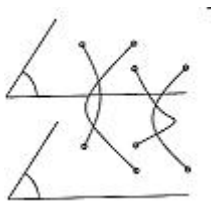
Renversement des tresses

Léo Ducas et Anne-Sophie de Suzzoni

Contents

1 Introduction

Soit $E = \{z_0...z_n\} \subset C$ et $\alpha_0...\alpha_n : [0,1] \to C$ tels que $\{\alpha_i(0)\} = \{\alpha_i(1)\} = E$ et que leurs graphes soient disjoints. L'ensemble des graphes des α_i représente ce que l'on appelle une tresse.



Les tresses sont les classes d'équivalence de ces graphes où la relation d'équivalence est le passage d'un graphe à un autre par transformation continue. On verra que les tresses à n brins forment un groupe, noté $\mathcal{B}_n$. On cherche à déterminer si deux ensembles de graphes représentent la même tresse, c'est-à-dire si l'on peut passer de l'un à l'autre sans

détacher les extremités. Ce problème admet une traduction algèbrique qui s'inscrit plus généralement dans la théorie des groupes à complément. On travaillera pour cela sur les mots d'un alphabet à $n-1$ lettres positives et leurs inverses. On se donne une relation d'équivalence sur ces mots qui correspond à l'équivalence topologique. De cette façon, on se rammène à décider de l'équivalence de deux mots. La solution de ce problème prend la forme d'un algorithme dit de réduction. Le calcul sur les tresses s'implémente alors simplement à travers la représentation en mots, l'avantage de cette représentation étant la simplicité de la multiplication, celle-ci se rammenant à la concaténation des mots.
Les tresses apparaissent en informatique dans le domaine de la cryptologie. Le groupe peut en effet servir à décider d'une clé partagée: soit ρ une tresse publique dans $\mathcal{B}_{2n}$, Alice choisit une tresse a sur les n premiers brins et envoie à Bob $A = a\rho a^{-1}$, Bob choisit quant à lui, une tresse b sur les n derniers et envoie à Alice $B = b\rho b^{-1}$, la clé partagée est alors $k = ab\rho a^{-1}b^{-1}$. Etant donné que a et b sont à support disjoints, ils commutent, ils est donc facile pour Bob et Alice de calculer k. Par contre, si A et B ont été réduits avant d'être envoyés, factoriser A connaissant ρ est un problème supposé difficile.
Tout d'abord, on présentera le groupe des tresses de façon algébrique et topologique. On présentera ensuite l'algorithme de réduction dont on montrera qu'il est solution du problème. On utilisera pour cela la notion de cohérence ainsi que l'existence d'un élément de Garside.

2 Présentation des tresses et du problème de mot

2.1 Groupe $\mathcal{B}_n$, définition algèbrique

Dans toute la suite, on travaillera sur l'alphabet A à n éléments, notés $\sigma_1, ..., \sigma_n$. De plus, ϵ représente le mot vide sur A^*.

Définition On appelle mots positif sur A toute suite finie $\sigma_{i_1}...\sigma_{i_r}$, $i_1...i_r \in [[1..n]]$. Leur ensemble A^+ forme un monoïde libre pour la concaténation.

Pour obtenir un groupe, on a besoin d'une copie de A, notée A^{-1}:

$$A^{-1} = \{\sigma_1^{-1}, ..., \sigma_n^{-1}\}$$

On forme alors A^*, l'ensemble des mots sur $A \cup A^{-1}$.
En quotientant le monoïde A^* par la relation d'équivalence engendrée par concaténation et transitivité par les couples $\{(xx^{-1}, \epsilon) | x \in A \cup A^{-1}\}$, on obtient le groupe libre de rang n.

Définition Soit $\equiv^+$ la relation d'équivalence sur A^+ engendrée par concaténation et transitivité par les couples $(\sigma_i\sigma_j, \sigma_j\sigma_i)$ si $|i-j| > 1$, $(\sigma_i\sigma_j\sigma_i, \sigma_j\sigma_i\sigma_j)$ sinon.
Soit $\equiv$ celle engendrée par $\equiv^+$ et les couples (xx^{-1}, ϵ), pour $x \in A \cup A^{-1}$.
Le groupe $\mathcal{B}_{n+1}$ est le quotient de A^* par $\equiv$. On note $\mathcal{B}_{n+1}^+$ le monoïde quotient de A^+ par $\equiv^+$.

Remarque On constate qu'il existe une fonction $f : A \times A \to A^+$ telle que $\equiv^+$ soit

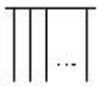
engendrée par les couples $(xf(x,y), yf(y,x))$, avec ici

$$f(\sigma_i, \sigma_j) = \begin{cases} \sigma_j & \text{si } |j-i| > 1 \\ \sigma_i\sigma_j & \text{si } |j-i| = 1 \\ \epsilon & \text{sinon.} \end{cases}$$

Un tel f est appellé complément du groupe $\mathcal{B}_n$. A partir de la deuxième partie, on s'interessera plus généralement aux groupes qui admettent un complément.

