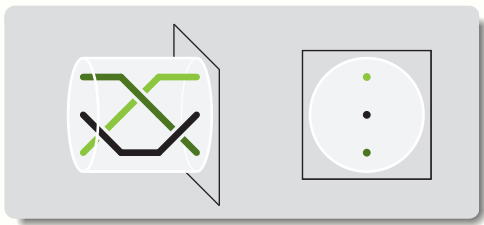


Des solutions au problème d'isotopie des tresses

- Grand nombre de solutions basées sur des approches variées, **algèbre, topologie, géométrie, combinatoire**, dont certaines sont très efficaces algorithmiquement.
- Intérêt : **variété** des solutions :
 - ↪ richesse de la structure des groupes de tresses,
 - ↪ grand nombre de généralisations possibles.

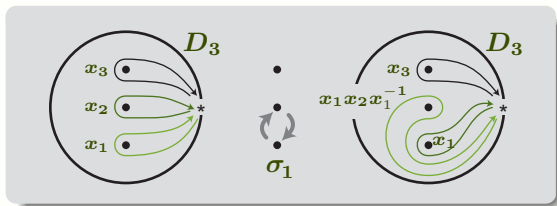
- Diagramme de tresse à n brins \rightsquigarrow mouvement de n points :



... \rightsquigarrow homéomorphisme de D_n laissant ∂D_n fixe
 disque avec n points marqués \uparrow bord de D_n

- Donc : $B_n \simeq$ groupe des classes d'isotopie d'homéomorphismes de D_n laissant ∂D_n fixe
 \uparrow
 «mapping class group» de D_n
 (ou groupe de difféotopie de D_n)

- Tresse = homéomorphisme du disque troué D_n
 \rightsquigarrow action de B_n sur le groupe fondamental de D_n ,
 qui est un groupe libre de rang n .



- De là, homomorphisme ρ de B_n dans $\text{Aut}(F_n)$:

$$\rho(\sigma_i) : \begin{cases} x_i \mapsto x_i x_{i+1} x_i^{-1}, \\ x_{i+1} \mapsto x_i, \\ x_k \mapsto x_k \text{ pour } k \neq i, i+1. \end{cases}$$

Théorème (Artin) : L'homomorphisme ρ est injectif.

Algorithme : Partant de w mot de tresse à n brins,

- (i) Calculer, pour $i = 1, \dots, n$, l'image de x_i par $\rho(w)$;
- (ii) Alors $w \equiv \varepsilon$ si et seulement si $\rho(w)(x_i) = x_i$ pour tout i .

Exemple : $w = \sigma_2^{-2} \sigma_1^{-2} \sigma_2^2 \sigma_1^2$.



- (i) $\rho(w)(x_1) = x_1 x_2 x_1^{-1} x_3 x_1 x_2^{-1} x_1^{-1} x_3^{-1} x_1 x_3 x_1 x_2$
 $\dots x_1^{-1} x_3^{-1} x_1 x_2^{-1} x_1^{-1}$;
- (ii) $\dots = x_1?$ non, donc $w \neq \varepsilon$.

- Complexité exponentielle : toujours aussi calamiteux ;
- Lien avec le coloriage des tresses ;
- Calcul différentiel libre \rightsquigarrow représentation de Burau, **non** fidèle.

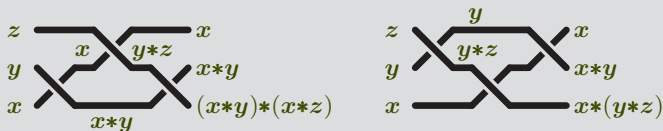
Solution : Représentation linéaire

- Domaine : **algèbre linéaire**
- Point de vue : **trousse = matrice**
- Méthode : **représentation**
- Auteurs : **Burau '36, Krammer '00, Bigelow '00**
- Mots-clés : coloriage de trousse, opérations auto-distributives
- Arrière-plan : théorie de Coxeter
- Extensions : groupes et monoïdes d'Artin–Tits



- **Colorier** un diagramme : propager les couleurs depuis la gauche ;
- Règle 1 :  \rightsquigarrow projection de B_n sur \mathfrak{S}_n
pas de solution au problème d'isotopie
- Règle 2 :  \rightsquigarrow compatibilité avec relations de tresse ?

