RAIRO. INFORMATIQUE THÉORIQUE ET APPLICATIONS

BRIGITTE ROZOY

Outils et résultats pour les transducteurs boustrophédons

RAIRO. Informatique théorique et applications, tome 20, n° 3 (1986), p. 221-250

<http://www.numdam.org/item?id=ITA_1986__20_3_221_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 20, n° 3, 1986, p. 221 à 249)

OUTILS ET RÉSULTATS POUR LES TRANSDUCTEURS BOUSTROPHEDONS (*)

par Brigitte ROZOY (1)

Communiqué par J. BERSTEL

Résumé. – *On montre deux lemmes d'itération pour les langages produits par des transducteurs boustrophédons, déterministes ou non, ainsi que la non-appartenance du Dyck $D_2'^*$* à la famille non déterministe.

Abstract. – *We consider the families of languages produced by two-way transducers, and prove iteration lemmas for them. Thus, it is shown that the Dyck language $D_2'^*$ doesn't belong to the non deterministic family.*

1. INTRODUCTION

Des machines telles que les transducteurs, ainsi que l'objet algébrique associé, les transductions rationnelles, ont, dans les dernières années, fait l'objet de nombreuses et fécondes études. La raison en est certainement leur parfaite adéquation à la hiérarchie de Chomsky-Schutzenberger, qui en a fait un outil privilégié dans l'étude des propriétés structurelles des langages, qu'ils soient rationnels ou algébriques.

Il est naturel d'introduire d'autres machines, les transducteurs boustrophédons : à la différence des premiers, ils lisent leur ruban d'entrée aussi bien de gauche à droite que de droite à gauche, mais ne peuvent parcourir leur ruban de sortie que dans un seul sens, sans jamais revenir en arrière sur les caractères précédemment écris.

Un certain nombre de leurs propriétés provient alors de ce qu'ils n'ont qu'une mémoire finie (la tour de contrôle) et pas de mémoire auxiliaire

(*) Reçu en avril 1985, révisé en novembre 1985.
(1) L.I.T.P. et Laboratoire d'Informatique, Université, 14032 Caen Cedex.

Informatique théorique et Applications/Theoretical Informatics and Applications
0296-1598 86/03 221 29/$4.90/

(ruban, pile...). Si on leur donne, comme mot d'entrée, n'importe quel mot de A^*, (où A est l'alphabet, fini, des entrées), ils n'accepteront que ceux d'un sous-ensemble rationnel.

Par contre, la capacité à revenir en arrière munit — ou afflige! — ces machines de pouvoirs très différents de ceux connus pour leurs parents transducteurs droits; l'ensemble des mots produits en sortie peut très bien ne pas être rationnel : ainsi l'ensemble $\{a^n b^n c^n \mid n \in N\}$, de même que $D_1'^*$, le dyck sur $(a+a')$, sont-ils images d'un langage d'entrée égal à A^* (dès que l'alphabet A a au moins deux éléments).

Par ailleurs, un résultat de Rajlich [17], utilisant des grammaires absolument parallèles, établit que la famille des langages produit par des transducteurs boustrophédons déterministes (DTWT) est égale à la famille des langages produit par une grammaire matricielle d'index fini (MLIF), et de façon équivalente, égale à celle des langages images d'un système EDTOL d'index fini [13]. Une caractérisation grammaticale de la famille des langages produit par des transducteurs boustrophédons non déterministes reste encore inconnue, et certainement problématique (*voir* [7]).

De plus, les transducteurs boustrophédons forment un modèle adapté à l'étude de problèmes liés aux compilateurs, et efficaces pour résoudre certaines questions de complexité; ils sont également assez semblables aux machines de Turing qui ne visitent qu'un nombre borné de fois leur ruban d'entrée ([9] 1).

Mais, bien qu'ayant fait l'objet de nombreuses études, les transducteurs boustrophédons n'ont jamais acquis de statut aussi clair et puissant que leurs aînés.

Notre propos est de délimiter plus soigneusement les classes considérées en associant aux facteurs des mots des objets algébriques qui faciliteront l'étude des langages produits. Ainsi, en définissant pour chaque facteur Y d'un mot d'entrée un polynôme formel $CH(Y)$, nous avons pu prouver que le langage $D_1'^*$, s'il appartient à la famille non déterministe NDTWT, n'est pas élément de la famille déterministe DTWT [20].

De même, dans cet article nous associons à chaque facteur d'un mot une fonction d'un ensemble fini dans un ensemble fini, et une relation d'équivalence liée au langage produit. Ces outils conduisent à la démonstration de lemmes d'itération pour les langages considérés, et permettent de conclure que $D_2'^*$ n'appartient pas à la famille non déterministe NDTWT.

Nous pensons qu'ils pourraient également se révéler fort utiles dans une attaque de la vieille question ouverte : la famille des langages quasi rationnels est-elle caractérisable en terme de finitude d'index de langages matriciels? (*cf.* Beauquier, Paun, Salomaa).

Le plan de cet article est le suivant

2. DESCRIPTION ET FONCTIONNEMENT DES MACHINES

Un transducteur boustrophédon est une machine qui est capable de lire son ruban d'entrée à la fois de droite à gauche et de gauche à droite, qui n'écrit sur son ruban de sortie que dans un seul sens, et qui ressemble à un automate fini, dans le sens où elle possède une tour de contrôle finie. Formellement, un transducteur boustrophédon sera un 6-uple $\tau=(\Sigma, \Delta, Q, q(0), F, \mathscr{C})$ où :

- Σ est l'alphabet d'entrée; il est fini;
- Δ est l'alphabet de sortie, également fini;
- Q est l'ensemble, fini, des états de la machine;
- $q(0)$, élément de Q, est l'état de départ;
- F, sous-ensemble de Q, est l'ensemble des états terminaux;
- $\mathscr{C}$ est un sous-ensemble fini de $Q \times \Sigma \times Q \times \Delta^* \times \{-1, 0, 1\}$.

Si l'automate est déterministe, $\mathscr{C}$ est le graphe d'une fonction partielle δ de $Q \times \Sigma$ dans $Q \times \Delta^* \times \{-1, +1, 0\}$, sinon dans $P_f(Q \times \Delta^* \times \{-1, 1, 0\})$. On peut interpréter $\delta(q, x)=(q', w', d)$ comme : étant en l'état q et lisant la lettre x, le transducteur passe en état q', écrit w' sur le ruban de sortie, place la tête d'écriture immédiatement à droite de w', et déplace la tête de lecture de d case.

Intuitivement, on peut décrire le mode de travail de ces machines comme suit :

— Un mot R est écrit sur le ruban d'entrée, la tête de lecture est placée sur la première lettre de R; le ruban d'écriture est vide; l'état initial est $q(0)$.

— Les mouvements commandés par la fonction de transition se succèdent alors. Si la fonction de transition n'est pas définie, ou si la tête de lecture pointe sur une case à gauche du mot, alors le calcul s'arrête, aucun mot n'est retenu à la sortie; c'est aussi le cas si les calculs bouclent indéfiniment à l'intérieur du mot; on dit que l'entrée est refusée.

— On dira que le mot d'entrée est accepté, soit qu'il est dans le domaine du transducteur, si le calcul ne boucle ni ne se bloque, et qu'il termine à la droite du mot en état d'acceptation final. On dit que le calcul a alors produit le mot R', écrit sur le ruban de sortie.

Une configuration est un 3-uple $(q, \ldots \uparrow x_i \ldots, w)$, où q est un état, x_i une occurrence de l'alphabet, alors pointée par la tête de lecture, et w la chaîne de caractères déjà portée sur le ruban de sortie.

Pour toute configuration, on notera

$$(q, \ldots \uparrow x(i) \ldots, w) \vdash (q', \ldots \uparrow x(i+d) \ldots, ww')$$

si l'ordre utilisé a été $\delta(q, x(i)) = (q', w', d)$; $\overset{*}{\vdash}$ sera la clotûre transitive de $\vdash$.

Le domaine et l'image du transducteur seront définis respectivement comme l'ensemble des mots acceptés et produits par les calculs; on supposera toujours l'ensemble des mots d'entrée égal à Σ^*, ce qui est d'ailleurs équivalent — au sens des familles produites — à le supposer égal à un rationnel quelconque (*voir* [11]).

