Quelques exemples d'ordres sur le monoïde plaxique

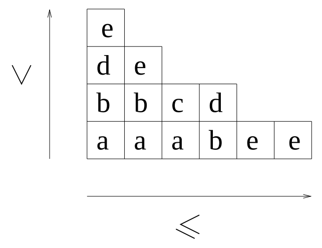
Eric Thierry

Résumé

De récents travaux ont montré l'intérêt de munir certains monoïdes d'une structure d'ordre. Ce rapport présente des essais d'ordonnancement du monoïde plaxique. Les ordres étudiés ici sont construits en comparant les formes des tableaux de Young associés à chaque classe du monoïde plaxique.

Introduction

Le monoïde plaxique trouve son origine dans l'étude des tableaux de Young. Rappelons qu'un tableau de Young est un arrangement de lettres prises dans un alphabet totalement ordonné. Ces lettres sont disposées sur des lignes de tailles décroissantes, selon un ordre croissant large de la gauche vers la droite, et strictement croissant du bas vers le haut, comme le montre l'exemple suivant pour un alphabet $A = \{a < b < c < d < e\}$:



Introduits en 1900 par Alfred Young dans l'étude des représentations matricielles du groupe symétrique, ces objets ont pris une nouvelle dimension à la suite d'un algorithme de Schensted publié en 1961 (cf [9]). Cet algorithme construit, pour tout mot u sur un alphabet A totalement ordonné, un tableau de Young $P(u)$, le *tableau d'insertion* de u. Initialement utilisé pour déterminer la longueur de la plus longue sous suite croissante d'un mot u (égale à la longueur de la première ligne de $P(u)$), il a ensuite connu un champ d'applications beaucoup plus large. L'algorithme de Schensted a permis en particulier de retrouver une bijection fondamentale entre les mots et les paires (P, Q) de tableaux de Young de même forme, établie en 1938 par Robinson, connue aujourd'hui sous le nom de correspondance de Robinson-Schensted.

En 1970, Knuth démontre que l'on peut caractériser de manière locale les mots ayant le même tableau d'insertion : les tableaux d'insertion de deux mots u et v sont identiques si et seulement si ils sont équivalents pour la congruence sur A^* engendrée par les relations

$$\begin{array}{l} acb \equiv cab,\ bca \equiv bac \text{ pour } a < b < c, \\ aba \equiv baa,\ bba \equiv bab \text{ pour } a < b \end{array}$$

Lascoux et Schützenberger remarquent que ces relations définissent un monoïde, auquel ils donnent le nom de *monoïde plaxique*. Leurs travaux sur ce monoïde trouvent de nombreuses applications : interprétation de la règle de Richardson-Littlewood pour les fonctions de Schur, polynômes de Foulkes-Green (cf [6]), charge et cocharge d'un tableau de Young (cf [8]), bases standards (cf [7]) ...

Bien entendu, il existe de nombreuses manières d'ordonner le monoïde plaxique, ou les tableaux de Young, ce qui revient au même du fait de la bijection entre classes plaxiques et tableaux de Young.

Ce rapport ne s'intéresse qu'à une catégorie particulière d'ordres. Leur définition ressemble de loin aux ordres syntaxiques utilisés en théorie algébrique des langages. Pour tout tableau de Young, la suite des longueurs de chaque ligne forme une suite décroissante d'entiers. Ainsi, pour un mot u, si $P(u)$ admet une 1^{re}ligne de longueur n_1, une 2^{e}ligne de longueur n_2, ... , une k^{e}et dernière ligne de longueur n_k, la suite $n_1 \geq n_2 \geq \ldots \geq n_k$ sera notée $g(u)$. Une telle notation provient de l'interprétation donnée par Greene des sommes $n_1, n_1 + n_2, \ldots, n_1 + n_2 + \ldots + n_k$ en termes de longueurs maximales de shuffles de sous-suites croissantes. Tout ordre $\leq$ sur les suites d'entiers induit alors un ordre sur les classes du monoïde, défini par :

$$\bar{u} \leq \bar{v} \quad ssi \quad \forall s, t \in A^*,\ g(sut) \leq g(svt)$$

Dans un premier temps, après avoir rappelé les définitions exactes des termes qui ont été mentionnés dans l'introduction, nous justifierons qu'un tel procédé définit bien un ordre. Puis, dans les parties 2,3 et 4, trois cas particuliers associés à trois ordres usuels sur les suites d'entiers seront étudiés :

- l'ordre d'inclusion
- l'ordre produit
- l'ordre sur les sommes partielles

L'objectif sera de donner une caractérisation intéressante des ordres induits sur le monoïde plaxique. Puisque chaque classe du monoïde plaxique est représentée par un unique tableau de Young, dans chaque cas, nous aurons le choix de décrire les ordres en travaillant sur les tableaux de Young ou sur les mots.

1 Définitions et rappels

Les définitions suivantes sont extraites des ouvrages [1], [5] et [8].

1.1 Le monoïde plaxique

Définition 1.1 *Soit* $(A, <)$ *un alphabet totalement ordonné. La congruence plaxique est la congruence définie par les relations, pour tout* $a < b < c$,

$$cab \equiv acb \tag{1}$$
$$bca \equiv bac \tag{2}$$
$$baa \equiv aba \tag{3}$$
$$bba \equiv bab \tag{4}$$

Le monoïde plaxique $Pla(A, <)$ *est le monoïde quotient de* A^* *par la congruence plaxique. La classe d'équivalence d'un mot* u *sera notée* $\bar{u}$.

Chaque classe du monoïde plaxique est caractérisée par un unique tableau de Young, obtenu par l'algorithme de Schensted.

L'algorithme de Schensted construit pour tout mot un tableau de Young. C'est un algorithme d'insertion : on part du tableau vide, puis on insère une à une les lettres du mot voulu au tableau de Young déjà construit. Deux variantes donnant le même résultat sont possibles suivant que les lettres sont insérées par la gauche ou par la droite. Le tableau obtenu après avoir inséré toutes les lettres d'un mot u, appelé *tableau d'insertion de* u, est noté $P(u)$.

Algorithme 1 (Insertion à droite)
Entrée : T tableau de Young, x lettre de A,
Sortie : $T.x$ tableau de Young,
Exécution :

- si x est plus grand que tous les éléments de la première ligne, ou si cette ligne est vide, ajouter une case à la fin de la première ligne et y placer la lettre x pour obtenir $T.x$,
- sinon, soit α la lettre strictement supérieure à x la plus à gauche de la premiére ligne, remplacer α par x, et insérer α en 2ème ligne selon le même procédé (algorithme récursif)

Algorithme 2 (Insertion à gauche)
Entrée : T tableau de Young, x lettre de A,
Sortie : $x.T$ tableau de Young,
Exécution :

- si x est plus grand que tous les éléments de la première colonne, ou si cette colonne est vide, ajouter une case en haut de la première colonne et y placer la lettre x pour obtenir $x.T$,
- sinon, soit β la lettre supérieure ou égale à x la plus basse de la première colonne, remplacer β par x, et insérer β à la 2ème colonne selon le même procédé (algorithme récursif)

Exemple : considérons l'alphabet $A = \{a < b < c\}$, nous décrivons à la page suivante le déroulement des algorithmes d'insertion à droite, puis à gauche pour le mot $u = bcabba$. Les lettres remplacées au cours des algorithmes sont signalées par une case en gras.

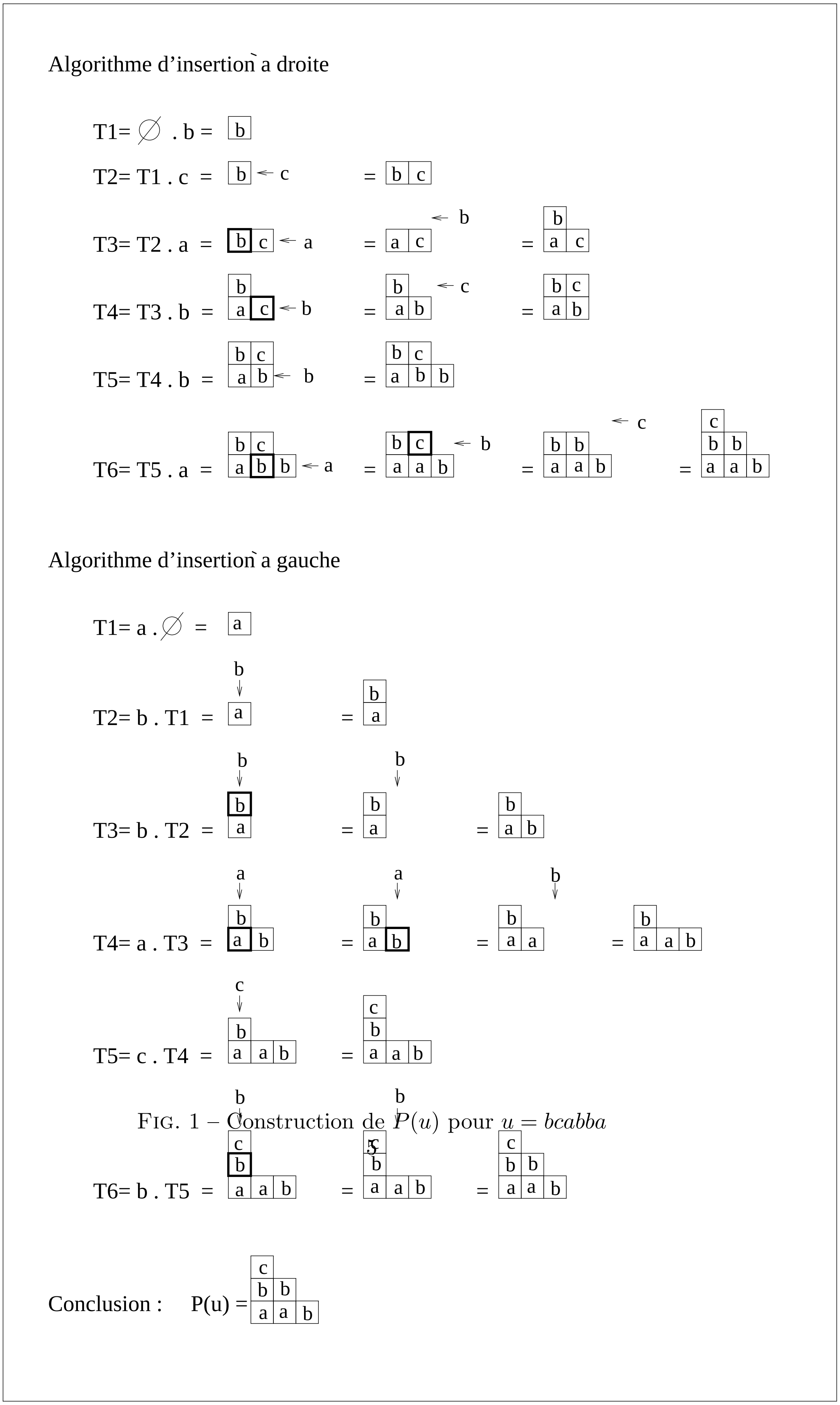


FIG. 1 – Construction de $P(u)$ pour $u = bcabba$

c		
b	b	
a	a	b

P(u)

6		
3	4	
1	2	5

Q(u)

FIG. 2 – Couple de tableaux associés au mot $u = bcabba$

Cet algorithme a permis de retrouver une correspondance fondamentale formulée en 1938 par Robinson. En effet, un deuxième tableau de Young peut être construit en même temps que $P(u)$, lorsqu'on exécute l'algorithme de Schensted *à droite* (un résultat identique existe pour l'algorithme à gauche). Noté $Q(u)$, ce tableau prend ses valeurs dans les entiers : $Q(u)$ est le tableau de même forme que $P(u)$, où les cases sont numérotées par leur ordre d'apparition dans $P(u)$. La figure 2 donne le couple $(P(u), Q(u))$ associé au mot $u = bcabba$, dont l'insertion est décrite à la figure précédente.

Théorème 1.1 (Correspondance de Robinson-Schensted) *Soit A un alphabet totalement ordonné, l'application $u \to (P(u), Q(u))$ est un bijection entre les mots de A^* et les couples de tableaux de Young de même forme, l'un à valeurs dans A, l'autre à valeurs dans les entiers.*

L'application réciproque qui reconstruit un mot à partir d'un couple de tableaux (P, Q) est donnée par un algorithme inverse de l'algorithme de Schensted, qui réduit P avec Q. Cet algorithme est, par exemple, décrit dans [5]. Pour un tableau P à valeurs dans A, il permet donc de trouver un mot u de A^* tel que $P(u) = P$, en se fixant un tableau Q.

En fait, pour un tableau P à valeurs dans A, il existe toujours deux mots simples dont le tableau d'insertion est P.

Proposition 1.1 (Codage) *Soit P un tableau de Young à valeurs dans A, on appelle* codage ligne *de P le mot obtenu en lisant les lignes de P de haut en bas (chaque ligne est lue de gauche à droite), on appelle* codage colonne *de P le mot obtenu en lisant les colonnes de P de gauche à droite (chaque colonne est lue de haut en bas), ces mots ont, tous les deux, P comme tableau d'insertion.*

La preuve consiste juste à appliquer l'algorithme de Schensted à droite. Par exemple, le codage ligne du tableau $P(u)$ de la figure 2 est *cbbaab*, son codage colonne est *cbabab*.