2.2 Représentation topologique

Chaque mot de A^* se représente de façon topologique par $n+1$ brins entremêlés. Le mot vide correspond au brin délié :



Quant à σ_i (resp. σ_i^{-1}), il représente le passage du brin i au-dessus (resp. en dessous) de $i+1$:

Les relations algébriques définies ci-dessus proviennent de relations topologiques. Une tresse est l'ensemble des graphes de $n+1$ fonctions continues de $[0,1]$ dans C telles que:

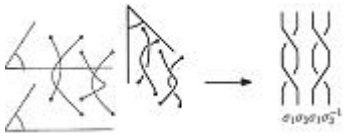
- les brins ne se croisent pas : $i \neq j \Rightarrow a_i(t) \neq a_i(t)$,
- $\{a_i(0)\} = \{a_i(1)\} = E$

quotienté par la relation d'isotopie : deux ensembles α et β de $n+1$ brins sont isotopes s'il existe une fonction $H : [0,1] \times [0,1] \to C^{n+1}$ continue telle que

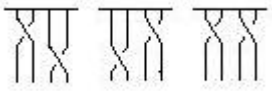
$$\begin{cases} \forall x \in [0,1]\ H(x,0) = (\alpha_0(x), ..., \alpha_n(x)) \\ \forall x \in [0,1]\ H(x,1) = (\beta_0(x), ..., \beta_n(x)) \\ \forall t \in [0,1]\ H(.,t) \text{ vérifie les conditions données ci-dessus} \end{cases}$$

Cette définition permet de vérifier que :

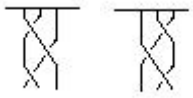
- l'ensemble des tresses forme un groupe : en effet, la composition des chemins fait des ensembles de $n+1$ brins un monoïde, qui, quotienté par la relation d'sotopie, forme un groupe ;
- que ce groupe se plonge dans le groupe libre de rang n : projeter les brins sur un plan vertical en gardant l'information "dessus/dessous" pour les intersection créées renvoit un mot de A^* ;



- que le plongement vérifie bien les relations algébriques définies: en effet, $\sigma_i\sigma_j$ et $\sigma_j\sigma_i$ pour $|i-j|>1$ peuvent s'obtenir par projection de la même classe d'isotopie:



Il en est de même pour $\sigma_i\sigma_{i+1}\sigma_i$ et $\sigma_{i+1}\sigma_i\sigma_{i+1}$:



- de plus, on peut représenter chaque mot par un ensemble de $n+1$ brins.

La raison pour laquelle ces relations algébriques suffisent à engendrer toutes les isotopies n'est pas évidente, et ne fait pas l'objet de cet exposé.

2.3 Définition du problème de mot

Notre objectif est de déterminer si deux mots x et y de A^* sont $\equiv$-équivalents. Pour cela, on forme le mot xy^{-1} et on va construire un algorithme agissant sur ce mot renvoyant ϵ si et seulement si $x \equiv y$.
Le problème n'est pas trivial: le langage L des mots représentants la tresse neutre n'est

- ni rationel $L \cap \sigma_1^*(\sigma_1^{-1})^* = \{\sigma_1^n(\sigma_1^{-1})^n | n \in N\}$
- ni algébrique $L \cap \sigma_1^*\sigma_3^*(\sigma_1^{-1})^*(\sigma_3^{-1})^* = \{\sigma_1^n\sigma_3^m(\sigma_1^{-1})^n(\sigma_3^{-1})^m | n, m \in N\}$.

3 Algorithme et phase de redressement

3.1 Présentation de l'algorithme

Définition Un groupe à complément est un quadruplet (G, A, f, ϕ) où G est un groupe, A un alphabet fini, f un application de $A \times A$ dans A^+ et $\phi : A^* \to G$ telle que

- $\phi(ww') = \phi(w')\phi(w')$, pour tout $w, w' \in A^*$,
- $\phi(\epsilon) = 1_G$,
- $\phi(xx^{-1} = 1_G$ pour tout $x \in A \cup A^{-1}$,

- $\phi(xf(x,y)) = \phi(yf(y,x))$, pour tous $x, y \in A$.

On note $\equiv^+$ la relation d'équivalence sur le monoïde A^+ engendrée par les couples $xf(x,y), yf(y,x)$.

Définition L'algorithme de redressement consiste à parcourir le mot w de départ et à transformer chaque sous-mot $x^{-1}y$, où x et $y \in A$ par $f(x,y)f(y,x)^{-1}$ jusqu'à ce qu'il n'y ait plus de sous-mot de cette forme. Chaque étape de cet algorithme, c'est-à-dire chaque remplacement d'un sous-mot $x^{-1}y$ sera appellé un pas de redressement. On notera $u \hookrightarrow v$ le fait que u se redresse en v en un pas, et $u \hookrightarrow^n v$ en n pas. Si cette opération termine, on obtient un mot de la forme uv^{-1} où $u, v \in A^+$.