Si R est un mot de Σ^*, on notera $\tau(R)$ l'ensemble de ses images par τ :

$$\tau(R) = \{ W \mid W \in \Delta^* \text{ et } \exists q(f) \in F : (q(0), \uparrow R, \varepsilon) \overset{*}{\vdash} (q(f), R\uparrow, w) \}.$$

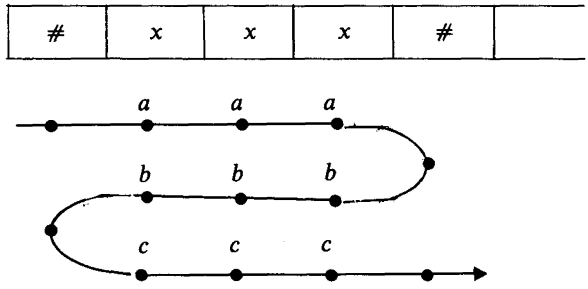
On notera $\operatorname{dom}\tau$ et $\operatorname{Im}\tau$ le domaine et l'image de τ :

$$\operatorname{dom}\tau = \{ R \mid R \text{ dans } \Sigma^* \text{ et il existe } w \text{ dans } \Delta^* \text{ tel que } w = \tau(R) \}.$$

$$\operatorname{Im}\tau = \{ w \mid w \text{ dans } \Delta^* \text{ et il existe } R \text{ dans } \Sigma^* \text{ tel que } w = \tau(R) \}.$$

Voici quelques exemples de transducteurs boustrophérons :

Exemple 1 : $\tau(1) = \{\Sigma, \Delta, Q, q(0), F, \mathscr{C}\}$



$\Sigma = \{\#, x\}$; $\Delta = \{a, b, c\}$; $Q = \{q_0, q_1, q_2, q_3\}$; $F = \{q_3\}$;

$\delta(q_0) = (q_1, \varepsilon, +1)$;

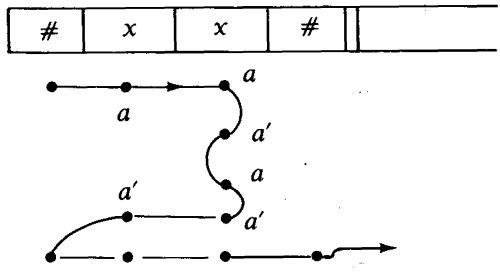
$\delta(q_1 x) = (q_1, a, +1)$; $\delta(q_1, \#) = (q_2, \varepsilon, -1)$;

$\delta(q_2, x) = (q_2, b, -1)$; $\delta(q_2, \#) = (q_3, \varepsilon, +1)$;

$\delta(q_3, x) = (q_3, C, +1)$; $\delta(q_3, \#) = (q_3, \varepsilon, +1)$.

Le transducteur $\tau(1)$ est déterministe, et son image est $\{a^n b^n c^n \mid n \in \mathbb{N}\}$.

Exemple 2 : $\tau(2) = \{\Sigma, \Delta, Q, q_0, F, \delta\}$



$\Sigma = \{\#, x\}$; $\Delta = \{a, a'\}$; $Q = \{q_0, q_1, q_2, q_3\}$; $F = \{q_3\}$;

$\delta(q_0 \#) = (q_1, \varepsilon, +1)$;

$\delta(q_1 x) = \{(q_1, a, +1); (q_2, a, 0)\}$; $\delta(q_1, \#) = (q_2, \varepsilon, -1)$;

$\delta(q_2, x) = \{(q_2, a', -1); (q_1, a', 0)$; $\delta(q_2, \#) = (q_3, \varepsilon, +1)$;

$\delta(q_3, x) = \delta(q_3, \#) = (q_3, \varepsilon, +1)$.

Le transducteur (2) est non déterministe, et produit le dyck $D_1'^*$.

Il a été démontré ailleurs que le *domaine* des transducteurs boustrophédons est toujours un *ensemble rationnel* et que *leurs images* sont toutes *récursives* (et même context-sensitives; *voir* par exemple [6] ou [16]).

Nous utiliserons ici toujours *une forme normale* des machines :

(i) le mot écrit à chaque étape est de longueur au plus égale à 1;

(ii) sous chaque lettre de R est écrit au moins une fois un mot non vide de Δ^* (*cf.* [6]).

3. QUELQUES OUTILS ASSOCIÉS AUX MOTS

Soit τ un transducteur boustrophédon, *donné sous forme normale.*

On se propose, dans ce paragraphe, de décrire les calculs de τ sur les mots d'entrée, mettant tour à tour en relief les états de la machine, ses mouvements, les lettres lues associées aux états, les lettres écrites pour une lettre donnée lue... Ceci nous conduira à définir plusieurs alphabets, et à introduire les définitions qui suivent.

NOTATIONS : Si A est un alphabet et W un mot dans A^*, on notera $W(i)$ la i-ième occurrence de lettre de A dans W : W s'écrit $W = W(1) \ldots W(n)$;

Soit R un mot de dom τ; on considère la suite d'éléments de $\mathscr{C}$, relative à un calcul réussi sur R.

Soit X un facteur de R.

3.1. Diverses notions de chemins

— On appelle chemin de R, noté $CH(R)$, le mot de $\mathscr{C}^*$ définit par :

• $CH(R) = c(1) \ldots c(n)$ et quelque soit i, $c(i)$ est dans $\mathscr{C}$.

• $c(1) = (q(0), R(1), \ldots)$ et $c(n) = (., R(n), q(f), +1)$, où $R(1)$ et $R(n)$ sont la première et la dernière occurrence de $\mathscr{C}$ dans R.

• Quelque soit i, si $c(i) = (q(i), x(j), q(i+1), y(i), d(i))$ alors $(q(i+1), y(i), d(i))$ est dans $(q(i), x(j))$.

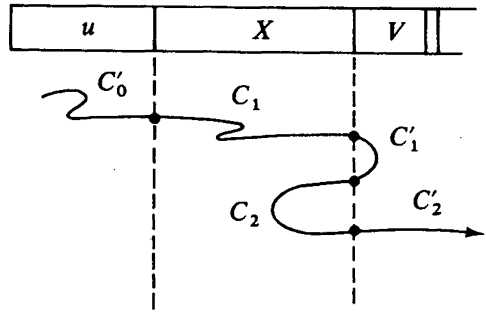
— On appelle chemin du facteur X de R, noté $CH(X)$, le mot de $(\mathscr{C}^*)^*$ définit par :

• $CH(X) = (C1, \ldots, Ch)$ et, quelque soit i, Ci est dans $\mathscr{C}^*$.

• Il existe $C'0, \ldots, C'h$ dans $\mathscr{C}^*$, tous non vides sauf peut être $C'0$ et $C'h$, tels que

$$CH(R) = C'0\ C1\ C'1\ C2 \ldots Ci\ C'i \ldots C'h.$$

• Si $c=(q, x, q', y, d)$ est une occurrence de $\mathscr{C}$ dans $CH(R)$ avec x occurrence de Σ dans X, alors c occurre dans l'un des $C(i)$.



Dans l'exemple ci-contre on a :

$R=UXV$;

$CH(R)=C'0\ C1\ C'1\ C2\ C'2$ dans $\mathscr{C}^*$;

$CH(U)=(C'0)$ dans $(\mathscr{C}^*)$;

$CH(X)=(C1, C2)$ dans $(\mathscr{C}^*)^2$ $CH(V)=(C'1, C'2)$ dans $(\mathscr{C}^*)^2$.

En substance le chemin d'un mot et le chemin d'un facteur contiennent tous les renseignements relatifs aux calculs effectués par le transducteur boustrophédant sous ce mot ou ce facteur, et ceci en conservant l'ordre des calculs.

La notion qui suit le chemin parallèle recèle également toutes les informations mais respecte cette fois l'ordre des occurrences de lettre de Σ dans le mot d'entrée, R.

Soit $R(i)$ une occurrence de lettre de Σ dans R.

— On appelle *chemin parallèle sous* $R(i)$ le mot $CP(R(i)$ de $\mathscr{C}^*$ définit par :

• $CPR(i))=c(1)\ldots c(k)$ avec

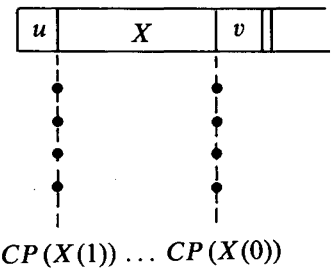
• Quelque soit j, $c(j)$ est dans $\mathscr{C}$ et

• $c(j)=(q, R(i), q', w, d)$ avec (q', w, d) dans $\delta(q, R(i))$. Ce qui signifie que, pointant sous la i-ième lettre de R, pour la j-ième fois, la machine était en état q, est passée en état q', a écrit w sur le ruban de sortie, et s'est déplacée de d.

— On appelle chemin parallèle sous le facteur X de R, le mot $CP(X)$ de $(\mathscr{C}^*)^*$ dont la i-ième lettre de $\mathscr{C}^*$ est le chemin parallèle $CP(X(i))$ sous la

i-ième lettre $X(i)$ de Σ dans X :

$$CP(X)=CP(X(1))\ldots CP(X(n)).$$



On a bien sur :

si $R=UXV$

alors $CP(R)=CP(U)\ CP(X)\ CP(V)$

Soit $\mathscr{S}$ le sous-ensemble de $\mathscr{C}^*$ minimal tel que, quelque soit R dans dom τ, alors $CP(R)$ soit dans $\mathscr{S}^*$: on appellera $\mathscr{S}$ *l'alphabet des chemins parallèles* du transducteur boustrophédon.

Il est aisé de constater que, si l'automate est déterministe, alors l'alphabet est fini : on ne peut passer deux fois dans le même état sous une même occurrence, sous peine de boucler indéfiniment. Réciproquement, un résultat de S. Greibach sur les machines de Turing permet de montrer que, pour qu'un langage produit par un transducteur boustrophédon soit dans DTWT, la famille déterministe, il suffit que l'alphabet $\mathscr{S}$ soit fini (*voir* [9] 1 et 4).

PROPOSITION 1 : *Soit τ un transducteur boustrophédon.*

Soit R un mot de domaine de τ tel que XY soit facteur de R. Soit $CL(R)$ un calcul de R; alors on a :

(i) $CP(XY)=CP(X)\ CP(Y)$;

(ii) $CH(XY)=(C1, C2, \ldots, Cr)$, *où les Ci sont des concatènes de $CP(X)j$ et des $CP(Y)k$.*

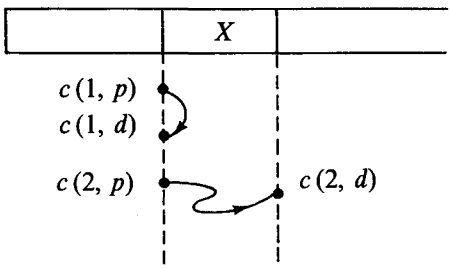
Les notions que nous allons introduire maintenant « oublient » une grande partie de l'information relative à l'intérieur des facteurs, retenant seulement les réactions de la machine à l'entrée et à la sortie de celui-ci, c'est-à-dire sa sensibilité au contexte.