Venons en au théorème qui relie l'algorithme de Schensted au monoïde plaxique.

Théorème 1.2 (Knuth) *Soit u,v deux mots de A^* et $P(u)$,$P(v)$ les tableaux obtenus par l'algorithme de Schensted,*

$$u \equiv v \Longleftrightarrow P(u) = P(v)$$

Ce théorème, allié à la proposition précédente, établit la bijection fondamentale entre les classes du monoïde plaxique $Pla(A,<)$ et les tableaux de Young à valeurs dans A.À partir de maintenant, on pourra identifier les classes de $Pla(A,<)$ et les tableaux de Young sur A, via l'algorithme de Schensted.

Depuis, d'autres algorithmes équivalents à l'algorithme de Schensted, ont été découverts : la construction de $P(u)$ peut s'effectuer avec un jeu de taquin (cf [10]), les ombres de Viennot (cf [11]), ou un algorithme parallèle de Fomin (cf [2]), dont le principe est proche de celui des ombres de Viennot.

1.2 Invariant de Greene

Définition 1.2 *Soit u un mot de A^* et $P(u)$ le tableau de Young obtenu par l'algorithme de Schensted, soient n_1 la longueur de la 1^{re}ligne de $P(u)$, n_2 la longueur de la 2^eligne, ... , n_k la longueur de la k^eet dernière ligne, alors $n_1 \geq n_2 \geq \ldots \geq n_k$ et la suite décroissante d'entiers $g(u) = (n_1, n_2, \ldots, n_k)$ est appelée* invariant de Greene *de u.*

L'invariant de Greene d'un mot u caractérise totalement la forme du tableau de Young $P(u)$. Greene a donné une interprétation de cette forme à l'aide des sous-suites croissantes de u (cf [4]).

Théorème 1.3 (Greene) *Soit $g(u) = (n_1, n_2, \ldots, n_k)$ l'invariant de Greene du mot u, alors,*

$$\begin{aligned}
n_1 &= \textit{longueur maximum d'une sous-suite croissante de } u, \\
n_1 + n_2 &= \textit{longueur maximum de l'union} \\
&\qquad\qquad \textit{de 2 sous-suites croissantes de } u, \\
&\vdots \\
n_1 + n_2 + \ldots + n_k &= \textit{longueur maximum de l'union} \\
&\qquad\qquad \textit{de } k \textit{ sous-suites croissantes de } u.
\end{aligned}$$

Tous les mots d'une classe plaxique ont le même invariant de Greene, et donc la propriété remarquable d'avoir les mêmes cardinalités pour les unions de sous-suites croissantes. La figure 3 présente un exemple pour l'alphabet $A = \{a < b < c\}$: soient $u = ccbabb$ et $v = cbcbab$, leur tableau d'insertion est identique, donc $u \equiv v$, leur invariant de Greene vaut $(3, 2, 1)$. Le théorème est bien vérifié puisque u (resp. v) admet par exemple comme sous-suite croissante maximale abb (resp. bbb), comme union maximale de deux sous-suites cc avec abb (resp. cc avec bbb).

c		
b	c	
a	b	b

FIG. 3 – Tableau d'insertion de *ccbabb* et *cbcbab*

1.3 Involution de Schützenberger

Notons $Pla(A, >)$ le monoïde plaxique construit avec A muni de l'ordre opposé à $<$. Considérons l'application $u \to \tilde{u}$ qui envoie un mot $u = a_1a_2 \dots a_m$ sur son retourné $\tilde{u} = a_m \dots a_2a_1$, $(a_i \in A)$. Cette application respecte la structure de monoïde plaxique, en effet

$$u \equiv v \text{ dans } Pla(A, <) \Longleftrightarrow \tilde{u} \equiv \tilde{v} \text{ dans } Pla(A, >)$$

Il suffit de vérifier que les relations plaxiques (1),(2),(3),(4) sont conservées par cette application.

Une autre manière de présenter ce résultat consiste à introduire $\#$ l'involution $A \to A$ telle que $x \leq y$ ssi $\#y \leq \#x$ (c.à.d $\#(i) = n - i + 1$ si $A = \{1 < 2 < \dots n\}$), puis l'étendre à un mot $u = a_1a_2 \dots a_m$ par $\#u = \#a_m \dots \#a_2\#a_1$. En restant dans $Pla(A, <)$, on a l'équivalence :

$$u \equiv v \Longleftrightarrow \#u \equiv \#v$$

$\#$ définit donc une involution sur $Pla(A, <)$, appelée *involution de Schützenberger* (elle se définit par $\#\bar{u} = \overline{\#u}$).

Concernant les tableaux de Young, il existe un algorithme permettant de passer directement de $P(u)$ à $P(\#u)$. Cet algorithme de Schützenberger, parfois appelé algorithme de "vidage-remplissage", est par exemple présenté

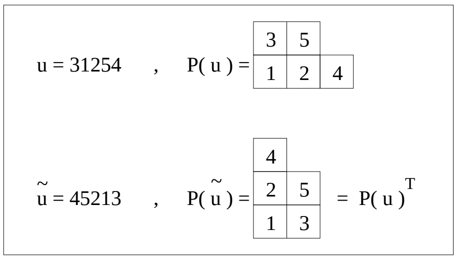


FIG. 4 – Un tableau et son transposé pour l'alphabet $A = \{1 < \ldots < 5\}$

dans [11]. Nous ne retiendrons que le résultat suivant sur la forme de ces tableaux : pour tout mot u, $g(u) = g(\#u)$.

1.4 Standardisation

L'opération de *standardisation* associe à tout mot u de A^* un *mot standard*, c.à.d. un mot dont toutes les lettres sont distinctes. Pour cela, on indexe les diverses occurences de chaque lettre dans u par un indice $i = 1, 2, \ldots$ de gauche à droite. Le mot résultant, $Stand(u)$, est un mot standard sur l'alphabet $A \times \mathbf{N}$ ordonné lexicographiquement.

Par exemple, si $u = baabbcab$, on a $Stand(u) = b_1a_1a_2b_2b_3c_1a_3b_4$. Les quatre relations plaxiques sont conservées après standardisation, et donc,

$$u \equiv v \Longrightarrow Stand(u) \equiv Stand(v).$$

De plus, les formes des tableaux $P(u)$ et $P(Stand(u))$ sont identiques, c.à.d. $g(u) = g(Stand(u))$.

Les mots standards se distinguent aussi par les propriétés suivantes. Si u et v sont deux mots standards plaxiquement équivalents, alors leurs retournés $\tilde{u}$ et $\tilde{v}$ sont plaxiquement équivalent. Si u est un mot standard, alors $P(\tilde{u}) = P(u)^T$, où $P(u)^T$ est le tableau de Young obtenu en inversant les lignes et les colonnes de $P(u)$, comme le montre l'exemple de la figure 4.

1.5 Restriction

Soit I un intervalle de A. Il définit un morphisme $\pi_I : u \rightarrow u \bigcap I^*$ envoyant chaque mot u sur son plus long sous-mot dans I^* ('effacement' des lettres de $A \setminus I$). Un tel morphisme, dit *morphisme de restriction*, est compatible avec la congruence plaxique :

$$u \equiv v \Longrightarrow \pi_I(u) \equiv \pi_I(v).$$

π_I agit ainsi sur $Pla(A, <)$ par $\pi_I(\bar{u}) = \overline{\pi_I(u)}$.

1.6 Contraction d'intervalle

Soit I un intervalle de A, i une lettre de I et considérons le nouvel alphabet $A_I = A \setminus I \bigcup \{i\}$ totallement ordonné en restreignant $<$ à $A \setminus I$ et en fixant pour tout $\alpha \in A \setminus I$

$$\alpha < i \text{ ssi } \alpha < \beta \text{ pour un } \beta \in I$$

Cette définition a un sens puisque I est un intervalle. On peut alors définir le morphisme χ_I de A^* dans A_I^* par :

$$\chi_I(\alpha) = \begin{cases} \alpha \text{ si } \notin I \\ i \text{ si } \in I \end{cases}$$

En d'autres termes, χ_I consiste à remplacer dans un mot chaque lettre de I par la lettre i. Ce morphisme est compatible avec la congruence plaxique :

$$u \equiv v \Longrightarrow \chi_I(u) \equiv \chi_I(v).$$

χ_I agit ainsi sur $Pla(A, <)$ par $\chi_I(\bar{u}) = \overline{\chi_I(u)}$.

1.7 Définition des ordres

A partir de maintenant, nous ne travaillerons plus que sur des alphabets *finis*. Sans perdre de généralité, on pourra parfois considérer que ces alphabets sont de la forme $A = \{1 < 2 < \ldots < n\}$. Cette notation permet d'écrire $i+1$ pour désigner la lettre consécutive à la lettre i, ou encore i-1 i-2 le mot constitué des deux lettres précédant i.
Nous adopteront aussi les notations suivantes :

- pour un mot u et un entier k, $u^k = \underbrace{uu \ldots u}_{k\ fois}$.
- soit T un tableau de Young, si elle existe, la valeur située en ligne i et en colonne j sera notée $T[i, j]$.

Etant donné un alphabet totallement ordonné $A = \{1 < 2 < \ldots < n\}$, voici les trois lemmes qui vont permettre de définir les ordres présentés dans l'introduction :

Lemme 1.1 *Soient T un tableau de Young à valeurs dans A, et α une lettre de A. On s'intéresse à la i^eligne de T, pour $i \leq \alpha$. On définit l'entier i_α par : $T[i, i_\alpha]$ est la valeur inférieure ou égale à α la plus à droite de cette ligne, en notant $i_\alpha = 0$ si elles sont toutes strictement supérieures. Considérons le tableau T' obtenu après avoir inséré à T le mot α α-1 ... α-i+1, T'=T.α α-1 ... α-i+1, les modification des valeurs sont les suivantes,*

$$\begin{cases} \forall 1 \leq k \leq i_\alpha,\ T'[i,k] = T[i,k] \\ T'[i, i_\alpha + 1] = \alpha \end{cases}$$

De plus, si $N_{i,\alpha}(T) = \#\{T[l,k] > \alpha,\ \text{avec } l \leq i \text{ et } k \geq i_\alpha\} > 0$ et $N_{i,\alpha}(T') = \#\{T'[l,k] > \alpha,\ \text{avec } l \leq i \text{ et } k \geq i_\alpha\}$, alors :

$$N_{i,\alpha}(T') < N_{i,\alpha}(T).$$

Preuve - Il suffit d'appliquer l'algorithme de Schensted, en étudiant précisément les effets de l'insertion à droite. La première lettre insérée à T est α : dans tous les cas de figure, elle se place en ligne 1, éventuellement elle évacue une lettre $\beta > \alpha$ qui sera insérée en deuxième ligne. Cette lettre β peut à son tour faire monter d'autres lettres, mais elles seront toutes $> \alpha$; et donc elles ne viendront pas remplacer l'un des $T[i,k]$, $1 \leq k \leq i_\alpha$, car $T[i,k] \leq T[i,i_\alpha] \leq \alpha$. Ajoutons ensuite la lettre α-1, elle vient se placer en ligne 1, en évacuant un α puisqu'il en existe un. Cet α s'insère en ligne 2, éventuellement il fait remonter des lettres $> \alpha$ qui ne modifient pas les $T[i,k]$, $1 \leq k \leq i_\alpha$.

Ainsi de suite, on insère α-2, ..., α-i+1, faisant à chaque fois remonter d'un niveau toutes les lettres précédemment insérées. Finallement, la lettre α a franchi $i-1$ niveaux à partir de la ligne 1, elle vient donc se placer en i^eligne, en position $i_\alpha + 1$ par définition de $T[i, i_\alpha]$ (dont la valeur n'a pas changé depuis le début). Les lettres $T[i,k]$, $1 \leq k \leq i_\alpha$, n'ont pas bougé. Par contre, s'il existait des lettres $> \alpha$, situées en ligne $< i$ ou en ligne i à droite de $T[i,i_\alpha]$, alors au moins l'une d'elle aura été repoussée en ligne $> i$ par l'ascension de α. C'est ce que signifie la deuxième partie du lemme. La signification de ce lemme est très simple : il s'agit de montrer que l'on peut remplir n'importe qu'elle ligne avec la lettre de son choix, et autant de fois que l'on veut (quitte à répéter plusieurs fois l'insertion de α α-1 ... α-i+1). ■

La figure 5 représente les effets de ce procédé, après plusieurs insertions. La position dans le tableau des lettres insérées est signalée par des cases en gras, l'alphabet considéré est $A = \{1 < 2 < 3 < 4 < 5\}$.

