Groupes des tresses
&
Algorithme de réduction des poignées.

Nicolas Curien
&
Louis-Hadrien Robert

Sujet proposé par Patrick Dehornoy

Juin 2006

Mémoire de maitrise. FIMFA 2006

Table des matières

Remerciements

Les auteurs de ce travail tiennent à remercier chaleureusement leur encadrant Patrick Dehornoy pour ses conseils avisés, sa rigueur autant mathématique que typographique, sa disponibilité et sa bonne humeur.

Introduction

La théorie des tresses, dont les fondements on été jetés par Emil Artin (1898-1962) dans un article datant de 1926, provient de l'intuition naturelle d'une tresse. Les groupes des tresses ont des applications dans plusieurs domaines : géométrie, algèbre, topologie mais aussi physique théorique, cryptographie ou encore théorie des noeuds. Bien que simple d'accès car très intuitive, la théorie des tresses regorge de résultats séduisants. Ce travail proposé par P.Dehornoy, est axé sur la preuve de la convergence d'un algorithme dit de réduction des poignées[1].

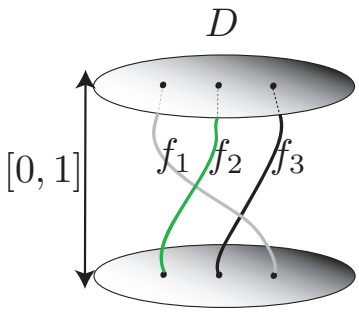
1 Présentation des groupes de tresses

1.1 Présentation géométrique

Comment donner une représentation mathématique de notre idée de tresse ?

Soit $n \geqslant 1$. Définissons une *tresse géométrique* à n brins.

Définition 1.1. *Une tresse géométrique à n brins est un n-uplet de fonctions continues $f_1, ..., f_n$ définies sur $[0,1]$ et à valeurs dans D (D est le disque unité de $\mathbb{C}$) qui vérifient*
- *l'ensemble $\{f_1(0), ..., f_n(0)\} = \{f_1(1), ..., f_n(1)\} = \{x_0, ..., x_n\}$: ce sont les encrages de la tresse ;*
- *pour tout $t \in [0,1]$ et pour tout $i \neq j$ on a $f_i(t) \neq f_j(t)$: les brins ne se rencontrent pas.*



C'est l'idée géométrique d'une tresse classique : n brins dans le cylindre $[0,1] \times D$ qui partent de n trous, s'entrecroisent sans se rencontrer, pour enfin se rattacher aux points d'ancrages initiaux. (voir dessin ci-contre)

L'ensemble des tresses géométriques sera noté $\mathcal{B}_n^g$. Deux tresses géométriques $\alpha = (f_1, ..., f_n)$ et $\beta = (g_1, ..., g_n)$ sont dites *isotopes* (notées $=$) lorsque l'on peut déformer continûment l'une en l'autre, c'est à dire s'il existe ϕ une fonction continue de $[0,1]$ à valeurs dans les tresses géométriques telle que $\phi(0) = \alpha$ et $\phi(1) = \beta$. On munit l'ensemble des tresses géométriques de la concaténation : si $\alpha = (f_1, ..., f_n)$ et $\beta = (g_1, ..., g_n)$ sont deux tresses géométriques alors $\alpha.\beta = (h_1, ..., h_n)$ avec

$$\forall i \in [|1, n|] \quad h_i = \begin{cases} f(2t) \text{ si } 0 \leqslant t \leqslant 1/2, \\ g(2t-1) \text{ si } 1/2 \leqslant t \leqslant 1. \end{cases}$$

Soit B_n^g le quotient de $\mathcal{B}_n^g$ par isotopie, alors $(B_n^g, .)$ est un groupe ; l'inverse de la classe de α est la classe de son image par la réflexion dans le disque D, c'est à dire si $\alpha = (f_1, ..., f_n)$ alors sa réflexion est $\alpha^r = (g_1, ..., g_n)$ avec $g_i(t) = f_i(1-t)$. L'élément neutre est la classe de $\epsilon = (\overline{x_0}, ..., \overline{x_n})$ où $\overline{x_i}$ est la fonction constante égale à x_i sur $[0,1]$. Un élément de B_n^g est appelé une *tresse*. Une tresse est donc une classe d'équivalence de tresses géométriques. On appelle $(B_n^g, .)$ *le groupe des tresses à n brins.*

[1] Algorithme dû à P.Dehornoy

1.2 Présentation algébrique

On caractérise par générateurs et relations le groupe des tresses de façon algébrique.

Définition 1.2. *Introduisons le groupe abstrait suivant :*

$$B_n^a = \left\langle \sigma_1, ..., \sigma_n \;\middle|\; \begin{array}{ll} (1) & \sigma_i\sigma_j = \sigma_j\sigma_i \text{ si } |i-j| \geqslant 2 \\ (2) & \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \text{ si } |i-j| = 1 \end{array} \right\rangle.$$

C'est à dire que B_n^a est isomorphe en tant que groupe au quotient du groupe libre de rang n, F_n, par la plus petite relation d'équivalence compatible avec le produit et contenant les relations (1) *et* (2)*.*

On dira qu'un mot sur l'alphabet $\sigma_1^{\pm 1}, ...\sigma_n^{\pm 1}$ est un *mot de tresse*, l'ensemble des mots de tresse est noté $\mathcal{B}_n^a$ ou plus simplement $\mathcal{B}_n$. Deux mots de tresse équivalents (ils sont dans la même classe de B_n^a) seront notés $w \equiv w'$.

FIG. 1 – Voici un dessin de deux générateurs : à gauche σ_1 à droite σ_1^{-1} qui permutent les deux premiers brins. Les générateurs s'interprètent géométriquement : l'élément σ_i correspond à une tresse géométrique (fixée dans toute la suite) qui permutent les brins i et $i+1$ en faisant passer le premier par dessus le second.

Soit w un mot de tresse, grâce à l'interprétation géométrique des générateurs, nous pouvons lui associer de façon injective une tresse géométrique que nous notons $\phi(w)$. Nous avons donc une injection

$$\phi : \mathcal{B}_n^a \hookrightarrow \mathcal{B}_n^g.$$

On peut vérifier que les relations (1) et (2) sont vraies dans B_n^g, c'est-à-dire que les tresses géométriques représentées par les mots intervenant dans les relations (1) et (2) sont isotopes. Voir le dessin ci-dessous.

Il s'en suit que ϕ passe au quotient :

$$\overline{\phi} : B_n^a \hookrightarrow B_n^g$$

Nous disons que $w \in \mathcal{B}_n^a$ représente la tresse α si $\overline{\phi(w)} = \alpha$ (la tresse géométrique associée à w est un représentant de la classe α), nous notons $\alpha = \overline{w}$. Cependant nous écrivons σ_i pour $\overline{\sigma_i}$. Attention, σ_i peut donc signifier soit la lettre σ_i d'un mot de tresse w, soit la tresse que représente σ_i.

Théorème 1.3 (Artin)**.** *L'injection $\overline{\phi}$ est un isomorphisme de groupe. Ce fait peut se traduire ainsi : soient deux mots w et w' qui représentent deux tresses géométriques α et β ; alors $\alpha = \beta$ si et seulement $w \equiv w'$; c'est à dire s'il existe une suite finie de mot $w = w_0, ..., w_n = w'$ et l'on passe du mot w_i à w_{i+1} avec l'aide d'une relation* $(1), (2)$ *ou $\sigma_i^{\pm 1}\sigma_i^{\mp 1} = \epsilon$.*

Un élément de B_n^a est appelé *tresse* (tout comme un élément de B_n^g), c'est une classe d'équivalence de mot de tresse. À partir de maintenant nous appelons B_n le groupe des tresses à n

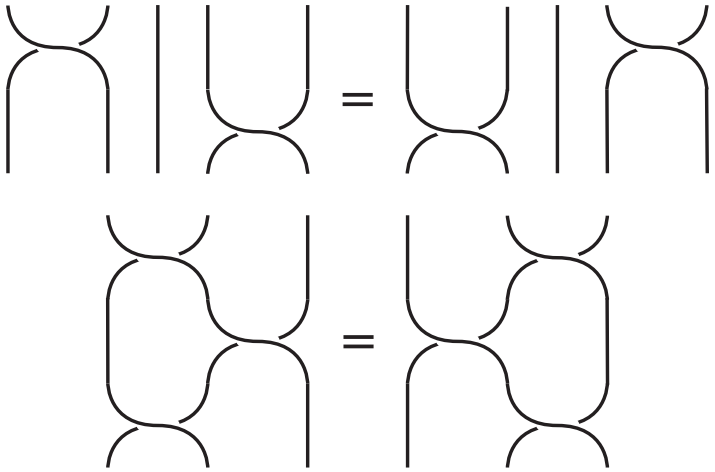


FIG. 2 – Les relations (1) et (2) sur les exemples $\sigma_1\sigma_4 = \sigma_4\sigma_1$ et $\sigma_1\sigma_2\sigma_1 = \sigma_2\sigma_1\sigma_2$

brins (que ce soit B_n^g ou B_n^a). Une tresse est donc une classe d'équivalence de mots de tresse ou une classe d'isotopie de tresses géométriques.

Définition 1.4. *Définissons également le monoïde B_n^+ par la présentation :*

$$B_n^+ = \left\langle \sigma_1, ..., \sigma_n \;\middle|\; \begin{array}{ll} (1) & \sigma_i\sigma_j = \sigma_j\sigma_i \text{ si } |i-j| \geqslant 2 \\ (2) & \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \text{ si } |i-j| = 1 \end{array} \right\rangle^+ .$$

Remarque 1.5. *Attention, cette présentation, contrairement à celle faite pour B_n^a est une présentation de monoïde.*

Un mot sur l'alphabet $\sigma_1, ..., \sigma_n$ (resp. $\sigma_1^{-1}, ..., \sigma_n^{-1}$) est appelé mot de tresse *positif* (resp. *négatif*), il représente une tresse appelée *positive* (resp.*negative*). Nous notons B_n^+ l'ensemble des tresses positives de B_n. Nous verrons ultérieurement des résultats profonds sur ce monoïde B_n^+.

Remarque 1.6 (Plongement dans $\mathcal{S}_n$)**.** *Chaque tresse induit une permutation des n brins, on a donc un homomorphisme de groupe*

$$\phi : B_n \longrightarrow \mathcal{S}_n.$$

Le morphisme ϕ est surjectif, mais pas injectif : la tresse σ_1^2 induit la permutation triviale sur les deux brins sans être isotope à la tresse triviale. [2]

1.3 Injection de B_n^+ dans B_n

Nous donnons sans démonstration quelques résultats sur les groupes de tresses.

[2] Une présentation du groupe symétrique peut être déduite de la présentation de B_n en ajoutant $\sigma_i^2 = 1$

Définition 1.7. *Soit w un mot positif (resp négatif). On dit qu'un mot w' est* positivement équivalent *(resp négativement) à w, ou déduit de w par* équivalence positive *(resp* négative*) si l'on peut passer du mot au mot w' en utilisant que les relations $\sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j$ et $\sigma_k\sigma_l = \sigma_l\sigma_k$ (resp $\sigma_i^{-1}\sigma_j^{-1}\sigma_i^{-1} = \sigma_j^{-1}\sigma_i^{-1}\sigma_j^{-1}$ et $\sigma_k^{-1}\sigma_l^{-1} = \sigma_l^{-1}\sigma_k^{-1}$) pour $|i-j| = 1$ et $|k-l| \geqslant 2$.*

Théorème 1.8 (Garside)**.** *L'application naturelle $B_n^+ \to B_n$ est une injection :*

$$B_n^+ \hookrightarrow B_n.$$

En d'autres termes, si deux mots de tresse positifs w et w' sont équivalents dans B_n, alors ils sont aussi positivement équivalents. L'équivalence positive est plus restrictive que l'équivalence dans B_n car rester dans B_n^+ interdit certaines opérations comme l'ajout de séquences $\sigma_i\sigma_i^{-1}$. Cette injection permet de voir B_n^+ comme un sous monoïde de B_n, ce qui n'est pas du tout évident.

On peut maintenant définir plusieurs notions :

Définition 1.9. *Soit α une tresse positive représentée par un mot w positif, alors $|w|$ ne dépend pas de w : c'est la* longueur *de la tresse α.*

Démonstration. Soit w et w' deux mots représentant α. D'après l'injection $B_n^+ \hookrightarrow B_n$, w et w' sont équivalents dans B_n^+. On conclut en remarquant que les relations dans B_n^+ conservent la longueur. □

Définition 1.10. *Soit γ une tresse positive. Une tresse α est un* diviseur à gauche *de γ si α est une tresse positive et s'il existe β une tresse positive telle que $\alpha\beta = \gamma$.*

Remarque 1.11. *La notion de divisibilité n'a de sens que pour les tresses positives. Si α est une tresse positive de longueur m, alors α admet au plus n^m diviseurs. (Nous reverrons cette inégalité plus tard)*

L'étude de B_n^+ est importante, nous verrons plus tard que toute tresse α se met sous la forme $\alpha = \beta_1^{-1}\beta_2$ avec β_1 et β_2 des tresses positives. Il y a même unicité si β_1 et β_2 n'admettent pas de facteurs communs. C'est à dire que B_n est le groupe des fractions du monoïde B_n^+.

1.4 Problème de mot

La présentation algébrique du groupe B_n est commode car manipuler une tresse revient à manipuler un mot. Un problème réside dans l'identification de deux tresses : soit w et w' deux mots de tresse représentant deux tresses α et β ; alors, d'après les résultats d'Artin, $\alpha = \beta$ (les deux tresses sont identiques) si et seulement si $ww'^{-1} \equiv \epsilon$. Savoir si un mot de tresse w représente la tresse triviale s'appelle le *problème de mot.*

Le problème de mot est fondamental pour les groupes présentés par générateurs et relations. Il existe des exemples de présentations finies pour lesquelles le problème de mot n'est pas décidable. Dans les groupes automatiques (les groupes B_n en sont un exemple), ce problème a une solution efficace. Nous nous proposons dans la suite d'expliquer le fonctionnement d'un algorithme proposé par P. Dehornoy permettant de résoudre le problème de mot dans les groupes de tresses en s'appuyant sur des propriétés spécifiques de ces groupes.

1.5 Intérêt des groupes de tresses

Les groupes de tresses apparaissent dans de nombreuses branches des mathématiques (en physique, biologie...) et sous différents aspects, d'où leur importance.

On note D_n le disque à n trous. Alors le groupe des tresses est homéomorphe au quotient par isotopie du groupe des homéomorphismes de D_n sur lui-même préservant l'orientation, laissant fixe ∂D_n et préservant globalement les n trous. [3]

Les groupes de tresses apparaisent de bien d'autres façons et tous ces nouveaux aspects des groupes de tresses permettent de mieux en comprendre la structure.

2 Résultats sur les groupes de tresses

On présente ici divers résultats sur les groupes de tresses qui sont nécessaire pour établir la convergence de l'algorithme de réduction des poignées (ARDP).

Définition 2.1. *On note $\mathcal{B}_n$ l'ensemble de mots de tresse, et $\mathcal{B}_n^+$ l'ensemble des mots de tresse positifs.*

2.1 Résultats de Garside

Les résultats présentés ici sont tirés d'un article écrit en 1966 par Garside, on raisonne essentiellement sur les mots de tresse positifs, sur leurs multiples et sur leurs diviseurs.

On introduit quelques notations pour les mots de tresse : si w et w' sont des mots de tresse, on dira que $w \equiv w'$ si ils représentent la même tresse, on dit que $w = w'$ si ils sont égaux lettre à lettre, enfin si w et w' sont deux mots positifs et qu'ils sont positivement équivalents , c'est à dire qu'ils représentent le même élément dans le monoïde des tresses positives, on notera $w \equiv^+ w'$[4] cette dernère notation est provisoire).