On a alors $w \equiv \epsilon \Leftrightarrow uv^{-1} \equiv \epsilon \Leftrightarrow v^{-1}u \equiv \epsilon$.

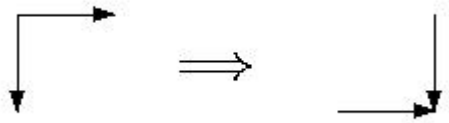
Définition L'algorithme de réduction consiste à , dans une première étape, appliquer l'algorithme de redressement à un mot w. Si ce redressement termine, on obtien un mot de la forme uv^{-1} où $u, v \in A^+$. On applique ensuite un nouveau redressement à $v^{-1}u$, et on renvoit son résultat.

On se heurte à deux problèmes : tout d'abord, $f(x,y)$ n'étant pas nécessairement de longueur 1, la taille du mot peut augmenter au cours du redressement, rien n'assure a priori la convergence; ensuite, on doit être certain d'obtenir ϵ et non un mot équivalent à la fin de la réduction.

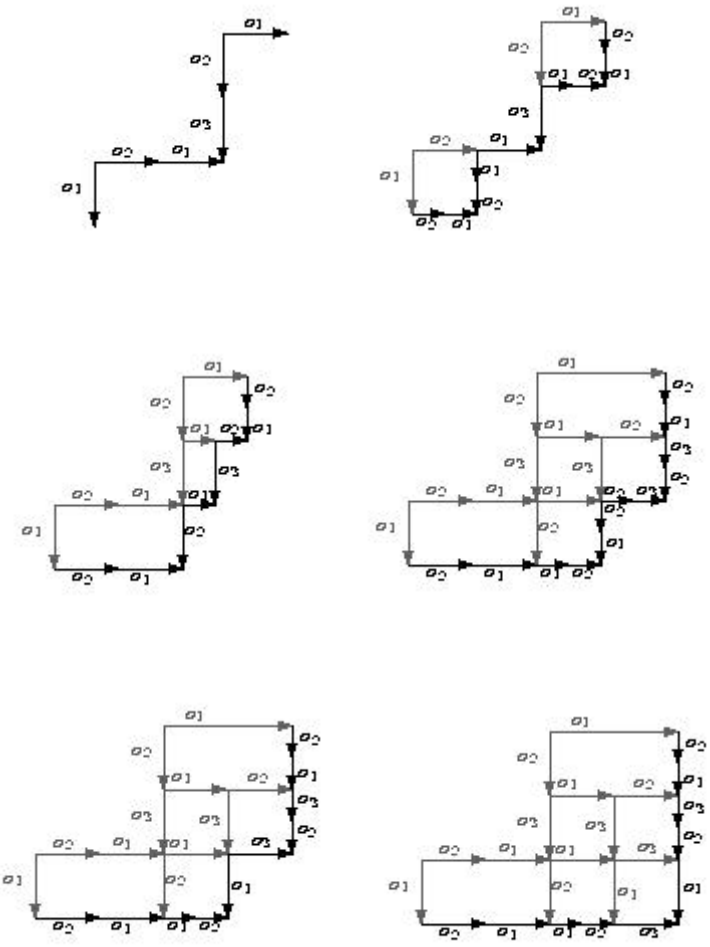
3.2 Graphe de redressement

On peut représenter le redressement d'un mot par un graphe. Chaque mot est représenté par un escalier défini par récurrence par:

Chaque pas de redressement se traduit graphiquement par une transformation d'un sous-graphe de la forme suivante:



Exemple pour $\sigma_1^{-1}\sigma_2\sigma_3^{-1}\sigma_2^{-1}\sigma_1$



3.3 Confluence

L'algorithme de redressement se formalise en tant que système de réécriture. Pour de tels systèmes, on cherche généralement des propriétés de confluence : à chaque étape de calcul, on peut faire différents choix sur le sous-mot à réécrire. La propriété de confluence nous assure que le résultat final est indépendant de ces choix.

Propriété Si $w \hookrightarrow^{n'} w'$ et $w \hookrightarrow^{n"} w"$ alors il existe n tel que $sup(n', n") \leq n \leq n' + n"$ et v tel que $w' \hookrightarrow^{n-n'} v$ et $w" \hookrightarrow^{n-n"} v$.

Preuve Cette confluence est en fait équivalente à la confluence forte : il suffit de prouver que si $w \hookrightarrow w'$ et $w \hookrightarrow w"$ et $w' \neq w"$, alors il existe v tel que $w' \hookrightarrow v$ et $w" \hookrightarrow v$. Dans ce cas, il existe deux facteurs $x^{-1}y$ et $x'^{-1}y'$ de w sur lesquels on a appliqué le pas de renversement pour obtenir respectivement w' et $w"$. Ces facteurs sont soit situés au même endroit dans w, soit disjoints. Dans le premier cas, $w' = w"$. Dans le deuxième cas, w, w', $w"$ s'écrivent:

- $w = w_0...w_i x^{-1} y...x'^{-1} y'...w_k$

- $w' = w_0...w_i f(x,y) f(y,x)^{-1}...x'^{-1}y'...w_k$
- $w" = w_0...w_i x^{-1}y...f(x',y')f(y',x')^{-1}...w_k$

Ainsi, $v = w_0...w_i f(x,y)f(y,x)^{-1}...f(x',y')f(y',x')^{-1}...w_k$ convient.