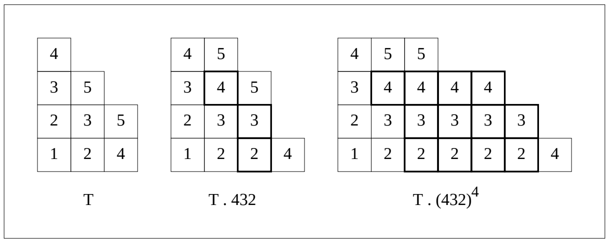


FIG. 5 – Insertion répétée d'une suite de lettres consécutives

Lemme 1.2 *Les notations sont celles du lemme précédent. Etant donné un tableau de Young T, l'insertion d'un mot du type α α-1 ... α-i+1, répétée un très grand nombre de fois conduit à une forme stable de tableau de Young dont on connait les longueurs de toutes les lignes.*
Précisément, soit $N \geq N_{i,\alpha}(T)$, soit $T' = T.(\alpha\ \alpha\text{-}1\ \ldots\ \alpha\text{-}i\text{+}1)^N$, alors les longueurs des i premières lignes sont les suivantes :

$$\begin{array}{rcl} \textit{ligne } 1 & : & \textit{card}\{T[1,k] \leq \alpha\} + N, \\ \textit{ligne } 2 & : & \textit{card}\{T[2,k] \leq \alpha\} + N, \\ \vdots & & \\ \textit{ligne } i & : & \textit{card}\{T[i,k] \leq \alpha\} + N, \end{array}$$

De plus, soit S le tableau de Young composé des lignes de T situées au dessus de la ligne i ; soit u le mot constitué des lettres $> \alpha$, lues sur chaque ligne de la gauche vers la droite, en partant de la ligne i et jusqu'à la ligne 1. Alors, dans T', le tableau constitué des lignes au-dessus de la ligne i, est $S.u$.

Preuve - Le lemme précédent donnait une idée de ce que produisait l'insertion d'un mot α α-1 ... α-i+1 ; celui-ci donne une description complète du tableau $T' = T.(\alpha\ \alpha\text{-}1\ \ldots\ \alpha\text{-}i\text{+}1)^N$, pour N très grand. Si $N \geq N(T)$, d'après ce qui a été vu dans le lemme précédent, toutes les lettres de T, $> \alpha$ et située sur les lignes $\leq i$ vont être réinsérées au fur et à mesure en ligne i+1, selon l'ordre de lecture indiqué pour le mot u. C'est pourquoi les lignes $> i$ de T' forment le tableau $S.u$. L'autre conséquence de l'application de l'algorithme de Schensted est la configuration des i premières lignes : soit $1 \leq l \leq i$, la ligne l de T' est la ligne l de T dont on a enlevé les lettres $> \alpha$, et où on a rajouté N fois la lettre α-i+l, en respectant la condition de croissance de la ligne. ■

Toutes ces modifications se constatent sur l'exemple de la figure 5.

Lemme 1.3 *Soient u et v deux mots de A^*,*

$$u \equiv v \iff \forall\, t \in A^*,\ g(ut) = g(vt)$$

Preuve - Le sens $\Rightarrow$ est évident, puisque :

$$u \equiv v \Rightarrow ut \equiv vt \Rightarrow g(ut) \equiv g(vt)$$

L'autre sens nécessite l'utilisation du lemme précédent, qui permet de faire varier de manière arbitraire les longueurs des lignes d'un tableau.

Soit u, v, deux mots tels que $u \not\equiv v$, c.à.d. $P(u) \neq P(v)$, cherchons un mot t tel que les formes de $P(ut)$ et $P(vt)$ divergent. Si, dès le départ, $P(u)$ et $P(v)$ n'ont pas la même forme, $g(u) \neq g(v)$, il suffit de prendre le mot vide.

Sinon, comme $P(u) \neq P(v)$, il existe une case telle que $P(u)[i,k] \neq P(v)[i,k]$. Supposons que $P(u)[i,k] > P(v)[i,k]$, notons $\alpha = P(v)[i,k]$ et $N = \max\{N_{i,\alpha}(P(u)), N_{i,\alpha}(P(v))\}$. Prenons $t = (\alpha\ \alpha\text{-}1\ ...\ \alpha\text{-}i\text{+}1)^N$.

D'après le lemme précédent, les longueurs de la i^{e}ligne de $P(ut) = P(u).t$ et $P(vt) = P(v).t$ valent respectivement card$\{P(u)[i,k] \leq \alpha\}$+$N$ et card$\{P(v)[i,k] \leq \alpha\}$+$N$. Or, par définition de α et comme les lignes sont croissantes,

$$\text{card}\{P(u)[i,k] \leq \alpha\} + N < \text{card}\{P(v)[i,k] \leq \alpha\} + N,$$

donc la longueur de la i^{e}ligne n'est pas la même pour $P(ut)$ et $P(vt)$, et $g(ut) \neq g(vt)$. L'équivalence est démontrée. ■

Nous pouvont maintenant aborder le procédé générique de construction d'ordres sur le monoïde plaxique, présenté dans l'introduction.

Définition 1.3 *Soit* $A = \{1 < 2 < \ldots < n\}$ *un alphabet fini totallement ordonné, et* $\leq$ *un ordre sur les suites finies d'entiers. Cet ordre induit trois ordres sur le monoïde plaxique, de la manière suivante : soient* u *et* v *deux mots de* A^**, et* $\bar{u}$,$\bar{v}$ *leurs classes,*

$$\begin{array}{llll} \bar{u} \leq_d \bar{v} & si & \forall\, t \in A^*, & g(ut) \leq g(vt) \quad \textit{(ordre à 'droite')} \\ \bar{u} \leq_g \bar{v} & si & \forall\, s \in A^*, & g(su) \leq g(sv) \quad \textit{(ordre à 'gauche')} \\ \bar{u} \leq \bar{v} & si & \forall\, s,t \in A^*, & g(sut) \leq g(svt) \quad \textit{(ordre 'bilatère')} \end{array}$$

Ces trois relations sont bien des relations d'ordre sur le monoïde plaxique $Pla(A,<)$.

Preuve - La relation $\leq_d$ est réflexive et transitive, car $\leq$ est un ordre sur les suites d'entiers. Elle est antisymétrique, du fait du lemme 1.3. Le raisonnement est le même pour l'ordre 'bilatère'.
Pour la relation $\leq_g$, on utilise l'involution de Schützenberger pour inverser le sens droite et gauche : d'après la proposition du 1.3 , pour tous mots u et t, $g(ut) = g(\#t\#u)$, donc, pour u et v deux mots de A^*,

$$\begin{array}{rcl} \forall\, t \in A^*, g(ut) \leq g(vt) & \iff & \forall\, s \in A^*, g(s\#u) \leq g(s\#v) \\ \bar{u} \leq_d \bar{v} & \iff & \overline{\#u} \leq_g \overline{\#v} \end{array}$$

Par conséquent, la relation $\leq_g$ est un ordre. ■

L'objet de ce rapport sera d'étudier ces relations dans trois cas particuliers d'ordres sur les suites finies d'entiers. A la suite de ces définitions, quelques remarques s'imposent :

Remarque 1.1 : on a conservé la même notation pour désigner l'ordre sur les entiers et l'ordre induit sur le monoïde plaxique. En effet, il n'y aura pas de risque d'ambiguité.

Remarque 1.2 : du fait de leurs similitudes, les trois définitions semblent avoir des liens étroits, et on pourrait supposer qu'elles définissent un seul ordre. En fait, elles sont bien distinctes, comme le montreront les exemples étudiés. Les seules liens simples, déductibles des définitions, sont :
- $\bar{u} \leq \bar{v} \Longrightarrow \bar{u} \leq_d \bar{v}$,
- $\bar{u} \leq \bar{v} \Longrightarrow \bar{u} \leq_g \bar{v}$,

– $\bar{u} \leq_g \bar{v} \iff \overline{\#u} \leq_d \overline{\#v}$.
– $\bar{u} \leq \bar{v} \iff \overline{\#u} \leq \overline{\#v}$.

Deux propositions s'ajoutent aux précédentes :

Proposition 1.2 *Si l'ordre $\leq_d$ ou l'ordre $\leq_g$ se réduit à la relation d'égalité, alors les trois ordres $\leq_d$, $\leq_g$, $\leq$, sont identiques, réduits à l'égalité.*

Preuve - Si, par exemple, l'ordre $\leq_d$ se réduit à l'égalité :

$$\bar{u} \leq_d \bar{v} \iff \bar{u} = \bar{v}$$

L'ordre bilatère $\leq$ est aussi l'égalité, puisque :

$$\bar{u} \leq \bar{v} \Rightarrow \bar{u} \leq_d \bar{v} \Rightarrow \bar{u} = \bar{v}$$

De plus, avec l'involution de Schützenberger,

$$\begin{array}{rcl} \bar{u} \leq_g \bar{v} & \Rightarrow & \overline{\#u} \leq_d \overline{\#v} \\ & \Rightarrow & \overline{\#u} = \overline{\#v} \\ & \Rightarrow & \bar{u} = \bar{v} \end{array}$$

■

Proposition 1.3 *Si les ordres $\leq_d$ et $\leq_g$ sont égaux, alors les trois ordres $\leq_d$, $\leq_g$, $\leq$, sont égaux.*

Preuve - Soient u,v,t trois mots de A^*, par définition,
$\bar{u} \leq_d \bar{v} \Rightarrow \overline{ut} \leq_d \overline{vt}$,
$\bar{u} \leq_g \bar{v} \Rightarrow \overline{tu} \leq_g \overline{tv}$,
Supposons maintenant que $\leq_d$ et $\leq_g$ sont identiques,
soient u et v deux mots de A^*, tels que $\bar{u} \leq_d \bar{v}$, montrons que cela implique $\bar{u} \leq \bar{v}$:

soient s,t deux mots de A^*, alors,

$$\begin{array}{rcl} \bar{u} \leq_d \bar{v} & \Rightarrow & \overline{ut} \leq_d \overline{vt} \\ & \Rightarrow & \overline{ut} \leq_g \overline{vt} \\ & \Rightarrow & \overline{sut} \leq_g \overline{svt} \\ & \Rightarrow & \forall\, \alpha, \beta \in A^*,\ g(\alpha sut\beta) \leq g(\alpha svt\beta) \end{array}$$

Et donc, en particulier, $g(sut) \leq g(svt)$. On en déduit l'implication cherchée, et celle-ci suffit à démontrer que $\leq_d$,$\leq_g$,$\leq$ sont bien identiques. ■

D'autres implications se sont pas vraies en général, comme nous le verrons par exemple, $\bar{u} \leq_d \bar{v}$ et $\bar{u} \leq_g \bar{v}$ n'implique pas $\bar{u} \leq \bar{v}$.

Remarque 1.3 : L'ordre bilatère est plus particulièrement intéressant, puisqu'il confère au monoïde plaxique une véritable structure de monoïde ordonné. En effet, cet ordre vérifie :

$$\left.\begin{array}{l}\overline{u} \leq \overline{v}\\ \overline{u'} \leq \overline{v'}\end{array}\right\} \Longrightarrow \overline{uu'} \leq \overline{vv'}.$$

Les trois ordres sur les suites d'entiers qui vont être considérés, sont : l'ordre d'inclusion, l'ordre produit, l'ordre sur les sommes partielles (un ordre qui intervient fréquemment en combinatoire). Il s'agit de donner une caractérisation intéressante des ordres induit sur $Pla(A,<)$ (caractérisation facilement traitable, faisant appel à d'autres domaines, similitude avec des ordres déjà connus ...). Le facteur déterminant de l'étude sera le cardinal de l'alphabet A. Dans les trois cas, le plan de travail consistera à donner une définition simple des ordres induits pour un petit alphabet ($|A| = 2$ ou $|A| = 3$), puis à tenter de trouver une caractérisation élégante ou efficace des ordres pour $|A|$ quelconque. Une question se posera au cours de toute l'étude :
Question 1 : soit A un alphabet fini totallement ordonné fixé, soit $\leq$ un des ordres induits sur $Pla(A,<)$, le problème consistant à déterminer si deux classes données $\bar{u}$ et $\bar{v}$ sont comparables pour $\leq$, est-il décidable ?

2 L'ordre d'inclusion

2.1 Définition

Définition 2.1 *soit* $(m_1 \geq m_2 \geq \ldots \geq m_k)$ *et* $(n_1 \geq n_2 \geq \ldots \geq n_{k'})$ *deux suites décroissantes d'entiers, on appelera* ordre d'inclusion *l'ordre défini par :*

$$(m_1, m_2, \ldots, m_k) \sqsubseteq (n_1, n_2, \ldots, n_{k'}) \text{ si } \{m_1, m_2, \ldots, m_k\} \subset \{n_1, n_2, \ldots, n_{k'}\}$$

Soit A alphabet fini totalement ordonné, l'ordre d'inclusion induit sur $Pla(A,<)$ trois ordres $\sqsubseteq_d$, $\sqsubseteq_g$, $\sqsubseteq$, conformément à la définition 1.3.

Exemple : Soient l'alphabet $A = \{1 < 2 < 3\}$, $u = 3122$, $v = 323113$. Les deux tableaux suivants $P(u)$ et $P(v)$, représentant respectivement les classes $\bar{u}$ et $\bar{v}$, vérifient bien : $g(u) \sqsubseteq g(v)$, en effet $\{1,3\} \subset \{1,2,3\}$. Cependant, pour $t = 1$, $g(ut) \not\sqsubseteq g(vt)$, car $\{1,1,3\} \not\subset \{1,3,3\}$. Par conséquent, $\bar{u} \not\sqsubseteq_d \bar{v}$.

$$\begin{array}{lll} 3 & & \\ 1 & 2 & 2 \end{array}$$

P(u)

$$\begin{array}{lll} 3 & & \\ 2 & 3 & \\ 1 & 1 & 3 \end{array}$$

P(v)

$$\begin{array}{lll} 3 & & \\ 2 & & \\ 1 & 1 & 2 \end{array}$$

P(ut)

$$\begin{array}{lll} 3 & & \\ 2 & 2 & 3 \\ 1 & 1 & 1 \end{array}$$

P(vt)

2.2 Caractérisation générale

Nous allons montrer que, dans le cas de l'ordre d'inclusion, les trois ordres $\sqsubseteq_d$, $\sqsubseteq_g$, $\sqsubseteq$ de $Pla(A,<)$ sont identiques, et sont tout simplement réduits à la relation d'égalité.