2.1.1 Les mots de tresse positifs

Remarque 2.2. *Deux mots positifs, équivalents, ont même longueur (on pourra aller voir la définition 1.9*

Théorème 2.3. *Soient w et w' deux mots de tresse positifs et i et k deux entiers compris entre 1 et $n-1$. On suppose que $\sigma_i w \equiv^+ \sigma_k w'$ alors :*

1. *si $k = i$, on a $w \equiv^+ w'$,*
2. *si $|k-i| > 1$, il existe un mot z tel que $w \equiv^+ \sigma_k z$ et $w' \equiv^+ \sigma_i z$,*
3. *si $|k-i| = 1$, il existe un mot z tel que $w \equiv^+ \sigma_k\sigma_i z$ et $w' \equiv^+ \sigma_i\sigma_k z$.*

Démonstration. Ce résultat se montre par récurrence sur la longueur des mots w et w', la preuve n'est pas très compliquée, mais un peu longue et fastidieuse. □

Définition 2.4. *Si $w = s_1 s_2 \dots s_t$ est un mot de tresse, on appelle* miroir de w *et on notera* $\mathrm{rev}(w)$ *le mot $s_t s_{t-1} \dots s_2 s_1$.*

[3] La propriété A énoncée ultérieurement se comprend plus facilement une fois cette identification faite.

[4] Nous allons voir dans la première sous partie que $\equiv^+$ est la restriction de $\equiv$ à $\mathcal{B}_n^+$

Remarque 2.5. *On a immediatement que* $w \equiv w'$ *si et seulement si* $\mathrm{rev}(w) \equiv \mathrm{rev}(w')$

En utilisant, les miroirs, on a facilement un théorème analogue au théorème 2.3.

Théorème 2.6. *Soient* w *et* w' *deux mots de tresse positifs et* i *et* k *deux entiers compris entre* 1 *et* $n-1$. *On suppose que* $w\sigma_i \equiv^+ w'\sigma_k$ *alors :*

1. *si* $k = i$, *on a* $w \equiv^+ w'$,
2. *si* $|k-i| > 1$, *il existe un mot* z *tel que* $w \equiv^+ z\sigma_k$ *et* $w' \equiv^+ z\sigma_i$,
3. *si* $|k-i| = 1$, *il existe un mot* z *tel que* $w \equiv^+ z\sigma_k\sigma_i$ *et* $w' \equiv^+ z\sigma_i\sigma_k$.

On utilise les deux théorèmes précédents pour avoir, un résultat plus compact :

Théorème 2.7. *Soient,* u, v, z, t, w *et* w' *des mots de tresse positifs tels que* $u \equiv z$, $v \equiv t$ *et* $uwv \equiv zw't$ *alors* $w \equiv w'$.

2.1.2 Le mot Δ

Intuitivement, le mot Δ représente la tresse de demi-tour (voir la figure 3). Ses propiétés remarquable nous permettent de montrer notamment que la tresse représentée par Δ^2 est dans le centre de B_n

Définition 2.8. *On appelle* Π_s *le mot,* $\sigma_1\sigma_2\ldots\sigma_s$, Δ_r *le mot* $\Pi_r\Pi_{r-1}\ldots\Pi_1$ *et on note la plupart du temps* Δ *pour* Δ_{n-1}.

Définition 2.9. *On appelle* reflexion, *le morphisme* $\mathfrak{R}$ *alphabétique définit par :*

$$\mathfrak{R}\sigma_i^{\pm 1} = \sigma_{n-i}^{\pm 1}.$$

Lemme 2.10. *Pour tout entiers* s *et* t *vérifiant,* $1 < s \leqslant t < n$, *on a :* $\sigma_s\Pi_t \equiv^+ \Pi_t\sigma_{s-1}$

Démonstration.

$$\begin{aligned}
\sigma_s\Pi_t &\equiv^+ \sigma_s(\sigma_1\ldots\sigma_{s-2})\sigma_{s-1}\sigma_s(\sigma_{s+1}\ldots\sigma_t)\\
&\equiv^+ (\sigma_1\ldots\sigma_{s-2})\sigma_s\sigma_{s-1}\sigma_s(\sigma_{s+1}\ldots\sigma_t)\\
&\equiv^+ (\sigma_1\ldots\sigma_{s-2})\sigma_{s-1}\sigma_s\sigma_{s-1}(\sigma_{s+1}\ldots\sigma_t)\\
&\equiv^+ (\sigma_1\ldots\sigma_{s-2})\sigma_{s-1}\sigma_s(\sigma_{s+1}\ldots\sigma_t)\sigma_{s-1}\\
&\equiv^+ \Pi_t\sigma_{s-1}.
\end{aligned}$$

□

Lemme 2.11. *Si* $t < n$, *on a les résultats suivants :*

(i) $\sigma_1\Delta_t \equiv^+ \Delta_t\sigma_t$,

(ii) $\sigma_s\Delta \equiv^+ \Delta\mathfrak{R}\sigma_s$,

(iii) $\sigma_s^{-1}\Delta \equiv \Delta(\mathfrak{R}\sigma_s)^{-1}$,

(iv) $\sigma_s\Delta^{-1} \equiv \Delta^{-1}\mathfrak{R}\sigma_s$,

(v) $\sigma_s^{-1}\Delta^{-1} \equiv \Delta^{-1}(\mathfrak{R}\sigma_s)^{-1}$.

Démonstration. Les démonstrations sont des calculs du même type que ceux fait dans la preuve du lemme 2.10, nous les laissons au lecteur, cependant, il est utile de visualiser la tresse correspondant au mot Δ, et ainsi comprendre d'où viennent concrètement les résultats.

□

En concaténant les résultats précédents, on obtient :

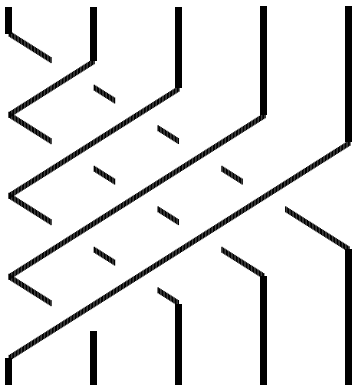


FIG. 3 – Le mot Δ.

Théorème 2.12. *1. Pour tout entier positif m et tout mot de tresse positif w, on a :*

$$w\Delta^{2m} \equiv^+ \Delta^{2m}w \quad et \quad w\Delta^{2m+1} \equiv^+ \Delta^{2m+1}\mathfrak{R}(w).$$

2. Pour tout entier m (positif ou negatif) et tout mot de tresse w, on a :

$$w\Delta^{2m} \equiv \Delta^{2m}w \quad et \quad w\Delta^{2m+1} \equiv \Delta^{2m+1}\mathfrak{R}(w).$$

Lemme 2.13. *On a* $\mathrm{rev}\Delta \equiv^+ \Delta$ *et* $\mathfrak{R}\Delta \equiv^+ \Delta$.

Démonstration. Pour la première affirmation le théorème 2.12 nous indique :

$$(\mathfrak{R}\Delta)\Delta \equiv^+ \Delta\mathfrak{R}(\mathfrak{R}\Delta)$$

et le théorème 2.7 permet de conclure.

Pour la deuxième affirmation, on raisonne par récurrence, on suppose que pour r $\mathrm{rev}\ \Delta_r \equiv^+ \Delta_r$. On a

$$\begin{array}{rcl} \mathrm{rev}\Delta_{r+1} & = & \mathrm{rev}((\sigma_1 \dots \sigma_{r+1})\Delta_r) \\ & = & \mathrm{rev}(\Delta_r)\mathrm{rev}(\sigma_1 \dots \sigma_{r+1}) \\ & \equiv^+ & \Delta_r(\sigma_{r+1} \dots \sigma 1) \\ & \equiv^+ & \Pi_r\Pi_{r-1} \dots \Pi_1(\sigma_{r+1} \dots \sigma 1) \end{array}$$

Or σ_{r+1} commute avec $\Pi_{r-1} \dots \Pi_1$, σ_r commute avec $\Pi_{r-2} \dots \Pi_1$, donc

$$\mathrm{rev}\Delta_{r+1} \equiv^+ \Pi_r\sigma_{r+1}\Pi_{r-1}\sigma_r \dots \Pi_2\sigma_3\Pi_1\sigma_2\sigma_1 = \Delta_{r+1}$$

C'est donc bon pour l'héridité, le résultat est vrai pour $r = 1$, le lemme est donc démontré. □

Lemme 2.14. *Pour tout entier r compris entre 1 et $n-1$, il existe u_r et v_r deux mots de tresse positifs tels que $\sigma_r v_r \equiv^+ u_r\sigma_r \equiv^+ \Delta$.*

Démonstration. On a $\Delta = \Pi_n\Pi_{n-1} \dots \Pi_2\Pi_1$, donc $u_1 = \Pi_n\Pi_{n-1} \dots \Pi_2$ convient. Soit t un entier strictement plus petit que n. Supposons qu'un mot de tresse positif w s'écrive en fonction des σ_i pour i allant de 1 à $t-1$, on peut écrire $w = f(\sigma_1, \sigma_2, \dots, \sigma_{t-1})$, alors le lemme 2.10 donne : $\Pi_t f(\sigma_1, \dots, \sigma_{t-1}) \equiv^+ f(\sigma_2, \dots, \sigma_t)\Pi_t$. On choisit $w = \Pi_{t-1} \dots \Pi_1$, on a donc :

$$\begin{array}{rcl} \Delta & = & \Pi_{n-1} \dots \Pi_{t+1}\Pi_t f(\sigma_1, \dots, \sigma_{t-1}) \\ & \equiv^+ & \Pi_{n-1} \dots \Pi_{t+1} f(\sigma_2, \dots, \sigma_t)\Pi_t \\ & \equiv^+ & \Pi_{n-1} \dots \Pi_{t+1} f(\sigma_2, \dots, \sigma_t)\sigma_1 \dots \sigma_{t-1}\sigma_t. \end{array}$$

Ainsi, $u_t = \Pi_{n-1} \dots \Pi_{t+1} f(\sigma_2, \dots, \sigma_t)\sigma_1 \dots \sigma_{t-1}$ convient. En posant $v_t = \mathrm{rev}(u_t)$, et en utilisant le lemme 2.13, on a le résultat. □

Corollaire 2.15. *Soit w un mot de tresse positif, alors pour tout entier r compris entre 1 et $n-1$, il existe un mot t_r tel que $\Delta w \equiv^+ t_r\sigma_r$*

Démonstration. En effet, on a :

$$\Delta w \equiv^+ (\mathfrak{R}w)\Delta \equiv^+ (\mathfrak{R}w)u_r\sigma_r \equiv^+ t_r\sigma_r$$

□

Corollaire 2.16. *Soit w et w' deux mots de tresse positifs, alors il existe v et v' deux mots de tresse positifs tels que $vw \equiv^+ v'w'$, autrement dit, il existe un multiple à gauche commun à w et w'.*

Démonstration. On écrit, $w = s_1s_2\dots s_k$ et $w' = s'_1s'_2\dots s'_m$, et on utilise le corollaire précédent m fois, ce qui donne :

$$\Delta^m w \equiv^+ \Delta^{m-1}t_1s'_m \equiv^+ \Delta^{m-2}t_2s'_{m-1}s'_m \equiv^+ \dots \equiv^+ t_mw'.$$

En prenant $v = \Delta^m$ et $v' = t_m$, on a le résultat : □

Remarque 2.17. *On a bien sûr un théorème analogue pour les facteurs à droite.*

Le corollaire 2.16 et le théorème 2.7 montre que le monoïde B_n^+ rentre dans les hypothèse du théorème de Öre et donc que B_n^+ s'injecte dans B_n, ce qui est signifié dans le théorème suivant :

Théorème 2.18 (Premier résultat de Garside)**.** *Deux mots de tresse positifs équivalents, sont positivement équivalents.*

Théorème 2.19. *Soit w un mot de tresse, il existe u et v deux mots de tresse positifs tels que*

$$w \equiv^+ uv^{-1}.$$

Démonstration. D'après le théorème 2.12, le mot représenté par Δ^2 appartient au centre de B_n. Soit w un mot de tresse et m le nombre de σ_i^{-1} dans le mot w. On peut donc écrire $w = t_1\sigma_{i_1}^{-1}t_2\sigma_{i_2}^{-1}\dots t_m\sigma_{i_m}^{-1}t_{m+1}$. On a :

$$\begin{aligned} w &\equiv w\Delta^{2m}\Delta^{-2m} \\ &\equiv t_1\sigma_{i_1}^{-1}\Delta^2t_2\sigma_{i_2}^{-1}\Delta^2\dots t_m\sigma_{i_m}^{-1}\Delta^2t_{m+1}\Delta^{-2m} \\ &\equiv t_1u_{i_1}\Delta t_2t_2u_{i_2}\Delta\dots t_mu_{i_m}\Delta t_{m+1}\Delta^{-2m} \end{aligned}$$

où les u_{i_s} sont ceux du lemme 2.14. On a donc bien le résultat annoncé. □

On constate que la méthode utilisée dans la preuve n'est pas optimale quant à la longueur des mots u et v. Il y un résultat plus fort que nous admettons :

Théorème 2.20 (Deuxième résultat de Garside)**.** *Soit α une tresse, alors, il existe un unique couple de tresses (β,γ) de tresses positives tel que $\alpha = \beta\gamma^{-1}$ et β et γ n'ont pas de facteur commun non trivial, c'est à dire que si δ, η et ν sont trois tresses positives telles que $\beta = \eta\delta$ et $\gamma = \nu\delta$ alors $\delta = \epsilon$.*

Remarque 2.21. *On a un résultat similaire à gauche ie $\alpha = \beta'^{-1}\gamma'$*

2.2 La propriété A

La propriété A donne une condition suffisante pour qu'un mot de tresse ne soit pas équivalent à ϵ, cette propriété est essentielle pour comprendre le sens de l'algorithme de réduction des poignées dans lequel on tente de se ramener au cas d'application de cette condition.

Définition 2.22. *Un mot de tresse w est σ_i-*positif *(resp* négatif*) si w ne contient aucun $\sigma_j^{\pm 1}$ pour $j < i$, contient au moins un σ_i mais aucun σ_i^{-1} (resp des σ_i^{-1} mais aucun σ_i). Une tresse α représentée par un mot σ_i-positif (resp négatif) est dite σ_i-*positive *(resp* négative*)*

Théorème 2.23 (Propriété A)**.** *Un mot de tresse σ_1-positif ne représente pas la tresse triviale.*

2.2.1 Coloriage de B_n^+

Pour montrer la propriété A, nous allons colorier les tresses. L'idée est d'attribuer une couleur – un élément d'un ensemble S – à chaque brin au départ, et de faire évoluer ces couleurs quand les brins se croisent. Il faut définir, la règle de changement de couleurs. On la définit sur la figure 4[5].

FIG. 4 – Premier coloriage.

Quelles les propriétés doit vérifier l'opérateur $\star$ pour être compatible avec $\equiv^+$. Pour cela il faut et il suffit que $\star$ soit compatible avec les relations de tresses :

$$\sigma_i\sigma_j = \sigma_j\sigma_i \text{ pour } |i-j| > 1 \quad \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \text{ pour } |i-j| = 1.$$

La première relation n'impose pas de condition à $\star$, en revanche, comme on le voit sur la figure 5, la deuxième impose :

$$\forall x, y, z \in S \quad x \star (y \star z) = (x \star y) \star (x \star z).$$

Définition 2.24. *Soit X un ensemble, et $\star$ une loi de composition interne sur X. On dit que $(X, \star)$ est un LD-*système *si :*

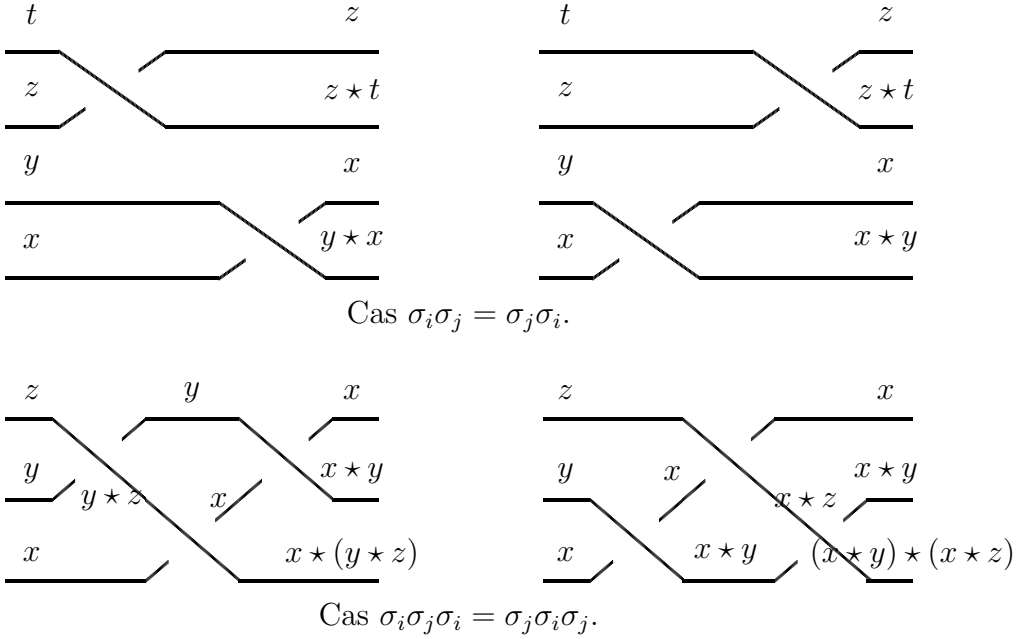
$$\forall x, y, z \in X \quad x \star (y \star z) = (x \star y) \star (x \star z).$$

Exemple 2.25.
- *$S = \{x\}$ et $x \star x = x$ définit le LD-système trivial.*
- *$S = \mathbb{Z}$ et $x \star y = \max(x, y)$ définit un LD-système*
- *Soit T_1 l'ensemble des termes bien formés sur l'alphabet $\{x, \star, (,)\}$ et soit $\mathcal{F}_1$ l'ensemble T_1 quotienté par les les relations des LD-systèmes. $\mathcal{F}_1$ est appelé LD-système libre à un générateur.*

Si S est un LD-système, on va définir le coloriage de B_n^+ par S : On considère l'action (à droite) de $\mathcal{B}_n$ sur S^n définit récursivement par :

$$\vec{x}.\epsilon = \vec{x} \text{ et } \vec{x}.\sigma_i w = (x_1, \ldots, x_i \star x_{i+1}, x_i, x_{i+2}, \ldots, x_n).w$$

[5]On constate que sur le dessin, on construit le coloriage qu'avec les σ_i^{+1}, c'est pour cela qu'on se contente de colorier B_n^+ pour l'instant



Cas $\sigma_i\sigma_j = \sigma_j\sigma_i$.