4 Cohérence et tresses

On s'intéresse maintenant à une notion appelée la cohérence, propriété du complément f, qui nous servira par la suite. Pour la définir, on donnera quelques notations, puis on vérifiera que le complément des tresses est cohérent.

4.1 Définitions

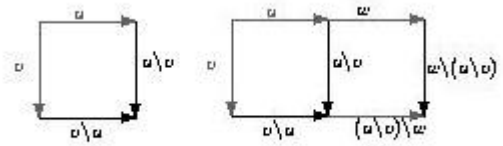
Introduisons tout d'abord quelques notations.

Définition Soit $\equiv^{++}$ la relation sur A^* telle que $u \equiv^{++} v$ si et seulement si $v^{-1}u$ se redresse en ϵ.
On définit aussi l'opération partielle de redressement à gauche $\backslash$: $u\backslash v$ est, s'il existe, le mot de A^+ tel qu'il existe $y \in A^+$ tel que $v^{-1}u$ se redresse en $y(u\backslash v)^{-1}$. On a alors $y = v\backslash u$.

Propriété Pour $u, v, w \in A^+$, on a les égalités suivantes dès lors que l'un des membres de l'égalité existe:

- $w\backslash(u\backslash v) = (uw)\backslash v$
- $v\backslash(uw) = (v\backslash u)((u\backslash v)\backslash w)$

Ces formules sont représentées par les graphes de redressement suivants:

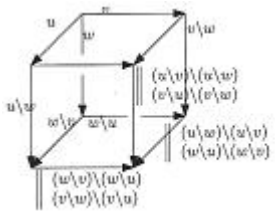


On note que $\backslash$ est une extension sur A^+ de f: pour $x, y \in A$, on a $f(x,y) = x\backslash y$.

Définition Le complément f est dit cohérent si, pour tout $u, v, w \in A^+$,

- soit $((u\backslash v)\backslash(u\backslash w)) = ((v\backslash u)\backslash(v\backslash w))$
- soit aucun des termes de l'égalité n'existe.

La cohérence est parfois appelée condition du cube, dûe à sa représentation graphique:



4.2 Le cas des tresses

Pour les cas particuliers, il est préférable de trouver une condition équivalente, mais plus aisément vérifiable.

Propriété Le complément f est cohérent si et seulement si pour tout $x, y, z \in A$, $f(x, y)\backslash f(x, z) = f(y, x)\backslash f(y, z)$.

Il ne reste plus qu'à faire une vérification au cas par cas pour s'assurer que le complément f défini pour le groupe des tresses $\mathcal{B}_n$, est bien cohérent.

5 Cohérence, complétude

5.1 Lien entre la cohérence et la complétude

A partir de maintenant, on suppose que $\equiv^+$ conserve la longueur des mots.

Définition Un groupe à complément est dit complet si la relation d'équivalence sur A^+ est égale à la restriction de $\equiv^{++}$, c'est-à-dire, si pour tous mots positifs u et v, $u \equiv^+ v \Leftrightarrow u^{-1}v$ se redresse en ϵ.

Propriété On a équivalence entre complétude et cohérence. Plus précisément, on a équivalence entre les trois assertions suivantes:

(i) le groupe à complément f est complet,

(ii) la relation $\equiv^+$ est compatible avec l'opération $\backslash$, autrement dit si $u \equiv^+ u'$ et $v \equiv^+ v'$, alors soit $u\backslash v$ et $u'\backslash v'$ existent et $u\backslash v \equiv^+ u'\backslash v'$, soit aucun des deux n'existent,

(iii) le complément f est cohérent.

5.2 Simplification à gauche

On a également un lien entre la cohérence du complément f et la simplification à gauche du monoïde quotient.

Propriété Soit $u, v \in A^+$. Supposons le complément f cohérent. S'il existe $w \in A^+$ tel que $wu \equiv^+ wv$, alors $u \equiv^+ v$.

Preuve. On a $wu \equiv^+ wv$, donc, comme f est cohérent (donc complet), on a $wu \equiv^{++} wv$. Autrement dit, $v^{-1}w^{-1}wu = (wv)^{-1}wu$ se redresse en ϵ. Or, $w^{-1}w$ se redresse en ϵ. On en déduit que $v^{-1}w^{-1}wu$ se redresse après un certain nombre de pas en $v^{-1}u$, alors,

par propriété de confluence, $v^{-1}u$ se redresse en ϵ, c'est-à-dire $u \equiv^{++} v$, et donc, par complétude de f, $u \equiv^{+} v$.