Proposition 2.1 *L'ordre à droite $\sqsubseteq_d$, induit par la relation d'inclusion, se réduit à la relation d'égalité.*

Preuve - Sans perdre de généralité, $A = \{1 < 2 < \ldots < n\}$. Soient u et v deux mots de A^*, $\bar{u}$, $\bar{v}$ leurs classes respectives, supposons $\bar{u} \sqsubseteq_d \bar{v}$ et $\bar{u} \neq \bar{v}$. Soit $g(u) = (m_1, m_2, \ldots, m_k)$, et $g(v) = (n_1, n_2, \ldots, n_{k'})$, $g(u) \sqsubseteq g(v)$, donc $\{m_1, m_2, \ldots, m_k\} \subset \{n_1, n_2, \ldots, n_{k'}\}$.

Si $g(u) \neq g(v)$, cela implique $k < k'$. Dans ce cas, considérons le mot t formé de la suite décroissante des lettres de A, $t = n...2\ 1$. Alors, par une simple application de l'algorithme de Schensted,

$$g(ut) = (m_1 + 1, m_2 + 1, \ldots, m_k + 1, \underbrace{1, \ldots, 1}_{|A|-k \text{ fois}})$$

$$g(vt) = (n_1 + 1, n_2 + 1, \ldots, n_{k'} + 1, \underbrace{1, \ldots, 1}_{|A|-k' \text{ fois}}).$$

Donc $g(ut) \not\sqsubseteq g(vt)$, car il y a plus de 1 dans $g(ut)$ que dans $g(vt)$. Ceci est en contradiction avec $\bar{u} \sqsubseteq_d \bar{v}$.

Si $g(u) = g(v)$, $P(u)$ et $P(v)$ ont la même forme. Lorsqu'on insère une lettre à un tableau , l'algorithme de Schensted ajoute une nouvelle case au tableau. Soit x une lettre quelconque de A, si la nouvelle case de $P(ux)$ et celle de $P(vx)$ n'apparaissent pas sur la même ligne, alors, comme on avait $g(u) = g(v)$ et par définition de l'ordre d'inclusion, $g(ux) \not\sqsubseteq g(vx)$. Ce qui serait en contradiction avec $\bar{u} \sqsubseteq_d \bar{v}$. Si les nouvelles cases s'ajoutent sur la même ligne, $P(ux)$ et $P(vx)$ conservent la même forme, $g(ux) = g(vx)$. Dans ce cas, on peut ajouter une autre lettre quelconque de A, et effectuer un raisonnement identique.

Pour éviter d'entrer en contradiction avec $\bar{u} \sqsubseteq_d \bar{v}$, il faut que quelles que soient les lettres ajoutées, les tableaux conservent la même forme, c.à.d. $\forall\, t \in A^*,\ g(ut) = g(vt)$. D'après le lemme 1.3, cela signifie $\bar{u} = \bar{v}$, ce qui est en contradiction avec les hypothèses initiales.

Conclusion : $\bar{u} \sqsubseteq_d \bar{v} \Rightarrow \bar{u} = \bar{v}$. ■

Et d'après la proposition 1.2, les trois ordres sont identiques, réduits à la relation d'égalité. Cet exemple assez simple montre que les ordres induits selon la définition 1.3 peuvent très bien être réduits à la relation d'égalité. Au vu du rythme assez chaotique de la croissance des lignes d'un tableau de Young lorsqu'on y insère des lettres au hasard, l'intuition pousserait à penser que ce cas d'ordre réduit à l'égalité n'est pas si rare, notamment si l'ordre de référence sur les entiers est bizarre.

Question 2 : Est-ce que dans la majorité des cas (à définir), le procédé étudié ne génère que la relation d'égalité, et faudrait-il certaines propriétés sur l'ordre des suites d'entiers pour obtenir des ordres non triviaux sur le monoïde plaxique ?

Justement, l'exemple qui va suivre décrit un cas d'ordre induit non trivial sur le monoïde plaxique. La relation d'égalite n'est donc pas inéluctable.

3 L'ordre produit

3.1 Définition

Définition 3.1 *soit $(m_1, m_2, \ldots, m_k)$ et $(n_1, n_2, \ldots, n_{k'})$ deux suites d'entiers, on appelera* ordre produit *l'ordre défini par :*

$$(m_1, m_2, \ldots, m_k) \leq (n_1, n_2, \ldots, n_{k'}) \ si \ \begin{cases} k \leq k' \ et \\ \forall\, i,\ 1 \leq i \leq k,\ m_i \leq n_i \end{cases}$$

Soit A alphabet fini totallement ordonné, l'ordre produit induit sur $Pla(A, <)$ trois ordres $\leq_d$, $\leq_g$, $\leq$, conformément à la définition 1.3.

Pour commencer, nous allons étudier ces trois ordres pour un petit alphabet : $|A| = 2$. Auparavant, nous introduisont deux notations, utiles par la suite quand nous travailleront sur les tableaux de Young. Soit u un mot sur un alphabet A, soit x une lettre de A, alors :

– $|u|_x$ est le nombre de x dans le mot u,
– $|u|_{x,i}$ est le nombre de x en i^{e} ligne du tableau $P(u)$.

Le tableau $P(u)$ peut ainsi être vu comme un vecteur d'entiers, qui sont les $|u|_{x,i}$ pour x décrivant l'alphabet A et i énumérant les lignes de $P(u)$. Les conditions de croissance des lignes et colonnes dans un tableau de Young imposent certaines inégalités sur ces entiers : par exemple, si $A = \{a < b\}$, alors $|u|_{a,1} \geq |u|_{b,2}$.

3.2 Cas où $|A| = 2$

Considérons un alphabet $A = \{a < b\}$, et les trois ordres $\leq_d$, $\leq_g$, $\leq$, induits sur $Pla(A, <)$ par l'ordre produit. Nous disposons des trois caractérisations suivantes :

Proposition 3.1 *Soit $\leq_d$ l'ordre à droite sur $Pla(A, <)$, soient u et v deux mots de A^*, de classes respectives $\bar{u}$ et $\bar{v}$, alors :*

$$\bar{u} \leq_d \bar{v} \iff \begin{cases} g(u) \leq g(v) \\ |u|_a \leq |v|_a \\ |u|_b \leq |v|_b \end{cases}$$

On peut aussi écrire ces dernières inégalités de la manière suivante :

$$\bar{u} \leq_d \bar{v} \iff \begin{cases} |u|_{a,1} + |u|_{b,1} \leq |v|_{a,1} + |v|_{b,1} \\ |u|_{b,2} \leq |v|_{b,2} \\ |u|_{a,1} \leq |v|_{a,1} \\ |u|_{b,1} + |u|_{b,2} \leq |v|_{b,1} + |v|_{b,2} \end{cases}$$

Preuve - L'ordre $\leq_d$ est défini par :

$$\bar{u} \leq_d \bar{v} \iff \forall\, t \in A^*,\ g(ut) \leq g(vt)$$

En particulier, $\bar{u} \leq_d \bar{v} \Rightarrow g(u) \leq g(v)$. Pour montrer la proposition, on procède en deux temps :

– on montre que, si un couple de mots (u, v) vérifie le système d'inégalités, alors (ua, va) et (ub, vb) le vérifient aussi. Et, par conséquent, si (u, v) vérifie le système, pour tout mot t, (ut, vt) vérifie le système et en particulier l'inégalité $g(ut) \leq g(vt)$,
– on montre ensuite que, si une des inégalités du système n'est pas vérifiée, alors il existe un mot t tel que $g(ut) \not\leq g(vt)$, et donc $\bar{u} \not\leq_d \bar{v}$.

La conservation des inégalités du système se vérifie en distinguant plusieurs cas dans l'application de l'algorithme de Schensted (insertion par la droite). Soient (u, v) un couple de mots vérifiant les inégalités,
• supposons que l'on ajoute la lettre b à u et v, ce qui revient à insérer b dans $P(u)$ et $P(v)$, pour obtenir $P(ub)$ et $P(vb)$, alors l'algorithme de Schensted assure que :

$$\begin{aligned} |ub|_{a,1} &= |u|_{a,1} \\ |ub|_{b,1} &= |u|_{b,1} + 1 \\ |ub|_{b,2} &= |u|_{b,2} \end{aligned}$$

Les modifications sont identiques pour vb, et on peut alors vérifier que les inégalités du système se conservent,
• supposons que l'on ajoute la lettre a à u et v, il faut distinguer deux cas de figure :
-si $|u|_{b,1} = 0$, alors,

$$\begin{aligned} |ua|_{a,1} &= |u|_{a,1} + 1 \\ |ua|_{b,1} &= |u|_{b,1} \\ |ua|_{b,2} &= |u|_{b,2} \end{aligned}$$

-si $|u|_{b,1} > 0$, alors,

$$\begin{aligned} |ua|_{a,1} &= |u|_{a,1} + 1 \\ |ua|_{b,1} &= |u|_{b,1} - 1 \\ |ua|_{b,2} &= |u|_{b,2} + 1 \end{aligned}$$

Comme ces deux cas peuvent aussi se présenter pour v, il faut donc vérifier pour les quatre types de modification du couple (u, v) que le système se conserve. Cette vérification s'effectue sans difficulté.
Supposons maintenant que l'une des inégalités ne soit pas vérifiée,
• évidemment, si $g(u) \not\leq g(v)$, alors $\bar{u} \not\leq \bar{v}$,
• si $|u|_a \not\leq |v|_a$, c.à.d. $|u|_{a,1} > |v|_{a,1}$, alors en prenant $t = a^n$ avec $n \geq |u|_{b,1} + |v|_{b,1}$, on obtient :
$|ut|_{a,1} = |u|_{a,1} + n$, $|ut|_{b,1} = 0$,
$|vt|_{a,1} = |v|_{a,1} + n$, $|vt|_{b,1} = 0$,
et donc $g(ut) \not\leq g(vt)$,
• si $|u|_b \not\leq |v|_b$, c.à.d. $|u|_{b,1} + |u|_{b,2} > |v|_{b,1} + |v|_{b,2}$, alors en prenant $t = a^n$ avec $n \geq |u|_{b,1} + |v|_{b,1}$, on obtient :
$|ut|_{b,2} = |u|_{b,2} + |u|_{b,1}$,

$|vt|_{b,2} = |v|_{b,2} + |v|_{b,1}$,
et donc $g(ut) \not\leq g(vt)$. ■

Proposition 3.2 *Soit $\leq_g$ l'ordre à gauche sur $Pla(A,<)$, soient u et v deux mots de A^*, de classes respectives $\bar{u}$ et $\bar{v}$, alors :*

$$\bar{u} \leq_g \bar{v} \iff \bar{u} \leq_d \bar{v}$$

Preuve - En effet, d'après la remarque 1.2,

$$\bar{u} \leq_g \bar{v} \iff \overline{\#u} \leq_d \overline{\#v}$$

Or, pour tout mot u, $g(\#u) = g(u)$, et $|\#u|_a = |u|_b$, $|\#u|_b = |u|_a$, donc :

$$\begin{cases} g(\#u) \leq g(\#v) \\ |\#u|_a \leq |\#v|_a \\ |\#u|_b \leq |\#v|_b \end{cases} \iff \begin{cases} g(u) \leq g(v) \\ |u|_b \leq |v|_b \\ |u|_a \leq |v|_a \end{cases}$$

C'est à dire : $\bar{\#u} \leq_d \bar{\#v} \iff \bar{u} \leq_d \bar{v}$. La propriété est ainsi démontrée. ■

Proposition 3.3 *Soit $\leq$ l'ordre bilatère sur $Pla(A,<)$, soient u et v deux mots de A^*, de classes respectives $\bar{u}$ et $\bar{v}$, alors :*

$$\bar{u} \leq \bar{v} \iff \bar{u} \leq_d \bar{v}$$

Preuve - Ceci est une simple conséquence de la proposition 1.3. ■

Remarque 3.1 : Ici encore, les trois ordres $\leq_d$, $\leq_g$, $\leq$, sont identiques. Ceci est dû à la simplicité du système d'inégalités, invariant pour beaucoup de transformations. Le système proposé ne se généralise pas pour des alphabets avec $|A| > 2$. En effet, on aurait pu imaginer que :

$$\bar{u} \leq_d \bar{v} \iff \begin{cases} g(u) \leq g(v) \\ \forall\, x \in A,\ |u|_x \leq |v|_x \end{cases}$$

Mais cela signifierait que :

$$\begin{aligned} P(u) = P(v) \quad &\iff \quad \bar{u} \leq_d \bar{v} \text{ et } \bar{v} \leq_d \bar{u} \\ &\iff \quad \begin{cases} g(u) = g(v) \\ \forall\, x \in A,\ |u|_x = |v|_x \end{cases} \end{aligned}$$

Or ceci n'est vrai que dans le cas $|A| = 2$. Comme nous le verront, rien que le cas $|A| = 3$ est déjà beaucoup plus compliqué.