Cas $\sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j$.

FIG. 5 – Relation de coloriage.

Lemme 2.26. *Si S est un LD-système, le coloriage de $\mathcal{B}_n^+$ par S est compatible avec le relations de tresses, autrement dit, si w et w' sont deux mots de tresses positifs équivalents, et si $\vec{x}$ appartient à S^n, alors, $\vec{x}.w = \vec{x}.w'$. L'action de $\mathcal{B}_n^+$ sur S^n passe au quotient et induit une action de B_n^+ sur S^n.*

Définition 2.27. *Si S est un LD-système et β une tresse de B_n^+, on appelle* S-coloriage de β *tout couple $(\vec{x}, \vec{y})$ tel que $\vec{x}.w = \vec{y}$ où w est un mot de tresse représentant β.*

2.2.2 Coloriage de $\mathcal{B}_n$

On sait désormais colorier les tresses du monoïde B_n^+. Nous voulons étendre ce coloriage au groupe B_n. Pour que le coloriage obtenu soit cohérent, il faut imposer :

$$\forall i, \quad \vec{x}.\sigma_i\sigma_i^{-1} = \vec{x}.\sigma_i^{-1}\sigma_i = \vec{x}$$

Définition 2.28. *Si S est un LD-système et $w = s_1 s_2 \dots s_p$ un mot de tresse de $\mathcal{B}_n$, on appelle* S-coloriage de w *tout couple $(\vec{x}, \vec{y})$ tel qu'il existe $\vec{x_0}, \vec{x_1}, \dots, \vec{x_p}$ tels que $\vec{x_0} = \vec{x}$, $\vec{x_p} = \vec{y}$ et :*

$$\begin{cases} si\ s_k = \sigma_i^{+1}, & s_{k+1} = s_k.\sigma_i \\ si\ s_k = \sigma_i^{-1}, & s_k = s_{k+1}.\sigma_i \end{cases}$$

où l'on interprète . comme l'opérateur de l'action de $\mathcal{B}_n^+$ sur S^n.

Une première idée est de demander au LD-système qui colorie d'avoir une propriété d'inversibilité, par exemple :

$$\forall x, y \in S,\ \exists z \in S,\ y \star z = x.$$

Une telle méthode permet effectivement de colorier $\mathcal{B}_n$ et même B_n, mais, on le verra plus tard, elle n'est d'aucun secours pour montrer la propriété A.

La bonne technique consiste à utiliser la factorisation de w sous la forme $u^{-1}v$: Si $w \in \mathcal{B}_n$ s'écrit $u^{-1}v$, et si $\vec{z}$ appartient à S^n alors, $(\vec{z}.u, \vec{z}.c)$ est un coloriage de w.

Attention, $\mathcal{B}_n$ n'agit pas sur S^n, à strictement parler. En effet cette action est incomplète pour deux raisons :

FIG. 6 – Deux cas de figure possibles

On constate que, quand on concatène les deux diagrammes précédents, dans un ordre ou dans l'autre, on obtient, deux coloriages : $((x,y),(x,y))$ et $((x\star y,x),(x\star y,x))$ qui sont deux coloriages du mot vide.

- d'une part, pour $w\in\mathcal{B}_n$ et $\vec{x}\in S$ donné, rien ne prouve l'existence d'un $\vec{y}$ tel que $(\vec{x},\vec{y})$ soit un coloriage de w.
- d'autre part, et c'est même plus inquiétant, étant donné un mot w, rien n'indique l'existence d'un $\vec{x}$ et d'un $\vec{y}$ tels que $(\vec{x},\vec{y})$ soit un coloriage de w.

Plus précisément, l'écriture $w\equiv u^{-1}v$ ne garantit pas qu'un coloriage de $u^{-1}v$ convienne pour w. Il se trouve, que l'on peut toujours colorier w. Utilisons le retournement à gauche :

Définition 2.29. *Soient w et w' deux mots de tresse, on dit que w* est retournable à gauche en w'*, si on peut construire w' en remplaçant récursivement les facteurs de type $\sigma_i\sigma_j^{-1}$ par des facteurs $f(\sigma_j,\sigma_i)^{-1}f(\sigma_i,\sigma_j)$, où*

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

On note $w\looparrowright w'$.

On définit de même un retournement à droite.

Théorème 2.30. *Soit w un mot de tresse, alors il existe deux mots u et v (resp. u' et v') dans $\mathcal{B}_n^+$ tel que w se retourne à gauche en $u^{-1}v$ (resp. à droite en $u'v'^{-1}$).*

Démonstration. Voir l'annexe A. □

Lemme 2.31. *Soient w et w' deux mots de tresse. Si $w\looparrowright w'$ alors, si $(\vec{x},\vec{y})$ est un S-coloriage pour w' alors, $(\vec{x},\vec{y})$ est un S-coloriage pour w.*

Démonstration. Il suffit de démontrer le résultat dans le cas où l'on passe de w à w' en une étape. Si la transformation consiste à remplacer $\sigma_i\sigma_j^{-1}$ par $\sigma_j^{-1}\sigma_i$, (cas $|i-j|=1$), le résultat est évident. De même si l'étape consiste à remplacer $\sigma_i\sigma_i^{-1}$ par ϵ. Reste le cas où l'on remplace $\sigma_i\sigma_j^{-1}$ par $(\sigma_i\sigma_j)^{-1}\sigma_j\sigma_i$, alors on se convainc sur les diagrammes de la figure 7 que le résultat est vrai.

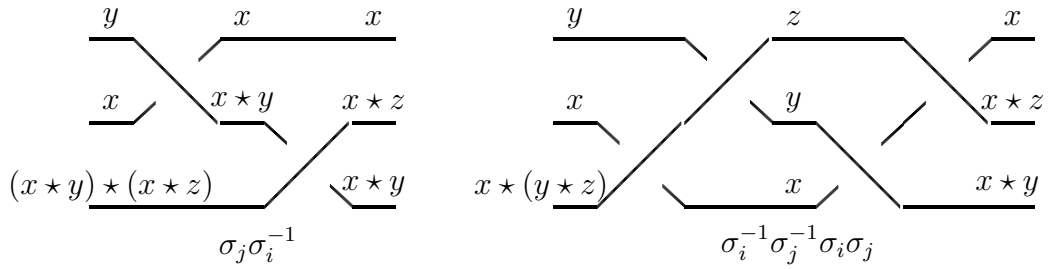


FIG. 7 – Coloriage de $(\sigma_j\sigma_i)^{-1}\sigma_i\sigma_j$et de $\sigma_j\sigma_i^{-1}$

On a le plus général des coloriages pour $(\sigma_i\sigma_j)^{-1}\sigma_j\sigma_i$, on constate qu'il convient pour $\sigma_i\sigma_j^{-1}$. □

Des deux résultats précédents on tire immédiatement :

Théorème 2.32. *Soit S un LD-système. Alors tout mot de tresse admet un S-coloriage.*

Ne perdons pas de vue que l'on veut montrer la propriété A, c'est à dire qu'un mot σ_1-positif ne représente pas la tresse triviale. Nous allons essayer de traduire cette propriété en terme de coloriage.

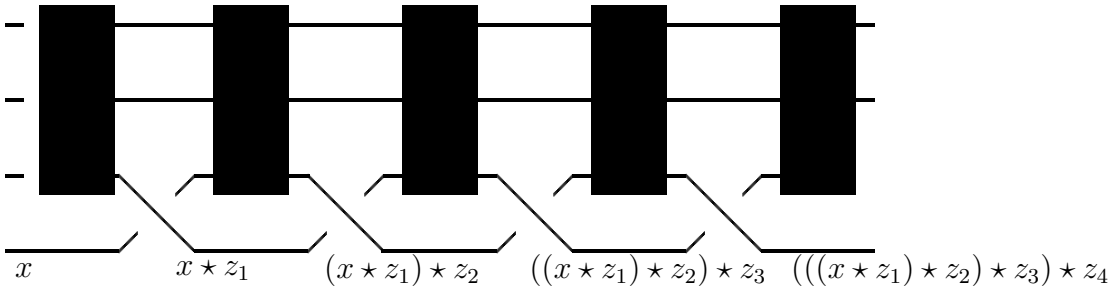


FIG. 8 – Coloriage d'un mot de tresse σ_1-positif

La figure 8 donne le lemme suivant.

Lemme 2.33. *Si w est un mot de tresse σ-positif et si $(\vec{x}, \vec{y})$ est un coloriage, alors : $\exists z_1, \ldots, z_k$ avec $k > 0$ tels que :*

$$y_0 = (((x_0 \star z_0) \star z_1) \star \cdots) \star z_k$$

Lemme 2.34. *On suppose que w se retourne à droite en uv^{-1} alors si $(\vec{x}, \vec{y})$ est un S-coloriage de w alors $(\vec{x}, \vec{y})$ est un S-coloriage de uv^{-1}.*

Pour s'en convaincre, il suffit de regarder le cas critique où on transforme $\sigma_i^{-1}\sigma_j$ en $\sigma_j\sigma_i\sigma_j^{-1}\sigma_i^{-1}$. La figure 9 montre la validité du resultat.

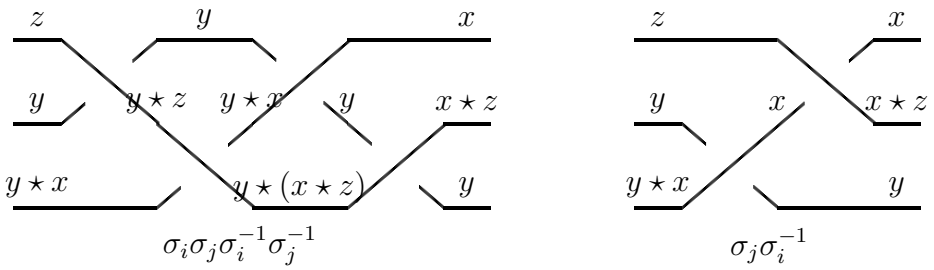


FIG. 9 – Coloriage et retournement à droite.

Lemme 2.35. *Soit S un LD$-$système simplifiable à gauche, on suppose que w et w' sont deux mots de tresse équivalents. Si $(\vec{x}, \vec{y})$ et $(\vec{x}, \vec{y'})$ sont des coloriages de w et respectivement de w', alors $\vec{y} = \vec{y'}$.*

Démonstration. Pour montrer l'existence des coloriages, on a utilisé la retournabilité à gauche, pour ce résultat qui est en quelque sorte un résultat d'unicité, on se sert du retournement à droite : on suppose que w et w' se retourne à droite respectivement en uv^{-1} et en $u'v'^{-1}$. D'après les résultats de Garside, on a l'existence de t et t' deux mots de tresse positifs tels que $ut \equiv u't'$ et $vt \equiv v't'$, on a alors, $\vec{x} \cdot ut = \vec{y} \cdot vt$ et $\vec{x} \cdot u't' = \vec{y'} \cdot v't'$ or de $ut \equiv u't'$ et $vt \equiv v't'$, on déduit $\vec{x} \cdot ut = \vec{x} \cdot u't'$ et $\vec{y} \cdot vt = \vec{y} \cdot v't'$, finalement, on a

$$\vec{y} \cdot v't' = \vec{y'} \cdot v't'.$$

Donc grâce à la simplifiabilité à gauche, on a $\vec{y} = \vec{y'}$.

□

On en tire donc immédiatement le lemme suivant :

Lemme 2.36. *Soit S un LD-système simplifiable à gauche. Si w est un mot de tresse positif et w' un mot de tresse équivalent à w, pas forcément positif, alors, tout coloriage de w' convient pour w.*

Si S est un LD-système simplifiable à gauche et si on a un mot à la fois σ-positif et représentant le mot vide, en combinant les deux lemmes précédents, on aurait : l'existence de $z_1, \ldots, z_k$ avec $k > 0$ tels que :

$$x_0 = (((x_0 \star z_0) \star z_1) \star \cdots) \star z_k.$$

On va montrer dans la suite qu'il existe des LD-systèmes tels qu'aucune égalité de ce type ne soient vérifiées.

2.2.3 Un peu plus avant sur les LD-système

On a définit un coloriage sur $\mathcal{B}_n$ grâce aux LD-systèmes et on a vu comment la propriété A se comprend en terme de coloriage. On va travailler sur un LD-système particulier pour montrer qu'il a les propriétés nécessaires pour conclure.

Définition 2.37. *Soit S un LD-système, soient x et y deux éléments de S, on dit que x est* un diviseur itéré de y *et on note $x \sqsubset y$ si il existe $z_1, \ldots, z_k$ dans S avec $k > 0$ tels que :*

$$y = (((x \star z_1) \star z_2) \star \cdots) \star z_k.$$

On dira que S est acyclique si on a jamais : $x \sqsubset x$ pour x dans S.

Vu les nouvelles définitions, et les lemmes 2.33 et 2.36 pour montrer la propriété A, il suffit de démontrer qu'il existe un LD-système acyclique et simplifaible à gauche.

De la même manière que pour les tresses il faut que l'on distingue les éléments des LD-systèmes des mots (appelés termes) qui les représentent.

Définition 2.38. *On note T_1 l'ensemble des termes bien formés sur l'alphabet $\{x, \star, (,)\}$ et $\mathcal{F}_1$ le LD-système libre à 1 générateur. Si on note $=_{LD}$ les relations vérifiées par un LD- système, on a $\mathcal{F}_1 = T_1/=_{LD}$. Si t et t' sont deux termes de T_1, on dit que $(t \sqsubset t')$ est vraie s'il existe $t_1, \ldots, t_k$ dans S avec $k > 0$ tels que :*

$$t' = (((t \star t_1) \star t_2) \star \cdots) \star t_k$$

.

Si t et t' sont deux termes de T_1, on dit que $(t \sqsubset_{LD} t')$ est vrai s'il existe t_1 et t'_1 deux termes tels que $t =_{LD} t_1$, $t' =_{LD} t'_{LD}$ et $t_1 \sqsubset t'_1$.

On remarque que $z \sqsubset_{LD} y$ implique $\bar{z} \sqsubset \bar{y}$ où $\bar{z}$ représente la $=_{LD}$-classe d'équivalence de z.

On admet que $\mathcal{F}_1$ est simplifiable à gauche. On s'attache désormais à montrer que $\mathcal{F}_1$ est acyclique c'est-à-dire que $(y \in \mathcal{F}_1)$ entraine $\neg(y \sqsubset y)$. Ceci revient à montrer que dans T_1 on a jamais $z =_{LD} t$ et $z \sqsubset_{LD} t$ simultanément.

