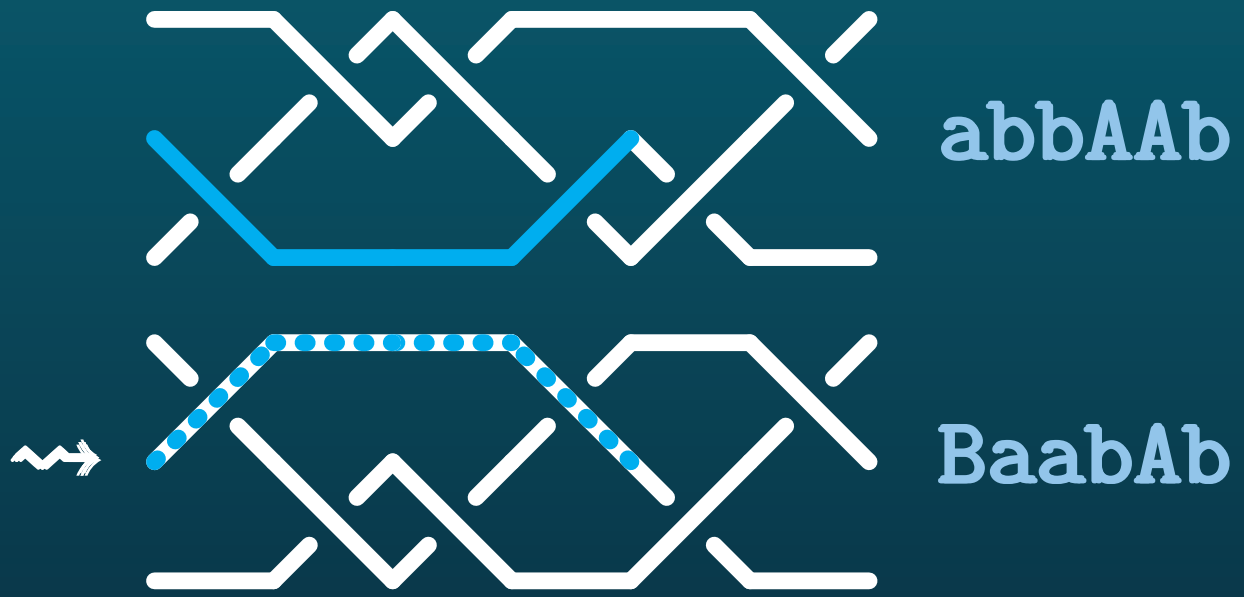
- **Un exemple :**

`abbAAb`

- **Un exemple :**

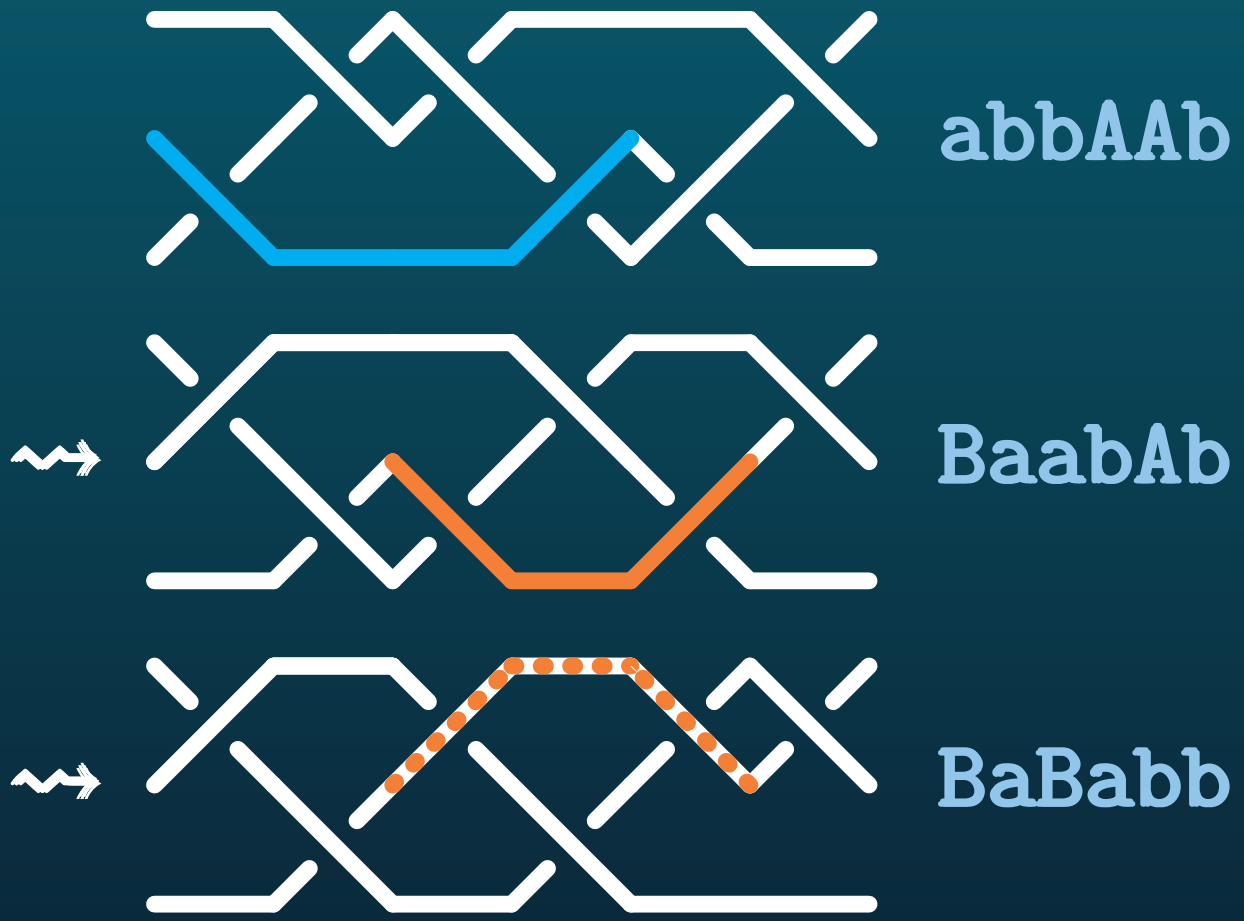
`abbAAb`

- **Un exemple :**

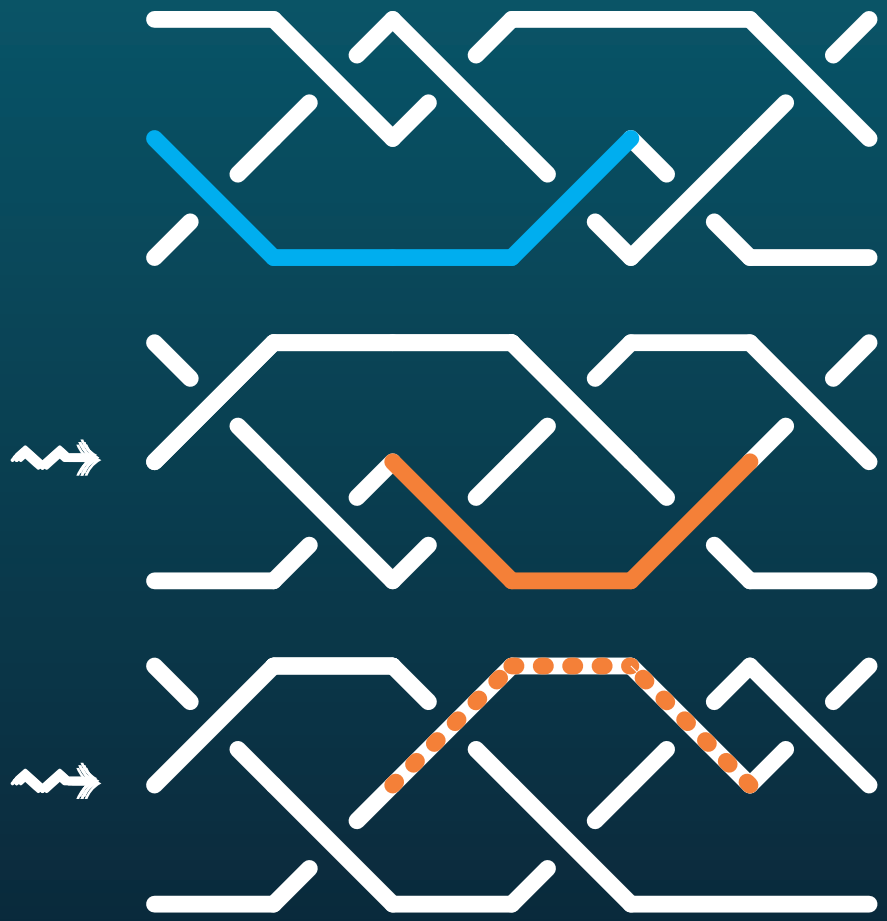


- **Un exemple :**

- **Un exemple :**



- **Un exemple :**



`abbAAb`

$\rightsquigarrow$

`BaabAb`

$\rightsquigarrow$

`BaBabb`

$\begin{cases} \text{deux } \mathtt{a} & (= \sigma_1) \\ \text{zéro } \mathtt{A} & (= \sigma_1^{-1}) \end{cases}$

- **Un exemple :**

`abbAAb`

⇝

`BaabAb`

⇝

`BaBabb`