Lemme : La règle 2 donne un coloriage de B_n^+ si et seulement si l'opération $*$ satisfait la loi $x * (y * z) = (x * y) * (x * z)$.



- Opérations satisfaisant $x * (y * z) = (x * y) * (x * z)$?

↑
« **autodistributive** »

- Opération triviale : $x * y = y$ (= règle 1) :
↪ projection sur une permutation — pas solution au pb. d'isot.
- **Conjugaison** sur un groupe : $x * y = xyx^{-1}$;
Pour F_n (groupe libre) : $B_n \rightarrow \text{Aut}(F_n)$:
↪ représentation d'Artin — solution du problème d'isotopie.
- **Moyenne** sur un $\mathbb{Z}[t]$ -module : $x * y = (1 - t)x + ty$.
Alors sorties = combinaison linéaire des entrées, donc **matrice** :
↪ représentation de **Burau** : $B_n \rightarrow \text{GL}_n(\mathbb{Z}[t, t^{-1}])$.

- La représentation de Burau résout-elle le problème d'isotopie ?
Est-ce une représentation fidèle (= injective) de B_n ?
- Oui pour $n = 3$, mais

Théorème (Moody '91, ...) La représentation de Burau de B_n
n'est pas fidèle pour $n \geq 5$.

- Autres représentations ?

Théorème (Krammer '00, Bigelow '00) : Il existe une représentation
linéaire fidèle de B_n dans $GL_{n(n-1)/2}(\mathbb{Z}[t, t^{-1}, q, q^{-1}])$.

- Donc solution au problème d'isotopie.
— complexité polynomiale, mais inefficace en pratique.

Théorème (Garside '67) : Le monoïde B_n^+ équipé de la relation de divisibilité à gauche est un treillis.

↑
existence de ppcm et de pgcd

- **Corollaire** : Pour tout β dans B_n^+ , il existe une unique tresse de permutation **maximale** divisant β à gauche, à savoir $\text{pgcd}(\beta, \Delta_n)$.
- **Décomposition distinguée en produit de tresses de permutation** :

$$\beta = [f_1] \cdot \beta' = [f_1][f_2] \cdot \beta'' = \dots = [f_1][f_2] \dots [f_d].$$

Proposition (Thurston, Morton ... '88) : Toute tresse de B_n admet une unique expression $\Delta_n^p [f_1][f_2] \dots [f_d]$ t.q. $f_1 \neq (n, \dots, 1)$, $f_d \neq \text{id}$, et, pour chaque r , tout recul de f_{r+1} est une descente de f_r .

↑

$$f_{r+1}^{-1}(i) > f_{r+1}^{-1}(i+1) \text{ entraîne } f_r(i) > f_r(i+1)$$

- Nombre **fini** de permutations + normalité **locale** (descentes)
 \rightsquigarrow $\text{NF}(w\sigma_i^{\pm 1})$ obtenu depuis $\text{NF}(w)$ et $\sigma_i^{\pm 1}$ par **transducteur** :
 structure **automatique** du groupe B_n .

Algorithme : Partant de w mot de tresse,

- (i) Calculer la forme normale $\text{NF}(w)$ de w incrémentalement ;
- (ii) Alors $w \equiv \varepsilon$ si et seulement si $\text{NF}(w) = [\text{id}]$.

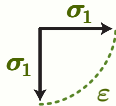
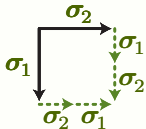
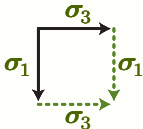
Exemple : $w = \sigma_2^{-2}\sigma_1^{-2}\sigma_2^2\sigma_1^2$.