Un terme w de T_1 peut être représenté par un arbre binaire $a(w)$: si $w = x$ c'est l'arbre vide, et si $w = y \star z$ on définit $a(w)$ comme l'arbre ayant pour fils gauche $a(y)$ et pour fils gauche $a(z)$. On constate facilement que l'application a est non seulement bijective mais aussi algorithmique et visualisable (figure 10). Il sera bon d'utiliser cette interprétation pour comprendre les calculs fait plus loin. Si t est un arbre dont la racine a pour fils gauche lt et pour fils droit rt on note $t = lt \star rt$.

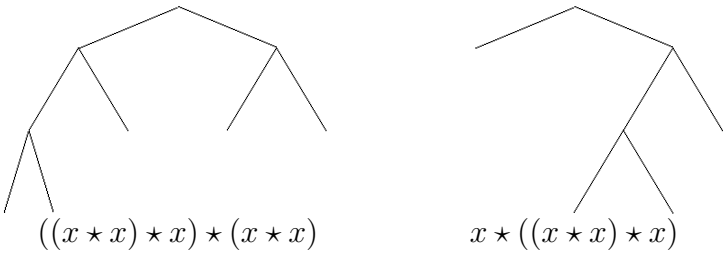


FIG. 10 – Exemple de la correspondance mot de T_1 / arbre binaire.

Définition 2.39. *Si $t = t_1 \star t_2$ est un terme, on définit left$(t) = t_1$. Et récursivement, si left$^{k-1}(t) = t_k \star t_{k+1}$ on définit left$^k(t) = t_k$.*

Définition 2.40. *On appelle* adresse *un mot sur $\Sigma = \{0, 1\}$. Si α est une adresse (on note ϵ le mot vide), on appelle LD_α l'opérateur partiel sur T_1 définit par les formules suivantes :*

$$\begin{aligned} LD_\epsilon(a \star (b \star c)) &= (a \star b) \star (b \star c)), \\ LD_{0\alpha}(a \star b) &= LD_\alpha(a) \star b, \\ LD_{1\alpha}(a \star b) &= a \star LD_\alpha(b). \end{aligned}$$

On définit aussi LD_α^{-1} l'opérateur partiel inverse de LD_α. On appelle $\mathcal{G}_{LD}^+$ le monoïde engendré par les LD_α et $\mathcal{G}_{LD}$ celui engendré les opérateurs LD_α et leurs inverses.

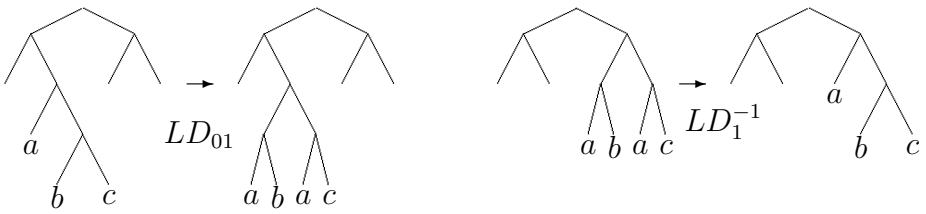


FIG. 11 – Actions des opérateurs LD_α

On fait agir les monoïdes $\mathcal{G}_{LD}^+$ et $\mathcal{G}_{LD}$ à droite c'est-à-dire que l'on note $w \cdot LD_\alpha$ pour $LD_\alpha(w)$.

On vérifie facilement que si $w = LD_\alpha(w')$ on a $w =_{LD} w'$. Pour cela on observe qu'appliquer LD_α revient à appliquer une fois l'identité LD.

Remarque 2.41. *Les opérateurs LD_α décrivent donc l'ensemble des identités LD que l'on peut appliquer. Donc si $t =_{LD} t'$ on peut trouver un f dans $\mathcal{G}_{LD}$ tel que $t =_{LD} t' \cdot f$.*

Insistons sur le fait que les opérateurs LD_α sont partiels (et donc $\mathcal{G}_{LD}$ n'est pas un groupe). Par exemple sur le mot $(x \star x) \star (x \star (x \star (x \star x)) \star x))$, seuls les opérateurs LD_ϵ, LD_1 et LD_{11} sont définis. Le fait qu'ils soient partiels ne va pas nous embêter car nous allons introduire un groupe abstrait vérifiant les même propriétés que $\mathcal{G}_{LD}$ pour pallier au problème. Regardons les relations vérifiées par les opérateurs LD_α.

Lemme 2.42. *Si α, β et γ sont des adresses alors :*

$$\begin{aligned} LD_{\alpha 0\beta} \cdot LD_{\alpha 1\gamma} &= LD_{\alpha 1\gamma} \cdot LD_{\alpha 0\beta} && R1 \\ LD_{\alpha 0\beta} \cdot LD_\alpha &= LD_\alpha \cdot LD_{\alpha 00\beta} \cdot LD_{\alpha 10\beta} && R2 \\ LD_{\alpha 10\beta} \cdot LD_\alpha &= LD_\alpha \cdot LD_{\alpha 01\beta} && R3 \\ LD_{\alpha 11\beta} \cdot LD_\alpha &= LD_\alpha \cdot LD_{\alpha 11\beta} && R4 \\ LD_{\alpha 1} \cdot LD_\alpha \cdot LD_{\alpha 1} \cdot LD_{\alpha 0} &= LD_\alpha \cdot LD_{\alpha 1} \cdot LD_\alpha && R5 \end{aligned}$$

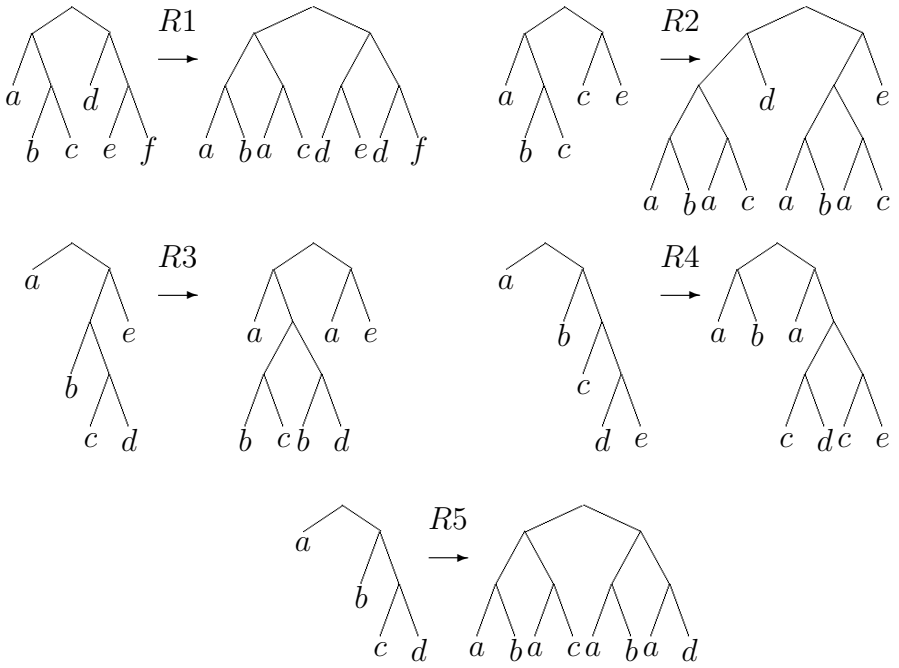


FIG. 12 – Démonstration graphique du lemme 2.42.

Démonstration. On regarde les dessins de la figure 12. □

Nous pouvons désormais introduire un groupe qui vérifie ces relations :

Définition 2.43. *On note G_{LD} le quotient du groupe libre engendré par les éléments τ_α où α est une adresse, par la plus petite relation d'équivalence compatible avec le produit contenant les relations énoncées dans le lemme précédent.*

L'idée est de construire une application de T_1 dans G_{LD}, ensuite un raisonnement dans G_{LD} nous permettra de conclure. Pour ça nous allons introduire la copie-carbone d'un mot. Mais il nous faut un lemme préliminaire.

Définition 2.44. *Soit k un entier non nul, pour $t \in T_1$ on définit inductivement $t^{[k]}$ par : $t^{[1]} = t$ et $t^{[k+1]} = t^{[k]} \star t$.*

Lemme 2.45. *Soit t un mot de T_1, alors il existe un entier n_t tel que pour tout $k \geqslant n_t$, on ait :*

$$t \star x^{[k]} =_{LD} x^{[k+1]}.$$

Remarque 2.46. *Mieux, le n_t est effectif : la démonstration donne une méthode pour calculer un n_t qui convient connaissant t, et mieux encore la démonstration montre comment il faut se servir des relations de LD-système pour faire passer de $t \star x^{[k]}$ à $x^{[k]}$.*

Démonstration. On montre ce résultat par récurrence : pour $t = x$ le résultat est évident. Pour $t = t_1 \star t_2$, soit $n_t = \max(n_{t_1}, n_{t_2}) + 1$, on a alors, pour tout $k \geqslant n_t$:

$$\begin{aligned} x^{[k+1]} &= t_1 \star x^{[k]} = t_1 \star (t_2 \star x^{[k-1]}) \\ &= (t_1 \star t_2) \star (t_1 \star x^{[k-1]}) \\ &= t \star x^{[k]}. \end{aligned}$$

□

Comme on l'a dit, la démonstration du lemme 2.45 nous montre qu'il existe f_t appartenant à $\mathcal{G}_{LD}$ telle que pour tout $k \geqslant n_t$, on ait : $x^{[k+1]} \cdot f_t = t \star x^{[k]}$: si $t = x$, l'élément neutre convient. Si $t = t_1 \star t_2$, on prend :

$$f_t = f_{t_1} \cdot \mathrm{sh}_1(f_{t_2}) \cdot LD_\epsilon \cdot \mathrm{sh}_1(f_{t_1})^{-1},$$

où sh_1 est le morphisme alphabétique qui a LD_α associe $LD_{1\alpha}$ Ce qui nous amène à la définition analogue dans G_{LD} :

Définition 2.47. *Soit $t \in T_1$, on appelle* copie-carbone de t, *et l'on note $[t]$ l'élément de G_{LD} définit par les formules :*

$$\begin{aligned} [x] &= e \text{ (neutre du groupe)}, \\ [t_1 \star t_2] &= [t_1] \cdot \mathrm{sh}_1([t_2]) \cdot \tau_\epsilon \cdot \mathrm{sh}_1([t_1])^{-1}. \end{aligned}$$

où sh_1 *est définit comme précédemment.*

Nous allons maintenant nous intéresser au propriétés des $[t]$.

Proposition 2.48. *1. Si $t =_{LD} t'$ alors $[t']^{-1} \cdot [t]$ appartient au sous groupe engendré par les $\mathrm{sh}_0(\tau_\alpha) = \tau_{0\alpha}$.*

2. Si $t \sqsubset_{LD} t'$ alors $[t']^{-1} \cdot [t]$ admet une écriture admettant au moins un τ_ϵ mais pas de τ_ϵ^{-1}.

Démonstration. 1. On peut se limiter au cas où $t = t' \cdot \tau_\alpha$ et conclure par induction. Par récurrence sur la longueur de l'adresse, on montre que si $t = t' \cdot \tau_\alpha$, alors $[t']^{-1} \cdot [t]$ appartient au sous-monoïde engendré par les $\tau_{0\alpha}$. Commençons par $\alpha = \epsilon$:
Notons $t' = t_1 \star (t_2 \star t_3)$ et donc $t = (t_1 \star t_2) \star (t_1 \star t_3)$. On prend notre courage à deux mains et on fait les calculs :

$$\begin{aligned} [t'] &= [t_1 \star (t_2 \star t_3)] \\ &= [t_1] \cdot \mathrm{sh}_1([t_2 \star t_3]) \cdot \tau_\epsilon \cdot \mathrm{sh}_1([t_1])^{-1} \\ &= [t_1] \cdot \mathrm{sh}_1([t_2]) \cdot \mathrm{sh}_{11}([t_3]) \cdot \tau_1 \cdot \mathrm{sh}_{11}([t_2])^{-1} \cdot \tau_\epsilon \cdot \mathrm{sh}_1([t_1])^{-1} \\ &= [t_1] \cdot \mathrm{sh}_1([t_2]) \cdot \mathrm{sh}_{11}([t_3]) \cdot \tau_1 \cdot \tau_\epsilon \cdot \mathrm{sh}_{11}([t_2])^{-1} \cdot \mathrm{sh}_1([t_1])^{-1} \end{aligned}$$

$$\begin{aligned} [t] &= [(t_1 \star t_2) \star (t_1 \star t_3)] \\ &= [t_1 \star t_2] \cdot \mathrm{sh}_1([t_1 \star t_3]) \cdot \tau_\epsilon \cdot \mathrm{sh}_1([t_1 \star t_2]) \\ &= [t_1] \cdot \mathrm{sh}_1([t_2]) \cdot \tau_\epsilon \cdot \mathrm{sh}_1([t_1])^{-1} \cdot \mathrm{sh}_1([t_1]) \cdot \mathrm{sh}_{11}([t_3]) \cdot \tau_1 \\ &\quad \cdot \mathrm{sh}_{11}([t_1])^{-1} \cdot \tau_\epsilon \cdot (\mathrm{sh}_1([t_1]) \cdot \mathrm{sh}_{11}([t_2]) \cdot \tau_1 \cdot \mathrm{sh}_{11}([t_1])^{-1})^{-1} \\ &= [t_1] \cdot \mathrm{sh}_1([t_2]) \cdot \tau_\epsilon \cdot \mathrm{sh}_1([t_1])^{-1} \cdot \mathrm{sh}_1([t_1]) \cdot \mathrm{sh}_{11}([t_3]) \cdot \tau_1 \\ &\quad \cdot \mathrm{sh}_{11}([t_1])^{-1} \cdot \tau_\epsilon \cdot \mathrm{sh}_{11}([t_1]) \cdot \tau_\epsilon \cdot \mathrm{sh}_{11}([t_2])^{-1} \cdot \mathrm{sh}_1([t_1])^{-1} \\ &= [t_1] \cdot \mathrm{sh}_1([t_2]) \cdot \mathrm{sh}_{11}([t_3]) \cdot \tau_\epsilon \cdot \tau_1 \cdot \tau_\epsilon \cdot \tau_1^{-1} \cdot \mathrm{sh}_{11}([t_2])^{-1} \cdot \mathrm{sh}_1([t_1])^{-1} \\ &= [t_1] \cdot \mathrm{sh}_1([t_2]) \cdot \mathrm{sh}_{11}([t_3]) \cdot \tau_1 \cdot \tau_\epsilon \cdot \tau_1 \cdot \tau_0 \cdot \tau_1^{-1} \cdot \mathrm{sh}_{11}([t_2])^{-1} \cdot \mathrm{sh}_1([t_1])^{-1} \\ &= [t_1] \cdot \mathrm{sh}_1([t_2]) \cdot \mathrm{sh}_{11}([t_3]) \cdot \tau_1 \cdot \tau_\epsilon \cdot \mathrm{sh}_{11}([t_2])^{-1} \cdot \mathrm{sh}_1([t_1])^{-1} \cdot \tau_0 \\ &= [t'] \cdot \tau_0 \end{aligned}$$

On a donc le résultat pour $\alpha = \epsilon$. Passons, au cas $\alpha = 0\beta$: posons $t = t' \cdot \tau_{0\alpha}$, on a $t = u \star v$ et $t' = u' \star v$ avec $u = u' \cdot \tau_\alpha$ et donc il existe r engendré positivement par les $(\tau_{0\gamma})_{\gamma \text{ adresse}}$, tel que $[u']^{-1} \cdot [u] = r$. Alors :

$$\begin{aligned} [t] &= [u \star v] \\ &= [u] \cdot \mathrm{sh}_1([v]) \cdot \tau_\epsilon \cdot \mathrm{sh}_1([u])^{-1} \\ &= [u'] \cdot r \cdot \mathrm{sh}_1([v]) \tau_\epsilon \cdot \mathrm{sh}_1(r)^{-1} \cdot \mathrm{sh}_1([u'])^{-1} \\ &= [u'] \cdot \mathrm{sh}_1([v]) \cdot r \tau_\epsilon \cdot \mathrm{sh}_1(r)^{-1} \cdot \mathrm{sh}_1([u'])^{-1} \\ &= [u'] \cdot \mathrm{sh}_1([v]) \tau_\epsilon \cdot \mathrm{sh}_0(r) \cdot \mathrm{sh}_1([u'])^{-1} \\ &= [u'] \cdot \mathrm{sh}_1([v]) \tau_\epsilon \cdot \mathrm{sh}_1([u'])^{-1} \cdot \mathrm{sh}_0(r) \\ &= [t'] \cdot \mathrm{sh}_0(r). \end{aligned}$$

Pour l'avant dernière égalité on utilise $R2$ inductivement. Le résultat passe donc à la récurrence. Si $\alpha = 1\beta$, posons $t = t' \cdot \tau_{1\alpha}$, on a $t = u \star v$ et $t' = u \star v'$ avec $v = v' \cdot \tau_\alpha$ et donc il existe r, engendré positivement par les $(\tau_{0\gamma})_{\gamma \text{ adresse}}$, tel que $[v']^{-1} \cdot [v] = r$ On a alors

$$\begin{aligned} [t] &= [u \star v] \\ &= [u] \cdot \mathrm{sh}_1([v]) \cdot \tau_\epsilon \cdot \mathrm{sh}_1([u])^{-1} \\ &= [v] \cdot \mathrm{sh}_1([v']) \cdot \mathrm{sh}_1(r) \cdot \tau_\epsilon \cdot \mathrm{sh}_1([u])^{-1} \\ &= [v] \cdot \mathrm{sh}_1([v]) \cdot \tau_\epsilon \cdot \tilde{r} \cdot \mathrm{sh}_1([u])^{-1} \\ &= [t'] \cdot \tilde{r}. \end{aligned}$$

On a noté $\tilde{r}$ l'élément engendré positivement par $(\tau_{0\gamma})_{\gamma \text{ adresse}}$, obtenu en utilisant récursivement la relation $R3$. Ainsi dans tous les cas, le résultat passe à la récurrence.