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

⇝ tresse non triviale

- **Un exemple :**

`abbAAb`

`BaabAb`

`BaBabb`

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

⇝ tresse non triviale

- **Juste chanceux, ou marche toujours ?**

... on ne sait pas (!)

... on ne sait pas (!)

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

⇝ solution locale

... on ne sait pas (!)

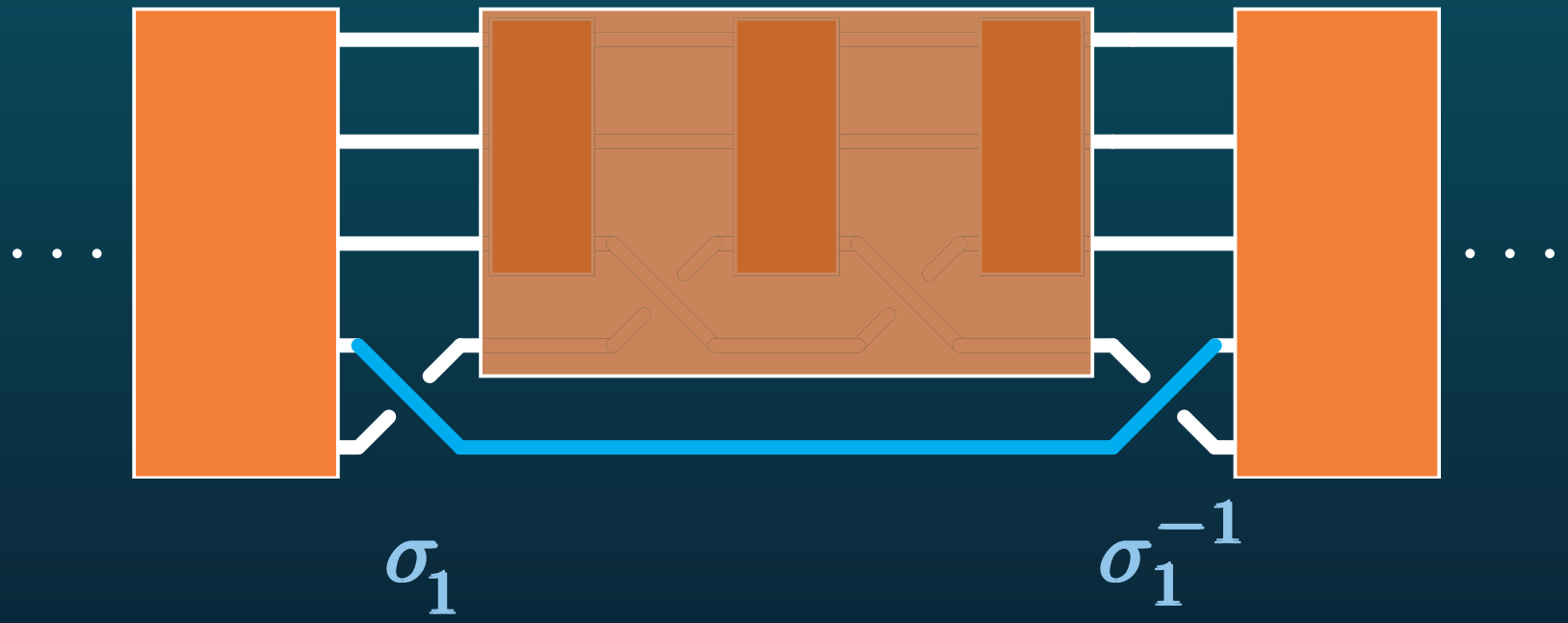
De toute façon, on peut faire (beaucoup) mieux :

⇝ solution locale

... on ne sait pas (!)