- (i) $\text{NF}(\sigma_2^{-1}) = \Delta_3^{-1} \cdot [3, 1, 2]$, $\text{NF}(\sigma_2^{-2}) = \dots$
 $\dots \text{NF}(w) = \Delta_3^{-3} \cdot [3, 1, 2] \cdot [2, 1, 3] \cdot [2, 3, 1] \cdot [1, 3, 2] \cdot [3, 1, 2] \cdot [2, 1, 3]$.
- (ii) $\dots = [1, 2, 3]$? non, donc $w \neq \varepsilon$.

- Complexité **quadratique** à n fixé — donc utilisable.
- Existe en version **symétrique** : deux suites de permutations.

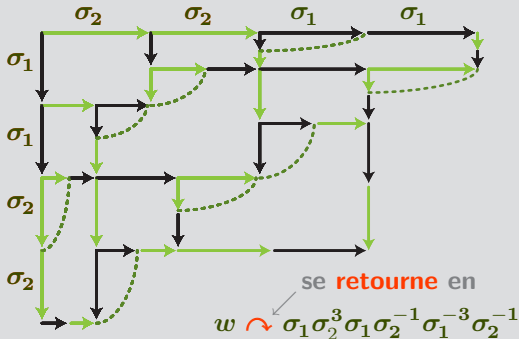
Retournement de sous-mot

- Associer à un mot de tresse un **escalier** : $\sigma_i \mapsto \xrightarrow{\sigma_i}$, $\sigma_i^{-1} \mapsto \downarrow \sigma_i$.
- Compléter jusqu'à obtenir une **équerre** \lrcorner à l'aide des règles



- **Exemple :**

$$w = \sigma_2^{-2} \sigma_1^{-2} \sigma_2^2 \sigma_1^2$$



- Où est la réécriture ? Lire les étiquettes de Pau à Strasbourg...
retourner les $-+$ en $+-$.
- Mots **terminaux** = uv^{-1} avec u, v positifs (équerres).
- **Confluence** (unicité d'un mot terminal)? Unicité du diagramme.
- **Terminaison** (existence d'un mot terminal) ?

Proposition : Supposons $u \begin{array}{c} \xrightarrow{v} \\ \curvearrowright \\ \xrightarrow{v'} \end{array} u'$ et $u_1 \equiv^+ u, v_1 \equiv^+ v$.

Alors il existe $u'_1 \equiv^+ u', v'_1 \equiv^+ v'$ t.q. $u_1 \begin{array}{c} \xrightarrow{v_1} \\ \curvearrowright \\ \xrightarrow{v'_1} \end{array} u'_1$.

Corollaire 1 : Pour u, v positifs, $u \equiv^+ v$ entraîne $u^{-1}v \curvearrowright \varepsilon$.

Corollaire 2 : Pour tout w , il existe u, v positifs t.q. $w \curvearrowright uv^{-1}$.

Algorithme : Partant d'un mot de tresse w ,

- (i) Retourner w en uv^{-1} avec u, v positifs ;
- (ii) Alors $w \equiv \varepsilon$ si et seulement si $v^{-1}u$ se retourne en ε .

• Justification :

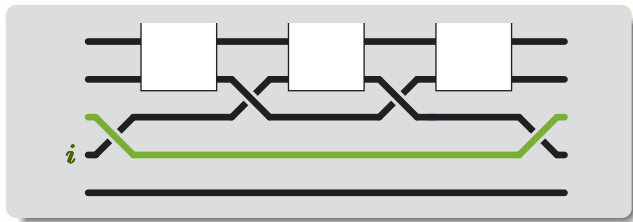
$$w \equiv \varepsilon \iff uv^{-1} \equiv \varepsilon \iff u \equiv v \iff u \equiv^+ v \iff v^{-1}u \curvearrowright \varepsilon. \quad \square$$

Exemple : $w = \sigma_2^{-2}\sigma_1^{-2}\sigma_2^2\sigma_1^2$: poser $\mathbf{a} = \sigma_1$, $\mathbf{b} = \sigma_2 \dots \mathbf{A} = \sigma_1^{-1} \dots$