2. Vu que τ_ϵ n'appartient pas au sous groupe engendré par les $\tau_{0\alpha}$ et le résultat, il suffit de traiter le cas $t' = t \star t_0$, or par définition,

$$[t'] = [t] \cdot \mathrm{sh}_1([_0]) \cdot \tau_\epsilon \cdot \mathrm{sh}_1([t])^{-1}$$

□

2.2.4 Une démonstration de la propriété A

On est désormais en mesure de conclure que $\mathcal{F}_1$ est acyclique.

Définition 2.49. *Soient t et t' des termes, on dira que t' est une LD-*extension *de t si $t' = t \cdot f$ pour un f dans G^+_{LD}.*

Lemme 2.50. *Soit c un élément de G_{LD} alors il existe a et b dans G^+_{LD} tels que $c = ab^{-1}$*

Démonstration. On admet ce lemme[6] mais, intuitivement, voici comment il faut comprendre ce résultat : si $t = t' \cdot c$, avec t et t' des termes alors t et t' admettent une LD-extension commune : $t \cdot a = t' \cdot b$. □

Lemme 2.51. *Il existe deux applications $d : \mathbb{N} \times G^+_{LD} \to \mathbb{N}$ et $p : G^+_{LD} \times \mathbb{N} \to G^+_{LD}$ telles que pour tout a dans G_{LD}, pour tout terme t et tout entier k suffisamment petit pour que les expressions qui suivent aient un sens, on ait :*

$$\mathit{left}^{d(k,a)}(t \cdot a) = \mathit{left}^k(t) \cdot p(a, k)$$

Démonstration. Regardons ce que signifie concrètement ce résultat. Si a est dans G^+_{LD}, $t \cdot a$ est une LD-extension de t, le résultat dit que de tout u facteur itéré à gauche[8] de t, et toute LD-extension t' de t on peut trouver u' un facteur itéré à gauche de t' tel qu'il soit une LD-extension de u. Ceci est (finalement) assez naturel : quand on LD-étend un terme t, tout facteur itéré à gauche reste facteur itéré à gauche, quitte à se LD-étendre lui aussi. Pour construire d et p on commence par constuire leurs analogues δ et π sur $\mathcal{G}^+_{LD}$. On contruit d'abord $\delta(k, LD_\alpha)$ et $\pi(LD_\alpha, k)$:

- si l'adresse α est constituée uniquement de zéros et que le nombre est compris entre 0 et $k-1$, on pose $\delta(k, LD_\alpha) = k+1$ et $\pi(LD_\alpha, k) = e$ (e représente l'identité).
- Si l'adresse α commence par plus de k zéros, on pose $\delta(k, LD_\alpha) = k$ et $\pi(LD_\alpha, k) = LD_{\tilde{\alpha}}$ où $\tilde{\alpha}$ est l'adresse α privée des k premier zéros.

[6] Pour le montrer on commence par se restreindre au cas où $t = t' \cdot g^{-1} \cdot h$ et on montre que pour t un terme il existe un ∂t qui soit une LD-extension de toutes les 1-LD-extensions[7]. On conclut en utilisant le terme $\partial^k t$ pour k suffisamment grand.

[8] On entend par là le résultat d'une application left^k

– si l'adresse α n'a aucune de ses deux formes, LD_α agit sur un domaine de t d'intersection vide avec $\text{left}^k(t)$ donc on pose $\delta(k, LD_\alpha) = k$ et $\pi(LD_\alpha, k) = e$.

On se convainc sur des dessins d'arbres que ces choix sont les bons. On pose donc $d(k, \tau_\alpha) = \delta(k, LD_\alpha)$ et $p(\tau_\alpha, k) = \pi(LD_\alpha, k)$. Il faut maintenant étendre d et p à G_{LD} tout entier : en pose pour tout a, b dans G_{LD}, $d(k, ab) = d(d(k, a), b)$ et $p(ab, k) = p(a, 1) \cdot p(g, d(k, a))$. Il faudrait vérifier que ces relations sont comptabibles avec $R1$, $\ldots R5$, mais on se contente ici de dire que les mêmes définitions pour δ et π donne le résultat escompté et on admet donc que p et d vérifient la propriété demandée.

□

Remarque 2.52. *Notons que pour tout a, b dans G_{LD} et tout entier k on a l'égalité $p(ab, k) = p(a, 1) \cdot p(g, d(k, a))$ et que $k \mapsto d(k, a)$ est strictement croissante (il faut commencer par voir que $d(k, a) \geqslant k$*

Théorème 2.53. *Pour tout t et t' des termes de T_1, $t =_{LD} t'$ entraine $\neg(t \sqsubset_{LD} t')$*

Démonstration. On introduit deux sous-ensemble de G_{LD} : on dit qu' un élément c de G_{LD} appartient à $P_<$ (resp. $P_=$), si il existe une décomposition de de $c = ab^{-1}$ avec a et b dans G_{LD}^+ telle que :

$$d(1, b) < d(1, a) \quad (\text{resp. } =)$$

Nous allons commencer par montrer que $P_=$ et $P_<$ sont disjoints, ensuite on s'interressera à leur stabilité par produit, enfin on caracterisera certains éléments de $P_<$ et de $P_=$ et on pourra conclure.

Il s'agit de montrer que si un élément c se décompose en $c = ab^{-1} = a'b'^{-1}$, alors, si $d(1, b) < d(1, a)$ on a $d(1, b') < d(1, a')$. On décompose $a'^{-1}a$ en $g'g^{-1}$ on a alors, $ag = a'g'$ et $bg = b'g'$ ce qui implique en particulier $d(1, ag) = d(1, a'g')$ et donc grâce à la remarque 2.52

$$d(d(1, a), g) = d(d(1, a'), g').$$

En faisant un raisonnement similaire on obtient

$$d(d(1, b), g) = d(d(1, b'), g').$$

En utilisant la stricte croissance de d on obtient $d(d(1, b), g) < d(d(1, a), g)$ donc $d(d(1, b'), g) < d(d(1, a'), g)$ et enfin $d(1, b') < d(1, a')$. Ainsi $P_=$ et $P_<$ sont disjoints.

Ensuite, on vérifie facilement que $P_=$ est stable par passage à l'inverse, en effet $(ab^{-1})^{-1} = ba^{-1}$. Montrons que $P_= \cdot P_< \subseteq P_<$. Soient u un élément de $P_=$ et v un élément de $P_<$, soient a, b, c et f des éléments de G_{LD} tels que $u =_{LD} ab^{-1}$ et $v =_{LD} cf^{-1}$, on peut choisir $b = c$, en effet quitte à remplacer b par bb', c par cc', a par ab' et f par fc' où b' et c' sont des éléments de G_{LD}^+ tels que $b^{-1}c = b'c'^{-1}$. On a alors $uv = af^{-1}$ et donc

$$d(1, f) < d(1, b) = d(1, a).$$

Donc on a bien l'inclusion $P_= \cdot P_< \subseteq P_<$. On montre de même $P_< \cdot P_= \subseteq P_<$, $P_= \cdot P_= \subseteq P_=$ et $P_< \cdot P_< \subseteq P_<$.

Si on décompose τ_ϵ, on obtient $\tau_\epsilon = \tau_\epsilon \cdot e^{-1}$ et on trouve $d(1, e) = 1 < d(1, \tau_\epsilon) = 2$ donc τ_ϵ appartient à $P_<$.

Vu le lemme 2.51 on a, pour toute adresse α, $d(1, 1) = 1 = d(1, \tau_{0\alpha}) = d(1, \tau_{1\alpha})$ et donc $\tau_{0\alpha}$ et $\tau_{1\alpha}$ appartiennent à $P_=$.

On peut donc conclure : si c appartient au sous-groupe $\text{sh}_0(G_{LD})$ alors par les stabilités par produits et inverse que l'on a vu, on a : c appartient à $P_=$, si par contre, c admet une écriture

dans laquelle τ_ϵ apparaît sans que τ_ϵ^{-1} apparaisse[9] alors c appartient à $P_<$, les deux ensembles étant disjoints, ces deux cas s'excluent. En prenant $c = [t']^{-1} \cdot [t]'$ et en invoquant le lemme 2.48, on a le résultat. □

3 L'algorithme de réduction des poignées

3.1 Présentation

3.1.1 Introduction

L'idée de l'algorithme des poignées provient de la propriété A. C'est en tentant de se ramener au cas d'application de cette propriété qu'on parvient à un algorithme que nous étudierons dans la suite de ce travail.

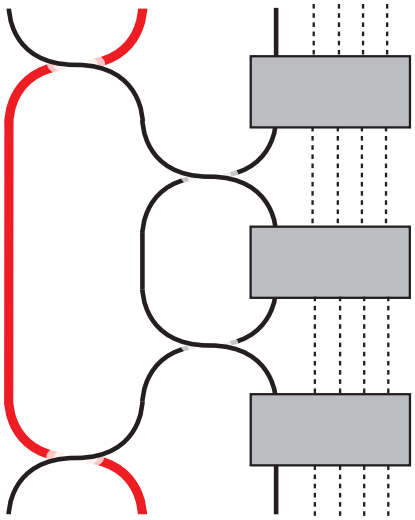
Définition 3.1 (Rappel)**.** *Un mot de tresse w est σ_i-*positif *(resp* négatif*) si w ne contient aucun $\sigma_j^{\pm 1}$ pour $j < i$, contient au moins un σ_i mais aucun σ_i^{-1} (resp des σ_i^{-1} mais aucun σ_i). Une tresse α représentée par un mot σ_i-positif (resp négatif) est dite σ_i-*positive *(resp* négative*).*

Une extension de la propriété A affirme qu'une tresse σ_i-positive (ou négative) n'est pas la tresse triviale. Tentons de résoudre le problème de mot dans B_n : donnons-nous une tresse représentée par un mot de tresse w. Deux cas de figure se présentent (nous supposons que w contient un $\sigma_1^{\pm 1}$ quitte à le translater) :

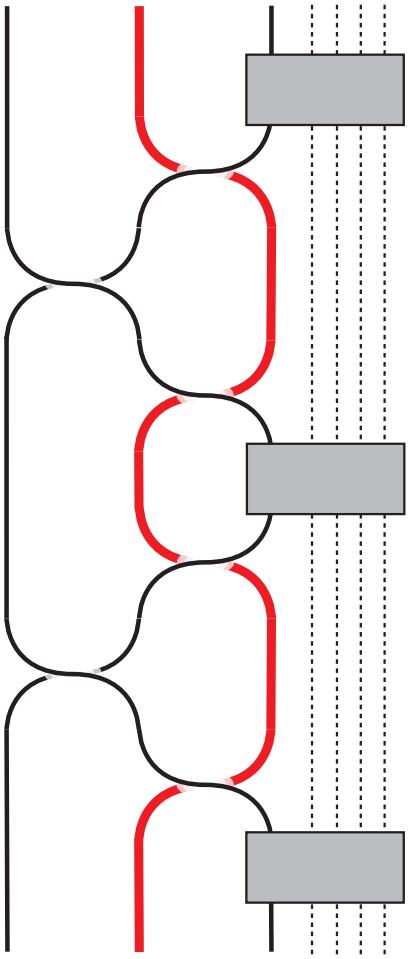
1. Soit w est σ_1-positif (ou négatif) alors $w \not\equiv \epsilon$
2. Soit w est de la forme $w = ... \ \sigma_1^{\pm 1} w' \sigma_1^{\mp 1} ...$ avec le mot w' ne contenant que des $\sigma_j^{\pm 1}$ avec $j > i$. Cette séquence sera appelée (dénomination de P.Dehornoy) une σ_1-*poignée* (positive ou négative).

Que faire face à une σ_1- poignée ? Le dessin ci-dessous le traduit

[9] On ne sait pas où est τ_ϵ^{-1}.



Ci-dessus une σ_1-poignée positive. On la réduit en la faisant glisser sous les brins suivants. Les grands rectangles noirs ne contiennent que des $\sigma_k^{\pm 1}$ avec $k \geqslant 3$. Ci-contre le résultat de la réduction.



Définition 3.2. *On appelle réduction d'une σ_i-poignée la transformation d'une séquence $\sigma_i^e v \sigma_i^{-e}$, v ne contenant que des $\sigma_j^{\pm 1}$ avec $j > i$, avec les opérations suivantes*

- $\sigma_i^{\pm 1} \to \epsilon$,
- $\sigma_{i+1}^{\pm 1} \to \sigma_{i+1}^{-e} \sigma_i^{\pm 1} \sigma_{i+1}^{e}$,
- $\sigma_k^{\pm 1} \to \sigma_k^{\pm}$ *si* $k \geqslant i+2$.

Le nouveau mot obtenu est équivalent à celui de départ. Il représente donc toujours la même tresse.

3.1.2 Poignées nichées

Hélas, cet algorithme ne converge pas. En effet le mot de tresse $w = \sigma_1\sigma_2\sigma_3\sigma_2^{-1}\sigma_1^{-1}$ est réduit en $w_1 = \sigma_2^{-1} w \sigma_2$, et l'itération du procédé ne conduit qu'à une augmentation de la taille du mot.

Le phénomène décrit ci-dessus provient du fait que dans le mot $w = \sigma_1\sigma_2\sigma_3\sigma_2^{-1}\sigma_1^{-1}$, à l'intérieur de la σ_1-poignée, il y a, nichée, une σ_2-poignée. Donnons l'algorithme de réduction des poignées :

(I) Quitte à translater, supposons que w contient des $\sigma_1^{\pm 1}$. On identifie alors la première σ_1-poignée

$$w = debut \quad \underbrace{p}_{\text{première } \sigma_1\text{-poignée}} \quad fin$$

Si il n'y a pas de σ_1-poignée, le mot est positif ou négatif, l'algorithme s'arrête et renvoir w.

(II) Soit $p = \sigma_1^{\pm 1} w' \sigma_1^{\mp 1}$, le mot w' (qui ne comporte que des $\sigma_k^{\pm 1}$ avec $k \geqslant 2$) ne possède donc plus que $n-1$ brins, on le réduit par induction en w'' qui sera σ-positif ou négatif.

(III) On réduit $p = \sigma_1^{\pm 1} w'' \sigma_1^{\mp 1}$ qui devient p'

(IV) On recommence l'algorithme sur $debut\ p'\ fin$.

L'algorithme peut être modifié : on peut choisir une autre poignée que la première, ou proposer un algorithme *divide and conquer*, mais le principe est là.

3.2 Preuve de la convergence

3.2.1 Opérations limitées

L'algorithme de réduction des poignées (ARDP) est intuitif. Cependant la preuve de sa convergence est loin d'être évidente. La démonstration provient du fait que la réduction des poignées n'utilise qu'un nombre restreint d'opérations.