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

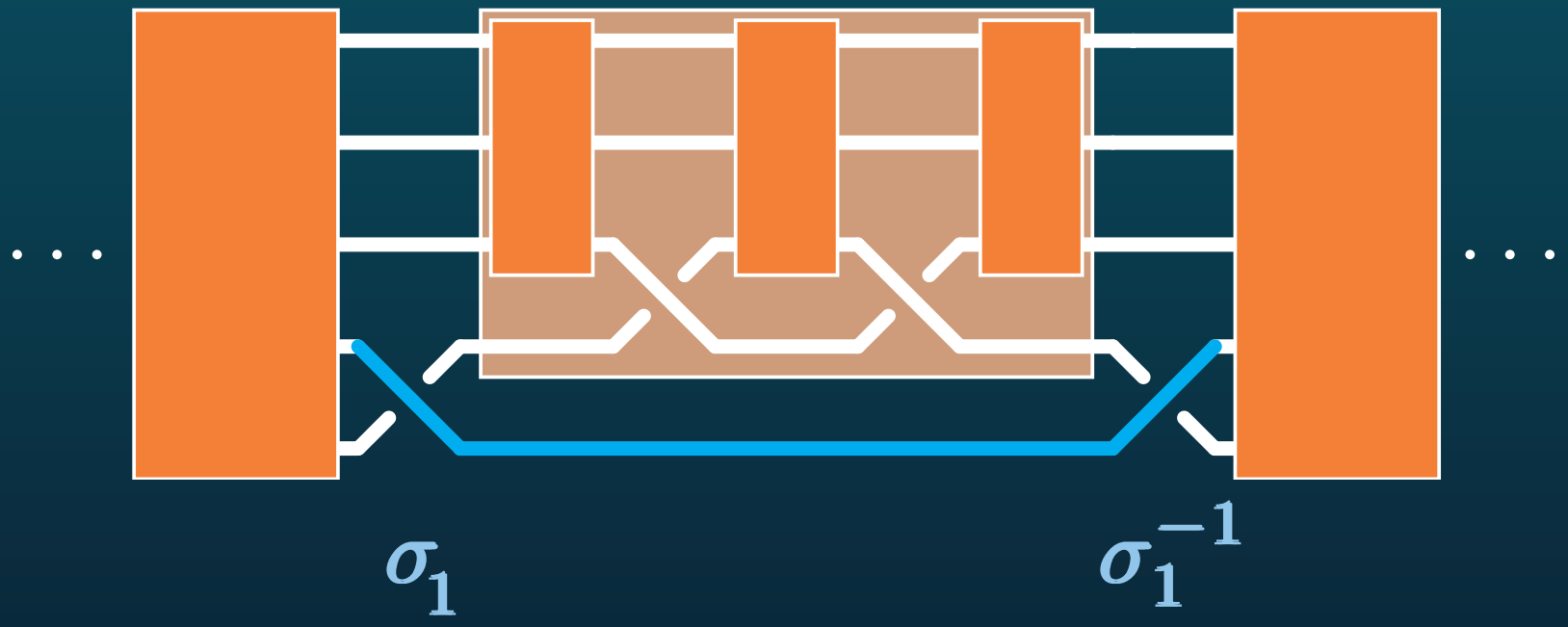
⇝ solution locale



... on ne sait pas (!)

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

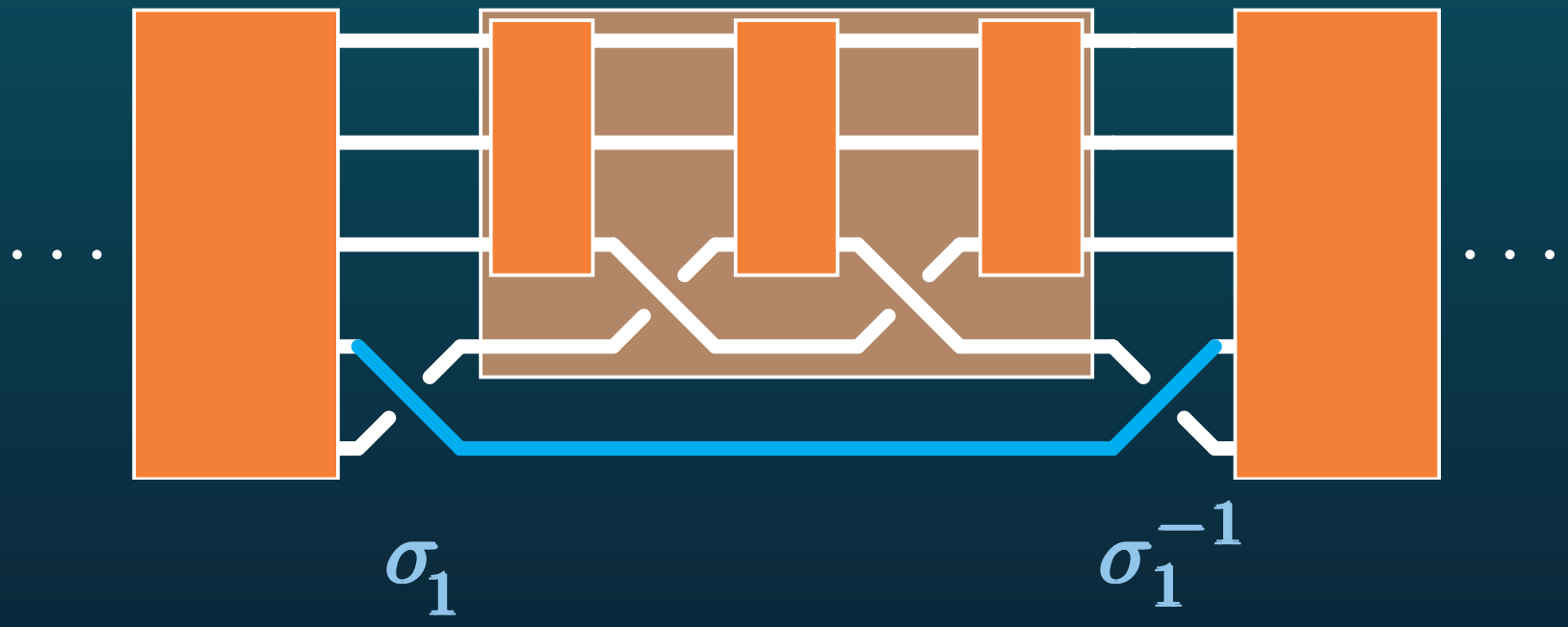
⇝ solution locale



... on ne sait pas (!)