3.2. Profil et image terminale associés à un facteur

Soit $CH(X)=(C1, \ldots, Cn)$ le chemin du facteur X. Soient, pour tout i, $c(i, p)$ et $c(i, d)$ la première et la dernière occurrence de lettre de $\mathscr{C}$ dans $C(i)$.

On appelle *profil de X, noté* $P(X)$, le mot de $(\mathscr{C}\mathscr{C})^*$ définit par les $c(i, p)$ et $c(i, d)$:

$$P(X)=(\ldots, c(i, p)\, c(i, d), \ldots)$$



Le profil de X donne tous les renseignements sur les calculs sous la première et la dernière lettre de X.

On dit que deux facteurs X *et* Y *ont le même profil si et seulement si* $P(X)=P(Y)$.

Comme l'énonce la proposition suivante, cette relation d'équivalence est fondamentale.

PROPOSITION 2 : *Soient* $R=U'XU''$ *et* $S=V'YV''$ *deux mots de* dom τ *tels que* X *et* Y *aient même profil. Alors il existe* $U0, \ldots, Ur$, $V0, \ldots, Vr$, $X1, \ldots, Xr$, $Y1, \ldots, Yr$, *dans* Δ^*, *tels que :*

(i) $\tau(R)=U0\ X1\ U1 \ldots Xr\ Ur$, $\tau(S)=V0\ Y1\ V1 \ldots Yr\ Vr$;

(ii) $U'YU''$ *et* $V'XV''$ *sont dans* dom τ;

(iii) $\tau(U'YU'')=U0\ Y1\ U1 \ldots Yr\ Ur$, $\tau(V'XV'')=V0\ X1\ U1 \ldots Xr\ Vr$.

La preuve de cette proposition — dite aussi *lemme de recollement* — est directe d'après les définitions. Elle exprime que X et Y sont substituables, dans le contexte U', U'' et V', V''.

Le mot $(X1, \ldots, Xr)$ de $(\Delta^*)^*$ sera appellé *l'image terminale* de X :

$$\mathrm{TIM}(X)=(X1, \ldots, Xr)$$

$\mathrm{Tim}(X)$ est clairement la projection du chemin de X sur sa quatrième composante; $\mathrm{Tim}(X)$ a « oublié » les informations relatives aux mouvements, pour ne retenir que celles liées aux lettres écrites sur le ruban de sortie.

L'équivalence de profil peut donc s'exprimer : si X et Y ont même profil, alors $\mathrm{Tim}(X)$ et $\mathrm{Tim}(Y)$ ont même longueur sur l'alphabet Δ^*, et dans toute image par le transducteur on peut remplacer les composantes de $\mathrm{Tim}(X)$ par celle de $\mathrm{Tim}(Y)$.

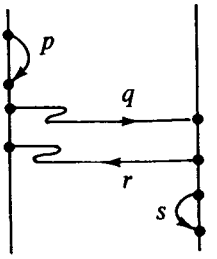
3.3. Canevas et fonctions associés à un facteur

Les deux notions équivalentes qui suivent sont assez proches du profil $P(X)$, mais autorisent la concaténation des facteurs. On utilise pour cela un nouvel alphabet, $G=\{p, q, r, s\}$, et l'on ne s'attache qu'aux mouvements de la tête de lecture.

— Soit $P(X)=(\ldots, c(i, p)\, c(i, d), \ldots)$ le profil de X, avec quelque soit i, $c(i, p)$ et $c(i, d)$ dans $\mathscr{C}$.

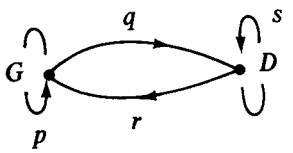
Soient $p(i)$ et $d(i)$ la cinquième composante de $c(i, p)$ et $c(i, d)$ respectivement. On pose :

si $p(i)=+1$ et $d(i)=-1$, $y(i)=p$,
si $p(i)=+1$ et $d(i)=+1$, $y(i)=q$,
si $p(i)=-1$ et $d(i)=-1$, $y(i)=r$,
si $p(i)=-1$ et $d(i)=+1$, $y(i)=s$.



On appelle *canevas* de X le mot $C(X)=y(1)\ldots y(h)$.

Remarquons qu'il aurait aussi été possible d'introduire le canevas $C(X)$ comme un chemin dans un graphe, où les sommets G et D symbolisent la partie gauche ou droite du facteur X :



— On associe alors à X une *fonction* f_x :

Soit n le nombre de q et de s qui occurent dans le canevas $C(X)$, et $(m-1)$ le nombre de p et de r dans $C(X)$.

(Intuitivement n compte le nombre de fois où la machine sort de X à droite, et $(m-1)$ comptabilise les sorties à gauche.)

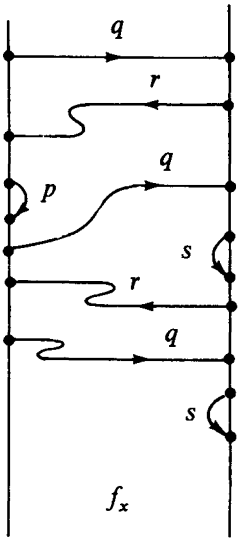
Soit alors la fonction f_x :

$$\{1, \ldots, n\} \to \{1, \ldots, m\},$$
$$i \mapsto j+1,$$

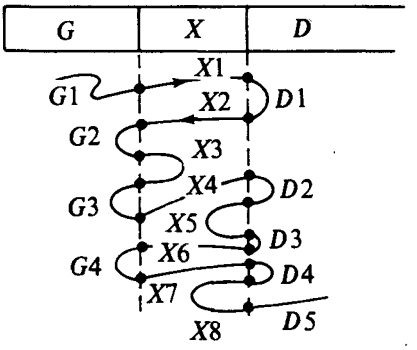
où : j est le nombre d'occurrences de p et de r avant la ième occurrence de q et de s dans $C(X)$.

Dans l'exemple ci-contre on a :
$C(X) = qrpqsrqs$
$n = 5$ $m = 3+1$
f_x: {1, ..., 5} → {1, ..., 4}

1	2	3	4	5
↓	↘	↙	↘	↙
1		3		4



Reprenons cet exemple avec tous les calculs faits sur l'entrée $R = GXD$, et les images terminales obtenues :



$\tau(GXD) = G1\ X1\ D1\ X2\ G2\ X3\ G3\ X4\ D2\ X5\ D3\ X6\ G4\ X7\ D4\ X8\ D5$, $D1$ est situé après $G1$, $D2$ et $D3$ après $G3$, $D4$ et $D5$ après $G4$, ceci dans $(\Delta^*)^*$.

— En général, on peut interpréter f_x comme suit :

Soit $R = GXD$ un mot du domaine du transducteur. Soient $\mathrm{Tim}(X) = (X1, \ldots, Xh)$ l'image terminale de X, $\mathrm{Tim}(D) = (D1, \ldots, Dm)$ et $\mathrm{Tim}(G) = (G1, \ldots, Gn)$ les images terminales de D et G. Alors $n+m = h+1$, et le mot produit est $\tau(GXD) = F0\ X1\ F1 \ldots Xh\ Fh$, avec quelque soit i, Fi égal à un Dj ou Gj; fx envoie $\{1, \ldots, n\}$ dans $\{1, \ldots, m\}$ et $fx(i) = j$ signifie que, dans l'image $\tau(GXD)$ Di occure à la droite de Gj et à la gauche de $Gj+1$:

$$\tau(R) = \ldots Gj \ldots Di \ldots Gj+1 \ldots$$

(le facteur entre Gj et $Gj+1$ peut contenir des Xh, éventuellement des Dk, mais pas d'autres Gk).

— la correspondance entre canevas $C(X)$ et fonction fx est une *bijection.*

— *Canevas et fonctions remarquables :*

• si R est dans dom τ, $C(R) = q$ et $f_x : \{1\} \to \{1\}$.

• si G est un facteur gauche de R, alors $C(G)$ est dans $q(s)^*$, et $f_x : \{1, \ldots, n\} \to \{1\}$.

• si D est un facteur droit de R, alors $C(D)$ est dans $(P)^*$ et $f_x : 1 \to \{1, \ldots, m\} : 1 \mapsto m$.

• si $C(X)$ est de la forme $qrqr \ldots qr$, alors fx est la fonction identité sur $\{1, \ldots, n\}$.

— Enfin l'application qui a X associe fx est un *morphisme* pour la concaténation des facteurs et la composition des fonctions.

C'est l'objet de la proposition suivante, et une clé pour la suite.

PROPOSITION 3 : *Si X et Y sont facteurs consécutifs de R, alors $fxy = fx \circ fy$.*

Preuve : Soit $R = GXYD$ un mot de dom τ. On pose, pour tout facteur F de R, $\mathrm{Tim}(X) = (F1, \ldots)$. (*Voir* définition de l'image terminale Tim, § 3-2.) Comme on s'en convaincra aisément à l'examen de l'exemple ci-dessous, on a : $\tau(R) = (GX)1\ (YD)1\ (GX)2 \ldots$; de plus GX étant un facteur gauche de R, le canevas de GX est de la forme $q(s)^*$, et chaque composante $(GX)i$ contient au moins un facteur égal à un Xj; de même YD est facteur droit de R, et chaque $(YD)i$ contient au moins un facteur égal à un Yj.