Pour l'instant, nous allons énoncer quelques conditions vérifiées par deux éléments comparables.

3.3 Quelques conditions nécessaires et leurs conséquences

3.3.1 Les morphismes de restriction

Les morphismes de restriction, présentés en 1.5, sont des applications croissantes pour les ordres induits sur les monoïdes plaxiques. Deux preuves seront nécessaires, suivant que l'on s'intéresse à l'ordre à droite ou à l'ordre bilatère.

Proposition 3.4 *Soit I un intervalle de l'alphabet fini $(A,<)$, et π_I le morphisme de restriction qui lui est associé, notons de la même manière, $\leq_d$, l'ordre à droite induit sur $Pla(A,<)$ et sur $Pla(I,<)$ par l'ordre produit, alors, soient u,v deux mots de A^*,*

$$\bar{u} \leq_d \bar{v} \text{ dans } Pla(A,<) \Rightarrow \pi_I(\bar{u}) \leq_d \pi_I(\bar{v}) \text{ dans } Pla(I,<)$$

Preuve - Soit M l'élément maximum de A, et m son minimum, nous allons montrer que la restriction à l'intervalle $A \setminus \{M\}$, ainsi que la restriction à l'intervalle $A \setminus \{m\}$, conservent la relation d'ordre à droite. Tous les intervalles pouvant s'obtenir en effectuant plusieurs retraits d'éléments maximum ou d'éléments minimum, la propriété sera démontrée.

Soient u et v deux mots de A^*, considérons l'intervalle $I = A \setminus \{M\}$. Remarquons d'abord que, pour tout mot w de A^*, $P(\pi_{A\setminus\{M\}}(w))$ est un sous-tableau de $P(w)$. Cela se constate avec l'algorithme de Schensted : $P(w)$ est constitué du tableau $P(\pi_{A\setminus\{M\}}(w))$ sur lequel les lettres M ont été "posées" (voir figure 6).

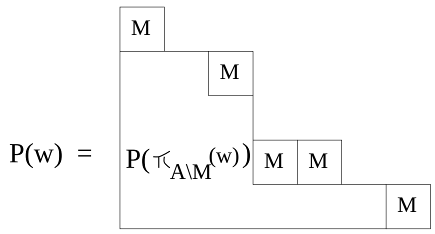


FIG. 6 – Sous-tableau

Si $\overline{\pi_I(u)} \not\leq_d \overline{\pi_I(u)}$, cela signifie qu'il existe un mot $t \in I^*$ tel que $g(\pi_I(u)t) \not\leq g(\pi_I(v)t)$, donc il existe une ligne i telle que la longueur l_1 de la ligne i de $P(\pi_I(u)t)$ soit strictement supérieure à la longueur l_2 de la ligne i de $P(\pi_I(v)t)$. Soit s le mot constitué de la suite décroissante des lettres de I ($s = n$-1 n-2 … 1, si $I = \{1 < 2 < \ldots < n-1\}$), soit $N = \max(|u|_M, |v|_M)$.

En utilisant la remarque sur les sous-tableaux et le lemme 1.2, la ligne i du tableau $P(uts^N)$ a pour longueur $l_1 + N$, et la ligne i du tableau $P(vts^N)$ a pour longueur $l_2 + N$, avec $l_1 + N > l_2 + N$. Donc $\bar{u} \not\leq_d \bar{v}$.

Considérons maintenant l'intervalle $I = A \setminus \{m\}$. Si $\overline{\pi_I(u)} \not\leq_d \overline{\pi_I(u)}$, cela signifie qu'il existe un mot $t \in I^*$ tel que $g(\pi_I(u)t) \not\leq g(\pi_I(v)t)$, donc il existe une ligne i telle que la longueur l_1 de la ligne i de $P(\pi_I(u)t)$ soit strictement supérieure à la longueur l_2 de la ligne i de $P(\pi_I(v)t)$. Soit $N1 = \max(\mathrm{card}\{P(u)[1,k] > m\}, \mathrm{card}\{P(u)[1,k] > m\})$, et $N2 =$longueur de t, alors l'algorithme de Schensted nous permet de décrire $P(um^{N1}tm^{N2})$ et $P(vm^{N1}tm^{N2})$, dont les lignes $i+1$ sont les lignes i de $P(\pi_I(u)t)$ et $P(\pi_I(v)t)$. Par conséquent, $\bar{u} \not\leq_d \bar{v}$. ■

Proposition 3.5 *En reprenant les mêmes hypothèses, et en considérant l'ordre à gauche, soient u,v deux mots de A^*,*

$$\bar{u} \leq_g \bar{v} \text{ dans } Pla(A,<) \Rightarrow \pi_I(\bar{u}) \leq_g \pi_I(\bar{v}) \text{ dans } Pla(I,<)$$

Preuve - Cette proposition se déduit de la précédente en utilisant l'involution # de Schützenberger, sachant que celle-ci commute avec le morphisme de restriction π_I. ■

Proposition 3.6 *En reprenant les mêmes hypothèses, et en considérant cette fois l'ordre bilatère, soient u,v deux mots de A^*,*

$$\bar{u} \leq \bar{v} \text{ dans } Pla(A,<) \Rightarrow \pi_I(\bar{u}) \leq \pi_I(\bar{v}) \text{ dans } Pla(I,<)$$

Preuve - La preuve utilisée pour l'ordre à droite ne s'applique plus. Ici, nous allons utiliser le lemme suivant qui fait appel à l'interprétation de l'invariant de Greene.

Lemme 3.1 *Soit I un intervalle de A, s le mot constitué de la suite décroissante des lettres de I. Soit u un mot de A^*, notons $\lg(u)$ la longueur de u, soit $N > \lg(u)$.*

Si $n_1 \geq \ldots \geq n_{|I|}$ sont les longueurs des $|I|$ premières lignes de $P(\pi_I(u))$ (éventuellement nulles pour les dernières),

Si $n'_1 \geq \ldots \geq n'_{|I|}$ sont les longueurs des $|I|$ premières lignes de $P(s^N us^N)$, Alors $\forall\, k,\ 1 \leq k \leq |I|,\ n'_k = n_k + 2N$.

Preuve - Elle utilise l'interprétation de Greene des longueurs des lignes.

• Pour toute lettre x de I, x^{2N} est bien une sous-suite croissante de $s^N us^N$ (x est présente N fois dans le préfixe s^N, et N fois dans le s^N suffixe). Soit $1 \leq k \leq |I|$, en prenant k lettres dans I, on obtient k sous-suites croissantes distinctes de $s^N us^N$, le cardinal de leur union vaut $2kN$, donc $n'_1 + n'_2 + \ldots + n'_k \geq 2kN$, d'après le théorème de Greene.

• Soit $1 \leq k \leq |I|$, il existe k sous-suites croissantes distinctes de $s^N us^N$, dont l'union a pour cardinal $n'_1 + n'_2 + \ldots + n'_k$. Considérons l'une de ces k sous-suites, montrons qu'elle ne peut contenir une lettre hors de I.
Une sous-suite croissante de $s^N us^N$ est toujours la concaténation d'une sous-suite croissante de s^N (de cardinal $\leq N$, par définition de s), d'une sous-suite de u, et d'une sous-suite s^N (de cardinal $\leq N$).
Supposons que l'une des k sous-suites contienne une lettre x strictement supérieure à toutes les lettres de I. L'union des $(k-1)$ autres sous-suites est nécessairement de cardinal $\leq 2N(k-1) + |$lettres de u utilisées$|$. Par contre, la sous-suite contenant x est la concaténation d'une sous-suite de s^N (de cardinal $\leq N$) et d'une sous-suite de u, et c'est tout, car les lettres du s^N suffixe sont toutes $< x$. Cette sous-suite a donc un cardinal $\leq N +$ $|$lettres de u utilisées$|$. Les k sous-suites sont distinctes, le cardinal de leur union est $\leq 2N(k-1) + N + \lg(u) < 2kN$. Ce résultat est en contradiction avec $n'_1 + n'_2 + \ldots + n'_k \geq 2kN$.

Le raisonnement est le même pour une lettre x strictement inférieure à toutes les lettres de I, sachant qu'une sous-suite croissante contenant x est nécessairement la concaténation d'une sous-suite croissante de u et d'une sous-suite croissante du s^N suffixe.

• Par conséquent, chacune de ces k sous-suites ne prend que des valeurs dans $s^N \pi_I(u) s^N$. Deux méthodes permettent alors de conclure.

<u>Méthode 1</u> : les k sous-suites de $s^N \pi_I(u) s^N$ dont le cardinal de l'union vaut $n'_1 + n'_2 + \ldots + n'_k$, forment en particulier k sous-suites croissantes de $\pi_I(u)$ dont le cardinal de l'union est forcément inférieur ou égal à $n_1 + \ldots + n_k$. Comme chaque sous-suite croissante de $s^N us^N$ à un cardinal $\leq$ $2N + |$lettres de u utilisées$|$, l'union des k sous-suites vérifie : $n'_1 + \ldots + n'_k \leq n_1 + \ldots + n_k + 2kN$.
Nous allons montrer l'égalité : soit s_1, ..., s_k k sous-suites croissantes distinctes de $\pi_I(u)$, dont le cardinal de l'union vaut $n_1 + \ldots + n_k$. Notons $debut(s_i)$ la première lettre de s_i, $fin(s_i)$ la dernière lettre de s_i. Il existe k sous-suites croissantes, distinctes, s'_1, ..., s'_k de $\pi_I(u)$ dont le cardinal de l'union vaut $n_1 + \ldots + n_k$, telles que les lettres $debut(s'_i)$ soient toutes distinctes et les lettres $fin(s'_i)$ soient toutes distinctes.

En effet, supposons par exemple que $fin(s_1) = fin(s_2)$, et que $fin(s_1)$ est située avant $fin(s_2)$ dans $\pi_I(u)$. On considère alors :
$s'_1 = s_1$ concaténé à $fin(s_2)$,
$s'_2 = s_2$ privé de $fin(s_2)$,
ces deux suites restent croissantes, distinctes. Et on réitère cet échange de lettre jusqu'à ce que $fin(s_1) \neq fin(s_2)$.

Le même procédé s'applique aux $debut(s_i)$ avec l'échange suivant : si $debut(s_1)$ est situé avant $debut(s_2)$ dans $\pi_I(u)$, prendre :
$s'_1 = s_1$ privé de $debut(s_1)$,
$s'_2 = debut(s_1)$ concaténé à s_2,
ceci assure que les nouvelles suites restent croissantes.

Pour chaque s'_i ainsi obtenu, on considère la sous-suite croissante de $s^N us^N$ suivante : $debut(s'_i)^N s'_i fin(s'_i)^N$. Ces sous-suites sont bien distinctes, donc le cardinal de leur union vaut $n_1 + \ldots + n_k + 2kN$, et est par définition inférieur ou égal à $n'_1 + \ldots + n'_k$.

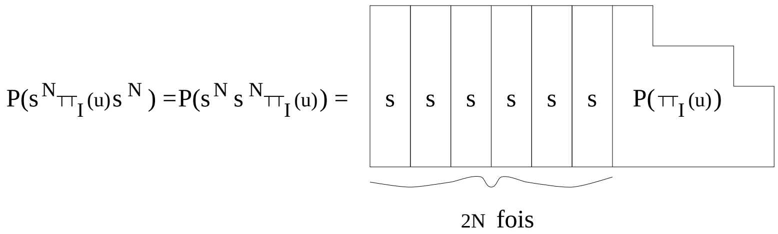
Conclusion :

$$\forall\, k,\ 1 \leq k \leq |I|,\ n'_1 + \ldots + n'_k = n_1 + \ldots + n_k + 2kN,$$

donc,

$$\forall\, k,\ 1 \leq k \leq |I|,\ n'_k = n_k + 2N.$$

<u>Méthode 2</u> : pour $1 \leq k \leq |I|$, les ensembles de k sous-suites de $s^N us^N$ dont l'union a un cardinal maximum prennent leurs valeurs dans $s^N \pi_I(u) s^N$. Donc n'_1, ..., n'_k sont les longueurs des lignes de $P(s^N \pi_I(u) s^N)$. Or, avec l'algorithme de Schensted, on constate que ce tableau se déduit simplement de $P(\pi_I(u))$, comme le montre la figure suivante :



Conclusion : $\forall\, k,\ 1 \leq k \leq |I|,\ n'_k = n_k + 2N$. ∎

L'application du lemme est immédiate : s désigne toujours le mot formé des lettres de I dans l'ordre décroissant,

$$\bar{u} \leq \bar{v} \Longleftarrow \forall\, \alpha, \beta \in I^*,\ g(\alpha u \beta) \leq g(\alpha v \beta)$$

Ceci implique que $\forall\, \alpha, \beta \in I^*$, $\forall\, k,\ 1 \leq k \leq |I|$, la longueur de la k^{e}ligne de $P(\alpha u \beta)$ est inférieure ou égale à la longueur de la k^{e}ligne de $P(\alpha v \beta)$.
Donc : $\forall\ \alpha, \beta \in I^*$, $\forall\ k,\ 1 \leq k \leq |I|$, la longueur de la k^{e}ligne de $P(s^N \alpha u \beta s^N)$ est inférieure ou égale à la longueur de la k^{e}ligne de $P(s^N \alpha v \beta s^N)$, pour $N > \max(\lg(\alpha u \beta), \lg(\alpha v \beta))$.
Avec le lemme : $\forall\, \alpha, \beta \in I^*$, $\forall\, k,\ 1 \leq k \leq |I|$, la longueur de la k^{e}ligne de $P(\alpha \pi_I(u) \beta)$ est inférieure ou égale à la longueur de la k^{e}ligne de $P(\alpha \pi_I(v) \beta)$.
C'est à dire : $\forall\, \alpha, \beta \in I^*$, $g(\alpha \pi_I(u) \beta) \leq g(\alpha \pi_I(v) \beta)$,

$$\overline{\pi_I(u)} \leq \overline{\pi_I(v)} \text{ pour } Pla(I, <).$$

Les restrictions conservent bien les comparaisons. ■

Question 3 : quelles autres transformations conservent les comparaisons ?