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

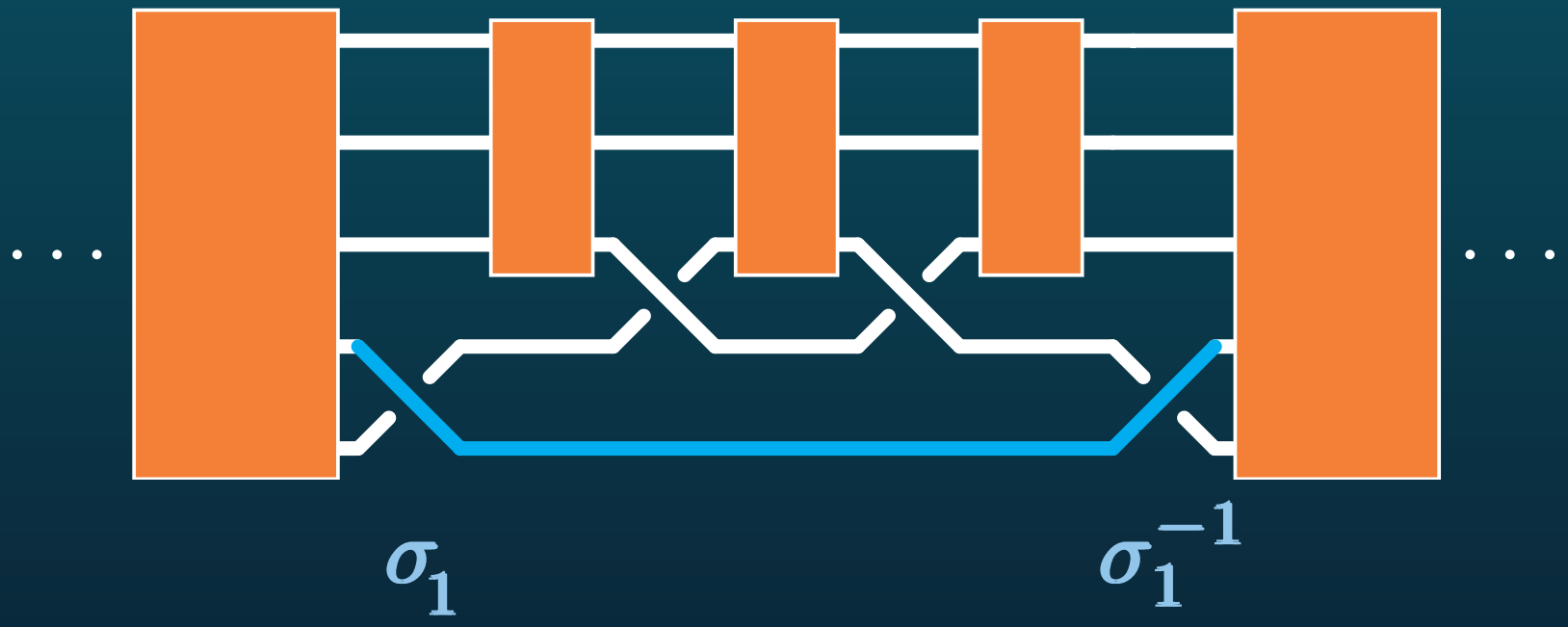
⇝ solution locale



... on ne sait pas (!)

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

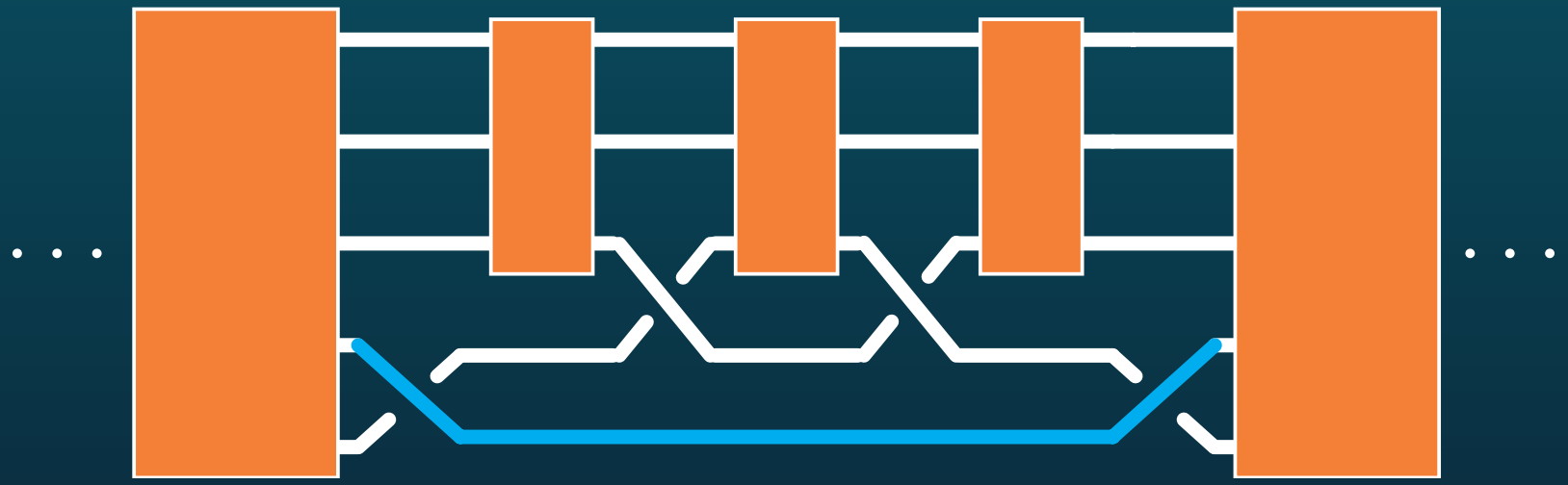
⇝ solution locale



... on ne sait pas (!)

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

⇝ solution locale



... on ne sait pas (!)

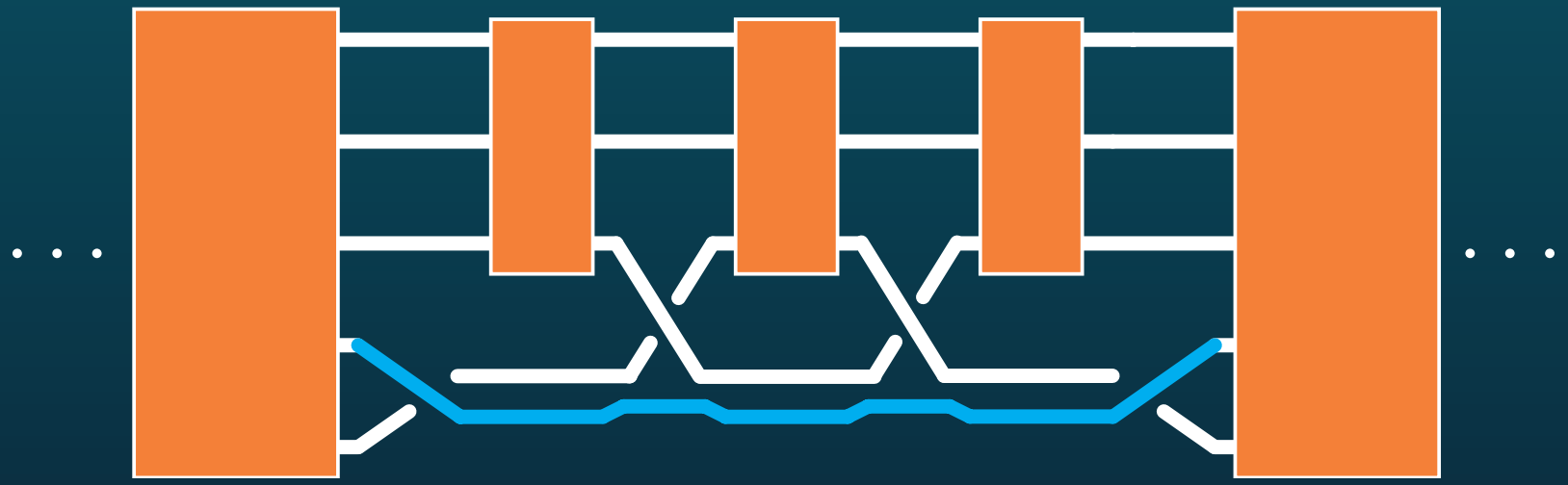
De toute façon, on peut faire (beaucoup) mieux :

⇝ solution locale

... on ne sait pas (!)