6 Cohérence et convergence, élément de Garside

A partir de maintenant, on suppose que f est cohérent.
Dans cette partie, on va donner une condition pour que l'algorithme termine : l'existence d'un élément de Garside. On définira d'abord un tel élément, puis on verra en quoi il assure la convergence de l'algorithme de redressement, enfin, on cherchera cet élément dans le groupe des tresses.

6.1 Définition

Définition Soit M un monoïde et $\Delta \in M$. On dit que Δ est un élément de Garside si l'ensemble de ses diviseurs à gauche coïncide avec l'ensemble de ses diviseurs à droite, et que ceux-ci engendrent M

On commence par prouver un lemme.

Lemme Soit A un alphabet et f un complément sur A, f est convergent si et seulement s'il existe $P \subseteq A^+$ tel que $A \subseteq P$ et que P soit clos par $\backslash$.

Preuve. En effet, si f est convergent, $P = A^+$ convient. Réciproquement, si un tel P existe, alors, quel que soit le mot $w \in A^*$, w est de la forme:

$$w = u_1^{e_1}...u_r^{e_r},$$

avec pour tout i, $u_i \in P$ et $e_i = \pm 1$. On raisonne alors par récurrence sur r. Soit $I = \{i \in [|1, r|] | e_i = -1, e_{i+1} = 1\}$. Si I est vide, alors w se redresse en lui-même. Sinon, on pose $i_0 = minI$. La première étape renvoie alors un mot de la forme $u_0^{e_0}...u_{i_0-1}^{e_{i_0-1}} v_{i_0} v_{i_0+1}^{-1}...u_r^{e_r}$. Par hypothèse de récurrence, $u_0^{e_0}...u_{i_0-1}^{e_{i_0-1}} v_{i_0}$ et $v_{i_0+1}^{-1}...u_r^{e_r}$ se réduisent respectivement en xy^{-1} et ts^{-1}. Par confluence, il reste à réduire $xy^{-1}ts^{-1}$. Or, comme P est clos sous $\backslash$, $y^{-1}t$ est un mot de la forme $a_1^{b_1}...a_m^{b_m}$, avec $a_i \in P$ et $m < r$, on peut donc le redresser en $y't'^{-1}$, ce qui assure la convergence de f.

Propriété Soit f un complément cohérent et M le monoïde associé. Si M admet un élément de Garside Δ, alors f est convergent.

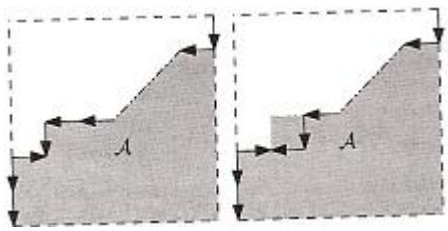
Preuve. Soit S l'ensemble des représentants des diviseurs de Δ. L'ensemble des diviseurs de Δ engendre M, donc S contient A. Il reste à montrer que S est clos sous $\backslash$. Soit $u, v \in S$ il existe u' et v' tels que uv' et vu' représentent Δ. On a donc $uv' \equiv^{+} vu'$ et comme f est cohérent, $uv' \equiv^{++} vu'$. Autrement dit, $u'^{-1}v^{-1}uv'$ se redresse en ϵ. On en déduit d'une part, que $u\backslash v$ existe et d'autre part que

$$\begin{aligned} u(u\backslash v)((u\backslash v)\backslash v')(((u\backslash v)\backslash v')\backslash((v\backslash u)\backslash u')) &\equiv^{+} u(u\backslash v)((v\backslash u)\backslash u')(((v\backslash u)\backslash u')\backslash((u\backslash v)\backslash v')) \\ &\equiv^{+} u(u\backslash(vu'))(((v\backslash u)\backslash u')\backslash((u\backslash v)\backslash v')) \\ &\equiv^{+} vu'(vu'\backslash u)(((v\backslash u)\backslash u')\backslash((u\backslash v)\backslash v')) \end{aligned}$$

$$\equiv^+ vu'(u'\backslash(v\backslash u))(((v\backslash u)\backslash u')\backslash((u\backslash v)\backslash v'))$$
$$\equiv^+ vu'(u'\backslash((v\backslash u)v')) \equiv^+ vu'(vu'\backslash uv').$$

Or, $vu'\backslash uv' = \epsilon$, donc $u(u\backslash v)$ représente un diviseur à gauche de Δ. Les diviseurs à gauche de Δ étant égaux aux diviseurs à droite, $u(u\backslash v)$ représente également un diviseur à droite, donc $u\backslash v$ aussi. On a bien $u\backslash v \in S$, S est clos par $\backslash$.

Remarque On se rammène à construire sur l'ensemble S un nouveau complément $\overline{f}$ tel que $\overline{f}(u,v) = u\backslash v$ qui est donc compatible avec f. En considérant les mots de A^* comme des mots construits sur l'alphabet S, on a construit à partir de $\overline{f}$ un algorithme qui conserve la longueur des mots, ce qui assure la convergence de l'algorithme. Plus précisément, on a si l est la longueur d'un mot de S^*, la terminaison en au plus $l^2/4$ étapes. En effet, si $\mathcal{A}$ est l'aire sous le graphe de redressement d'un mot u, une étape de calcul réduit cette aire d'au moins une unité (plus si c'est une simplification de xx^{-1}) et l'aire initiale est bornée par $l^2/4$.