3.3.2 La condition de superposition

Proposition 3.7 *Soient u et v deux mots tels que $u \leq_d v$, alors, pour tout i,j tels que $P(u)[i,j]$ et $P(v)[i,j]$ existent : $P(u)[i,j] \geq P(v)[i,j]$.*

Preuve - Autrement dit, si l'on superpose les tableaux $P(u)$ et $P(v)$, pour n'importe quelle case, la lettre de $P(u)$ est supérieure ou égale à celle de $P(v)$. La preuve de cette propriété est exactement celle du lemme .... En effet, on y montrait que s'il existe une case telle que $P(v)[i,j] > P(u)[i,j]$, avec $\alpha = P(u)[i,j]$, $N = \max\{N_{i,\alpha}(P(u)), N_{i,\alpha}(P(v))\}$, et $t = (\alpha\ \alpha\text{-}1\ ...\ \alpha\text{-}i\text{+}1)^N$, les longueurs de la i^{e}ligne de $P(ut) = P(u).t$ et $P(vt) = P(v).t$ valaient respectivement $\text{card}\{P(u)[i,k] \leq \alpha\}+N$ et $\text{card}\{P(v)[i,k] \leq \alpha\}+N$. Or, par définition de α et comme les lignes sont croissantes,

$$\text{card}\{P(u)[i,k] \leq \alpha\} + N > \text{card}\{P(v)[i,k] \leq \alpha\} + N,$$

donc $g(ut) \not\leq g(vt)$, et $\bar{u} \not\leq_d \bar{v}$. ■

3.3.3 Une application

Les conditions présentées permettent déjà d'éliminer certains cas de comparabilité. La proposition suivante en donne un exemple.

Proposition 3.8 *Pour l'ordre à droite $\leq_d$, induit par l'ordre produit sur les suites d'entiers, deux classes distinctes associées à deux mots de même longueur sont toujours incomparables.*

Preuve - Soient u et v deux mots de A^* de même longueur, dont les classes $\bar{u}$ et $\bar{v}$ sont distinctes.

Supposons que $\bar{u} \leq_d \bar{v}$. Les tableaux $P(u)$ et $P(v)$ ont le même nombre de cases, or $g(u) \leq g(v)$ pour l'ordre produit, donc nécessairement $g(u) = g(v)$, les deux tableaux ont la même forme. Soit x une lettre de A, considérons l'intervalle réduit à un singleton $I = \{x\}$, la condition sur les restrictions impose : $\overline{\pi_{\{x\}}(u)} \leq_d \overline{\pi_{\{x\}}(v)}$ pour l'alphabet réduit à $\{x\}$, ce qui signifie $|u|_x \leq |v|_x$.

$$\forall\, x \in A,\ |u|_x \leq |v|_x$$

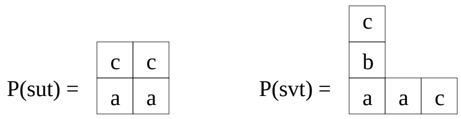
Or u et v ont la même longueur, donc,

$$\forall\, x \in A,\ |u|_x = |v|_x,$$

et par conséquent, u est une permutation de v. Les tableaux $P(u)$ et $P(v)$ ont la même forme, les mêmes lettres mais arrangées différemment puisque $P(u) \neq P(v)$. Ils ne peuvent donc pas vérifier la condition de superposition, ce qui est en contradiction avec $\bar{u} \leq_d \bar{v}$.

De même, $\bar{v} \leq_d \bar{u}$ aboutit à une contradiction. $\bar{u}$ et $\bar{v}$ sont incomparables.
■

Les conditions nécessaires qui viennent d'être présentées, ne sont pas suffisantes, comme le montre l'exemple suivant : considérons l'ordre à droite sur l'alphabet $A = \{a < b < c\}$, les mots $u = ca$ et $v = cab$ vérifient la condition de superposition, et les conditions de restrictions aux alphabets $\{a, b\}$ et $\{b, c\}$ (vérifiable grâce au système du ...). Cependant, pour $t = ca$, $g(ut) \not\leq g(vt)$ comme on le voit sur la figure, donc $\bar{u} \not\leq_d \bar{v}$. Elles sont encore moins suffisantes pour l'ordre bilatère, puisque $\bar{u} \leq \bar{v} \Rightarrow \bar{u} \leq_d \bar{v}$.



Nous allons maintenant étudier le cas $|A| = 3$, où les résultats qui ont été obtenus sont bien plus complexes que dans le cas $|A| = 2$.

3.4 Cas où $|A| = 3$

3.4.1 L'ordre à droite

Pour $|A| = 3$, il existe aussi un système d'inégalités caractérisant l'ordre à droite.

Proposition 3.9 *Soit* $A = \{a < b < c\}$ *un alphabet à trois lettres, et* $\leq_d$ *l'ordre à droite induit sur* $Pla(A,<)$*, alors* $\bar{u} \leq_d \bar{v}$ *ssi le couple* (u,v) *vérifie le système d'inégalités suivant :*

$$
\begin{aligned}
&|u|_{a,1} \leq |v|_{a,1} && (1)\\
&|u|_{a,1} + |u|_{b,1} \leq |v|_{a,1} + |v|_{b,1} && (2)\\
&|u|_{a,1} + |u|_{b,1} + |u|_{c,1} \leq |v|_{a,1} + |v|_{b,1} + |v|_{c,1} && (3)\\
&|u|_{b,2} \leq |v|_{b,2} && (4)\\
&|u|_{b,2} + |u|_{c,2} \leq |v|_{b,2} + |v|_{c,2} && (5)\\
&|u|_{c,3} \leq |v|_{c,3} && (6)\\
&|u|_{b,1} + |u|_{b,2} \leq |v|_{b,1} + |v|_{b,2} && (7)\\
&|u|_{b,2} + |u|_{c,2} + |u|_{c,1} \leq |v|_{b,2} + |v|_{c,2} + |v|_{c,1} && (8)\\
&|u|_{b,2} + \max(|u|_{b,1}, |u|_{c,2}) + |u|_{c,1} - |u|_{b,1} \leq \\
&\qquad |v|_{b,2} + \max(|v|_{b,1} + |u|_{c,1}, \min(|u|_{c,1}, |v|_{c,1}) + |v|_{c,2}) - |v|_{b,1} && (9)\\
&|u|_{b,2} + \max(|u|_{b,1}, |u|_{c,2}) + |u|_{c,1} \leq \\
&\qquad |v|_{b,2} + \max(|v|_{c,2}, \min(|u|_{b,1}, |v|_{b,1})) + \min(|u|_{c,1}, |v|_{c,1}) && (10)\\
&|u|_{c,3} + \min(|u|_{b,1}, |u|_{c,2}) \leq \\
&\qquad |v|_{c,3} + \min(|u|_{b,1}, |u|_{c,2}, |v|_{b,1}, |v|_{c,2}) && (11)\\
&|u|_{c,1} + |u|_{c,2} + |u|_{c,3} \leq |v|_{c,1} + |v|_{c,2} + |v|_{c,3} && (12)
\end{aligned}
$$

Preuve - A l'image du cas où $|A| = 2$, nous procédons en deux temps :

• Démontrons que le système (1) $\rightarrow$ (12) reste invariant après l'ajout à u et v d'une même lettre (c.à.d. pour toute lettre x, si (u,v) vérifie le système, alors (ux, vx) le vérifie aussi). Les relations (3),(5),(6) nous intéressent plus particulièrement, puisqu'elles comparent les formes des tableaux : elles signifient $g(u) \leq g(v)$, sachant que $g(u) = (|u|_{a,1} + |u|_{b,1} + |u|_{c,1}, |u|_{b,2} + |u|_{c,2}, |u|_{c,3})$.
Remarque : les relations (1),(2),(3) restent invariantes pour l'ajout de a,b ou c ; ceci est facile à montrer car l'ajout en 1^{re}ligne traité par l'algorithme de Schensted est identique pour u et v (les problèmes interviendront pour les autres lignes où les lettres insérées ne seront pas les mêmes pour u et v).
Démonstration pour les autres relations :
nous utiliserons le schema de la figure 7 pour résumer les différentes modifications possibles des $|u|_{a,i}$ après insertion à droite d'une lettre.

Soit u un mot de A^*, et $P(u)$ son tableau d'insertion, les valeurs des $|u|_{x,i}$ vont être modifiées suivant la lettre ajoutée à u, qui sera donc insérée à droite de $P(u)$. L'algorithme de Schensted (insertion à droite) nous donne les cas suivants (à chaque fois, nous ne donnons que les valeurs des $|u|_{x,i}$ qui changent) :
• La lettre ajoutée est un **c** : un seul cas,

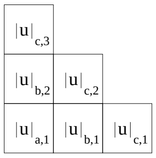


FIG. 7 – Tableau caractérisant un tableau de Young sur A

-cas 1 :

$$|uc|_{c,1} = |u|_{c,1} + 1$$

• La lettre ajoutée est un **b** : deux cas se présentent,
-cas 2 : si $|u|_{c,1} = 0$, alors,

$$|ub|_{b,1} = |u|_{b,1} + 1$$

-cas 3 : si $|u|_{c,1} > 0$, alors,

$$\begin{aligned} |ub|_{b,1} &= |u|_{b,1} + 1 \\ |ub|_{c,1} &= |u|_{c,1} - 1 \end{aligned}$$

• La lettre ajoutée est un **a** : quatre cas se présentent,
-cas 4 : si $|u|_{b,1} = 0$ et $|u|_{c,1} = 0$, alors,

$$|ua|_{a,1} = |u|_{a,1} + 1$$

-cas 5 : si $|u|_{b,1} = 0$ et $|u|_{c,1} > 0$, alors,

$$\begin{aligned} |ua|_{a,1} &= |u|_{a,1} + 1 \\ |ua|_{c,1} &= |u|_{c,1} - 1 \end{aligned}$$

-cas 6 : si $|u|_{b,1} > 0$ et $|u|_{c,2} = 0$, alors,

$$\begin{aligned} |ua|_{a,1} &= |u|_{a,1} + 1 \\ |ua|_{b,1} &= |u|_{b,1} - 1 \\ |ua|_{b,2} &= |u|_{b,2} + 1 \end{aligned}$$

-cas 7 : si $|u|_{b,1} > 0$ et $|u|_{c,2} > 0$, alors,

$$\begin{aligned} |ua|_{a,1} &= |u|_{a,1} + 1 \\ |ua|_{b,1} &= |u|_{b,1} - 1 \\ |ua|_{b,2} &= |u|_{b,2} + 1 \\ |ua|_{c,2} &= |u|_{c,2} - 1 \\ |ua|_{c,3} &= |u|_{c,3} + 1 \end{aligned}$$

Ces différents cas sont symbolisés dans la figure 8 par de petits schemas.