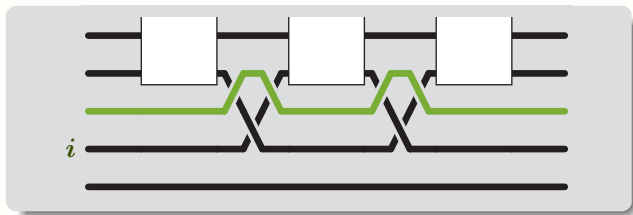
- (i) $\mathbf{BBAAbbaa} \curvearrowright \mathbf{abbbaBAAAB}$;
- (ii) $\mathbf{BAAABabbba} \curvearrowright \mathbf{BBAAbbaa} (= w)$;
- (iii) $\mathbf{BBAAbbaa} = \varepsilon$? ... non, donc $w \neq \varepsilon$.

• Complexité **quadratique** à n fixé, mais inconnue par rapport à n .

- Une σ_i -poignée dans un diagramme de tresse :



- La **réduction** d'une poignée :



- En termes de mots :

σ_i -poignée = $\sigma_i^e v \sigma_i^{-e}$, avec $e = \pm 1$ et que des $\sigma_j^{\pm 1}$, $j \geq i$, dans v ,
réduire = supprimer $\sigma_i^{\pm 1}$, remplacer chaque σ_{i+1}^d par $\sigma_{i+1}^{-e} \sigma_i^d \sigma_{i+1}^e$.

- La réduction des poignées étend la **réduction libre** $\sigma_i \sigma_i^{-1} \mapsto \varepsilon$:



- Réduire donne un mot **équivalent** (donc $w \mapsto \varepsilon$ entraîne $w \equiv \varepsilon$).
- Mots terminaux = mots **sans poignée** (= où le σ_i de plus petit indice n'apparaît qu'avec un seul signe).

Théorème : (i) Un mot sans poignée n'est jamais trivial.
 (ii) Toute suite de réductions est finie.

Algorithme : Partant d'un mot de tresse w ,

- (i) Réduire la première poignée de w ,
et itérer jusqu'à obtenir w' sans poignée ;
- (ii) Alors $w \equiv \varepsilon$ si et seulement si $w' = \varepsilon$.

• Justification : Pour w' sans poignée, $w' \equiv \varepsilon$ équivaut à $w' = \varepsilon$. \square

Exemple : $w = \sigma_2^{-2} \sigma_1^{-2} \sigma_2^2 \sigma_1^2$: poser $\mathbf{a} = \sigma_1$, $\mathbf{b} = \sigma_2 \dots$ $\mathbf{A} = \sigma_1^{-1} \dots$

- **BBAAbbaa**

- **BBAb**a**BbaBa**

- **BBAbaaBa**

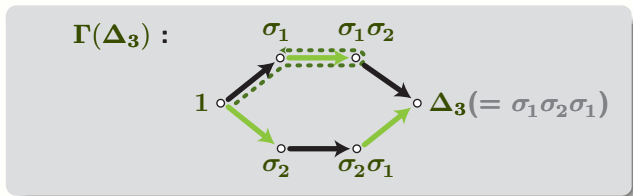
- **BBbaBaBa**

- **BaBaBa,**

sans poignée et non vide, donc $w \neq \varepsilon$.

- Stratégies de réduction, par exemple diviser pour régner.
- Complexité : **inconnue** (\leq exponentielle, conjecturée quadratique)

- **Graphe de Cayley** de B_n :
sommets = tresses ; arête $\beta \xrightarrow{\sigma_i} \beta'$ pour $\beta' = \beta\sigma_i$.
- $\Gamma(\Delta_n^d)$ = restriction du g. de Cayley aux **diviseurs** de Δ_n^d dans B_n^+ .



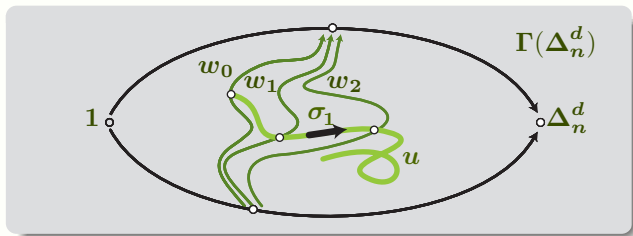
- Mot **tracé** à partir de β dans $\Gamma(\Delta_n^d)$:
 $\sigma_1\sigma_2\sigma_2^{-1}$ tracé à partir de 1 dans $\Gamma(\Delta_3)$, mais σ_1^2 non tracé.