Théorème 3.3. *Si un mot de tresse w' est obtenu à partir de w par réduction de poignées, alors w' dérive de w par équivalence positive-négative et retournement à gauche-droite.*

Démonstration. Prenons le cas d'une poignée $\sigma_1 \cdots \sigma_1^{-1}$: il faut transformer le mot $\sigma_1 w_1 \sigma_2^d w_2 \cdots w_i \sigma_2^d w_{i+1} \sigma_1^{-1}$ les σ_2 intervenant dans la poignée sont tous du même signe (on supposera 1) car on a déjà réduit les poignées nichées, les w_i intervenant sont des mots de tresse ne comportant que des $\sigma_j^{\pm 1}$ avec $j \geqslant 3$. Grâce aux relations $\sigma_i^{\pm 1}\sigma_j^{\pm 1} = \sigma_j^{\pm 1}\sigma_i^{\pm 1}$ si $|i-j| \geqslant 2$, relations permises par l'équivalence positive-négative ou par retournement, on transforme

$$\sigma_{\mathbf{1}} w_1 \sigma_2 w_2 \cdots w_i \sigma_2 w_{i+1} \sigma_{\mathbf{1}}^{\mathbf{-1}} \equiv w_1 \sigma_{\mathbf{1}} \sigma_2 w_2 \cdots w_i \sigma_2 \sigma_{\mathbf{1}}^{\mathbf{-1}} w_{i+1}.$$

Puis on effectue un retrounement à gauche

$$w_1 \sigma_1 \sigma_2 w_2 \cdots w_i \sigma_{\mathbf{2}} \sigma_{\mathbf{1}}^{\mathbf{-1}} w_{i+1} \equiv w_1 \sigma_1 \sigma_2 w_2 \cdots w_i \sigma_{\mathbf{1}}^{\mathbf{-1}} \sigma_{\mathbf{2}}^{\mathbf{-1}} \sigma_{\mathbf{1}} \sigma_{\mathbf{2}} w_{i+1}.$$

Puis on continue...

$$\underbrace{w_1 \sigma_1 \sigma_2 w_2 \cdots w_i \sigma_1^{-1}}_{\text{On continue les mêmes opérations}} \quad \underbrace{\sigma_2^{-1} \sigma_1 \sigma_2}_{\text{changement voulu}} \quad w_{i+1}.$$

À la fin de l'opération on a bien le mot de tresse voulu, à savoir

$$w_1\sigma_2^{-1}\sigma_1\sigma_2 w_2 \cdots \sigma_2^{-1}\sigma_1 w_i\sigma_2 w_{i+1}$$

□

Remarque 3.4.
- *Par rapport aux transformations possibles dans le groupe des tresses, cet algorithme n'utilise pas la relation* $\sigma_i^{\pm 1}\sigma_i^{\mp 1} \equiv \epsilon$.
- *La convergence de l'algorithme de réduction des poignées prouve qu'un mot de tresse trivial peut être transformé en le mot* ϵ *en utilisant uniquement l'équivalence positive-négative et le retournement.*
- *Le mot de tresse* $\sigma_1^{-1}\sigma_7\sigma_4\sigma_7^{-1}$ *est équivalent au mot* $\sigma_1^{-1}\sigma_4$ *mais le premier ne peut être déduit du second par équivalence* $\pm$ *et retournements.*

3.2.2 Graphe de Cayley

Cette limitation des opérations suffisantes pour effectuer la réduction d'une poignée permet de borner la taille des mots réduits successifs.

Définition 3.5. *Soit* β *une tresse positive. Le* graphe de Cayley *de* β *noté* $\Gamma(\beta)$ *est le graphe dont les sommets* $\alpha \in B_n^+$ *sont les diviseurs à gauche de* β

$$\underbrace{\alpha}_{\text{div gauche}} \quad \gamma = \beta \text{ avec } \alpha \text{ et } \gamma \text{ des tresses positives,}$$

le sommet α *est relié au sommet* β *s'il existe* σ_i *tel que* $\alpha\sigma_i = \beta$.

Remarque 3.6. *Les sommets du graphe sont des tresses positives, les arrêtes sont étiquetées par des lettres de l'alphabet* $\sigma_i^{\pm 1}$.

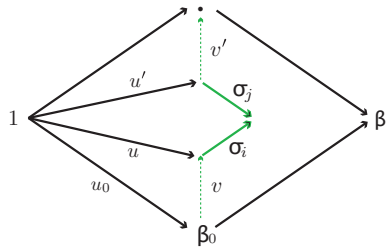
Comme on ne considèrent que des tresses positives, le graphe de Cayley d'une tresse positive est fini : nous avons vu que le nombre de diviseurs d'une tresse positive α est borné par n^m où m est la longueur de α.

Une fois défini le graphe d'une tresse positive, nous avons une notion naturelle de mot tracé dans ce graphe à partir d'un sommet ; en effet supposons par récurrence que l'on se trouve au sommet β et
- nous lisons la lettre σ_i, nous nous dirigeons alors vers l'arrête $\beta\sigma_i$.
- nous lisons la lettre σ_i^{-1}, alors c'est qu'une arrête pointe vers β c'est à dire qu'il existe β' un sommet du graphe tel que $\beta'\sigma_i = \beta$, dans ce cas on se dirige vers β'.

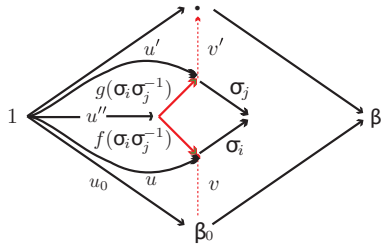
Si l'on ne peut pas effectuer ces déplacements, c'est que le mot n'est pas tracé dans le graphe.

Théorème 3.7. *Soit* β *une tresse positive et* $\gamma \in \Gamma(\beta)$. *Alors l'ensemble de tous les mots tracés depuis* γ *dans* $\Gamma(\beta)$ *est clos par retournement à gauche et à droite et par équivalence positive et négative.*

Démonstration. Commençons par le retournement à gauche. Supposons que le mot $w = v\sigma_i\sigma_j^{-1}v'$ est tracé depuis β_0 dans $\Gamma(\beta)$, nous voulons prouver que le mot $w = vf(\sigma_i\sigma_j^{-1})^{-1}g(\sigma_i\sigma_j^{-1})^{-1}v'$ avec les notations des retournements de mots, est traçable depuis β_0 dans $\Gamma(\beta)$.



Ci-contre, une illustration des hypothèses. La lecture du mot w dans le graphe de Cayley se fait en lisant v puis σ_i et σ_j^{-1} en suivant l'arrête à rebrousse-poil et en terminant par v'. Tous les sommets visités appartiennent au graphe de Cayley de β c'est à dire que u_0, u et u' sont des mots de tresse positifs.



Nous avons donc l'équivalence $u\sigma_i \equiv u'\sigma_j$ ou encore $\sigma_i\sigma_j^{-1} \equiv u^{-1}u'$ d'après les résultats de Garside 2.6, il existe $u''' \geqslant 0$ facteur gauche de u et de u' tel que $u''f(\sigma_i\sigma_j^{-1}) \equiv u$ et $u'''g(\sigma_i\sigma_j^{-1}) \equiv u'$. Le résultat en découle.

En ce qui concerne l'équivalence signée, il suffit (les autres cas seront similaires) de prouver que si on peut tracer le mot $v\sigma_i\sigma_{i+1}\sigma_i v'$ à partir de β_0 dans $\Gamma(\beta)$ alors on peut également tracer $v\sigma_{i+1}\sigma_i\sigma_{i+1}v'$ à partir de β_0 dans $\Gamma(\beta)$. En reprenant les mêmes notations que ci-dessus et par définition du graphe de Cayley d'une tresse positive, il existe un mot positif u tel que $u \equiv u_0 v$ alors on a $u\sigma_{i+1}\sigma_i\sigma_{i+1} \equiv u\sigma_i\sigma_{i+1}\sigma_i$ donc toujours par définition du graphe de Cayley, on peut tracer le mot $v\sigma_{i+1}\sigma_i\sigma_{i+1}v'$ depuis β_0 dans $\Gamma(\beta)$. □

3.2.3 Borne sur les itérées de ARDP

Dorénavant, nous possédons une structure finie stable par des opérations suffisantes pour la mise en jeu de l'algorithme de réduction des poignées. Si w est un mot de tresse, il nous faut trouver un graphe de Cayley dans lequel w est tracé ; alors à chaque étape de la réduction, le mot obtenu est tracé dans ce graphe.

Théorème 3.8. *Soit w un mot de tresse. Alors il existe β une tresse positive et $\alpha \in \Gamma(\beta)$ tels que w est tracé à partir de α dans $\Gamma(\beta)$.*

Démonstration. Soit w un mot de tresse, l'algorithme de retournement à gauche et à droite des mots de tresse nous fournit deux équivalents du mots w :

$$w \equiv D(w)^{-1}N(w),$$

$$\text{et } w \equiv \tilde{N}(w)\tilde{D}(w)^{-1}.$$

avec $D(w), N(w), \ldots$ des mots de tresse positifs. On note $D(w)\tilde{N}(w) = |w|$. Si $\beta = \overline{D(w)\tilde{N}(w)}$ alors w est tracé dans $\Gamma(\beta)$ depuis le sommet $\overline{D(w)}$:

- Pour le mot vide le résultat est vide de sens.
- Supposons le résultat vrai pour $w \in \mathcal{B}_n$ et prouvons le pour $w' = w\sigma_i$ (les autres cas sont similaires) ; par construction on a

$$D(w\sigma_i) = D(w) \quad N(w\sigma_i) = N(w)\sigma_i$$

$$\tilde{N}(w\sigma_i) = \tilde{N}(w)\tilde{N}(\tilde{D}(w)^{-1}\sigma_i) \quad \tilde{D}(w\sigma_i) = \tilde{D}(\tilde{D}(w)^{-1}\sigma_i)$$

Ainsi on a $|w\sigma_i| = |w|\tilde{N}(\tilde{D}(w)^{-1}\sigma_i)$ et donc le graphe $\Gamma(\overline{|w|})$ est un sous graphe de $\Gamma(\overline{|w\sigma_i|})$, on peut donc tracer w depuis $\overline{D(w\sigma_i)} = \overline{D(w)}$ dans $\Gamma(\overline{|w\sigma_i|})$; une fois w tracé,

nous arrivons au sommet $\overline{D(w)w} = \overline{N(w)}$, or $D(w\sigma_i)\tilde{N}(w\sigma_i) = N(w\sigma_i)\tilde{D}(w\sigma_i)$ [10] donc $\overline{N(w)\sigma_i}$ est un diviseur à gauche de $\overline{|w\sigma_i|}$ ce qui signifie bien (modulo l'injection de B_n^+ dans B_n) que l'arrête σ_i est tracée dans le graphe $\Gamma(\overline{|w\sigma_i|})$ depuis le sommet $\overline{N(w)}$. On peut donc finir de tracer $w\sigma_i$.

□

3.2.4 Absence de boucles : preuve de la convergence

Soit w un mot de tresse. Nous possédons désormais un graphe fini $\Gamma(\beta)$ et un sommet α de ce graphe tels que tous les itérées de w par RDP sont tracés à partir de α dans $\Gamma(\beta)$. Mais cela ne suffit pas à prouver la convergence de l'algorithme...

Revenons au fonctionnement de l'ARDP. Soit w un mot de tresse possédant un certain nombre de σ_1-poignées que nous notons $\mathrm{np}_1(w) > 0$. Nous supposons que w ne possède pas de poignées nichées. Alors si l'on note $w_1, ..., w_k, ...$ les itérés de w par RDP alors on a

$$\mathrm{np}_1(w) \geqslant \mathrm{np}_1(w_1) \geqslant ... \geqslant \mathrm{np}_1(w_k) \geqslant ...$$

(en effet par récurrence nous savons réduire les poignées nichées, donc la réduction d'une σ_1-poignée ne fait apparaître au plus qu'une σ_1-poignée). Raisonnons par l'absurde et supposons que l'ARDP ne converge pas, alors il existe un mot de tresse w tel que pour tout k, $\mathrm{np}_1(w_k) > 0$, c'est à dire que les itérés de w possèdent toujours des poignées σ_1. Quitte à choisir $w = w_{k_0}$ pour k_0 assez grand, on supposera dans la suite que pour tout k, $\mathrm{np}_1(w_k) = \mathrm{np}_1(w) > 0$. En réduisant la première poignée, on fait apparaître une nouvelle poignée, qui une fois réduite en fait apparaître une nouvelle ... *ad vitam eternam* : on n'arrive pas à chasser la première poignée. Nous faisons l'hypothèse dans la suite que cette poignée est σ_1-positive. Si tel est le cas alors :

Théorème 3.9. *Définissons $\pi(w_k)$ comme le sous-mot de w_k s'arrêtant à la première lettre de la première poignée de w_k. Alors pour tout k, le mot*

$$(\pi(w_{k+1}))^{-1}\pi(w_k)$$

est équivalent à un mot σ_1-positif tracé dans le graphe de Cayley $\Gamma(\beta)$.

Démonstration. Voir le dessin ci-dessous. □

[10] ceci est même vrai pour un autre mot que $w\sigma_i$

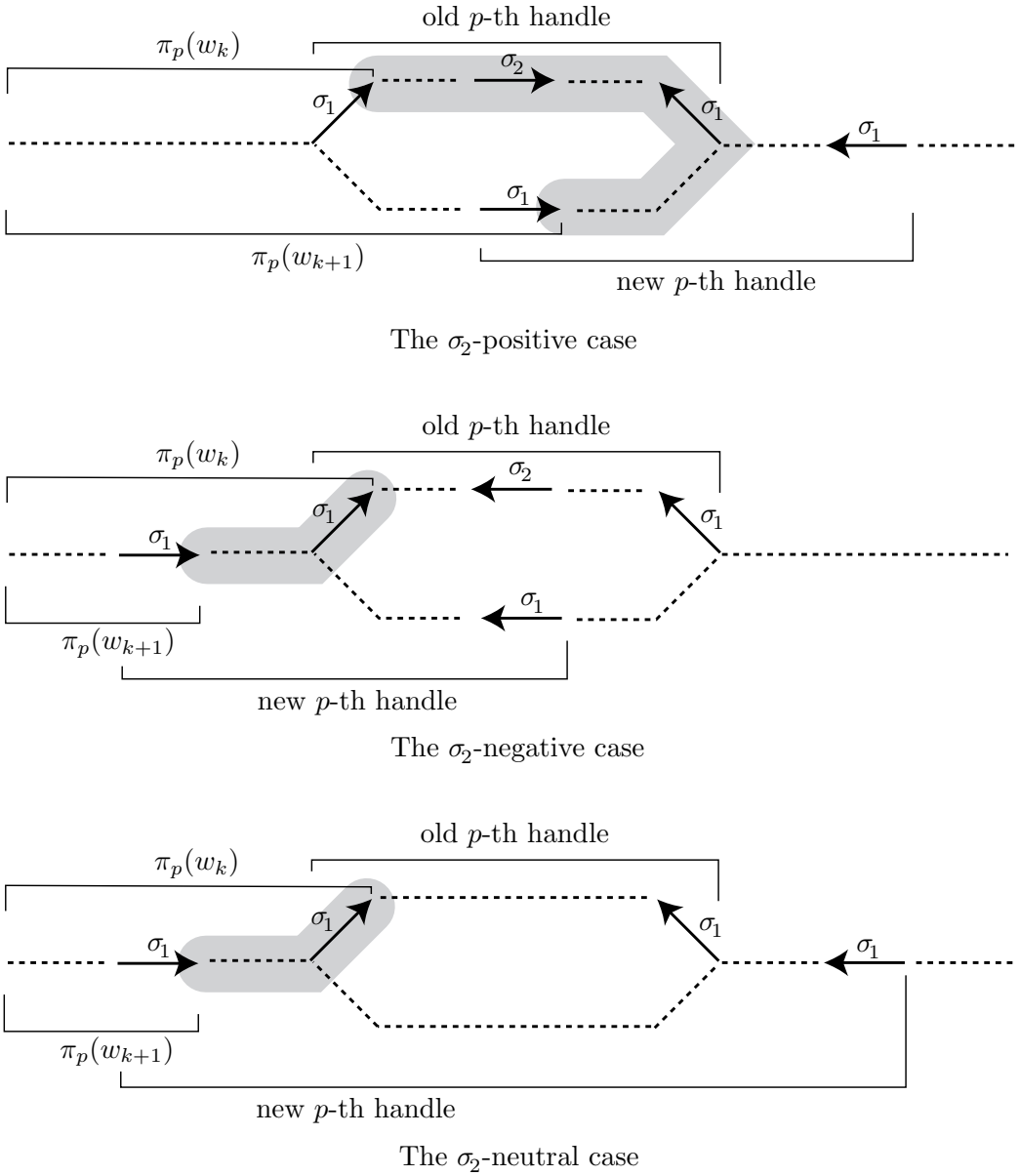


FIG. 13 – Le dessin ci-dessus tiré de *Why are braids orderable ?* illustre, dans tout les cas de figure d'une σ_1-poignée, la poignée obtenue après réduction.