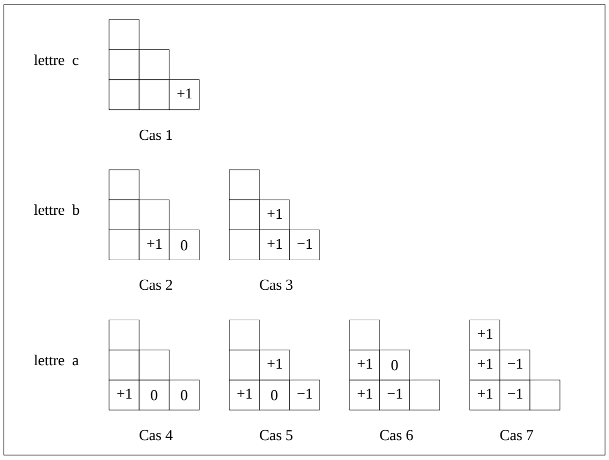
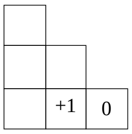
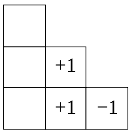
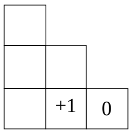
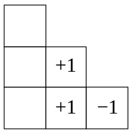


FIG. 8 – Modifications possibles des $|u|_{x,i}$

Le mot v présente les mêmes cas de modification. Si l'on considère le système vérifié par le couple (u, v), il y a donc 16 cas à étudier si on ajoute la lettre a, 4 cas à étudier si on ajoute un b, et un seul cas si on ajoute un c. On remarque aisément que si (u, v) vérifie le système, alors (uc, vc) le vérifie aussi, car seules les relations (8) et (10) se modifient et elles se conservent pour le couple (uc, vc). Sans détailler l'ensemble des vérifications, faciles mais fastidieuses, les figures 9, 10 résument comment se conservent les inégalités, après l'ajout de b ou a. Dans ces tableaux, chaque case correspond à un cas de modification du système. Les commentaires s'interprètent de la manière suivante, appelons x la lettre insérée :

- *(i) directement* signifie que, dans le cas présent, le couple (ux, vx) vérifie l'ingalité (i), car (u, v) vérifiait déjà cette inégalité,
- *(i) avec (j)* signifie que le couple (ux, vx) vérifie l'inégalité (i), car (u, v) vérifiait l'inégalité (j),
- si une inégalité n'est pas mentionnée, c'est qu'aucune de ses valeurs n'ont été modifiées après l'ajout de x.

u \ v		
	(9) avec (4) (10) avec (4) (11) avec (12)	(5) directement (8) directement (9) directement (10) directement (11) directement
	(5) avec (8) (8) directement (9) directement (10) directement (11) avec (12)	(5) directement (8) directement (9) directement (10) directement (11) directement

Dans tous les cas , (7) et (12) se conservent directement

FIG. 9 – Lettre ajoutée : b

u \ v	$\begin{array}{ccc} \square & & \\ \square & \square & \\ +1 & 0 & 0 \end{array}$	$\begin{array}{ccc} \square & & \\ \square & +1 & \\ +1 & 0 & -1 \end{array}$	$\begin{array}{ccc} \square & & \\ +1 & 0 & \\ +1 & -1 & \square \end{array}$	$\begin{array}{ccc} +1 & & \\ +1 & -1 & \\ +1 & -1 & \square \end{array}$
$\begin{array}{ccc} \square & & \\ \square & \square & \\ +1 & 0 & 0 \end{array}$		(5) directement (8) directement (9) directement (10) directement (11) directement	(4) directement (5) directement (8) directement (9) directement (10) directement (11) directement	(4) directement (5) directement (6) directement (8) directement (9) directement (10) directement (11) directement
$\begin{array}{ccc} \square & & \\ \square & +1 & \\ +1 & 0 & -1 \end{array}$	(5) avec (8) (8) directement (9) directement (10) directement (11) directement	(5) directement (8) directement (9) directement (10) directement (11) directement	(4) directement (5) directement (8) directement (9) directement (10) directement (11) directement	(4) directement (5) avec (9) (6) directement (8) directement (9) directement (10) directement si $\lvert u\rvert_{c,1} > \lvert v\rvert_{c,1}$ sinon avec (9) (11) directement
$\begin{array}{ccc} \square & & \\ +1 & 0 & \\ +1 & -1 & \square \end{array}$	(4) avec (7) (5) avec (8) (8) avec (10) (9) avec (8) (10) directement (11) directement	(4) avec (7) (5) directement (8) directement (9) avec (10) (10) directement (11) directement	(4) directement (5) directement (8) directement (9) directement (10) directement (11) directement	(4) directement (5) avec (4) (6) directement (8) avec (10) (9) directement (10) directement (11) directement
$\begin{array}{ccc} +1 & & \\ +1 & -1 & \\ +1 & -1 & \square \end{array}$	(4) avec (7) (5) directement (6) avec (11) (8) directement (9) avec (8) (10) directement (11) directement	(4) avec (7) (5) directement (6) avec (11) (8) directement (9) avec (10) (10) directement (11) directement	(4) directement (5) directement (6) avec (11) (8) directement (9) directement (10) directement (11) directement	(4) directement (5) directement (6) directement (8) directement (9) directement (10) directement (11) directement

Dans tous les cas, (7) et (12) se conservent directement.

FIG. 10 – Lettre ajoutée : a

Dans un deuxième temps, pour chaque inégalité, nous donnons un mot t tel que, si (u, v) ne respecte pas l'inégalité, alors $g(ut) \not\leq_d g(vt)$ (nous préciseront exactement la ligne où l'ordre produit n'est plus respecté). Voici la liste des inégalités, et le mot t à insérer si le couple (u, v) ne vérifie pas l'inégalité : x^∞ signifie prendre x^n avec n'importe quel n très grand,

(1) : ajouter $t = a^\infty$	(problème en ligne 1)
(2) : ajouter $t = b^\infty$	(problème en ligne 1)
(3) : $g(u) \not\leq_d g(v)$, prendre le mot vide	(problème en ligne 1)
(4) : ajouter $t = (ba)^\infty$	(problème en ligne 2)
(5) : $g(u) \not\leq_d g(v)$, prendre le mot vide	(problème en ligne 2)
(6) : $g(u) \not\leq_d g(v)$, prendre le mot vide	(problème en ligne 3)
(7) : ajouter $t = a^\infty(ba)^\infty$	(problème en ligne 2)
(8) : ajouter $t = b^\infty$	(problème en ligne 2)
(9) : ajouter $t = b^{\lvert u\rvert_{c,1}}(ca)^{\lvert v\rvert_{b,1}+\lvert u\rvert_{c,1}}$	(problème en ligne 2)
(10) : ajouter $t = a^{\lvert u\rvert_{b,1}}b^{\lvert u\rvert_{c,1}}$	(problème en ligne 2)
(11) : ajouter $t = a^{\min(\lvert u\rvert_{b,1},\lvert u\rvert_{c,2})}$	(problème en ligne 3)
(12) : ajouter $t = (ba)^\infty$	(problème en ligne 3)

■

Ce système a été trouvé empiriquement, en ajoutant aux conditions simples et intuitives (comme par exemple les relations (1),(2),(4),(7) qui signifient $\pi_{\{a,b\}}(\bar{u}) \leq_d \pi_{\{a,b\}}(\bar{v})$) des inégalités traitant les cas critiques trouvés au fur et à mesure. Si le système admettait 4 inégalités pour $|A| = 2$, il en comporte 12 pour $|A| = 3$. En plus, de cette nette augmentation du nombre d'inégalités, plusieurs d'entre elles ne présentent plus de symétrie entre u et v. Aussi, ce système n'est pas directement généralisable pour $|A| > 3$.

Question 4 : Existe-il un moyen de symétriser ou reécrire ce système de manière plus élégante, et généralisable ?

Question 5 : Pour un alphabet fixé, étant donné que l'on peut déterminer les différents cas d'éxécution de l'algorithme de Schensted, est-il possible de générer automatiquement le système d'inégalités sur les $|u|_{x,i}$?

3.4.2 L'ordre à gauche

Le système précédent permet de déterminer si deux classes $\bar{u}$ et $\bar{v}$ sont comparables, puisque : $\bar{u} \leq_g \bar{v} \iff \overline{\#u} \leq_d \overline{\#v}$. Ceci nécessite de calculer $P(\#u)$ et $P(\#v)$ pour obtenir les valeurs des $|\#u|_{x,i}$ et $|\#v|_{x,i}$. Et, s'il existe probablement un système pour $\leq_g$ du même ordre de complexité que celui de $\leq_d$, je n'ai pas trouvé de moyen de le déduire du système de $\leq_d$.

3.4.3 L'ordre bilatère

Remarque : Comme nous l'avions annoncé, voilà un exemple de couple (u, v) tel que $\bar{u} \leq_d \bar{v}$ et $\bar{u} \leq_g \bar{v}$, mais $\bar{u} \not\leq \bar{v}$. Considérons $u = a$ et $v = ab$: $\bar{u} \leq_d \bar{v}$, car (u, v) vérifie le système ; $\bar{u} \leq_g \bar{v}$, car $\#u = c$ et $\#v = bc$, $(\#u, \#v)$ vérifie le système. Cependant, en prenant $s = c$ et $t = ca$, l'algorithme de Schensted donne les tableaux ci-dessous, et donc $g(sut) \not\leq g(svt)$.

$$\mathrm{P(sut)} = \begin{array}{|c|c|}\hline c & c \\ \hline a & a \\ \hline\end{array} \qquad \mathrm{P(svt)} = \begin{array}{|c|cc}\cline{1-1} c & & \\ \cline{1-1} b & & \\ \hline a & \multicolumn{1}{c|}{a} & \multicolumn{1}{c|}{c} \\ \hline\end{array}$$

4 L'ordre sur les sommes partielles

4.1 Définition

Définition 4.1 *soit* $(m_1, m_2, \ldots, m_k)$ *et* $(n_1, n_2, \ldots, n_{k'})$ *deux suites d'entiers, on appelera* ordre sur les sommes partielles *l'ordre défini par :*
$(m_1, m_2, \ldots, m_k) \preceq (n_1, n_2, \ldots, n_{k'})$ *si*

$$\begin{cases} k \leq k' \text{ et} \\ \forall\, i,\ 1 \leq i \leq k,\ m_1 + \ldots + m_i \leq n_1 + \ldots + n_i \end{cases}$$

Soit A alphabet fini totallement ordonné, l'ordre sur les sommes partielles induit sur $Pla(A, <)$ trois ordres $\leq_d$, $\leq_g$, $\leq$, conformément à la définition 1.3.

Remarque 4.1 : soient u et v deux mots de A^*, avec le théorème de Greene 1.3, $g(u) \leq g(v)$ s'interprète par : le cardinal maximum de l'union de k sous-suites croissantes de v est toujours supérieur au cardinal de l'union de k sous-suites croissantes de u.
Nous commençons d'abord par étudier le cas $|A| = 2$.

4.2 Cas où $|A| = 2$

Considérons un alphabet $A = \{a < b\}$, et les trois ordres $\leq_d$, $\leq_g$, $\leq$, induits sur $Pla(A, <)$ par l'ordre sur les sommes partielles. Nous disposons des trois caractérisations suivantes :

Proposition 4.1 *Soit $\leq_d$ l'ordre à droite sur $Pla(A, <)$, soient u et v deux mots de A^*, de classes respectives $\bar{u}$ et $\bar{v}$, alors :*

$$\bar{u} \leq_d \bar{v} \iff \begin{cases} g(u) \leq g(v) \\ |u|_a \leq |v|_a \end{cases}$$

Comme nous travaillons avec l'ordre sur les sommes partielles, ce système se réécrit :

$$\bar{u} \leq_d \bar{v} \iff \begin{cases} |u|_{a,1} + |u|_{b,1} \leq |v|_{a,1} + |v|_{b,1} \\ |u|_{a,1} + |u|_{b,1} + |u|_{b,2} \leq |v|_{a,1} + |v|_{b,1} + |v|_{b,2} \\ |u|_{a,1} \leq |v|_{a,1} \end{cases}$$

Preuve - La preuve est identique à celle effectuée pour l'ordre produit (proposition 3.1). On procède par cas, pour montrer que le système se conserve après l'ajout d'une lettre. Puis, comme nécessairement la condition $g(u) \leq g(v)$ est vérifiée, il suffit de montrer que si la dernière inégalité n'est pas respectée, il existe un mot t tel que $g(ut) \not\leq g(vt)$: si $|u|_{a,1} > |v|_{a,1}$, prendre $t = a^n$ avec $n \geq |v|_{b,1}$. ∎

Proposition 4.2 *Soit $\leq_g$ l'ordre à gauche sur $Pla(A, <)$, soient u et v deux mots de A^*, de classes respectives $\bar{u}$ et $\bar{v}$, alors :*

$$\bar{u} \leq_g \bar{v} \iff \begin{cases} g(u) \leq g(v) \\ |u|_b \leq |v|_b \end{cases}$$

Preuve - En effet, avec l'involution de Schützenberger,

$$\bar{u} \leq_g \bar{v} \iff \overline{\#u} \leq_d \overline{\#v},$$

et $|\#u|_a = |u|_b$, $|\#v|_a = |v|_b$. ∎

Proposition 4.3 *Soit $\leq$ l'ordre bilatère sur $Pla(A, <)$, soient u et v deux mots de A^*, de classes respectives $\bar{u}$ et $\bar{v}$, alors :*

$$\bar{u} \leq \bar{v} \iff \begin{cases} g(u) \leq g(v) \\ |u|_a \leq |v|_a \\ |u|_b \leq |v|_b \end{cases}$$

Preuve - Ceci se démontre aussi au cas par cas. ∎

4.3 Quelques propriétés intéressantes

4.3.1 L'ordre à droite et l'ordre à gauche

L'ordre à droite et l'ordre à gauche ne sont pas conservés pour les morphismes de restrictions. En effet, pour l'alphabet $A = \{a < b\}$, d'après les systèmes précédents, $\bar{b} \leq_d \bar{a}$ et $\bar{a} \leq_g \bar{b}$. Or $\pi_{\{b\}}(b) = b$, $\pi_{\{b\}}(a)$ est le mot vide, et donc $\pi_{\{b\}}(b) \not\leq_d \pi_{\{b\}}(a)$ dans $Pla(\{b\}, <)$. Pour $\leq_g$, le même résultat est obtenu avec $\pi_{\{a\}}$.

4.3.2 L'ordre bilatère

Contrairement aux deux ordres précédents, les morphismes de restriction sont des applications croissantes dans le cas de l'ordre bilatère.