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

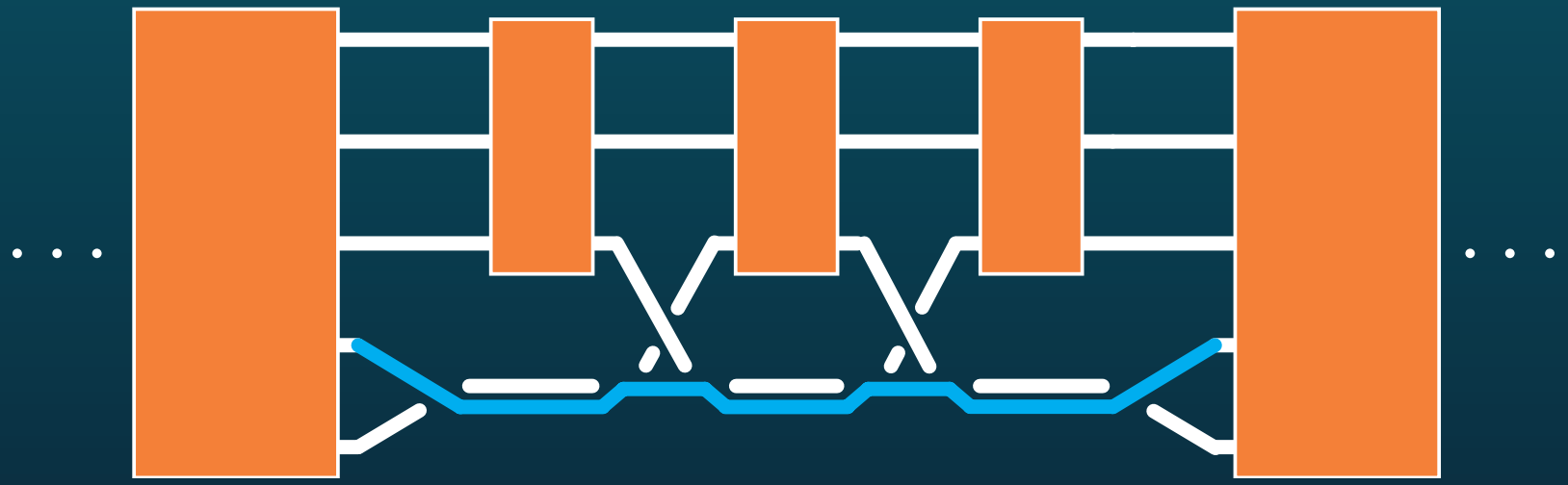
⇝ solution locale



... on ne sait pas (!)

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

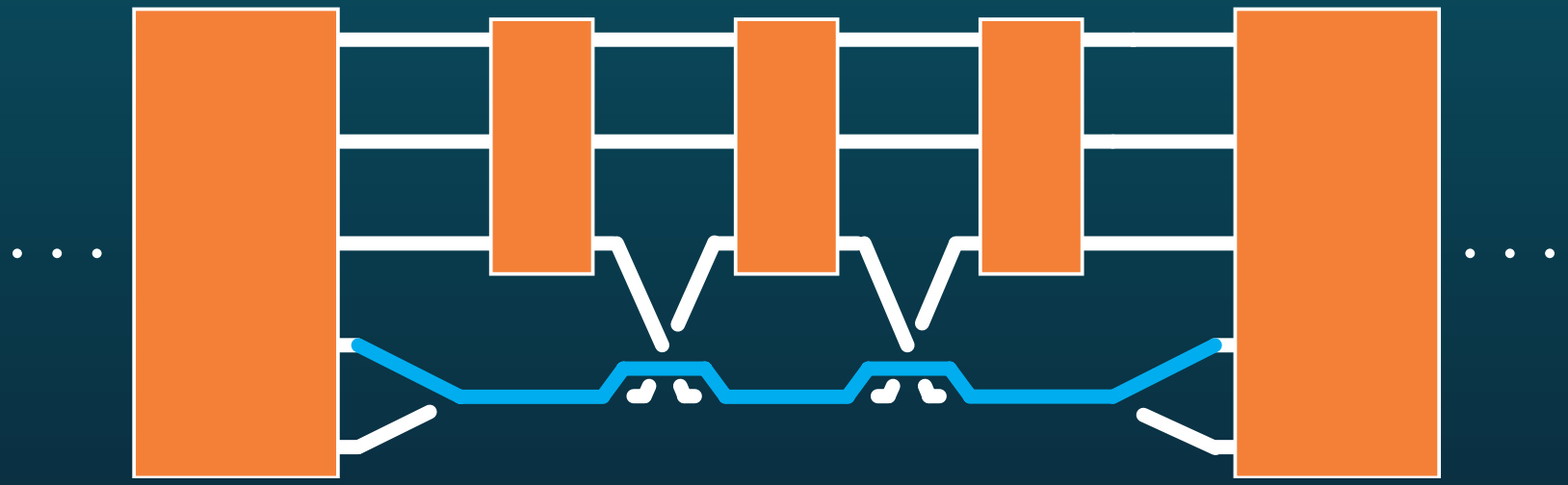
⇝ solution locale



... on ne sait pas (!)

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

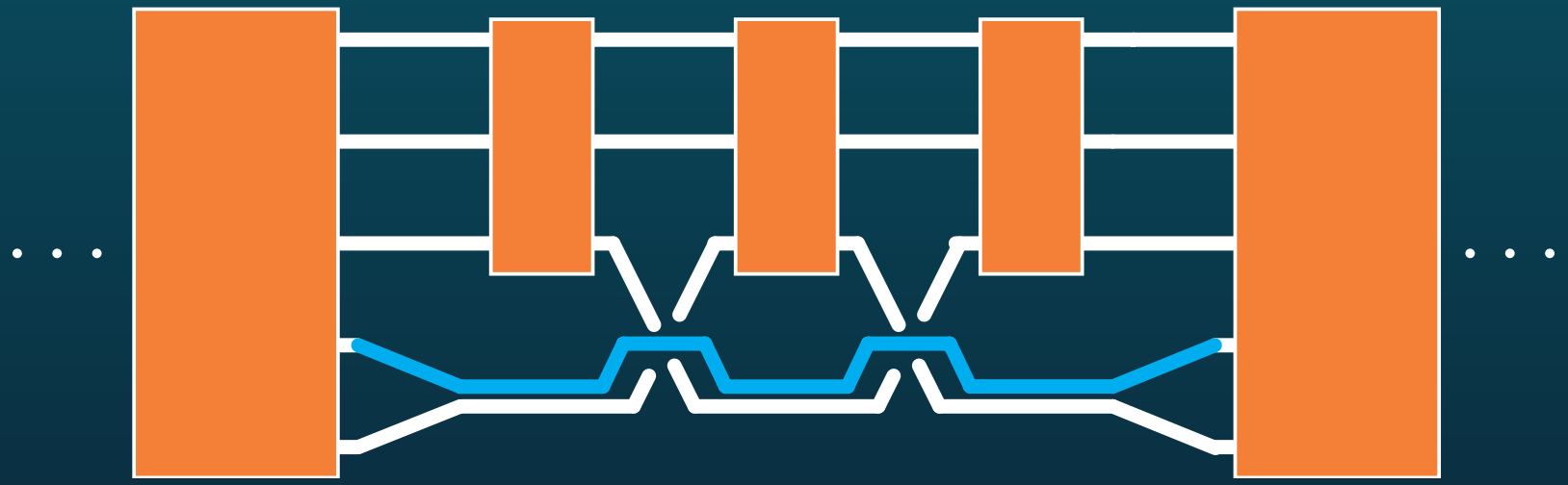
⇝ solution locale



... on ne sait pas (!)