Les mots en gris sur le dessin ci-dessus sont tracés dans $\Gamma(\beta)$ depuis le sommet $\alpha\overline{\pi(w_k)}$ jusqu'au sommet $\alpha\overline{\pi(w_{k+1})}$ dans $\Gamma(\beta)$. Remarquons de plus que tous ces mots sont σ_1 négatifs. Donc si l'ARDP ne converge pas, on peut tracer dans $\Gamma(\beta)$ un mot σ_1-négatif contenant autant de σ_1^{-1} que l'on souhaite, mais ceci est impossible :

Théorème 3.10. *Soit $\Gamma(\beta)$ le graphe de Cayley d'une tresse positive, alors il existe un k tel que : si m est tracé dans $\Gamma(\beta)$ et ne contient pas de σ_1^{-1} alors m contient moins de k σ_1. (le résultat est similaire pour le cas σ_1^{-1})*

Démonstration. Traçons dans $\Gamma(\beta)$ un mot m comportant des σ_1 mais pas de σ_1^{-1}. L'idée est qu'à chaque lecture d'un σ_1 on ne peut plus revenir sur nos pas, c'est à dire repasser par une arrête déjà parcourue : ce qui prouve le théorème car le graphe est fini d'après une remarque

précédente. Supposons par l'absurde que l'on emprunte deux fois la même arrête avec entre temps la lecture d'un σ_1, c'est à dire qu'il existe une tresse positive α divisant β et un mot m' σ_1-positif tel que $\alpha\overline{m'} = \alpha$ ou encore $m' \equiv 1$ ce qui est exclu d'après la propriété A. □

Nous obtenons une contradiction, donc il n'existe pas de mot de tresse w tel que pour tout k, $\mathrm{np}_1(w_k) > 0$, par conséquent pour tout mot w, $\mathrm{np}_1(w_k) = 0$ pour un certain k, ce qui démontre la convergence de l'algorithme de réduction des poignées.

3.2.5 Analyse de la complexité

La démonstration de la convergence fournie une borne supérieure très grande pour la complexité de l'algorithme. Or des mesures empiriques montrent un bien meilleur comportement.

La borne supérieure pour l'ARDP fournit par cette démonstration est exponentielle : cela provient de l'utilisation du graphe de Cayley $\Gamma(\beta)$ et de l'inégalité $\#\Gamma(\beta) \leqslant n^l$, où l est la longueur de β.
Or la pratique semble indiquer un nombre d'étapes de l'ARDP au plus quadratique en la longueur du mot de tresse. À ce jour ceci reste une conjecture.

4 Applications

4.1 L'ordre de Dehornoy

Définition 4.1. *Soit β_1 et $\beta_2 \in B_n^g$ ($n \geqslant 2$), on dit que β_1 est plus petite que β_2, ce que l'on note $\beta_1 < \beta_2$ si la tresse $\beta_1^{-1}\beta_2$ est σ_i-positive.*

Alors le théorème suivant est une conséquence de la convergence de l'ARDP :

Théorème 4.2 (Dehornoy)**.** *La relation $<$ sur B_n est un ordre total compatible avec la multiplication à gauche.*

Remarque 4.3. *L'ordre n'est pas compatible par multiplication à droite :*
$\sigma_1^{-1} < \sigma_2^{-1}$ et pourtant on a pas $\sigma_1^{-1}\sigma_1 < \sigma_2^{-1}\sigma_1$.

Démonstration. Commençons par montrer que $<$ est un ordre : c'est une relation transitive et compatible par multiplication à gauche de manière évidente, elle est anti-réflexive grâce à la propriété A. Le fait que $<$ soit un ordre total est impliqué par le fait que toute tresse soit σ_i-positive, négative ou la tresse triviale. Ce fait est une conséquence de la convergence de l'ARDP et de la propriété A. □

On peut même caractériser l'ensemble ordonné $(B_\infty, <)$ où $B_\infty = \cup_{n=0}^{\infty} B_n$:

Théorème 4.4. *L'ensemble ordnonné $(B_\infty, <)$ est isomorphe (en tant qu'en ensemble ordonné) à $(\mathbb{Q}, <)$.*

Démonstration. D'après un critère de Cantor, pour qu'un ensemble bien ordonné soit isomorphe à $(\mathbb{Q}, <)$ il faut et il suffit que celui-ci soit dénombrable, dense et n'ayant ni élément maximal ni élément minimal. On sait déjà que $(B_\infty, <)$ est dénombrable et bien ordonné de plus il est
- non borné car si $\beta \in B_\infty$ alors $\beta\sigma_1^{-1} < \beta < \beta\sigma_1$
- dense car à translation près, si $1 < \beta$ et $\beta \in B_n$ alors $1 < \beta\sigma_n^{-1} < \beta$.

□

4.2 Conséquence de l'ordre

Théorème 4.5. *Les groupes de tresses B_n sont sans torsion.*

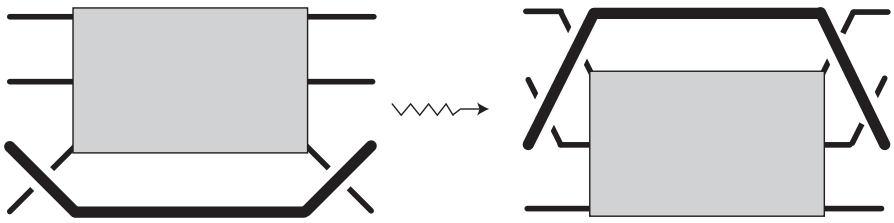
Démonstration. Supposons que l'on ait un élément de torsion c'est à dire une tresse $\beta \neq Id$ telle que $\beta^n = Id$. Supposons, sans perte de généralité que $\beta > 1$, alors par compatibilité de l'ordre avec la multiplication à gauche on a

$$1 < \beta < \beta^2 < ... < \beta^n$$

ce qui est absurde. □

Conclusion

En guise de conclusion, voici une conjecture de P.Dehornoy :
Au lieu de réduire les poignées comme dans cet exposé, mais en les faisant passer par au-dessus de tous les brins, l'algorithme converge-t-il ?



A Programmation

Nous incluons ici le code Caml de l'implémentation de l'ARDP.

```
(*-----------------------------preliminaires-----------------------------*)

let rec print_list l = match l with
  |[] -> print_newline();
  |((a,b)::suite) -> print_int(a); print_string(","); print_int(b);
print_string("|"); print_list suite;;

let inv l =
  let rec inv0 l = match l with
    |[] -> []
    | ((a,b)::suite) -> (a,-b)::(inv0 suite) in
List.rev (inv0 l);;

let impr i n s e =
  begin
match e with
| -1 -> for j = 1 to i-1 do print_string "|   " done;
print_string "\   ";
for j= i+1 to n do print_string "|   " done;
print_newline();
| 1 ->  for j = 1 to i do print_string "|   " done;
print_string "/   ";
                if (i+2) <= n then
for j= i+2 to n do print_string "|   " done else ();
print_newline();
| 0 ->  for i = 1 to n do print_string "|   " done;
print_newline();
  end;

for j= 1 to i-1 do print_string "|   " done;
print_string " \ /    ";
if (i+2) <= n then
for j=  i+2 to n do print_string "|   " done else ();
print_newline ();
 for j= 1 to i-1 do print_string "|   "  done;
if s = 1 then print_string "  /     " else print_string  "  \     ";
if (i+2) <= n then
for j=  i+2 to n do print_string "|   " done else ();
print_newline ();
 for j= 1 to i-1 do print_string "|   " done;
print_string " / \    ";
if (i+2) <= n then
for j=  i+2 to n do print_string "|   " done else ();
print_newline();;


let nettoi l =
  let rec nettoi0 l (a,b) =match l with
  |[] -> [(a,b)];
  |((c,d)::suite) when (a=c)&&(b=(-d)) -> if suite = [] then []
else nettoi0 (List.tl suite) (List.hd suite)
  |((c,d)::suite) -> (a,b)::(nettoi0 suite (c,d)) in
nettoi0 (List.tl l) (List.hd l);;
```

```
let rec print_braid n m l = match l with
  |[] -> for i = 1 to n do print_string "|   " done
  |((i,s)::suite) when i= (m+1)  -> (impr i n s (-1); print_braid n i suite;)
  |((i,s)::suite) when i= (m-1) -> (impr i n s 1 ; print_braid n i suite;)
  |((i,s)::suite) -> impr i n s 0 ; print_braid n i suite;;


print_braid 5 (-2) [(3,1);(2,-1);(2,1);(2,1);(3,-1);(3,1);(1,1);(2,1);(1,-1);(2,1);(1,1)];;

(*----------------------------reduction des poignees-----------------------------*)

let rec poigne0 l l1 poi (i,j) = match l with
  |((a,b)::suite) when (a=i && b= (-j)) -> (List.rev l1,List.rev ((a,b)::poi),suite)
  |((a,b)::suite) when (a=i && b = j) -> poigne0 suite (poi@l1) [(a,b)] (i,j)
  |((a,b)::suite) when (a=i && j = 0) -> poigne0 suite (poi@l1) [(a,b)] (a,b)
  |((a,b)::suite) when (a>i) -> poigne0 suite  l1 ((a,b)::poi) (i,j)
  |((a,b)::suite) when (a<i) -> poigne0 suite (poi@l1) [(a,b)] (a,b)
  | [] -> invalid_arg ("pdp") ;; (* pas de poignee*)

let poigne l =
  let mintr = List.fold_left min max_int (List.map (fun (a,b) -> a) l) in
 poigne0 l [] [] (mintr,0);;


let reduc l=
  let (i,j) = List.hd l in
let rec reduc0 l (i,j) = match l with
  |[] -> []
  | ((a,b)::suite) when a=i -> reduc0 suite (i,j)
  | ((a,b)::suite) when (a=(i+1)) -> (i+1,-j)::(i,b)::(i+1,j)::(reduc0 (suite) (i,j))
  |((a,b)::suite) -> (a,b):: (reduc0 suite (i,j)) in
reduc0 l (i,j);;

let bouture l =
  let (a,b)::suite = l in
  let (c,d)::suite2 = List.rev suite in
  ((a,b),List.rev suite2,(c,d));;

let rec reduipoi (l1,poi,l2) =
  try  let ((a,b),suite,(c,d)) = bouture poi in
  let (l1b,poib,l2b) = poigne suite in
  dehor (l1@[(a,b)]@l1b@(dehor poib)@l2b@[(c,d)]@l2) ;
  with Invalid_argument("pdp") -> dehor (l1 @ (reduc(poi)) @ l2);
and dehor l = match List.length l with
  |0 -> []
  |1 -> l
  |2 -> let [a,b;c,d]= l in if (a=c)&&(b= (-d)) then [] else l
  | n when n > 5000 -> failwith "tropgro"
  | n ->(
      (* let l1 = nettoi l in
      if l1 <> l then dehor l1 else*)
      begin
      (* print_int(n);
        print_string ("|");*)
      try  let (l1,poi,l2) = poigne l in
      reduipoi (l1,poi,l2)
      with Invalid_argument("pdp") -> l; end);;

(*------------------------------tests--------------------------------*)
```

```
let rec rand_braid n l = match l  with
  | 0 -> []
  | l -> let e = Random.int(2) in if e=1 then (Random.int(n)+1,1)::rand_braid n (l-1)
else (Random.int(n)+1,-1)::rand_braid n (l-1);;

let test f n l=
  let a = Sys.time() in
  f (rand_braid n l);
  Sys.time() -. a;;

let bigtest f n l m =
  let mini = ref max_float in
  let maxi = ref 0. in
  let sum = ref 0. in
  for i = 0 to (m-1) do
    let ne = test f n l in
    print_float(ne);
    sum := (!sum +. ne);
    mini := min (!mini) ne;
    maxi := max (!maxi) ne;
  done;
(!mini, (!sum /. float_of_int(m)) , !maxi);;

let a = rand_braid 10 600;;
let b = dehor a;;
let c= dehor (a @ (inv b));;
```

B Retournement à gauche

B.1 Retournements, diagrammes de retournements

Le but de cette annexe (largement tirée de [3]) est de démontrer le théorème suivant :

Théorème B.1. *Soit, w un mot de tresse, on a :*

$$\exists!(u,v) \in (\mathcal{B}_n^+)^2 \text{ tel que } w \looparrowright u^{-1}v$$

On rappelle que le retournement à gauche consite à remplacer récursivement dans un mot de tresse, les sous-mots de type $\sigma_i\sigma_j^{-1}$ par des mots de type $f(\sigma_i,\sigma_j)^{-1}f(\sigma_i,\sigma_j)$ où

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

Il s'agit de savoir si on peut faire terminer le processus.

On définit de la même manière, le retournement à droite qui consiste à remplacer récursivement les sous-mots de type $\sigma_i^{-1}\sigma_j$ par un mot de type $\tilde{f}(\sigma_i,\sigma_j)\tilde{f}(\sigma_i,\sigma_j)^{-1}$ où

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

Commençons par remarquer que le théorème B.1 se reformule pour le retournement à droite de la même manière, et que ce qu'on démontre pour le retournement à gauche a son équivalent à droite et réciproquement grâce au diagramme commutatif suivant :

$$\begin{array}{ccc} \mathcal{B}_n & \xrightarrow{\text{rev}} & \mathcal{B}_n \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle \tilde{f}} \\ \mathcal{B}_n & \xleftarrow[\text{rev}]{} & \mathcal{B}_n \end{array}$$

Culturellement il est plus classique de s'intéresser au problème de retournement à droite, comme les deux sont équivalents, nous nous empressons de renommer $\tilde{f}$ en f et de garder la notation $\looparrowright$ pour le retournement à droite. Nous nous intéressons donc maintenant au problème de retournement à droite. Ainsi le théorème B.1 devient :

Théorème B.2. *Soit, w un mot de tresse, on a :*

$$\exists!(u,v) \in (\mathcal{B}_n^+)^2 \text{ tel que } w \looparrowright uv^{-1}$$

avec $\looparrowright$ qui signifie désormais "se retourne à droite en".

Le problème de retournement peut se voir graphiquement comme la construction d'un graphe appelé diagramme de retournement (bien qu'il lui ressemble, ce n'est pas le graphe de Cayley), nous présentons ce point de vue pour rendre la suite plus lisible.

On considère les points du plan de base orthonormée $(\vec{e_1}, \vec{e_2})$ aux coordonnées rationnelles. Soit w un mot de tresse, on commence par tracer une ligne brisée en orientant et en étiquettant les segments : on part du point $(0,0) = (x_0, y_0)$, et si on a $w = \sigma_i^{+1} w'$, on trace le segment $[(x_0,y_0), (x_0,y_0) + \vec{e_1}]$ en orientant dans le sens de $\vec{e_1}$; on pose alors $(x_1, y_1) = (x_0, y_0) + \vec{e_1}$. Si on a $w = \sigma_i^{-1} w'$, on trace le segment $[(x_0,y_0), (x_0,y_0) + \vec{e_2}]$ en orientant dans le sens de $-\vec{e_2}$] ; on pose alors $(x_1, y_1) = (x_0, y_0) + \vec{e_2}$. On continue de même avec w' en partant de (x_1, y_1)

On a donc une ligne brisée orientée, allant de $(0,0)$ à (p,q) si on a w qui contient p termes positifs et q termes négatifs.

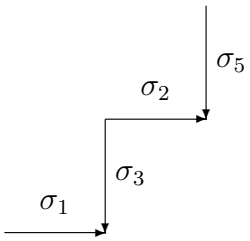


FIG. 14 – Ligne brisée correspondant à $\sigma_1\sigma_3^{-1}\sigma_2\sigma_5^{-1}$.

On peut lire le mot w sur le graphe an allant de $(0,0)$ à (p,q) en regardant l'étiquettage des segments et leurs orientations.