Ceci n'est, par contre, pas nécessairement le cas pour $(XY)i$.

Exemple :

$$(GXYD) = G1\ X1\ Y1\ X2\ Y2 \ldots$$

$$(GX)1 = G1\ X1;$$

$$(GX)2 = X2;$$

$$(GX)3 = X3\ G2\ X4;$$

$$(XY)1 = X1\ Y1\ X2\ Y2;$$

$$(XY)2 = Y3;$$

$$(XY)3 = Y4\ X3;$$

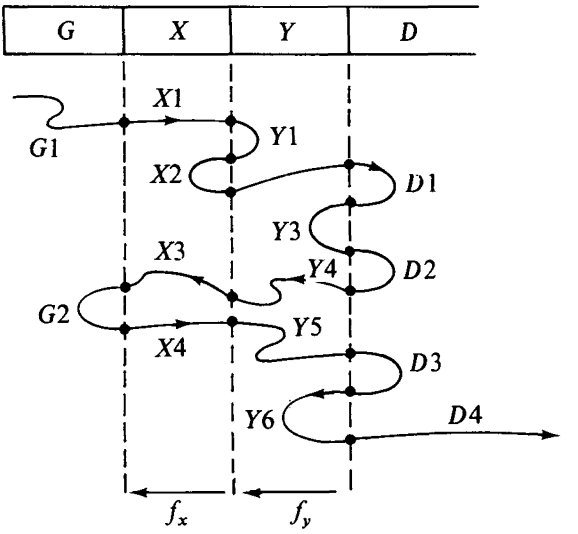
$$(XY)4 = X4\ Y5;$$

$$(XY)5 = Y6;$$

$$(YD)1 = Y1;$$

$$(YD)2 = Y2\ D1\ Y3\ D2\ Y4;$$

$$(YD)3 = Y5\ D3\ Y6\ D4.$$



Supposons alors que $f_y(i) = k$ et que $f_x(k) = j$; $(YD)_k$ occure dans $\tau(R)$ à la droite de Gj, et le facteur qui les sépare contient un $X1$; celui-ci est facteur de $(GX)k'$, et $(GX)k'$ est à la gauche de $(YD)k$; donc $k' < k$; Gj est donc ou un facteur de $(GX)k$, ou à sa gauche; comme $fy(i) = k$, Di est à la droite de $(GX)k$, et donc à la droite de Gj : $fxy(i) > = j$.

De la même façon $G(j+1)$ occure à la droite de $(YD)k$, et le facteur qui les sépare contient un $X1$, lequel est facteur de $(GX)k'$, avec $k'>k+1$; donc $D\,i$ occure à la gauche de $G(j+1)$ et $fxy(i)<j$.

On obtient donc, quelque soit i, $f_{xy}(i)=f_x f_y(i)$.

C.Q.F.D.

— La remarque suivante rattache profil et fonctions, et sera très utile.

PROPOSITION 4 : *Soient X et Y deux facteurs de mots de domaine de* τ. *Alors X et Y ont même profil si et seulement si* (1) *et* (2) *sont vrais :*

(1) *X et Y ont même fonction :* $f_x=f_y$,

(2) *X et Y ont mêmes chemins parallèles sous la première et la dernière occurrence de leurs lettres :*

$$CP(X(1))=CP(Y(1)),$$
$$CP(X(n))=CP(Y(m))$$

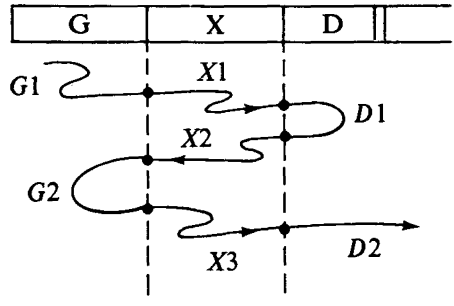
(*évident*).

4. LEMMES D'ITÉRATION

On veut, dans ce paragraphe, établir des lemmes d'itération pour les langages produits par des transducteurs boustrophédons.

Une idée simple — mais qui va se révéler simpliste — est de tirer partie de la rationnalité de l'ensemble donné en entrée, dom τ, en pompant $(i=0)$ ou en ajoutant $(i>1)$ un facteur du mot considéré, dans le domaine du transducteur. L'examen des deux exemples qui suivent dévoile l'insuffisance du procédé.

Exemple 1 : *Où cela se passe bien;*



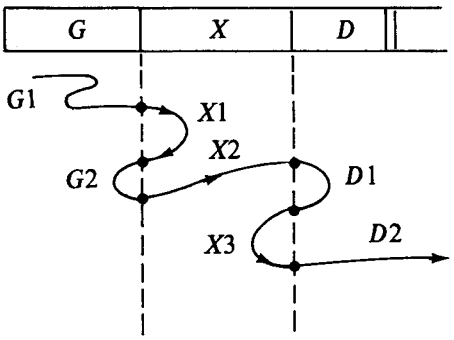
(On suppose les chemins parallèles sous la première lettre de X et D égaux)

$$\tau(GXD) = G1\ X1\ D1\ X2\ G2\ X3\ D2,$$

$$\tau(GD) = G1\ D1\ G2\ D2$$

$$K > 1, \quad \tau(GX^kD) = G1\,(X1)^k\ D1\,(X2)^k\ G2\,(X3)^k\ D3.$$

Exemple 2 : *Où cela se passe mal :*



(On suppose toujours $CP(X(1)) = CP(D(1))$.

$$\tau(GXD) = G1\ X1\ G2\ X2\ D1\ X3\ D2,$$

$$\tau(GD) = G1\ D1\ G2\ D2.$$

Un « shuffle » s'est produit entre les facteurs du contexte de X.

Dans le premier cas on avait $f_x = 1$, ce qui n'est pas dans le second : X ne peut être remplacé par le mot vide.

On s'aperçoit assez vite qu'il est vain de chercher des facteurs X tels que f_x soit l'identité; on est conduit donc à chercher des facteurs Z tels que $(1 - fz)$ soit un diviseur de zéro; on aura alors, pour un certain X, $f_x(1 - f_z) = 0$, et on cherchera une substitution faisant intervenir X et Z.

Ceci nous conduira au théorème suivant :

4.1. THÉORÈME 1 : Cas déterministe

Soit L un langage de DTWT, c'est-à-dire produit par un transducteur boustrophédon déterministe.

Alors il existe deux entiers p et N tels que :

Si W est un mot de L de longueur supérieure à N, alors il existe un entier q inférieur ou égal à p, des mots Ui et Vi de Δ^*, tels que

(i) quelque soit i, la longueur de Vi est inférieure à N et $U(0)$ ainsi que $q/2$ des $U(i)$ sont également de longueur inférieure à N;

(ii) un Vi au moins est non vide;

(iii) $W = U0\ V1\ U1\ V1\ U2 \ldots Vq\ Uq$;

(iv) quelque soit l'entier h, positif ou nul, le mot $W(h) = U0\ (V1)^h\ U1\ (V2)^h \ldots (Vq)^h\ Uq$, est dans le langage L.

Nous donnerons plus loin d'autres variantes de ce théorème, que nous allons nous attacher à montrer maintenant.

LEMME 1 : *Soient* τ *un transducteur boustrophédon déterministe et* X *et* Y *deux facteurs de mots de* dom τ, *tels que, les deux occurrences de* X *et* XYX *aient même profil. Soit* $\mathrm{Tim}(X) = (X1 \ldots Xq)$ *l'image terminale de* X, *et pour tout facteur* F, Fi *la i-ième composante de son image terminale.*

Alors, quelque soit i *entre* 1 *et* q, *quelque soit* k *positif ou nul* $(X(YX)^k)i = Xi\ Mi$, *où* Mi *est un facteur de* Δ^* *éventuellement vide).*

Exemple :

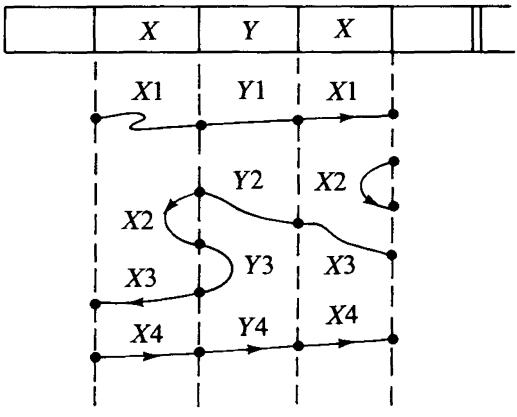


fig. 1

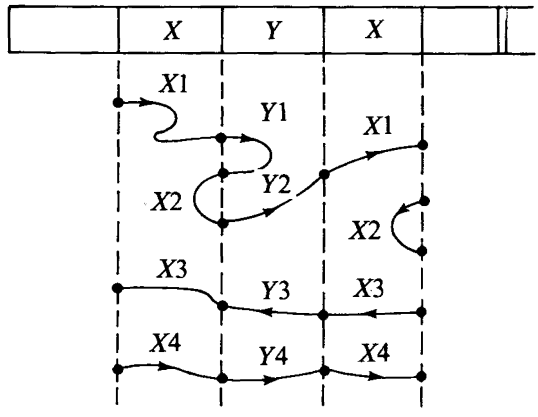


fig. 2

Les figures 1 et 2 illustrent le cas où XYX et les deux occurrences de X ont le même profil, ce qui n'est pas le cas dans la figure 3. La figure 4 met en évidence l'itération du facteur Mi, associé au cas de la figure 1.