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

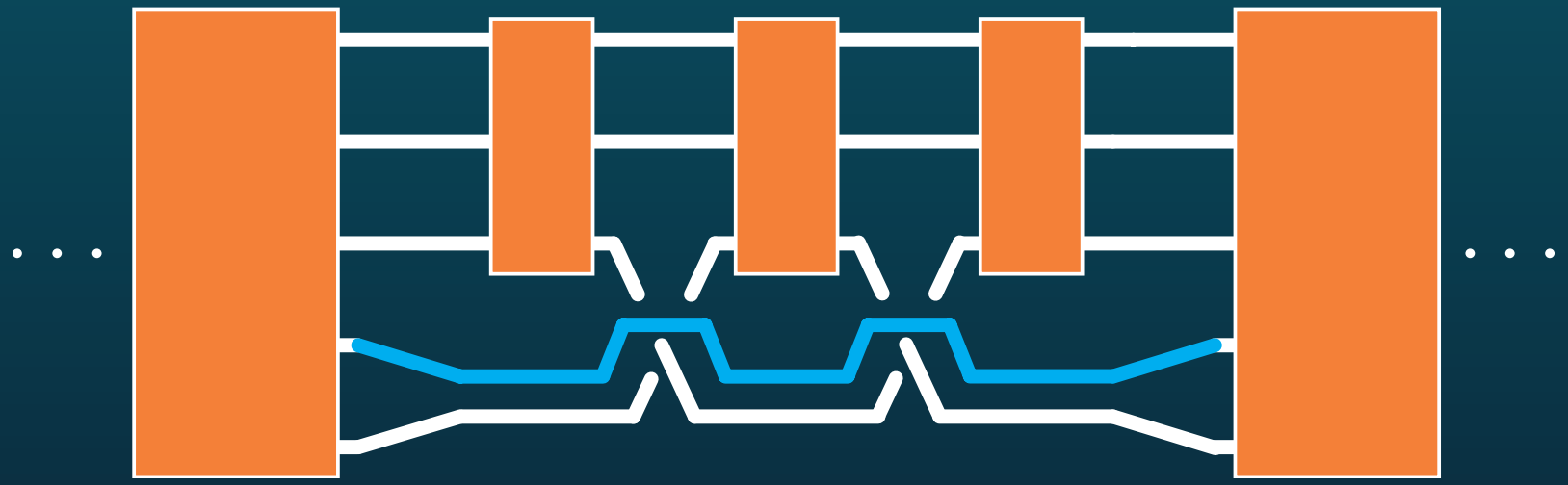
⇝ solution locale



... on ne sait pas (!)

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

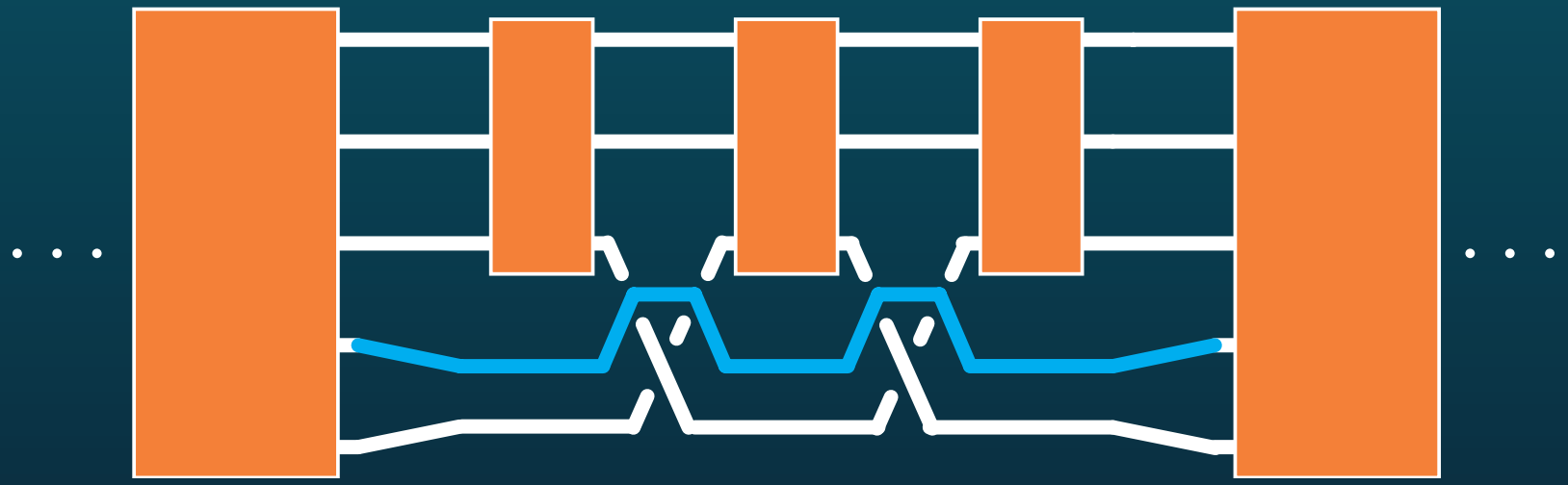
⇝ solution locale



... on ne sait pas (!)