Lemme : (i) Tout mot est tracé dans $\Gamma(\Delta_n^d)$ pour $n, d \gg 0$.
(ii) L'ensemble des mots tracés dans $\Gamma(\Delta_n^d)$ est clos par réduction.

- Démonstration du point (ii) :

« Toute suite de réductions est finie. »

- Soit w_0, w_1, \dots une suite de réductions, les w_r tracés dans $\Gamma(\Delta_n^d)$. Alors il existe $u = u_1 u_2 \dots$ tracé dans $\Gamma(\Delta_n^d)$ t.q.,
 si w_r, w_{r+1} est la réduction de la première σ_1 -poignée,
 alors u_r contient un σ_1 et zéro σ_1^{-1} , sinon u_r contient zéro $\sigma_1^{\pm 1}$.



- Par (i), le mot u traverse au plus une fois chaque σ_1 de $\Gamma(\Delta_n^d)$, et il n'y a qu'un nombre fini d'arêtes σ_1 dans $\Gamma(\Delta_n^d)$. \square

- Pour $x \in \mathbb{Z}$, on pose $x^+ = \max(0, x)$, $x^- = \min(x, 0)$, et

$$F^+(x_1, y_1, x_2, y_2) = (x_1 + y_1^+ + (y_2^+ - z_1)^+, y_2 - z_1^+, x_2 + y_2^- + (y_1^- + z_1)^-, y_1 + z_1^+),$$

$$F^-(x_1, y_1, x_2, y_2) = (x_1 - y_1^+ - (y_2^+ + z_2)^+, y_2 + z_2^-, x_2 - y_2^- - (y_1^- - z_2)^-, y_1 - z_2^-),$$

où $z_1 = x_1 - y_1^- - x_2 + y_2^+$ et $z_2 = x_1 + y_1^- - x_2 - y_2^+$.

- On fait agir les mots de tresse à n brins sur \mathbb{Z}^{2n} par

$$(a_1, b_1, \dots, a_n, b_n) * \sigma_i^e = (a'_1, b'_1, \dots, a'_n, b'_n)$$

avec $a'_k = a_k$ et $b'_k = b_k$ pour $k \neq i, i+1$, et

$$(a'_i, b'_i, a'_{i+1}, b'_{i+1}) = F^e(a_i, b_i, a_{i+1}, b_{i+1}).$$

- Par définition, les **coordonnées** de w sont $(0, 1, 0, 1, \dots, 0, 1) * w$.

Théorème (Dynnikov '00) : Les coordonnées de w ne dépendent que de la tresse représentée par w , et caractérisent celle-ci.

Algorithme : Partant d'un mot de tresse w ,

- (i) Calculer ses coordonnées par les formules de Dynnikov ;
- (ii) Alors $w \equiv \varepsilon$ si et seulement si on obtient $(0, 1, 0, 1, \dots, 0, 1)$.

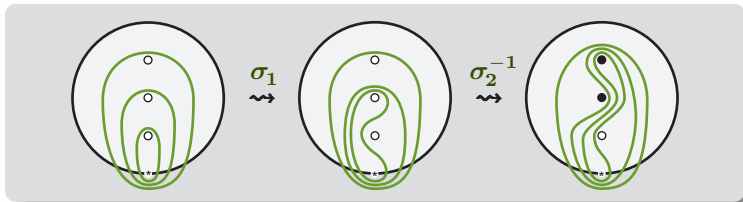
Exemple : $w = \sigma_2^{-2} \sigma_1^{-2} \sigma_2^2 \sigma_1^2$; coordonnées :