Dans le cas de figure 1, on a :

$M1 = Y1\ X1$; $M2 = \varepsilon$; $M3 = Y2\ X2\ Y3\ X3$; $M4 = Y4\ X4$.

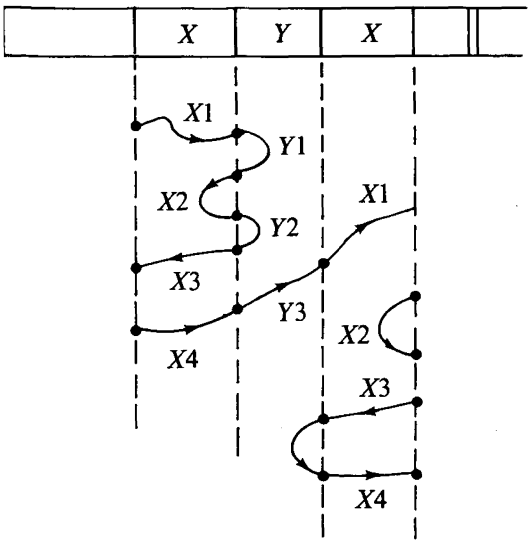


fig. 3

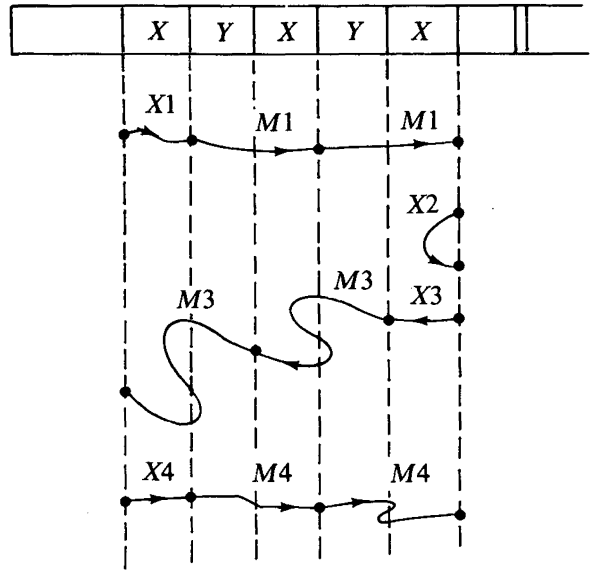


fig. 4

Pour la figure 2 on obtient :

$M1 = Y1\ X2\ Y2\ X1$; $M2 = \varepsilon$; $M3 = Y3\ X3$; $M4 = Y4\ X4$.

Preuve :

• Soit R un mot de dom τ, noté $R(1) \ldots R(n)$ dans Σ^*.

Soient $c(i)$ et $c(j)$ deux occurrences de $\mathscr{C}$ dans le chemin $CH(R)$ de R. On notera $CH(c(i) \to c(j))$ le chemin de $c(i)$ à $c(j)$, c'est-à-dire le facteur de $CH(R)$ compris entre $c(i)$ et $c(j)$. En ne conservant que la quatrième lettre de chaque $c(k)$, on obtient $P4(CH(c(i) \to c(j))) = \tau(c(i) \to c(j))$, dans Δ^*, qui sera dit image finale entre $c(i)$ et $c(j)$. On appellera domaine de $CH(c(i) \to c(j))$ le facteur $R(k) \ldots R(1)$ de R tel que

$$k = \text{Inf}\ \{ s \mid P2(c(r)) = X(s),\ i < r < j \}$$

et

$$1 = \text{Sup}\ \{ s \mid P2(c(r)) = X(s),\ i < r < j \}.$$

Si X est un facteur de R, égal à $R(n) \ldots R(m)$ dans Σ^*, on notera

$$CP(R(n)) = CP'(1) \ldots CP'(1) \ldots CP'(r)$$

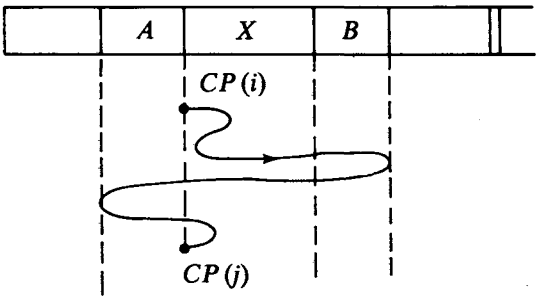
et

$$CP(R(m)) = CP''(1) \ldots CP''(s)$$

les chemins parallèles sous la première et la dernière lettre de R, avec $CP'(i)$ et $CP''(i)$ dans $\mathscr{C}$. On définira alors le *contexte de X entre CP(i) et CP(j)* de la façon suivante :

(i) $CP(i)$ et $CP(j)$ sont des occurrences de $\mathscr{C}$ dans $CH(X)$ et $CP(i)$ occurre à la gauche de $CP(j)$ dans $CH(R)$;

(ii) $\mathrm{CONT}(X \mid CP(i) \to CP(j))$ est égal au domaine du facteur $CP(i) \ldots CP(j)$ de $CH(R)$:



$\mathrm{CONT}(X \mid CP(i) \to CP(j)) = AXB =$ domaine de $CH(CP(i) \to CP(j))$.

Ceci signifie que le calcul, pour aller de l'étape $CP(i)$ à l'étape $CP(j)$, utilise des déplacements de la tête de lecture sous toutes les lettres de AXB, et seulement celles-ci.

On appellera $CONT\ (X)$, *contexte du facteur X*, l'union de tous les $CONT(X \mid CP(i) \to CP(j))$ pour $CP(i)$ et $CP(j)$ occurences de $\mathscr{C}$ dans la première et la dernière lettre du chemin parallèle de X.

Il découle directement des définitions que, si le chemin de X est $CH(X) = (C(1) \ldots, C(l))$, alors le contexte de X entre la première et la dernière lettre de chaque $C(i)$ est au plus égal à X, et que l'image finale de chaque $C(i)$ est la composante $X(i)$ de l'image terminale $\mathrm{Tim}(X)$.

• Supposons alors que $P(X) = P(XYX)$; alors quelque soit k, la première et la dernière lettre du chemin parallèle $CH(X(YX)^k)$ est égale à la première et la dernière lettre du chemin parallèle $CH(X)$; de plus $f_{xyx} = f_x$, donc, quelque soit k, $f_x(f_y f_x)^k = f_x$ et donc $f_{x(yx)^k} = f_x$: quelque soit k, $P(X(YX)^k) = P(X)$.

Ils sont tous profils équivalents;

il reste alors à examiner leurs images terminales.

• Soient $C(XYX)(i, p)$ et $C(XYX)(i, d)$ la première et la dernière occurence de $\mathscr{C}$ dans la i-ième composante du chemin de XYX, $CH(XYX)$.

Si le i-ième lettre du canevas de XYX est p, respectivement s, alors le contexte de XYX entre $C(XYX)$(i, p) et $C(XYX)(i, d)$ est la première occurence de X, respectivement la seconde, dans XYX. On a alors $C(XYX)(i) = X(i)$ et $M = \varepsilon$.

Si i-ième lettre du canevas de XYX est q, alors telle est la i-ième lettre du canevas de X, et le chemin entre $C(XYX)(i, p)$ et $C(XYX)(i, d)$ se décompose :

$$CH\,C(XYX)(i, p) \to C(XYX)(i, d)) = CH(C(X)(i, p) \to C(X)(i, d)).\, CH(s(C(X)(i, d)) \to C(XYX)(i, d))$$

[où $s(.)$ désigne l'occurrence qui succède à la lettre argument].

On obtient alors $C(XYX)(i) = C(X)(i).\ M$ où M est dans Δ^*. Comme le contexte de YX entre $s(C(X)(i, d))$ et $C(XYX)(i, d)$ est au plus égal à WYW, M ne contient que deux facteurs de la forme $X(j)$ ou $Y(j)$.

On trouve un résultat analogue si la i-ième lettre du canevas de XYX est r.

• Si k est strictement supérieur à 1, on trouve alors

$$C(X(YX)^{k-1}\,YX)(i) = C(X(YX)^{k-1})(i)\,M,$$

avec le même M (ou vide) que ci-dessus, puisque le contexte de YX est toujours au plus égal à XYX (dernière occurence).

• On a bien :

$$\forall i, \forall k, \quad (X(YX)^k)(i) = X(i).\,M^k.$$

C.Q.F.D.

Il nous reste donc à trouver, dans tout mot suffisament long, des facteurs consécutifs tels que X et XYX aient même profil. Nous allons déduire ceci de la finitude de l'ensemble des fonctions f_x, et de l'existence de quasi-puissance dans les mots.

On connaît en effet l'existence de mot de longueur arbitraire et ne contenant pas de carré, et donc *a fortiori* pas de puissance p-ième. Par contre ces mots renfermeront toujours une quasi-puissance, qui peut se révéler précieuse dans la démonstration de propriétés liées à l'itération (On trouve dans [22] une

première utilisation, semblable à celle-ci, des quasi-puissances, reprise dans [3] et [14].)