6.2 Construction de S et de $\overline{f}$, représentation dans le cas des tresses.

Définition Pour tout $i, j \geq 1$, on pose :

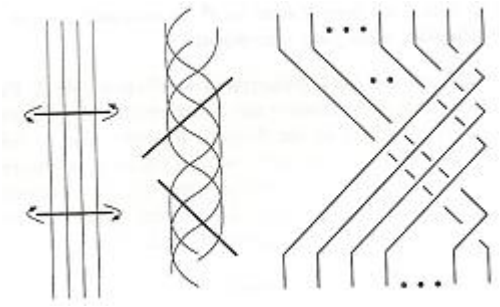
$$\sigma_{i,j} = \begin{cases} \sigma_i\sigma_{i+1}...\sigma_j \text{ si } i_j \geq 0 \\ \sigma_i\sigma_{i-1}...\sigma_j \text{ sinon} \end{cases}$$

Définition Soit p une permutation de $[|1, n+1|]$, on définit le mot $\widetilde{p}$ par récurrence sur l'entier minimal déplacé par p: $\widetilde{Id} = \epsilon$ et $\sigma_{p(k)-1,k}\widetilde{p'}$, où $k = min\{i \in [|1, n+1|] | p(i) \neq i\}$ et $p' = \begin{cases} i \text{ si } i \leq k \\ p(i)+1 \text{ si } k \leq p(i) < p(k) \\ p(i) \text{ sinon.} \end{cases}$ La classe de $\widetilde{p}$ est appellée tresse de permutation p.

On note $\Delta_{n+1} \in \mathcal{B}_{n+1}$ la classe de $\widetilde{\delta}$, $\delta : i \mapsto n+2-i$.

Propriété La tresse Δ_{n+1} est un élément de Garside de $\mathcal{B}^+_{n+1}$.

L'élément Δ_{n+1} s'illustre de la façon suivante : il s'agit d'un enroulement d'un demi-tour des $n+1$ brins.



On va, pour vérifier la proposition précédente, montrer que les tresses de permutation sont exactement les diviseurs à gauche et à droite de Δ_{n+1}.
On admet le lemme suivant :

Lemme Soit f une permutation de $[|1,n|]$, et $i \in [|1,n-1|]$. Si $f^{-1}(i) < f^{-1}(i+1)$, alors $si\widetilde{f}$ et $(i,\widetilde{i+1})f$ représentent la même tresse.

Propriété Soit b une tresse positive. On appelle $perm(b)$ la permutation de $[|1,n|]$ telle que $perm(b)(i) = j$ si et seulement si le brin partant de j arrive en position i. Soit f une permutation de $[|1,n|]$. On appelle $l(f)$ le cardinal de $\{(i,j)|i<j \wedge f(i) > f(j)\}$. Une tresse positive b est une tresse de permutation si et seulement si $l(perm(b))$ est égal à la longueur des mots représentant b.

Preuve. Soit f une permutation et F la tresse de permutation f. Montrons que $l(perm(F)) = |\widetilde{f}|$ par récurrence sur $l(f)$.
Si $l(f) = 0$, alors $f = Id$, $\widetilde{f} = \epsilon$ et $perm(F) = Id$, donc $l(perm(F)) = 0 = |\epsilon|$.
Si $l(f) > 0$, alors il existe i tel que $f^{-1}(i) > f^{-1}(i+1)$. On pose $g = (i,i+1)f$. On a donc $f = (i,i+1)g$ et $l(g) < l(f)$. Soit G la tresse de permutation g. Par hypothèse de récurrence, on a $l(perm(G)) = |\widetilde{g}|$. De plus, d'aprs le lemme, $\widetilde{f} = \sigma_i\widetilde{g}$, donc $|\widetilde{f}| = 1 + |\widetilde{g}| = 1 + l(perm(G)) = l(perm(F))$.

Réciproquement, soit b une tresse positive et B un de ses représentants. On suppose $l(perm(b)) = |B|$. Montrons par récurrence sur $l(perm(b))$ que b est la tresse de permutation $perm(b)$.
Si $l(perm(b)) = 0$, alors $perm(b) = Id$ et $|B| = 0$ donc $B = \epsilon$. On a b est la tresse de permutation Id.
Si $l(perm(b)) > 0$, alors $|B| > 0$, donc b se met sous la forme $b = cl(\sigma_i)c$. On a $perm(b) = (i,i+1)perm(c)$, et si C est un représentant de c, $|B| = |C| + 1$. On a donc $|B| = l(perm(b)) \leq 1 + l(perm(c)) \leq 1 + |C|$. On en déduit $l(perm(b)) = 1 + l(perm(c))$ et $l(perm(c)) = |C|$. Par hypothèse de récurence, c est la tresse de permutation $perm(c)$. Enfin, comme $l(perm(c)) < l((i,i+1)perm(c))$, on a $perm(c)^{-1}(i) > perm(c)^{-1}(i+1)$, et donc, d'après le lemme, b est la tresse de permutation $(i,i+1)perm(c) = perm(b)$, d'où le résultat.