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

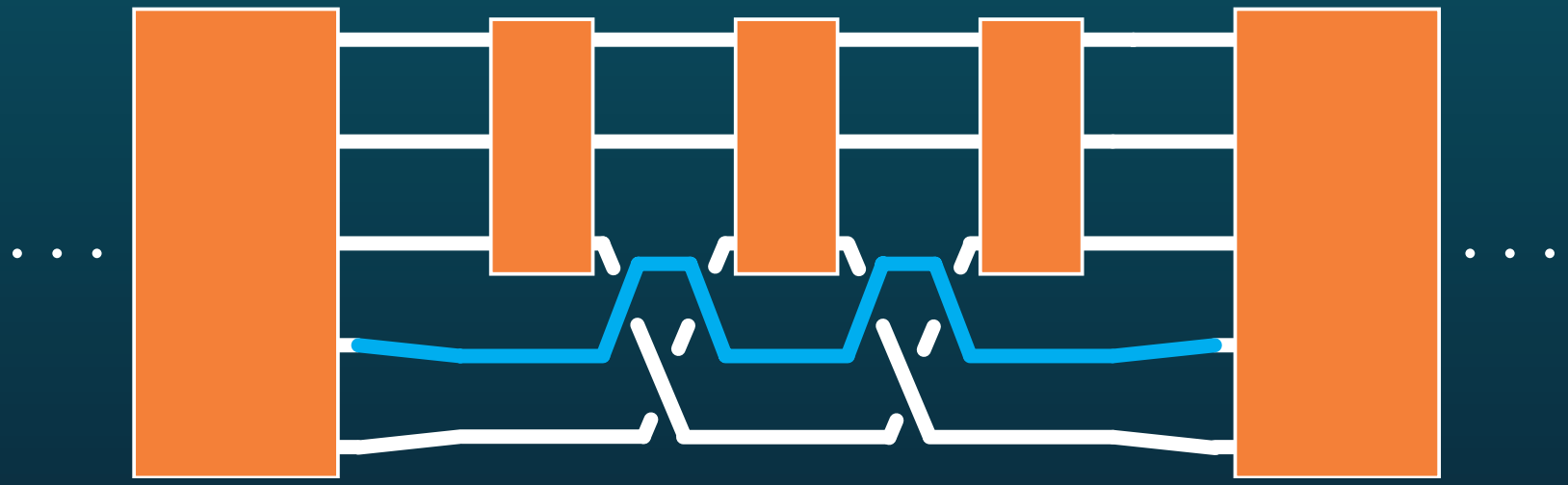
⇝ solution locale



... on ne sait pas (!)

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

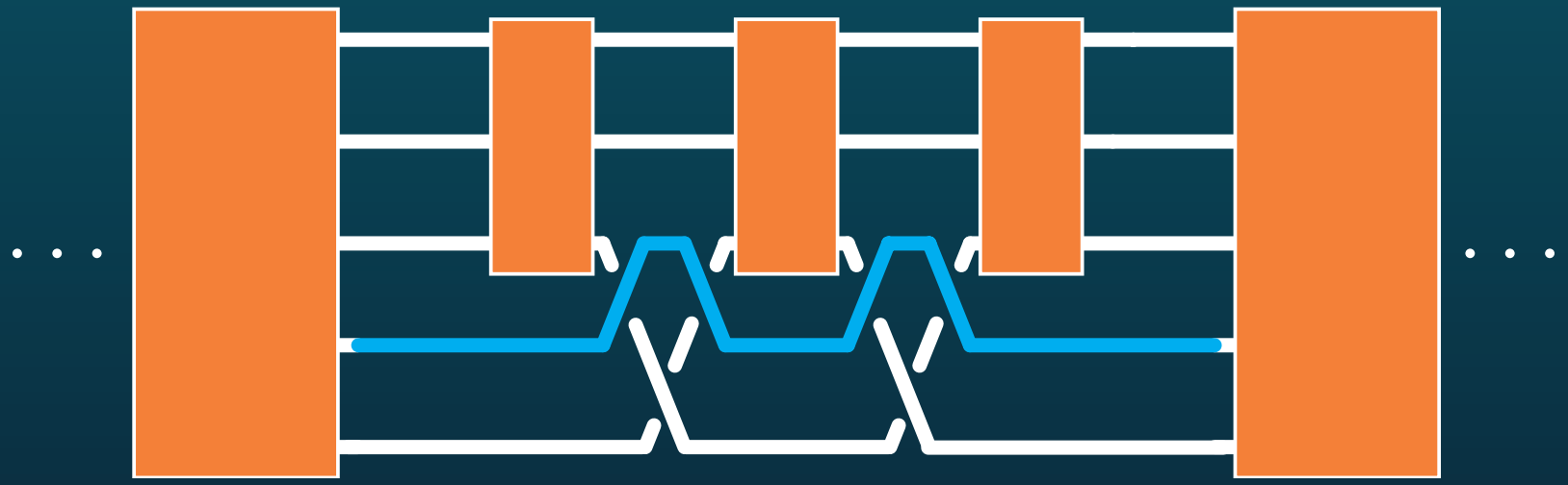
⇝ solution locale



... on ne sait pas (!)

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

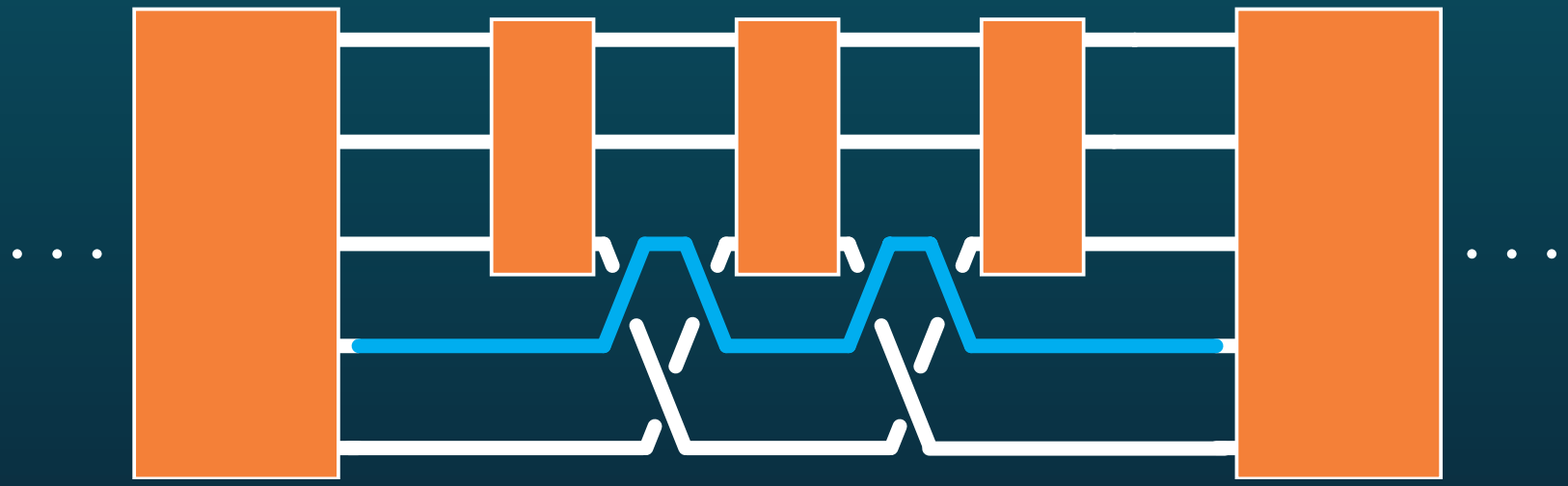
⇝ solution locale



... 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 $s_1, \ldots, s_n$:

- 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 $s_1, \ldots, s_n$:

- sommets = éléments de G,

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

- **Exemple : $\mathbb{Z}$ et 1**

- **Démonstration?**

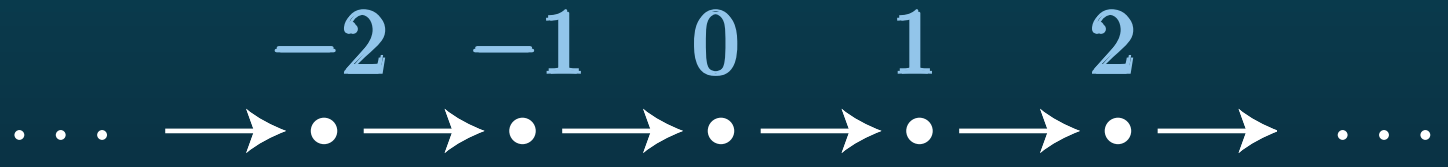
⇝ **Graphe de Cayley d'un groupe G**

relativement à des générateurs $s_1, \ldots, s_n$:

