RAIRO. INFORMATIQUE THÉORIQUE

J.-M. AUTEBERT
L. BOASSON
G. SÉNIZERGUES

Langages de parenthèses, langages N.T.S. et homomorphismes inverses

RAIRO. Informatique théorique, tome 18, nº 4 (1984), p. 327-344
<http://www.numdam.org/item?id=ITA_1984__18_4_327_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 18, n° 4, 1984, p. 327 à 344)

LANGAGES DE PARENTHÈSES, LANGAGES N.T.S. ET HOMOMORPHISMES INVERSES (*)

par J.-M. AUTEBERT [1], L. BOASSON [1], G. SÉNIZERGUES [1]

Communiqué par J.-E. PIN

Résumé. — *La famille des langages de parenthèses a été introduite par M. Takahashi comme généralisation de la famille des langages parenthétiques. Elle montre que les propriétés de celle-ci se généralisent à cette nouvelle famille. Par ailleurs a été introduite la famille des langages N.T.S. qui forment une grande sous-classe des langages déterministes et congruenciels. Parmi les questions ouvertes concernant les langages N.T.S., nous nous intéressons ici à la suivante : la famille des langages N.T.S. est-elle fermée par homomorphisme inverse ? Nous donnons une réponse partielle à cette question, en utilisant un résultat précédent : l'inclusion de la famille des langages de parenthèses dans la famille des langages N.T.S. Plus précisément, nous montrons que l'image d'un langage de parenthèses par un homomorphisme inverse est un langage N.T.S. Ce résultat est complété par une propriété de ces images qui permet de prouver que le cylindre engendré par la famille des langages de parenthèses est strictement inclus dans la famille des langages N.T.S.*

Abstract. — *Takahashi introduced the family of Nest-Sets as a generalization of the famous parenthesis languages. She showed that this family does satisfy the main properties of parenthesis languages. On the other hand, a large family of context-free congruencial languages has been introduced. It is called the family of N.T.S. languages. Among various open questions about N.T.S. languages, we are concerned here by the following: is the family of N.T.S. languages closed under inverse homomorphism?*

We give a partial answer, using a previous result stating that the family of Nest-Sets is included in the family of N.T.S. languages. Namely, we prove that any inverse homomorphic image of a Nest-Set in N.T.S. This result is completed by a property of such inverse homomorphic images which allows to show: the cylinder generated by the Nest-Sets is strictly included in the family of N.T.S. languages.

La famille des langages de parenthèses a été introduite par Takahashi [6, 7] comme une généralisation de la famille des langages parenthétiques [4]. Cette généralisation est remarquable en particulier parce qu'elle permet d'étendre les résultats connus concernant les parenthétiques à une famille

(*) Reçu en mars 1983, révisé en octobre 1983.
([1]) LITP, UER de Maths, Université Paris 7, 2, place Jussieu 75251 Paris Cedex 05, France.

0399-0540/84/04 327/18/$ 3,80/© AFCET-Bordas

notablement plus grande qui, en outre, contient tous les langages rationnels et tous les langages de Dyck. On notera que la clôture par intersection et la décidabilité de l'équivalence peuvent être étendues à cette nouvelle famille.

Récemment, il a été montré que la famille des langages de parenthèses était une sous-famille de celle des langages à non-terminaux séparés (N.T.S.) introduite dans [2]. La famille des langages N.T.S. jouit elle aussi de propriétés remarquables : elle est formée de langages congruenciels déterministes dont l'équivalence est décidable [5]. On sait que cette famille est fermée par intersection rationnelle, mais le problème est toujours ouvert de savoir si elle est fermée par homomorphisme inverse.

L'objet de cet article est de montrer que si l'on considère l'image inverse dans un morphisme d'un langage de parenthèses, on reste bien dans la famille des langages N.T.S. Ce résultat est complété d'une remarque qui montre que le problème général concernant les langages N.T.S. n'est pas pour autant résolu : il existe des langages N.T.S. qui ne sont pas image de langages de parenthèses dans un morphisme inverse.

Le présent article est divisé en 4 parties. La première est consacrée à des rappels concernant les langages de parenthèses et les grammaires N.T.S. La seconde est consacrée à la preuve d'un lemme sur les langages de Dyck qui sera la clef du résultat principal qui est établi dans la troisième partie. La dernière partie donne une propriété remarquable des langages concernés par notre résultat : tous leurs quotients sont N.T.S. Ceci permet de montrer que le cylindre engendré par les langages de parenthèses est strictement contenu dans la famille des langages N.T.S.

I. RAPPELS

Une grammaire algébrique G est donnée comme d'habitude par le quadruplet $\langle X, V, P, \sigma \rangle$ où X est l'alphabet terminal, V l'alphabet (disjoint de X) non-terminal, P l'ensemble des règles et σ l'axiome. Étant donné W inclus dans V, on note $L_G(W)$ et $\hat{L}_G(W)$ les langages sur X et sur $(X \cup V)$ engendrés par W.

DÉFINITION 1 [2] : Une grammaire algébrique $G=\langle X, V, P, \sigma \rangle$ est *N.T.S.* si et seulement si on a la propriété suivante :

$$(C) \quad \forall v, v' \in V, \forall \alpha, u, \beta \in (X \cup V)^* : \left.\begin{array}{l} v \xrightarrow{*} \alpha u \beta \\ v' \xrightarrow{*} u \end{array}\right\} \Rightarrow v \xrightarrow{*} \alpha v' \beta$$

DÉFINITION 2 [2] : Une grammaire algébrique $G=\langle X, V, P, \sigma\rangle$ est pré-N.T.S. si et seulement si on a la propriété suivante :

$$\text{(C1)}\quad \forall v, v'\in V, \forall\alpha, u, \beta, u'\in X^* :\quad \left.\begin{array}{l} v \xrightarrow{*} \alpha u\beta \\ v' \xrightarrow{*} u' \\ v' \xrightarrow{*} u \end{array}\right\} \Rightarrow v \xrightarrow{*} \alpha u'\beta$$

Clairement, une grammaire N.T.S. est pré-N.T.S.