$$(0, 1, 0, 1, \dots, 0, 1) * w = (1, -19, -12, 9, 0, 13, 0, 1)$$

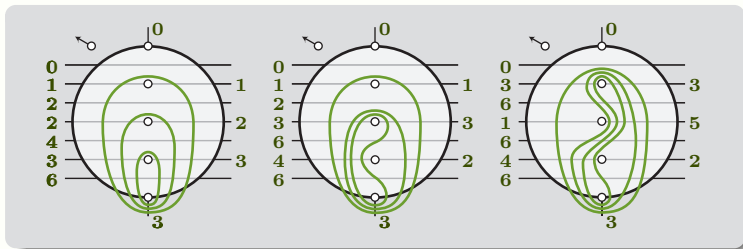
$$\neq (0, 1, 0, 1, \dots, 0, 1), \text{ donc } w \neq \varepsilon.$$

- Complexité : **quadratique** en temps, linéaire en espace,
indépendamment de n .

- Tresse=homéomorphisme \rightsquigarrow agit sur les courbes tracées sur D_n .



- Compter les intersections avec une triangulation fixée :



\rightsquigarrow $3n + 3$ nombres, déterminant la tresse

- Coordonnées = demi-différences entre nombres d'intersection ;
(passage de $3n + 3$ à $2n$ entiers)

Question : Coordonnées de $\beta\sigma_i$ à partir de celles de β et de i ?

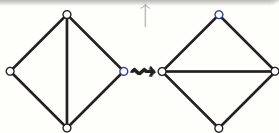
= comparer les intersections de C et $\sigma_i(C)$ avec la triangulation T
↙ courbe(s) fermée(s)

- Or on a

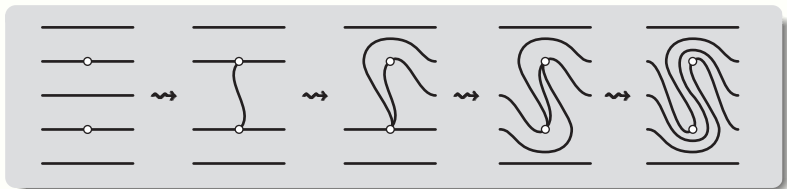
$$\#(\sigma_i(C) \cap T) = \#(C \cap \sigma_i^{-1}(T))$$

↔ comparer les intersections de C avec T et $\sigma_i^{-1}(T)$.

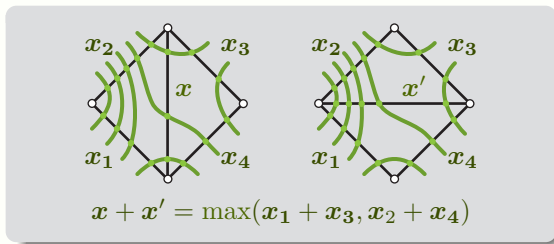
Proposition : Si T, T' sont deux triangulations (singulières) d'une surface, on peut passer de T à T' par une suite finie de **flips**.



- Donc on **doit** passer de T à $\sigma_i^{-1}(T)$ par une suite finie de flips :



- Pour **un** flip, la formule est



\rightsquigarrow formules de Dynnikov en itérant quatre fois.

Solution : Formes normales alternante et cyclante

- Domaine : algèbre / combinatoire
- Point de vue : monoïde de tresse
- Méthode : forme normale
- Auteurs : Burckel'94, D.'07, Fromentin'07
- Mots-clés : automorphisme, générateurs de Birman–Ko–Lee
- Arrière-plan : combinatoire des partitions non croisées
- Extensions : catégories de Garside



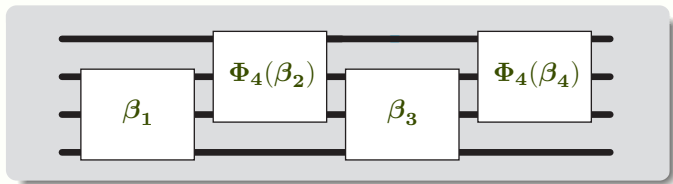
- Φ_n : automorphisme de B_n échangeant σ_i et σ_{n-i} pour $i < n$.
«flip»: symétrie horizontale des diagrammes

- Toute tresse de B_n^+ admet une unique décomposition

$$\beta = \beta_1 \cdot \Phi_n(\beta_2) \cdot \dots \cdot \Phi_n^{p-1}(\beta_p)$$

avec β_1, \dots, β_p dans B_{n-1}^+ t.q., pour tout $r \geq 2$,

aucun σ_i avec $i < n - 1$ ne divise $\Phi_n(\beta_r) \cdot \dots \cdot \Phi_n^{p-r+1}(\beta_p)$.