DÉFINITION : Soit A un alphabet fini donné, et $|A|$ son nombre d'éléments. On définit dans A^* les quasi-puissances d'ordre m récursivement comme suit :

(1) une quasi-puissance d'ordre 1 et un mot $\overline{W(1)}$ de A^*, de la forme $\overline{W(1)} = fw(1)f$, avec f dans A et $W1$ dans A^*;

(2) une quasi-puissance d'ordre m et un mot $\overline{w(m)}$ de A^*, de la forme $\overline{w(m)} = \overline{w(m-1)}\, w(m)\, \overline{w(m-1)}$, avec $W(m)$ dans A^*, et $W(m-1)$ quasi-puissance d'ordre $(m-1)$.

On a ainsi :

$$\overline{w(1)} = fw(1)f,$$

$$\overline{w(2)} = fw(1)fw(2)fw(1)f,$$

$$\overline{w(3)} = fw(1)fw(2)fw(1)fw(3)fw(1)fw(2)fw(1)f.$$

Toute quasi-puissance d'ordre m est de longueur supérieure ou égale à 2^m. On peut contrôler la longueur des quasi-puissances de diverses façons, dont voici un exemple.

LEMME 2 : *Soient A un alphabet fini et f la fonction définie récursivement par $f(1) = |A| + 1$ et $f(m+1) = f(m)[|A|^{f(m)} + 1]$.*

Soit g telle que $g(n) = m$, avec $f(m) \leqq n < f(m+1)$.

[On pose $f(0) = 1$.]

Soit W un mot de A^*, et soit $m = g(|w|)$, où $|W|$ est longueur de W.

Si la longueur de W est supérieure à $(|A| + 1)$, alors m est non nul, et il existe u, $W(m)$, v dans Δ^* tels que :

(i) $W = u\,\overline{W(m)}\,v$;

(ii) $\overline{W(m)}$ *est une quasi-puissance d'ordre m qui s'écrit :*

$$\overline{W(m)} = fw(1)f(w(2)fw(1)fw(3)\ldots$$

(iv) $\forall i$, $w(i)$ *est de longueur au plus égale à $f(i)$.*

(v) *la longueur de uv est bornée par $f(m+1) - f(m)$.*

Preuve : Si $m = 1$ alors la longueur de W est supérieure à $|A| + 1$; deux occurrences au moins de A dans W sont égales; on peut donc écrire $w = ufw'fv$, avec w' de longueur au plus égale à $|A|$, et W contient la quasi-puissance $w(1) = fw'f$, de longueur au plus égale à $f(1)$.

Si la propriété énoncée est vraie jusqu'à m, soit W tel que $g(|W|) = m + 1$, et soit $A(m)$ le nouvel alphabet formé des mots de A^* de longueur $f(m)$.

Considéré comme mot de $(A(m))^*$, w contient au moins $|A(m)|+1$ lettres, donc deux occurences d'entre elles au moins sont égales. En en choisissant deux distantes au plus de $|A(m)|$, on obtient $W=uqkqv$, avec q dans $A(m)$ et k dans $A(m)^*$, de longueur bornée par $|A(m)|$, d'où le résultat.

C.Q.F.D.

LEMME 3 : *Soit τ un transducteur boustrophédon déterministe.*

Soit $F(\tau)$ l'ensemble des fonctions f_x, pour X facteur de mots de dom τ.

Alors $F(\tau)$ est fini. On note k son cardinal.

Ceci est une conséquence directe, de la finitude de l'alphabet $\mathscr{S}$ des chemins parallèles (*cf.* paragraphe 3.1).

k est au plus égal au nombre de fonctions d'un ensemble à q' éléments dans un ensemble à q'' éléments, avec q' et q'' borné par le nombre d'états de l'automate.

LEMME 4 : *Soit τ un transducteur boustrophédon déterministe, et R un mot dans dom τ, de longueur au moins égale $f(k+1)$, où k est la constante définie lemme 3, et f est définie au lemme 2. Alors R s'écrit $R=UXYXV$ et $R=U'X'Y'X'V'$, avec :*

(i) *X et XYX ont même profil, la longueur de (YX) est borné par $f(k+1)$, ainsi que celle de U.*

(ii) *X' et $X'Y'X'$ ont même profil, et la longueur de $(Y'X')$ est supérieure à 2^{m-k} [où m est égal à $g(|R|)$].*

Preuve :

— On décompose w sur $(A_k)^*$, où A_k est l'alphabet défini au lemme 2, par $A_1=A$ et $A_{m+1}=\{\text{mot de } (A_m)^* \text{ de longueur } f(m)\}$. Alors le facteur droit de W, de longueur égale à $f(k+1)$ dans A^*, contient au moins une quasi-puissance d'ordre $k+1$ (lemme 2), et donc s'écrit :

$$ufw(1)fw(2)fw(1)fw(3)f\ldots v.$$

Posons

$$X1=fw(1)f,$$
$$X(i+1)=X(i)\,W(i+1)\,X(i).$$

Comme $F(\tau)=\{f_x \mid X \text{ facteur dans dom } \tau\}$ est de cardinal k, deux au moins des $(k+1)f_{xi}$ sont égales :

Il existe i et j : $f_{xi}=f_{xj}$; on peut alors écrire $Xj=XiYXi;\ YXi,\ Xi$ et u sont de longueur bornée par $f(k+1)$, et par construction Xi et $XiYXi$ ont même profil.

— En décomposant W sur $(A_m)^*$, on trouvera de même $R = u\,W(m)\,v$, où $W(m)$ est une quasi-puissance d'ordre m. En posant toujours $X1 = f\,W(1)\,f$ et $Xi+1 = Xi\,W(i+1)\,Xi$, on pourra choisir (puisque $m = g(|R|) \geqq f(k+1)$) i et j tels que $j-i > m-k$ et $f_{xi} = f_{xj}$; en écrivant $Xj = Xi\,Y\,Xi$, on trouvera alors Y contenant une quasi-puissance d'ordre au moins $(m-k)$, donc de longueur au moins égale à 2^{m-k}.

C.Q.F.D.

Démonstration du théorème 1 : Soit L un langage de DTWT, et soit τ un transducteur boustrophédon déterministe qui le produise. Soit p le nombre d'états de τ, et k le cardinal de $F(\tau)$ (*voir* lemme 3). Posons $N = 2pf(k+1)$.

L'automate étant sous forme normale (cf. § 2), toute lettre d'entrée produit au moins une lettre de sortie, et au plus p : quelque soit R, dans dom τ, on a

$$|R| \leqq |\tau(R)| \leqq p|R|.$$

Soit alors W un mot de L, de longueur au moins égale à N; W est l'image d'un mot R de dom τ, de longueur au moins égale à N/p, soit $2f(k+1)$. On peut donc appliquer à R le lemme 4(i).

On trouve $R = GXYXD$, avec les longueurs de G, de X, de XY, bornées par $f(k+1)$, et X et XY ont même profil. En appliquant alors le lemme 1, et la proposition 2 du paragraphe 3.2., on trouve le résultat cherché : les V_i sont les composantes de $\mathrm{Tim}(YX)$, de longueur bornée par $p|YX|$, donc par N; U_0 et $[q/2]$ parmi les U_i sont à prendre dans les composantes de $\mathrm{Tim}(GX)$, bornées par $p|GX|$, donc par N; q, qui est la longueur de $\mathrm{Tim}(YX)$, sur l'alphabet Δ^*, est bien sur borné par le nombre d'états p de l'automate.

COROLLAIRE. — *On peut, dans le théorème* 4.1., *remplacer* (i) *par* (i') : *il existe une fonction f croissante non bornée telle que, pour un i au moins, la longueur de* V_i *est supérieure à* $f(|W|)$.

Il suffit de refaire les mêmes constructions que précédemment, mais en appliquant la partie (ii) du lemme 4 du théorème 4.1.

4.2. Cas non-déterministe

THÉORÈME 2 : *Soit L un langage de NDTWT, c'est-à-dire produit par un transducteur boustrophédon non déterministe. Il existe des entiers N et p tels que, quelque soit le mot de W de longueur au moins égale à N, l'une de deux*

propriétés (A) ou (B) est vraie :

(A) W se factorise en $W = U' VU''$, avec V non vide, et quelque soit h positif ou nul $U'(V)^h U''$ est dans L.

(B) W se factorise en $W = U_0 V_1 U_1 V_2 U_2 \ldots V_q U_q$, avec q au plus égal à p, et les propriétés suivantes sont vraies :

(i) *Un V_i au moins est non vide; ils sont tous de longueur bornée par N;*

(ii) *quelque soit h positif ou nul,*

$$U_0 (V_1)^h . U_1 (V_2)^h . U_2 \ldots (V_q)^h . U_q \text{ est dans } L.$$

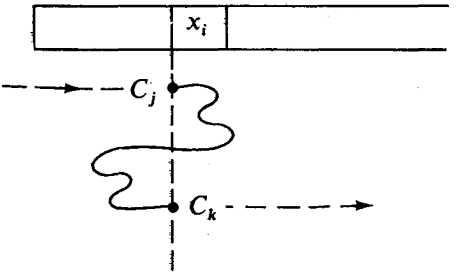
Preuve : On choisit un automate τ non déterministe, sous forme normale, à p états, et on pose $N = f(p^{p+2})$.

Soit W un mot de L, et R dans dom τ tel que W soit dans $\tau(R)$;

Examinons le calcul et le chemin parallèle de R, $CP(R)$.

Soit CP_i la i-ième lettre de $CP(R)$: la tête de lecture examine la i-ième lettre de R. CP_i est un élément de $\mathscr{C}^*$.