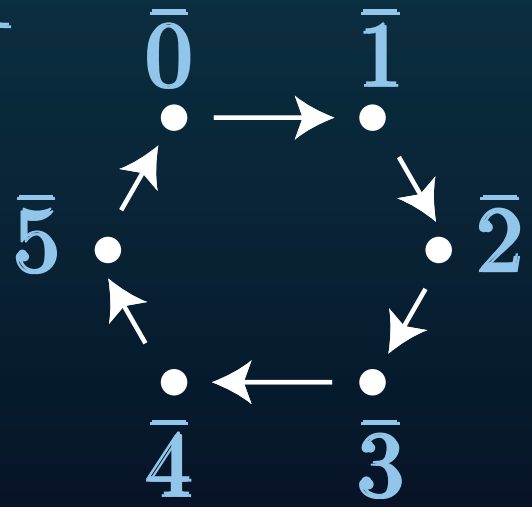
- sommets = éléments de G,

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

- **Exemple : $\mathbb{Z}$ et 1**



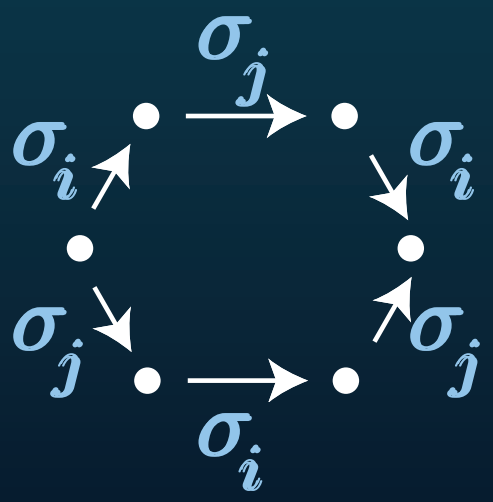
- **Exemple : $\mathbb{Z}/6\mathbb{Z}$ et $\bar{1}$**



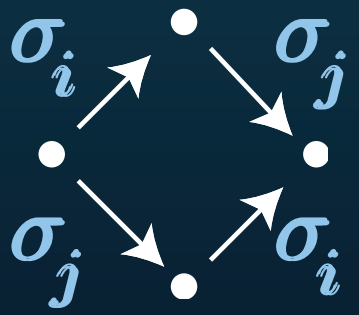
- **Exemple :** S_4 **et** $(12), (23), (34)$

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

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

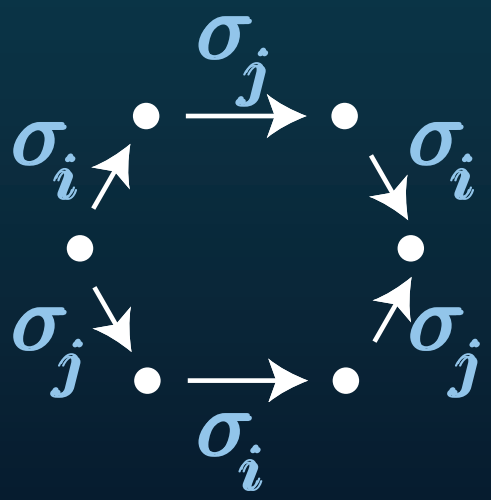


et de

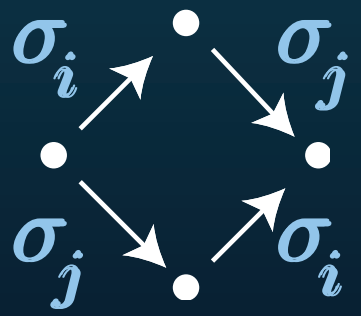


- Exemple : S_4 et $(12), (23), (34)$

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



et de



⇝ mot de tresse = chemin dans le graphe de Cayley de B_n...

- **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** { **preuve élémentaire de convergence** / **complexité algorithmique** } **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

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

où $R : V \otimes V \to V \otimes V$.

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

- Réponse 2 (physique) :

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

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

où $R : V \otimes V \to V \otimes V$.

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

$\rightsquigarrow$ représentations matricielles, cohomologie

- Réponse 2 (physique) :

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

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

où $R : V \otimes V \to V \otimes V$.

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

$\rightsquigarrow$ 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

- Réponse 4 (cryptographie) :

groupe de tresses = groupe simple et compliqué

on peut calculer

il y a des problèmes difficiles

- Pour $n \geqslant 3$, le groupe B_n est non commutatif :

(en général) on a $xyx^{-1} \neq y$.

- Réponse 4 (cryptographie) :

groupe de tresses = groupe simple et compliqué

on peut calculer

il y a des problèmes difficiles

- Pour $n \geqslant 3$, le groupe B_n est non commutatif :

(en général) on a $xyx^{-1} \neq y$.

⇝ Prendre xyx^{-1} comme crypté de y par x ?

- Réponse 4 (cryptographie) :

groupe de tresses = groupe simple et compliqué

on peut calculer

il y a des problèmes difficiles

- Pour $n \geqslant 3$, le groupe B_n est non commutatif :

(en général) on a $xyx^{-1} \neq y$.

(Prendre xyx^{-1} comme crypté de y par x)

⇝ Utiliser difficulté de retrouver x à partir de y et xyx^{-1}.

- Réponse 4 (cryptographie) :

groupe de tresses = groupe simple et compliqué

on peut calculer

il y a des problèmes difficiles

- Pour $n \geqslant 3$, le groupe B_n est non commutatif :

(en général) on a $xyx^{-1} \neq y$.

(Prendre xyx^{-1} comme crypté de y par x)

⇝ Utiliser difficulté de retrouver x à partir de y et xyx^{-1}.

V. Sidelnikov & al., K.Y. Ko & al., M. Anshel & al., ...

- **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 σ_{n+1} ... σ_{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 $s = ap_Ba^{-1}$; Bob calcule $s = bp_Ab^{-1}$.

tresses sur σ_{n+1} ... σ_{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 $s = ap_Ba^{-1}$; Bob calcule $s = bp_Ab^{-1}$.

tresses sur σ_{n+1} ... σ_{2n-1}

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

$$ap_Ba^{-1} = abpb^{-1}a^{-1} = bapa^{-1}b^{-1} = bp_Ab^{-1}.$$

- **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 $s = ap_Ba^{-1}$; Bob calcule $s = bp_Ab^{-1}$.

tresses sur σ_{n+1} ... σ_{2n-1}

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

$$ap_Ba^{-1} = abpb^{-1}a^{-1} = bapa^{-1}b^{-1} = bp_Ab^{-1}.$$

- 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/lmno

Merci !