Propriété Soit b une tresse positive. Les assertions suivantes sont équivalentes :

(i) la tresse b est une tresse de permutation,

(ii) b est un diviseur à droite de Δ_n,

(ii) b est un diviseur à gauche de Δ_n.

Preuve. $(i) \Rightarrow (ii)$ On suppose que b est la tresse de permutation f. Premier cas : $f = \delta$, alors $b = \Delta_n$, c'est un de ses diviseurs à droite. Sinon, il existe i tel que $f^{-1}(i) < f^{-1}(i+1)$. On a alors $\sigma_i \widetilde{f} \equiv^+ (i, \widetilde{i+1})f$ et $l((i,i+1)f) = l(f) + 1$. En raisonant par récurrence sur $\frac{n(n-1)}{2} - l(f)$, on construit une permutation g telle que $\widetilde{gf} \equiv^+ \widetilde{g}\widetilde{f}$ et $l(gf) = \frac{n(n-1)}{2}$, c'est-à-dire $gf = \delta$, d'où le résultat.
$(i) \Rightarrow (iii)$ On vient de montrer que, pour toute tresse de permutation b, il existe une tresse de permutation $\phi(b)$ telle que $\phi(b)b = \Delta_n$. Par simplification à gauche de $\mathcal{B}_n^+$, ϕ est injective. L'ensemble des tresses de permutation étant fini, ϕ est bijective. Pour toute tresse de permutation b, on a $b\phi^{-1}(b) = \Delta_n$ donc b est un diviseur à gauche de Δ_n.
$(ii) \vee (iii) \Rightarrow (i)$ On suppose que $\Delta_n = bb'$ avec b et b' des tresses positives, montrons que b et b' sont des tresses de permutation.
Soit f une permutation et i un entier, on a :

$$l((i,i+1)f) = \begin{cases} l(f)+1 \text{ si } f^{-1}(i) < f^{-1}(i+1), \\ l(f)-1 \text{ sinon.} \end{cases}$$

On en déduit que pour toute tresse positive a, $l(perm(a)) \leq |A|$, où A est un représentant de a. On a donc :

$$|bb'| = l(perm(bb')) = l(perm(b)perm(b')) \leq l(perm(b)) + l(perm(b')) \leq |b| + |b'| = |bb'|.$$

On en déduit les égalités : $l(perm(b)) = |b|$ et $l(perm(b')) = |b'|$, donc b et b' sont des tresses de permutation.

Remarque On a un nouvel ensemble de générateurs, les $\sigma_{i,j}$ et un nouveau complément $\overline{f}$ que l'on est capable de construire par récurrence sur la longueur des mots.

Propriété Pour tout $\alpha, \beta \in A^*$, $\alpha \equiv \beta$ est équivalent à l'existence de $u, v \in A^+$ tels que $\alpha_g u \equiv^+ \beta_g v$ et $\alpha_d u \equiv^+ \beta_d v$, où α se redresse en $\alpha_g \alpha_d^{-1}$ et β en $\beta_g \beta_d^{-1}$. En particulier, si $\alpha, \beta \in A^+$, on a : $\alpha \equiv \beta \Leftrightarrow \alpha \equiv^+ \beta \Leftrightarrow \alpha \equiv^{++} \beta$. Autrement dit, $\alpha \equiv \beta$ si et seulement si l'algorithme de redressement appliqué à $\alpha^{-1}\beta$ renvoit ϵ.

Preuve. Pour $\alpha, \beta \in A^*$, on note $\alpha \sim \beta$ si et seulement s'ils vérifient la première partie de la proposition. On vérifie que $\sim$ est une relation d'équivalence. Il suffit donc de prouver l'équivalence pour un ensemble de couples (α, β) générant $\equiv$ en tant que relation d'équivalence. On considèrera donc les paires $(\gamma\alpha\gamma', \gamma\beta\gamma')$, où α, β est de la forme $(xf(x,y), yf(y,x))$, (xx^{-1}, ϵ), ou $(x^{-1}x, \epsilon)$, avec $x, y \in A$.
Dans le premier cas, la compatibilité de $\equiv^+$ avec $\backslash$ donne : $(\epsilon\backslash\gamma\alpha\gamma') \equiv^+ (\epsilon\backslash\gamma\beta\gamma')$ et $(\gamma\alpha\gamma'\backslash\epsilon) \equiv^+ (\gamma\beta\gamma'\backslash\epsilon)$.
Dans le troisième cas, $x^{-1}x$ se redresse en ϵ, ainsi $(\epsilon\backslash\gamma\alpha\gamma') \equiv^+ (\epsilon\backslash\gamma\beta\gamma')$ et $(\gamma\alpha\gamma'\backslash\epsilon) \equiv^+ (\gamma\beta\gamma'\backslash\epsilon)$.
Dans le deuxième cas, notons $u = \epsilon\backslash\gamma$, $v = \gamma\backslash\epsilon$, $u' = \epsilon\backslash\gamma'$, et $v' = \gamma'\backslash\epsilon$. En appliquant les