(a) S'il existe un indice i, et une lettre c de $\mathscr{C}$ apparaissant deux fois dans CP_i, soit par exemple $c = (q, x, q', y, d)$ qui occure en C_j et C_k, alors le chemin entre C_j et C_k, $CH(c_j \to c_k)$, peut être soit enlevé, soit ajouté à $CH(R)$ sans quitter l'ensemble des chemins autorisés.



$$W = U' VU'',$$
$$U' = P_4(CH(c_0 \to pc_j)),$$
$$V = P_4(CH(c_j \to c_k)),$$
$$U'' = P_4(CH(sc_k \to c_f)),$$

(pc_j et sc_j sont les prédécesseurs de c_j et le successeur de c_j).

Quelque soit h, $U'(V)^h U''$ est dans L : c'est *(A)*.

Si V est vide, alors on enlève le chemin $CH(c_j \to c_k)$ de $CH(R)$, et on répète le procédé jusqu'à arriver à un V non vide ou à *(b)*.

(*b*) Si quelque soit i, CP_i ne contient jamais deux fois la même lettre, alors le cardinal de l'alphabet $\mathscr{S}_1$ des chemins parallèles utilisés, pour X facteur de R, est borné. La construction du théorème 1 appliqué au profil des facteurs, reste valable : c'est (*B*).

C.Q.F.D.

COROLLAIRE : *On peut, dans le théorème* 2, *remplacer* (*B*)(i) *par* (*B*)(i') : *il existe une fonction croissante non bornée f tel que, pour au moins un i, la longueur de* V_i *est plus grande que* $f(n)$, *où n est la longueur du mot considéré* (*conséquence directe du théorème* 2 *et du corollaire du théorème* 1).

On sait que la famille déterministe DTWT est strictement incluse dans la famille non déterministe NDTWT. Un résultat de J. Engelfriet affirme que tout langage de NDTWT-DTWT contient un langage infini régulier (*U*). Nous donnons maintenant une caractérisation des langages strictement non déterministe qui englobe ce résultat, et qui sera utilisé pour démontrer la non-appartenance du Dyck sur deux lettres à la famille NDTWT.

THÉORÈME 3 : *Soit L un langage de NDTWT.*

Alors L est dans NDTWT-DTWT si et seulement si, quelque soit l'entier n, il existe un mot W de L et une occurence x de lettre dans W telle que :

$$\text{Card}\{\, Y \mid W = X x YZ \text{ et } X x (Y)^* Z \subseteq L \,\} \geqslant n.$$

Preuve : Soit L dans NDTWT, et τ un transducteur sous forme normale qui produit L;

L est dans DTWT si et seulement si l'alphabet $\mathscr{S}$ des chemins parallèles (*cf.* § 3.1.) est fini. Si L n'est pas dans DTWT, les longueurs des chemins parallèles, sous les lettres de R dans dom τ, forment un ensemble non fini; on pourra toujours trouver un mot R de dom τ, une lettre A dans R, telle que le chemin parallèle $CP(AR)$ contienne $(n+1)$ occurrences d'une lettre $c=(q, x, q', y, d)$ de $\mathscr{C}$, occurant en $c(0), c(1), \ldots, c(n)$; alors les facteurs $P4(CH(c(0) \to c(i)))$ peuvent être enlevés ou ajoutés à l'image W de R.

C.Q.F.D.

On dit que L' est un *ensemble test* pour L si L' est inclu dans L, L est dans NDTWT, et la suite des mots w de L du théorème 3 peut être choisie dans L'.

COROLLAIRE : *Soit* $C(n)$ *la suite de* $D'_1 1$ *définie par*

$$C(1) = aa'; \qquad C(n+1) = a\,C(n)\,C(n)\,a';$$

alors D'^*_1 *est dans NDTWT-DTWT, et la suite* $C(n)$ *est un ensemble test pour* D'^*_1 (*voir* [20]).

5. LE LANGAGE $D_2'^*$ N'EST PAS DANS NDTWT

Pour établir ce résultat, nous allons utiliser certaines suites *ad hoc* de mots dans $D_2'^*$, qui apporteront une contradiction au théorème 3, en utilisant un résultat établit dans [20].

Dans ce qui suit, $D_2'^*$ sera le langage de Dyck d'ordre 2, c'est-à-dire la classe d'équivalence du mot vide dans la congruence définie dans $(a+a'+b+b')^*$ par $aa'=bb'=\varepsilon$ (*voir* [2] ou [12]).

Soit $AL=(a+b+a'+b')$.

Nous utiliserons le morphisme de réduction red, qui, à tout mot de AL^* associe sa forme réduite. Il vérifie :

(i) $\forall x \forall y \ \mathrm{red}(xy)=\mathrm{red}(\mathrm{red}(x)y)=\mathrm{red}(\mathrm{red}(x)\mathrm{red}(y))=\mathrm{red}(x\,\mathrm{red}(y))$.

(ii) $\forall W \ W\in D_2'^*$ ssi $\mathrm{red}(W)=\varepsilon$.

(iii) Soit

$$\mathrm{Init}(D_2'^*)=\{\,\mathrm{W}\mid \exists \mathrm{w}' : WW'\in D_2'^*\,\},$$
$$\mathrm{Fin}\,(D_2'^*)=\{\,W\mid \exists w' : W'W\in D_2'^*\,\}$$

Alors

$$W\in \mathrm{Init}(D_2'^*) \text{ ssi } \mathrm{red}(W)\in(a+b)^*,$$
$$W\in \mathrm{Fin}(D_2'^*) \textit{ ssi red}(W)\in(a'+b')^*.$$

5.1. Les suites $(A(n))$ et $(B(n))$ dans $D_2'^*$

On définit les deux suites $(A)_n$ et $(B)_n$ comme suit :

$$A(1)=aa', \quad \forall n\geqq 1, \qquad A(n+1)=a\,A(n)\,B(n)\,a',$$
$$B(1)=bb', \quad \forall n>1, \qquad B(n+1)=bB(n)\,A(n)\,b',$$

$A(1)=aa'$,

$A(2)=aaa'\,bb'\,a'$,

$A(3)=aaaa'\,bb'\,a'\,bbb'\,aa'\,b'\,a'$,

. . .

PROPOSITION 5 : *Quelque soit l'entier n et l'occurence x de AL dans* $A(n)$, *on a :* $\mathrm{Card}\{\,Y\mid A(n)=X\times YZ$ *et* $X\times(Y)^*Z$ *dans* $D_2'^*\,\}$ *est inférieur ou égal à deux.*

Preuve : On va prouver, de façon très combinatoire, que si l'on veut effacer un facteur dont la première lettre est donnée [dans un certain $A(n)$], alors on ne peut le faire au plus que de deux façons.

— *Exemple :*

$$A(3) = aaaa'bb'a'bb\ b'\ a\ a'\ \ b'\ a' \text{ deux possibilités,}$$

$$A(3) = a\ a\ a\ a'\ b\ b'\ a'\ b\ b\ b'\ a\ a'\ b'a' \text{ une possibilité}$$

$$A(3) = a\ a\ a\ a'\ b\ b'\ a'\ b\ b\ b'\ a\ a'\ b'\ a' \text{ aucune possibilité.}$$

— On écrira, quelque soit n :

$$A(n) = a(1)a(2)\ldots a(n)B(1)a'(n-1)B(2)\ldots a'(2)B(n-1)a'(1),$$
$$B(n) = b(1)b(2)\ldots b(n)A(1)b'(n-1)A(2)\ldots b'(2)A(n-1)b'(1),$$

où, quelque soit i, $a(i)$, $b(i)$, $a'(i)$, $b'(i)$ sont des occurrences de a, b, a', b', respectivement.

— Il est clair que si $A(n)$ se factorise en $A(n) = ww'$, avec w et w' non vide, alors red(W) est dans $a(a+b)^*$ et red(W') est dans $(a'+b')^*a'$. Un résultat analogue vaut pour B(n).

— Supposons maintenant la proposition vraie jusqu'à $(n-1)$, pour à la fois $A(i)$ et $B(i)$; (elle est trivialement vraie pour $n=1$).

Soit une factorisation de A (n) : $A(n) = X \times YZ$ avec $X.Z$ dans $D_2'^*$.

Cas 1 $x = a(i)$

Puisque red$(X) = (a)^{i-1}$ et red$(XZ) = \varepsilon$, on obtient red$(Z) = (a')^{i-1}$; si $i=1$, alors $Z = \varepsilon$ et $xy = A(n)$: une seule possibilité;

si $i \neq 1$, alors Z est non vide; soit x' sa première lettre.

si $x' = a$ alors

$$\text{red}(Z) = \text{red}(a(j)\ldots a(n)a'(n)B(1)\ldots a'(1)) = (a')^{j-1},$$

donc $i-1 = j-1$ et x est dans Z : impossible.

Si $x' = a'(j)$ alors

$$\text{red}(Z) = \text{red}(a'(j))B(n-j-1)a'(j-1)\ldots a'(1)) = (a')^j$$

donc $j=i-1$ et $Z=a'(j+1)\,B(n-j+2)\ldots a'(1)$ et

$$Y=a(i+1)\ldots a(n)\,a'(n)\,B(1)\ldots a'(i)\,B(n-i+1)$$

c'est ma première possibilité.

si x' est intérieur à un $B(j)$, alors

$$\mathrm{red}(Z)=\mathrm{red}(B'(j)\,a'(n-j)\ldots a'(1))=\mathrm{red}(B'(j))\,(a')^{n-j},$$

où $B'(j)$ est un facteur droit de $B(j)$. Si $B'(j)$ est un facteur droit strict de $B(j)$, alors $\mathrm{red}(B'(j))$ est dans $(a'+b')^*\,b'$ et $\mathrm{red}(Z)$ ne peut être dans $(a')^*$; donc $B'(j)$ est égal à $B(j)$, $n-j=i-1$, $Z=B(n-i-1)\,a'(i-1)\ldots a'(i-1)\ldots a'(1)$, et

$$Y=a(i+1)\ldots a(n)\,a'(n)\,B(1)\ldots a'(i-1)\,B(n-i)\,a'(i).$$

c'est la seconde possibilité.