Proposition 4.4 *Soit I un intervalle de l'alphabet fini $(A, <)$, et π_I le morphisme de restriction qui lui est associé, notons de la même manière $\leq$, l'ordre bilatère induit sur $Pla(A, <)$ et sur $Pla(I, <)$ par l'ordre sur les sommes partielles, alors, soient u,v deux mots de A^*,*

$$\bar{u} \leq_d \bar{v} \text{ dans } Pla(A, <) \Rightarrow \pi_I(\bar{u}) \leq_d \pi_I(\bar{v}) \text{ dans } Pla(I, <)$$

Preuve - La preuve est parfaitement identique à celle pour l'ordre produit, elle utilise le lemme 3.1 de la même manière, en comparant les sommes partielles des longueurs des lignes, au lieu de comparer directement les longueurs des lignes. ∎

4.4 Conjecture

La relation étudiée dans cette partie est l'ordre bilatère. Il existe deux manières simples de trouver des mots comparables pour cet ordre, elles sont présentées dans les deux propositions suivantes. L'alphabet considéré est $(A, <)$.

Proposition 4.5 (transposition)
Etant donné un mot $u = a_1 a_2 \dots a_i a_{i+1} \dots a_n$, $a_i \in A$, tel que $a_i > a_{i+1}$, en effectuant une transposition élémentaire sur ces deux lettres consécutives, on obtient le mot $u' = a_1 a_2 \dots a_{i+1} a_i \dots a_n$. Ce nouveau mot vérifie :

$$g(u) \leq g(u')$$

Et, par conséquent, $\overline{u} \leq \overline{u'}$.

Preuve - Ceci s'obtient à partir de l'interprétation de

$$g(u) \leq g(u')$$

en termes de sous-suites croissantes, sachant que toute sous-suite croissante de u est une sous-suite croissante de u'.

Maintenant, soient s et t deux mots quelconques, $su't$ est aussi obtenu après une transposition élémentaire de deux lettres consécutives, dans l'ordre décroissant, du mot sut, et donc $g(sut) \leq g(svt)$. Ceci prouve que $\overline{u} \leq \overline{u'}$. ■

Proposition 4.6 (insertion)
Etant donné un mot u et une lettre x de A, soit u' un mot obtenu en insérant x dans u, alors ce nouveau mot vérifie :

$$g(u) \leq g(u')$$

Et, par conséquent, $\overline{u} \leq \overline{u'}$.

Preuve - La preuve est identique à la précédente, car toute sous-suite croissante de u est une sous-suite croissante de u'. ■

Avec ces deux transformations, nous pouvont introduire une relation sur $Pla(A,<)$ définie par :

Définition 4.2 *Soient $\overline{u}$ et $\overline{u'}$ deux classes de $Pla(A,<)$, la relation $\overline{u} \to \overline{u'}$ signifie qu'il existe un représentant v de $\overline{u}$ et un représentant v' de $\overline{u'}$, tels que v' soit obtenu à partir de v par une transposition ou une insertion.*

Proposition 4.7 *La relation $\to$ n'admet pas de cycle.*

Preuve - En effet, supposons qu'il existe une suite de classes distinctes telles que :

$$\overline{u_1} \to \overline{u_2} \to \ldots \to \overline{u_k} \to \overline{u_1}$$

ce qui signifie :
il existe $u_1' \in \overline{u_1}$, $u_2'' \in \overline{u_2}$, tels que u_2'' est obtenu à partir de u_1',
il existe $u_2' \in \overline{u_2}$, $u_3'' \in \overline{u_3}$, tels que u_3'' est obtenu à partir de u_2',
$\vdots$

il existe $u'_k \in \overline{u_k}$, $u''_1 \in \overline{u_1}$, tels que u''_1 est obtenu à partir de u'_k, donc d'après les propriétés sur la transposition et l'insertion,

$$\begin{aligned}
\overline{u_1} = \overline{u'_1} &\leq \overline{u''_2} = \overline{u_2} = \overline{u'_2} \\
&\leq \overline{u''_3} = \overline{u_3} = \ldots \\
&\quad\vdots \\
&\leq \overline{u''_k} = \overline{u_k} = \overline{u'_k} \\
&\leq \overline{u''_1} = \overline{u_1}
\end{aligned}$$

c.à.d. $\overline{u_1} \leq \overline{u_2} \leq \overline{u_3} \leq \ldots \leq \overline{u_k} \leq \overline{u_1}$. Or $\leq$ est un ordre sur $Pla(A, <)$, donc

$$\overline{u_1} = \overline{u_2} = \ldots = \overline{u_k}.$$

Ceci est en contradiction avec l'hypothèse que toutes ces classes sont distinctes. ■

Question 6 : Est-il possible de démontrer directement (sans passer par l'ordre $\leq$) que la relation $\rightarrow$ est sans cycles ?

Définition 4.3 *La relation précédente n'admet pas de cycles, sa fermeture transitive* $\rightarrow^*$ *définit un ordre sur* $Pla(A, <)$.

Il est clair, d'après les propositions 4.5 et 4.6 , que :

$$\bar{u} \rightarrow^* \bar{v} \Longrightarrow \bar{u} \leq \bar{v}.$$

Aussi, la question suivante se pose :
Question 7 : l'ordre $\leq$ sur $Pla(A, <)$ est-il exactement l'ordre $\rightarrow^*$?

Sur de très nombreux exemples, ces deux ordres coïncident. Dans le cas $|A| = 2$, il y a effectivement égalité.

Vérification dans le cas simple $|A| = 2$

Soit $A = \{a < b\}$, u et v deux mots tels que $\bar{u} \leq \bar{v}$. Prouvons que $\bar{u} \rightarrow^* \bar{v}$.

Il suffit de considérer les codages lignes des tableaux $P(u)$ et $P(v)$. Soit $c(u)$ le codage ligne de $P(u)$, $c(u) \equiv u$ (ils ont le même tableau d'insertion),

$$c(u) = b^{|u|_{b,2}} a^{|u|_{a,1}} b^{|u|_{b,1}}.$$

Le codage ligne $c(v)$ de $P(v)$ vaut :

$$c(v) = b^{|v|_{b,2}} a^{|v|_{a,1}} b^{|v|_{b,1}}.$$

Supposons $|u|_{b,2} \geq |v|_{b,2}$. En effectuant des transpositions, comme $b > a$, on peut faire passer des b de gauche à droite :

$$\overline{b^{|u|_{b,2}}a^{|u|_{a,1}}b^{|u|_{b,1}}} \rightarrow^* \overline{b^{|v|_{b,2}}a^{|u|_{a,1}}b^{|u|_{b,1}+|u|_{b,2}-|v|_{b,2}}}$$

De plus, d'après le système de la proposition 4.3, $|u|_{a,1} \leq |v|_{a,1}$ et $|u|_{b,1} + |u|_{b,2} - |v|_{b,2} \leq |v|_{b,1}$. En insérant des a et des b, on obtient :

$$\overline{b^{|v|_{b,2}}a^{|u|_{a,1}}b^{|u|_{b,1}+|u|_{b,2}-|v|_{b,2}}} \rightarrow^* \overline{c(v)}$$

Par conséquent, $\overline{u} = \overline{c(u)} \rightarrow^* \overline{c(v)} = \overline{v}$.

Supposons maintenant $|u|_{b,2} < |v|_{b,2}$. Si $|u|_{b,1} \leq |v|_{b,1}$, alors, par insertions,

$$\overline{b^{|u|_{b,2}}a^{|u|_{a,1}}b^{|u|_{b,1}}} \rightarrow^* \overline{b^{|u|_{b,2}}b^{|v|_{b,2}-|u|_{b,2}}a^{|u|_{a,1}}a^{|v|_{a,1}-|u|_{a,1}}b^{|u|_{b,1}}b^{|v|_{b,1}-|u|_{b,1}}}$$

donc, $\overline{u} = \overline{c(u)} \rightarrow^* \overline{c(v)} = \overline{v}$.

Sinon, $|u|_{b,1} > |v|_{b,1}$. On insère alors des a :

$$\begin{aligned} \overline{b^{|u|_{b,2}}a^{|u|_{a,1}}b^{|u|_{b,1}}} &\rightarrow^* \overline{b^{|u|_{b,2}}a^{|u|_{a,1}}b^{|u|_{b,1}}a^{|u|_{b,1}-|v|_{b,1}}} \\ &= \overline{b^{|u|_{b,2}+|u|_{b,1}-|v|_{b,1}}a^{|u|_{a,1}+|u|_{b,1}-|v|_{b,1}}b^{|v|_{b,1}}} \end{aligned}$$

Les deux inégalités suivantes sont vraies :
$|u|_{b,2} + |u|_{b,1} - |v|_{b,1} \leq |v|_{b,2}$,
$|u|_{a,1} + |u|_{b,1} - |v|_{b,1} \leq |v|_{a,1}$.
Donc, en insérant les a et les b qui manquent, on obtient aussi :
$\overline{u} = \overline{c(u)} \rightarrow^* \overline{c(v)} = \overline{v}$.

Finalement, l'ordre bilatère est bien l'ordre $\rightarrow^*$, quand $|A| = 2$.

Pour A quelconque, la difficulté de la démonstration de $\bar{u} \leq \bar{v} \Rightarrow \bar{u} \rightarrow^* \bar{v}$, réside dans le choix des représentants des classes plaxiques qui vont relier $\bar{u}$ à $\bar{v}$. Pour $|A| = 2$, ce choix est assez restreint, c'est ce qui permet d'effectuer la preuve ci-dessus.

L'ordre d'Ehresmann faible

Cet ordre sur les permutations est lié à l'opération de transposition de la proposition 4.5. Soit σ et τ deux permutations de S_n, σ sera plus petit que τ, s'il existe une transposition élémentaire σ_i telle que

$$\tau = \sigma\sigma_i \text{ et } l(\tau) = l(\sigma) + 1$$

où $l(\sigma)$ désigne le nombre d'inversions de σ. C'est exactement l'opération de transposition décrite en 4.5, qui permet de passer du mot représentant σ au mot représentant τ. L'ordre d'Ehresmann est un treillis (cf par exemple [5] p19-20, pour une démonstration). Mais il semble difficile d'appliquer cette propriété à la recherche d'une démonstration de l'implication : $\bar{u} \leq \bar{v} \Rightarrow \bar{u} \rightarrow^* \bar{v}$.

Conclusion

À l'issue de cette étude, beaucoup de questions restent ouvertes. En fait, les liens entre les ordres introduits sur le monoïde plaxique et les ordres syntaxiques de la théorie des langages semblent difficiles à établir. Si cette étude ne traite pas des applications de tels ordres, elle reflète une partie de la richesse du domaine, et donne des exemples de résultats de combinatoire.

De nombreuses approches du problème étaient envisageables : travail sur les mots ou sur les tableaux de Young, recherche de transformations intéressantes dans le monoïde plaxique, choix de l'alphabet A (importance de $|A|$, éventuellement pour raisonner par récurrence), factorisations dans le monoïde plaxique, variantes de l'algorithme de Schensted, ...

Le problème de la décidabilité des comparaisons pour les ordres induits par l'ordre produit et l'ordre sur les sommes partielles, reste cependant ouvert. Une solution nécessiterait de mieux appréhender les effets de l'algorithme de Schensted pour un alphabet fixé. Considérer les tableaux de Young, sur un alphabet fixé, comme des vecteurs d'entiers (pour chaque lettre, on compte le nombre d'occurences sur chaque ligne) semble être un bon moyen de décrire l'algorithme de Schensted. Il serait par exemple intéressant de pouvoir générer automatiquement des systèmes d'inégalités tels que celui de la proposition 3.9. Une preuve de la conjecture de la partie 4, ou un contre exemple, apporterait aussi des informations sur le monoïde plaxique, et en particulier sur la structure des classes plaxiques.

Pour finir, je tiens à remercier Daniel Krob et Michel Morvan qui m'ont dirigé, pour leurs précieux conseils et leur soutien, ainsi que Alain Lascoux dont les renseignements sur le monoïde plaxique m'ont été très utiles.

Références

[1] J. Cassaigne, M. Espie, F. Hivert, D. Krob, J-C. Novelli, *The Chinese monoid*, Institut Blaise Pascal, Rapport LITP 95/46, 1995.

[2] Fomin S., *Schensted Algorithms for Dual Graded Graphs*, Institut Mittag-Leffler, Report 16, 1991/1992.

[3] Greene C., *An extension of Schented's theorem*, Adv. Maths., **14**, 254-265, 1974.

[4] Greene C., *Some partitions associated with a partially ordered set*, J. Combinatorial Theory, Serie A 20, 69-79, 1976.

[5] Knuth D.E, *The art of computer programming, vol. 3, Sorting and Searching*, Addison Wesley, 1976.

[6] Lascoux A., Schützenberger M.P., *Croissance des polynômes de Foulkes-Green*, C. R. Acad. Sc. Paris, t. 288, janvier 1979.

[7] Lascoux A., Schützenberger M.P., *Keys and standards bases*, Invariant theory, D. Stanton Ed., Springer, 1990.

[8] Lascoux A., Schützenberger M.P., *Le monoïde plaxique*, Noncommutative structures in Algebra and Geometric Combinatorics, Quaderni de la ricerca scientifica, **109**, 129-156, Rome, 1981.

[9] Schensted C., *Longest increasing and decreasing subsequences*, Canad. J. Math., **13**, 179-191, 1961.

[10] Schützenberger M.P., *La correspondamce de Robinson*, Combinatoire et représentation du groupe symétrique, Strasbourg 1976, D. Foata Ed., Lecture notes in Maths **579**, Springer-Verlag, 59-135, 1977.

[11] Viennot G., *Chain and antichain families, grids and young tableaux*, Annals of Discrete Mathematics, **23**, 409-464, 1984.