- En itérant, **forme normale**
— et solution quadratique au problème d'isotopie.

- (Birman–Ko–Lee '97) d'autres générateurs de B_n :

$$a_{i,j} = \sigma_{j-1} \dots \sigma_{i+1} \sigma_i \sigma_{i+1}^{-1} \dots \sigma_{j-1}^{-1} \text{ pour } 1 \leq i < j \leq n.$$



- Monoïde **dual** : le sous-monoïde B_n^{+*} de B_n engendré par les $a_{i,j}$.

Théorème : Le monoïde B_n^{+*} a une structure de Garside, où le rôle de Δ_n est joué par $\delta_n = \sigma_{n-1} \dots \sigma_2 \sigma_1$, et où les diviseurs de δ_n sont en bijection avec les **partitions non croisées** de $\{1, \dots, n\}$.

De là, solution du problème d'isotopie par « greedy normal form ».

- Automorphisme Φ_n de B_n^+ («flip») = conjugaison par Δ_n ;
 \rightsquigarrow automorphisme ϕ_n de B_n^{+*} («cyclage») = conjugaison par δ_n .

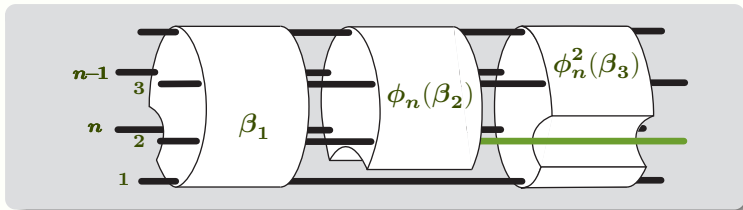
$$\phi_n(a_{i,j}) = a_{i+1 \bmod n, j+1 \bmod n}$$

- Toute tresse de B_n^{+*} admet une unique décomposition

$$\beta = \beta_1 \cdot \phi_n(\beta_2) \cdot \dots \cdot \phi_n^{p-1}(\beta_p)$$

avec β_1, \dots, β_p dans B_{n-1}^{+*} t.q., pour tout $r \geq 2$,

aucun $a_{i,j}$ avec $j < n$ ne divise $\phi_n(\beta_r) \cdot \dots \cdot \Phi_n^{p-r+1}(\beta_p)$.



- En itérant, **forme normale** — et solution au problème d'isotopie.

Solution : **Algorithme(s) de relaxation**

- **Domaine** : **topologie**
- **Point de vue** : **trousse = homéomorphisme**
- **Méthode** : **forme normale**
- **Auteur** : **X.Bressaud '05**
- **Mots-clés** : homéomorphisme, lacet, stratégie de relaxation
- **Arrière-plan** : systèmes dynamiques
- **Extensions** : frontière de Poisson, problèmes de stabilisation



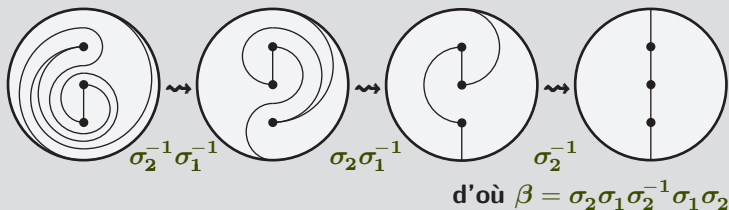
- Tresse = homéomorphisme.

Principe : Fixer une (ou des) courbe de base C , et définir une **stratégie** pour **relaxer** (= demêler) $\beta(C)$ et revenir à C : la suite des $\sigma_i^{\pm 1}$ utilisés donne une expression de β^{-1} .