formules de bases sur $\backslash$ ainsi que celles dûes à la cohérence, on trouve :

$$\begin{aligned}
&\epsilon\backslash(\gamma x x^{-1}\gamma') = u(v\backslash x)((x\backslash v)\backslash(x\backslash u')\\
&\equiv^{+} u(v\backslash x)((v\backslash x)\backslash(v\backslash u')\\
&\equiv^{+} u(v\backslash u')((v\backslash u')\backslash(v\backslash x)\\
&\equiv^{+} (\epsilon\backslash\gamma\gamma')((v\backslash u')\backslash(v\backslash x)
\end{aligned}$$

$$\begin{aligned}
&(\gamma x x^{-1}\gamma')\backslash\epsilon = v'(u'\backslash x)((x\backslash u')\backslash(x\backslash v)\\
&\equiv^{+} v'(u'\backslash x)((u'\backslash x)\backslash(u'\backslash v)\\
&\equiv^{+} v'(u'\backslash v)((u'\backslash v)\backslash(u'\backslash x)\\
&\equiv^{+} (\gamma\gamma'\backslash\epsilon)((u'\backslash v)\backslash(u'\backslash x)
\end{aligned}$$

Il reste à appliquer encore une fois la cohérence pour obtenir $((v\backslash u')v\backslash x) \equiv^{+} ((u'\backslash v)\backslash(u'\backslash x))$.
Dans le cas particulier $\alpha, \beta \in A^{+}$, il suffir d'écrire que $(\alpha\backslash\epsilon) = (\beta\backslash\epsilon) = \epsilon$.
On aboutit au résultat voulu : $x \in A^{*}$ est équivalent à ϵ si et seulement s'il se réduit en ϵ.
En effet, la première partie de l'algorithme de réduction (premier redressement) renvoit le mot $(\epsilon\backslash x)(x\backslash\epsilon)^{-1}$. Or, $x \equiv \epsilon$ est équivalent à $x\backslash\epsilon \equiv \epsilon\backslash x$, donc équivalent à $x\backslash\epsilon \equiv^{++} \epsilon\backslash x$, c'est-à-dire que le deuxième redressement renvoit ϵ si et seulement si $x \equiv \epsilon$.
Il n'est cependant pas nécessaire de travailler sur le système de générateurs S et le complément $\overline{f}$ pour obtenir la convergence de l'algorithme de redressement : si v est de longueur l sur A^{*}, on peut le transformer en un mot de longueur $l' \leq l$ sur S^{*}. Ce mot se redresse en au plus $l^2/4$ étapes par $\overline{f}$, et un pas de $\overline{f}$ peut se voir comme un nombre de pas fini sur f. Ainsi, si N est le nombre maximal de pas de redressement $v^{-1}u$ pour $u, v \in S$ (S est fini), on trouve qu'un redressement de v sur A termine en au plus $Nl^2/4$ pas de f. Par confluence, tout redressement de v se termine donc en au plus $Nl^2/4$.
L'algorithme de réduction se termine donc en $Nl^2/2$ étapes.
Notons que si l'on travaille dans $\mathcal{B}_{\infty}$, la convergence est toujours assurée, mais plus la borne N. Une étude plus précise permet de borner N et de prouver que la complexité du redressement est en $\mathcal{O}(l^2 n log(n))$ dans $\mathcal{B}_n$, et donc en $\mathcal{O}(l^3 log(l))$ dans $\mathcal{B}_{\infty}$.

Conclusion

On aboutit donc à une résolution du problème de mot, qui s'étend au-delà du groupe des tresses. En contrepartie, la solution n'est peut-être pas algorithmiquement optimale : il existe peut-être des algorithmes utilisant mieux la structure du groupe des tresses, ayant donc une meilleure complexité. En particulier, l'algorithme de réduction des poignées termine empiriquement en $\mathcal{O}(nl^2)$, mais on ne sait borner sa complexité qu'en $\mathcal{O}(2^{n^4 l})$.
D'autres approches consistent à dire que les mots ne sont pas des représentants efficaces des tresses et qu'il faut passer en dimension supérieure : il existe une représentation des tresses en tant que 3-polygraphes. Des recherches sont actuellement en cours pour implémenter les structures et les opérations des polygraphes dans des langages ML.

Références

[1] Patrick Dehornoy, *Complete positive group presentations*, Journal of Algebra 268 (2003) 156-197
[2] Patrick Dehornoy *Groups with a complemented presentation*, Journal of Pure and Applied Algebra 116 (1997) 115-137

[3] Patrick Dehornoy *Braids and self-distributivity*, Progress in Mathematics, volume 192; xvi + 624 pages. Birkhauser (2000).