Cas 2 $x=a'(i)$

Comme $\mathrm{red}(X)=(a)^i$ et $\mathrm{red}(XZ)=\varepsilon$, on obtient $\mathrm{red}(Z)=(a')^i$; soit x' la première occurence de Z; $x'=a(j)$ est impossible; $x'=a'(j)$ implique $j<i$, et $\mathrm{red}(Z)=(a')^j$: c'est impossible; x' intérieur $a\,B(j)$ implique

$$Z=B(j)\ldots a'(1) \qquad \text{et} \qquad \mathrm{red}(Z)=n-j,$$

ce qui est contradictoire avec $n-j<i$.

aucune possibilité.

Cas 3 *x intérieur a un* $B(j)$

$A(n)=X\,x\,YZ$ et XZ est dans $D_2'^*$, donc X admet un facteur droit qui est facteur gauche de $B(j)$, soit

$$X=a(1)\ldots a'(n)\,a(n)\,B(1)\ldots a'(n-j+1)\,B'(j);$$

alors $\mathrm{red}(X)=\mathrm{red}((a)^{n-j}\,red\,(B(j)))$.

– Si x est la première lettre de $B(j)$, alors $\mathrm{red}(X)=(a)^{n-j}$ et $\mathrm{red}(Z)=(a)^{n-j}$: Z est égal à $a(n-j)\ldots B(n-1)$, et $Y=B(j)$.

– Si x n'est pas la première lettre de $B(j)$, alors $\mathrm{red}(X)$ est dans $a(n-j)\,b(a+b)^*$; comme XZ est dans $D_2'^*$, $\mathrm{red}(Z)$ est dans $(a'+b')^*\,b'\,(a')^{n-j}$.

$X=a(1)\ldots a(n)\,a'(n)\,B(1)\,a'(n-1)\ldots a'(n-j+1)\,B'(j)$, avec $B'(j)$ facteur initial de $B(j)$. On trouve donc $B''(j)$ facteur final de $B(j)$ tel que

$$Z=B''(j)\,a'(n-j)\ldots a'(1).$$

Alors $B(j)=B'(j)$, avec $B'(j)B''(j)$ dans $D_2'^*$: par hyp d'induction, ceci est possible au plus deux fois.

C.Q.F.D.

5.2. THÉORÈME 4 : *Le langage de Dyck $D_2'^*$ n'est pas dans NDTWT.*

Preuve : Supposons que $D_2'^*$ soit dans NDTWT. Soit h le morphisme qui envoie $(a+b+a'+b')^*$ dans $(a+a')^*$, associant a à a et b, et a' à a' et b'. L'image de $D_2'^*$ est alors $D_1'^*$ et celle des $A(n)$ une suite $C(n)$. On sait que NDTWT est stable par morphisme ([6] ou [9]) : $D_2'^*$ est donc dans NDTWT, et la proposition 5 est vraie en remplaçant $A(n)$ par $C(n)$. Ceci contredit le corollaire du théorème 3.

C.Q.F.D.

6. CONCLUSION

En associant aux facteurs des mots une fonction, dépendante du contexte dans une certaine mesure, on prouve des théorèmes d'itération et la non-appartenance de $D_2'^*$ à la famille déterministe. Dans le cas déterministe, les lemmes obtenus ont une forme un peu plud précise que les versions connues, (contrôles divers sur la longueur des facteurs et leurs positions), et surtout la démonstration avancée ici est sans faille. Les formes non déterministes du lemme sont nouvelles; la non appartenance de $D_2'^*$ à NDTWT était déjà une conséquence de [9] et [7], mais la démonstration proposée ici est très différente, et beaucoup plus directe.

Je voudrais remercier ici J. Beauquier, D. Ferment, S. Grigorieff et C. Reutenauer, pour les discussions. les conseils, les critiques et les encouragements qu'ils n'ont cessés de me prodiguer, sur des versions beacoup plus hermétiques de ce papier. J. P. Pecuchet a assuré une relecture scrupuleuse de cet article, et détecté deux erreurs dans des lemmes. Qu'il soit ici loué pour sa diligence et son efficacité.

BIBLIOGRAPHIE

1. J. BEAUQUIER, *Deux familles de langages incomparables*, Inform and Control, vol. 43-2, 1979, p. 101-121.
2. J. BERSTEL, *Transductions and context-free languages*, Teubner Studien Bucher-Stuttgart, 1979.
3. J. BERSTEL et C. REUTENAUER, *Les séries rationnelles et leurs langages*, Masson, 1984.

4. B. BRAINERD, *An Analog of a Theorem about Context-Free Languages*, Inform and control, Vol. 11, 1968, p. 561-567.

5. A. EHRENFEUCHT et G. ROZENBERG, *On Some Context-Free Languages that Are Not Deterministic EDTOL Languages*, R.A.I.R.O. — Info. Théorique, vol. 11, n° 4, 1977, p. 273-191.

6. R. EHRICH et S. YAU, *Two-Way Sequential Transductions and Stacks Automata*, Inform and control, vol. 18, 1971, p. 404-446.

7. J. ENGELFRIET, *Two-Way Automata and Checking Automata*, Math-Centrum-Amsterdam, 1981.

8. D. FERMENT, *Principality Results About Some Matrix Languages Families*, Lecture note in computer Science, 172, I.C.A.L.P., juillet 1984.

9. S. GREIBACH, (1) *One Way Finit Visit Automata*, T.C.S., 6, 1978, p. 175-221; (2) *Remarks on Blind and Partially Blind One-Way Multicounter Machines*, Th. computer science, vol. 7, 1978, p. 311-324; (3) *Syntactic Operators on Full SemiAFLS*, J.C.S.S., vol. 6, 1972, p. 30-76; (4) *Checking Automata and One-Way Stack-Languages*, J.C.S.S., vol. 3, 1963, p. 196-217.

10. S. GREIBACH et J. HOPCROFT, *Scattered Context Grammars*, J.C.S.S., vol. 3, 1969, p. 233-247.

11. O. IBARRA, *On Two-Way Sequential Transductions of Full Semi-AFL'S*, T.C.S., vol. 7, 1978, p. 287-309.

12. M. HARRISON, *Introduction to Formal Language Theory*, Addison wesley, 1978.

13. M. LATTEUX, (1) *EDTOL, systèmes ultralinéaires et opérateurs associés*, T.R. 100, 1977, Université de Lille; (2) *Substitutions dans les EDTOL*, Inform and control, vol. 42, n° 2, 1979; (3) *Sur les générateurs algébriques ultralinéaires*, Acta Informatica, vol. 13, 1980, p. 347-363; (4) *A propos du lemme de substitution*, TCS, vol. 14, 1981, p. 119-123; (5) *Langages à un compteur*, J.C.S.S., vol. 26, n° 1, février 1983.

14. M. LOTHAIRE, *Combinatorics on Words*, Addison-Wesley Publ., 1983.

15. G. PAUN, (1) *On the Index of Grammars and Languages*, Inform and control, vol. 35, 1977, p. 259-266; (2) *Some Consequence of a Result of Ehrenfeucht and Rozenberg*, R.A.I.R.O., vol. 14, n° 1, 1980, p. 119-122.

16. J. PECUCHET, (1) *Automates boustrophédons, semi-groupes de Birget et monoïde inversif libre*, R.A.I.R.O.-Informatique théorique, vol. 19, n° 1, 1985, p. 71-100; (2) *Automates boustrophédons et mots infinis*, Th. Comput. Sc., vol. 35, 1982, p. 115-122.

17. V. RAJLICH, *Absolutely parallel grammars and two-way finite state transducers*, J.C.S.S., vol. 6, 1972, p. 324-342.

18. G. ROZENBERG et A. SALOMAA, *The Mathematical Theory of L-Systems*, Academic Press, 1980.

19. G. ROZENBERG et D. VERMEIR, (1) *On ETOL System of Finite Index*, Inform and control, vol. 38, 1978, p. 103-133; (2) *On the Effect of the Finite Index Restriction on Several Families of Grammars*, Inform and control, vol. 39, 1978, p. 284-302.

20. B. ROZOY, (1) *About two-Way Transducers*, FCT 85 et T.R. LITP 85-43- Université Paris-VII; (2) *The Dyck Language D'^*_1 Is Not Generated by Any Matrix Grammar of Funite Index*, T.R. L.I.T.P. 85, Université Paris-VII. To appear in Inform and Control.

21. A. SALOMAA, (1) *On the Index of Context Free Grammars and Languages*, Inform and control, vol. 14, 1969, p. 474-477; (2) *Formal Languages*, Academic Press, 1973.
22. M. P. SCHUTZENBERGER, *On a Special Case of Regular Events*, Annals of Math. Stat., vol. 32, 1961, p. 1201-1213.
23. E. WELZ et K. CULIK, *Two-Way Finite State Generators* (communicated by E. WELZ).