Une étape d'un retournement à droite consiste à transformer un angle orienté vers le bas à droite en un angle orienté vers le haut à gauche :

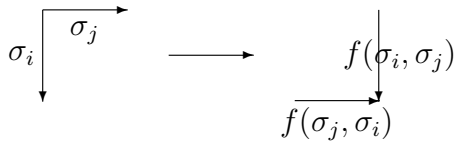


FIG. 15 – Passage de $\sigma_i^{-1}\sigma_j$ à $f(\sigma_j,\sigma_i)f(\sigma_i,\sigma_j)^{-1}$ sur le diagramme de retournement.

Il peut se passer que $f(\sigma_i,\sigma_j)^{-1}$ soit de longueur strictment plus grande que 1, on met alors des points intermediaires.

Par des retournements à droite, on complète donc le diagramme constitué de la ligne brisée représentant w, jusqu'à avoir un diagramme sur lequel il existe un chemin de $(0,0)$ à (p,q) représentant un mot uv^{-1}. Il reste à vérifier que le processus converge, en effet il se pourrait que, le pas des segment diminuant, on atteigne jamais la configuration recherchée En revanche, en cas d'existence, l'unicité des mots u et v est claire sur le diagramme.

Définition B.3. *Soient z et t deux mots de tresse positifs, on note $C_R(z,t)$ le mot de tresse positif u s'il existe tel que : $t^{-1}z \looparrowright uv^{-1}$ (avec v lui aussi positif).*

Remarquons sur la figure 18 qu'il suffit de prouver le théorème B.1 dans le cas où w s'écrit tz^{-1} ce qui constitue le cas le moins favorable.

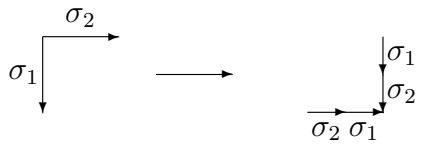


FIG. 16 – Passage de $\sigma_1^{-1}\sigma_2$ à $\sigma_2\sigma_1\sigma_2^{-1}\sigma_1^{-1}$ sur le diagramme de retournement.

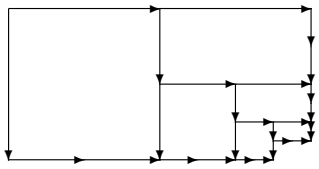


FIG. 17 – Cas hypothétique où le procédé ne convergerait pas.

B.2 Lemmes graphiques

Lemme B.4. *Si i, j et k sont des entiers alors les mots $C_R(f(\sigma_i,\sigma_j),f(\sigma_k,\sigma_j))$ et $C_R(f(\sigma_i,\sigma_k),f(\sigma_j,\sigma_k))$ existent et sont équivalents.*

Démonstration. Il y a beaucoup de cas (17) , mais en nombre fini. En faisant de petits dessins, on se rend compte que dans tous les cas l'existence est acquise et l'équivalence l'est aussi. □

Lemme B.5. *Si u, v et w sont des mots positifs alors $C_R(u,vw)$ existe si et seulement si $C_R(u,v)$ et $C_R(C_R(u,v),w)$ existent, et si c'est le cas on a*

$$C_R(u,vw) = C_R(C_R(u,v),w) \quad et \quad C_R(vw,u) = C(v,u)C_R(w,C_R(u,v)).$$

Démonstration. On se convainc en regardant la figure 19. □

Lemme B.6. *Si u et v sont deux mot de tresses positifs, alors $C_R(u,v)$ si et seulement si $C_R(v,u)$ est défini, et on a*

$$uC_R(u,v) \equiv vC_R(v,u).$$

Démonstration. On se convainc en regardant la figure 20. □

B.3 Fin de la démonstration

Définition B.7. *Si w et w' sont deux mots de tresse positifs et si $p \leqslant +\infty$, on note $w \equiv_p w'$, si on peut passer de w à w' en utilisant au plus p fois les relations du monoïde des tresses*

Théorème B.8. *Soient u, v, u' et v' quatres mots de tresse positifs, alors, si $uu' \equiv vv'$, alors les mots $C_R(u,v)$ et $C_R(v,u)$ existent et il existe un mot de tresse positif vérifiant $u' \equiv C_R(v,u)$ et $v' \equiv C_R(u,v)w$.*

Démonstration. On va prouver le résultat par récurrence : pour k, n et p des entiers positifs pouvant prendre la valeur $+\infty$, on appelle $\mathcal{P}^k_{n,p}$ la propiété suivante :

> "Supposons que les mots u, u', v et v' vérifient $uu' \equiv_p vv'$, $|u| \leqslant k$, $|v| \leqslant k$ et $|uu'| \leqslant n$, alors les mots $C_R(u,v)$ et $C_R(v,u)$ existent et il existe un mot de tresse positif vérifiant $u' \equiv C_R(v,u)$ et $v' \equiv C_R(u,v)w$."

Notre but est de montrer que $\mathcal{P}^\infty_{0,\infty}$ est vraie. On remarque tout d'abord que $\mathcal{P}^\infty_{\infty,\infty}$ est vraie, en effet, seul ϵ a une longueur nulle et sa seule décomposition dans $\mathcal{B}_n^+$ est $\epsilon \equiv \epsilon\epsilon$. Pour le reste de la démonstration nous procédons en trois étapes.

Lemme B.9. *La propriété $\mathcal{P}^1_{\infty,1}$ est vraie.*

Démonstration. Supposons $uu' \equiv_1 vv'$, $|u| \leqslant 1$ et $|v| \leqslant 1$, alors, soit u ou v est le mot vide et le résultat est trivial, soit $u = v = \sigma_i$ et $C_R(u,v) = C_R(v,u) = \epsilon$ et on peut prendre $w = u'$, soit $u = \sigma_i$, $v = \sigma_j$ et $j \neq i$, dans ce cas là, comme $uu' \equiv_1 vv'$, on a forcément u' qui commence par $f(\sigma_j,\sigma_i)$ $(= C_R(v,u))$ et v' qui commence par $f(\sigma_j,\sigma_i)$ $(= C_R(u,v))$, donc le résultat est vrai aussi. □

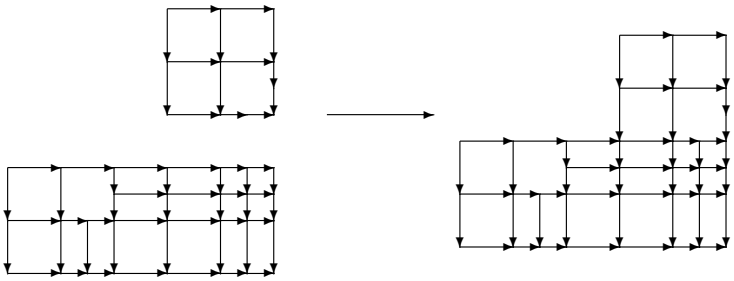


FIG. 18 – On empile les diagrammes.

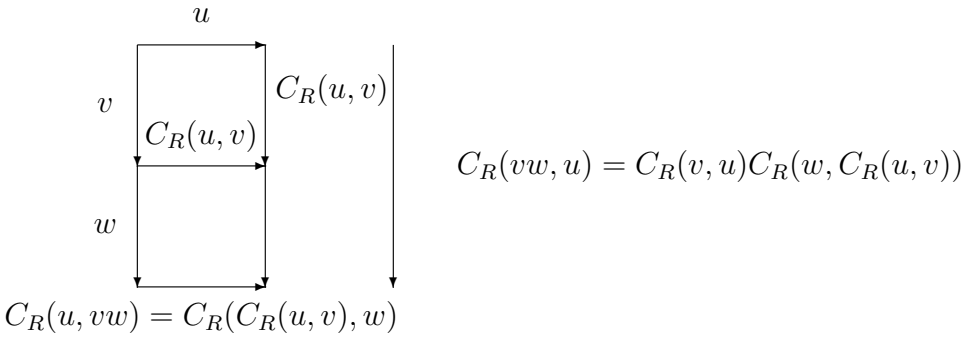


FIG. 19 – Démonstration graphique du lemme B.5.

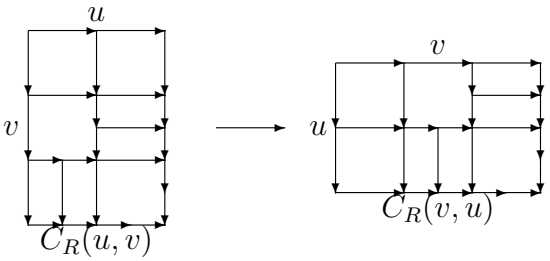


FIG. 20 – Démonstration graphique du lemme B.6. On retourne le diagramme.

Lemme B.10. *Si $\mathcal{P}^{\infty}_{n,\infty}$ et $\mathcal{P}^{1}_{n+1,1}$ sont vraies alors $\mathcal{P}^{1}_{n+1,\infty}$ est vraie.*

Démonstration. Supposant $\mathcal{P}^{\infty}_{n,\infty}$ et $\mathcal{P}^{1}_{n+1,1}$, on montre $\mathcal{P}^{1}_{n+1,p}$ par récurrence sur p. Supposons $\sigma_i u' \equiv_{p+1} \sigma_j v'$ et $|\sigma_i u'| \leqslant n+1$, vu la définition de $\equiv_p$, on peut se donner un l et un w' tels que

$$\sigma_i u' \equiv_p \sigma_l w' \equiv_p \sigma_j v'.$$

En supposant $\mathcal{P}^{1}_{n+1,p}$, on a l'existence de u'' et v'' tels que

$$\begin{cases} u' \equiv f(\sigma_l, \sigma_i) u'', \\ w' \equiv f(\sigma_i, \sigma_l) u'' \end{cases} \quad \text{et} \quad \begin{cases} w' \equiv f(\sigma_j, \sigma_l) v'', \\ v' \equiv f(\sigma_l, \sigma_j) v''. \end{cases}$$

On a $|w'| \leqslant n$, on est donc dans les hypothèses de $\mathcal{P}^{\infty}_{n,\infty}$, et donc, les mots $C_R(f(\sigma_j,\sigma_l), f(\sigma_i,\sigma_l))$ et $C_R(f(\sigma_i,\sigma_l) f(\sigma_i,\sigma_l))$ existent et il existe un mot positif w'' tel que

$$u'' \equiv C_R(f(\sigma_j,\sigma_l), f(\sigma_i,\sigma_l)) w'' \quad \text{et} \quad v'' \equiv C_R(f(\sigma_i,\sigma_l) f(\sigma_i,\sigma_l)) w'',$$

et donc

$$\begin{cases} u' \equiv f(\sigma_l,\sigma_i) C_R(f(\sigma_j,\sigma_l), f(\sigma_i,\sigma_l)) w'', \\ v' \equiv f(\sigma_l,\sigma_j) C_R(f(\sigma_i,\sigma_l) f(\sigma_j,\sigma_l)) w''. \end{cases}$$

On se sert des lemmes B.4, B.5 et B.6, on obtient

$$\begin{array}{rcl} f(\sigma_l,\sigma_i)C_R(f(\sigma_j,\sigma_l),f(\sigma_i,\sigma_l)) & \equiv & f(\sigma_l,\sigma_i)C_R(f(\sigma_j,\sigma_i),f(\sigma_l,\sigma_i)) \\ & \equiv & f(\sigma_j,\sigma_i)C_R(f(\sigma_l,\sigma_i),f(\sigma_j,\sigma_i)), \\ f(\sigma_l,\sigma_j)C_R(f(\sigma_i,\sigma_l)f(\sigma_i,\sigma_l)) & \equiv & f(\sigma_l,\sigma_j)C_R(f(\sigma_i,\sigma_j)f(\sigma_l,\sigma_j)) \\ & \equiv & f(\sigma_i,\sigma_j)C_R(f(\sigma_l,\sigma_i)f(\sigma_i,\sigma_j)) \\ & \equiv & f(\sigma_i,\sigma_j)C_R(f(\sigma_l,\sigma_i),f(\sigma_j,\sigma_i)). \end{array}$$

On a donc $u' \equiv f(\sigma_j,\sigma_i)w$ et $v' \equiv f(\sigma_i,\sigma_j)w$, en prenant $w = C_R(f(z,x),f(y,x))w''$. On a donc bien $\mathcal{P}^1_{n+1,p+1}$ et donc le lemme est démontré.

□

Lemme B.11. *Si $\mathcal{P}^\infty_{n,\infty}$ et $\mathcal{P}^1_{n+1,\infty}$ sont vraies alors $\mathcal{P}^\infty_{n+1,\infty}$ est vraie.*

Démonstration. On montre $\mathcal{P}^k_{n+1,\infty}$ par récurence sur k. On suppose que $\mathcal{P}^k_{n+1,\infty}$ est vérifiée, et que $uu' \equiv vv'$ avec $|uu'| \leqslant n+1$, $|u| \leqslant k+1$ et $|v| \leqslant k+1$. On écrit $u = u_1u_2$ et v=v_1v_2 avec $|u_e| \leqslant k$ et $|v_e| \leqslant k$. Comme $\mathcal{P}^k_{n+1,\infty}$ est vraie, les mots $C_R(u_1,v_1)$ et $C_R(u_2,v_2)$ existent et il existe un mot w tel que

$$u_2u' \equiv C_R(v_1,u_1) \quad \text{et} \quad v \equiv C_R(v_1,u_1).$$

Mais on a $|u_2u'| \leqslant n$ et $|v_2v'| \leqslant n$, et donc grâce à $\mathcal{P}^\infty_{n,\infty}$, on peut trouver u'_2 et v'_2 tel que

$$\begin{cases} u' \equiv C_R(C_R(v_1,u_1),u_2)u'_2, \\ w' \equiv C_R(u_2,C_R(v_1,u_1))u'_2, \end{cases} \quad \text{et} \quad \begin{cases} v' \equiv C_R(C_R(u_1,v_1),v_2)v'_2, \\ w' \equiv C_R(v_2,C_R(u_1,v_1))v'_2. \end{cases}$$

On a donc $C_R(u_2,C_R(v_1,u_1))u'_2 \equiv C_R(v_2,C_R(u_1,v_1))v'_2$ et ces deux mot ayant une longueur plus petite que n, on peut se servir de $\mathcal{P}^\infty_{n,\infty}$, on touve alors un w tel que

$$\begin{cases} u'2 \equiv C_R(C_R(v_2,C_R(u_1,v_1)),C_R(u_2,C_R(v_1,u_1))), \\ v'2 \equiv C_R(C_R(u_2,C_R(v_1,u_1)),C_R(v_2,C_R(u_1,v_1))). \end{cases}$$

Or le lemme B.5 donne

$$\begin{array}{rcl} C_R(u,v) & \equiv & C_R(C_R(v_1,u_1),u_2)C_R(C_R(v_2,C_R(u_1,v_1)),C_R(u_2,C_R(v_1,u_1))), \\ C_R(v,u) & \equiv & C_R(C_R(u_1,v_1),v_2)v'_2C_R(C_R(u_2,C_R(v_1,u_1)),C_R(v_2,C_R(u_1,v_1))). \end{array}$$

Ainsi on a bien $\mathcal{P}^{k+1}_{n+1,\infty}$, et donc le lemme est établi. □

La preuve du théorème est donc presque terminée. En effet, $\mathcal{P}^1_{\infty,1}$ est vraie ; donc, en vertu des deux lemmes précédents, $\mathcal{P}^\infty_{n+1,\infty}$ implique $\mathcal{P}^\infty_{n,\infty}$. Comme $\mathcal{P}^\infty_{0,\infty}$ est vraie, $\mathcal{P}^\infty_{\infty,\infty}$ est vraie, ainsi le théorème est démontré. □

Pour montrer les théorèmes B.1 et B.2, il suffit d'invoquer le résultat de Garside sur les multiples communs lequel, étant donné deux mots de tresse positifs u et v, donne l'existence de deux mots de tresse positifs u' et v' tel que $uu' \equiv vv'$.

Références

[1] *Why are braids orderable ?* Patrick Dehornoy, Ivan Dynnikov, Dale Rolfsen, Bert Wiest.

[2] *Braids, Links, and Mapping Class Groups.* J.Birman Annals of Maths. Sutides, vol 82 (1974)

[3] *Groups with a complemented presentation* Patrick Dehornoy J. P. Appl. Algebra 116 (1997) 115-137

[4] *Braids and Self-Distributivity* Patrick Dehornoy ISBN 3764363436

[5] *The braids group and other groups* F. A. Garside, Quart. J. Math Oxford Ser. (2) 30 (1969), p. 235-254