— suppose de définir une notion de **complexité**.

- Exemple 1 (Fenn et al. '97, Dynnikov–Wiest '06) :

C = diamètre principal de D_n , stratégie = « arc utile ».

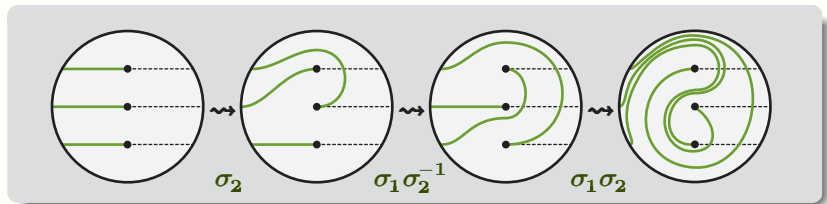


- Exemple 2 (Bressaud '05) :

C = axes des lacets standards (cf. représentation d'Artin) ;

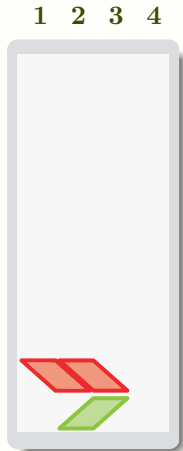
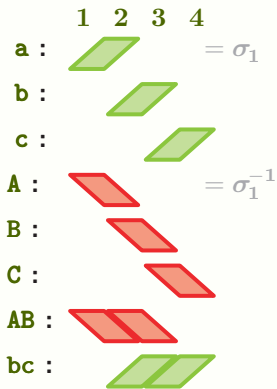
stratégie : relaxer $\beta(x_1)$, puis $\beta(x_2)$, etc.

en diminuant les intersections avec les demi-axes.



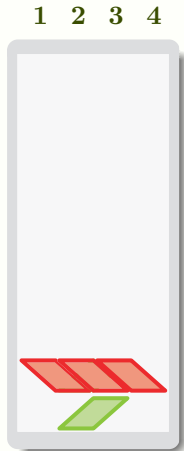
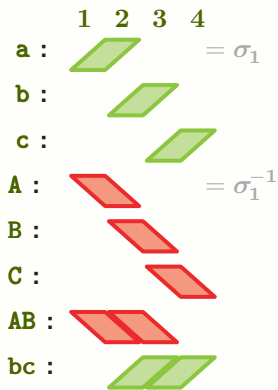
- Une forme normale – donc une solution au problème d'isotopie –
 ... mais surtout un **algorithme** calculant la forme normale de $w\sigma_i^{\pm 1}$
 à partir de celle de w et de i .

Algorithme de Bressaud

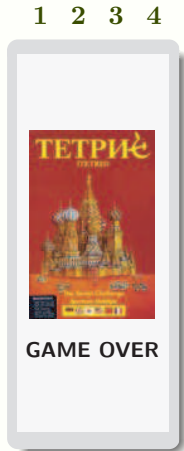
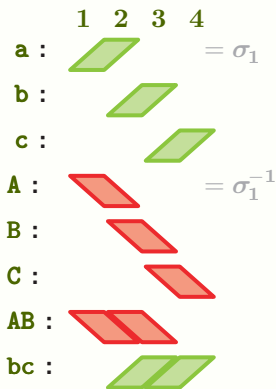


- Forme normale de Bressaud de **ABa** $= \mathbf{b.AB}$.

Algorithme de Bressaud



- Forme normale de Bressaud de **ABaaCA** = **b.ABC**.



- Forme normale de Bressaud de **ABaaCAbCc** = **b.c.ABC**.
- Question (J. Chamboredon) : Approche purement syntaxique ?

- On a survolé **des** solutions au problème d'isotopie des tresses, **variées**, mettant en jeu beaucoup de **jolies** mathématiques...

Y en a-t-il encore **d'autres** ?

Quid du **problème de conjugaison** ?

Quid des **groupes d'Artin–Tits** ?

Quid des **groupes de tresses de surface** ?

Quid des **mapping class groups** généraux ?

Quid de l'**isotopie des nœuds** ?

...