DÉFINITION 3 [2] : Un langage algébrique L est dit N.T.S. (resp. pré-N.T.S.) s'il existe une grammaire algébrique N.T.S. (resp. pré-N.T.S.) qui l'engendre.

Étant donné un alphabet de n lettres Z_n, on construit un alphabet de parenthèses $\hat{Z}_n=Z_n\cup\bar{Z}_n(\bar{Z}_n\cap Z_n=\emptyset)$ en définissant : $\bar{Z}_n=\{\bar{z}\mid z\in Z_n\}$.

DÉFINITION 4 [6] : Une grammaire algébrique $G=\langle\hat{Z}_n, V, P, \sigma\rangle$ est dite grammaire de parenthèses si toutes ses règles sont de l'une des deux formes suivantes :

(1) $v \to zv_1\bar{z}v_2$ où $z\in Z_n$ et $v, v_1, v_2\in V$
(2) $v \to 1$ (1 désigne le mot vide)

DÉFINITION 5 [7] : Une grammaire algébrique $G=\langle Y\cup\hat{Z}_n, V, P, \sigma\rangle$ $(Y\cap\hat{Z}_n=\emptyset)$ est dite grammaire de parenthèses généralisée si toutes ses règles sont de l'une des formes suivantes :

(1) $v \to zv_1\bar{z}v_2$ où $z\in Z_n$ et $v, v_1, v_2\in V$
(2) $v \to 1$
(3) $v \to bv_1$ où $b\in Y$ et $v, v_1\in V$

DÉFINITION 6 [6, 7] : Un langage algébrique L est dit langage de parenthèses (resp. langage de parenthèses généralisé) s'il existe une grammaire algébrique de parenthèses (resp. de parenthèses généralisée) qui l'engendre.

La remarque qui suit permet de restreindre l'étude des images par morphisme inverse des langages de parenthèses généralisés à celle des images par morphisme inverse des langages de parenthèses.

REMARQUE 1 : Si L est image dans un morphisme inverse d'un langage de parenthèses généralisé, il est l'image dans un morphisme inverse d'un langage de parenthèses.

Soit $L\subset X^*$, $L=\psi^{-1}(A)$ avec $\psi : X^*\to(Y\cup\hat{Z}_n)^*$ où A est un langage de parenthèses généralisé, engendré par $G=\langle Y\cup\hat{Z}_n, V, P, \sigma\rangle$. On construit $\bar{Y}=\{\bar{y}\mid y\in Y\}$ et $\hat{Y}=Y\cup\bar{Y}$. Soit Φ: $(Y\cup\hat{Z}_n)^*\to(\hat{Y}\cup\hat{Z}_n)^*$ défini par :

$\Phi(b) = b\bar{b}$ si $b \in Y$ et $\Phi(x) = x$ si $x \in \hat{Z}_n$. On ajoute un nouveau non-terminal $\bar{v}$ et la règle $\bar{v} \to 1$, et l'on obtient alors une grammaire

$$G' = \langle \hat{Y} \cup \hat{Z}_n, V \cup \{\bar{v}\}, P', \sigma \rangle$$

en remplaçant toutes les règles de type (3) de P : $v \to bv_1$ par la règle de type (1) : $v \to b\bar{v}\bar{b}v_1$. C'est alors une grammaire de parenthèses qui engendre un langage A' tel que $L = (\psi \circ \Phi)^{-1}(A')$.

Par ailleurs, on sait [1] que les langages de parenthèses sont des langages N.T.S.

Cette propriété est obtenue en exhibant pour tout langage de parenthèses L sur un alphabet $\hat{Z}_n$ une grammaire N.T.S. qui l'engendre. Nous rappelons ci-dessous comment est construite dans [1] cette grammaire N.T.S. qui l'engendre.

On désigne par $\mathscr{A}_n$ l'ensemble des arbres binaires dont les nœuds internes sont étiquetés par des lettres de Z_n. A chaque arbre de $\mathscr{A}_n$, on associe (bijectivement) un mot de la façon suivante :

. si l'arbre est l'arbre vide ε, le mot associé est le mot vide 1 : $\theta(\varepsilon) = 1$,

. si l'arbre t s'écrit $t = z(t_1, t_2)$, $z \in Z_n$ et si à t_1 et t_2 sont associés les mots m_1 et m_2, le mot associé à t s'écrit $zm_1\bar{z}m_2$:

$$\theta[z(t_1, t_2)] = z\,\theta(t_1)\bar{z}\theta(t_2).$$

On appellera $\theta(t)$ le mot associé à t. De la même façon, si m est un mot de Dyck sur $\hat{Z}_n$, il lui correspond un arbre associé $\theta^{-1}(m)$. On peut ainsi à une forêt F d'arbres de $\mathscr{A}_n$ associer un langage sur $\hat{Z}_n$ et à un sous-ensemble du langage de Dyck associer une forêt d'arbres de $\mathscr{A}_n$. On montre alors facilement [6] :

Fait : Le langage L sur $\hat{Z}_n$ est un langage de parenthèses si et seulement si il existe une forêt rationnelle F d'arbres de $\mathscr{A}_n$ telle $L = \theta(F)$.

(On trouvera dans [6] une définition des forêts rationnelles et des grammaires rationnelles d'arbres).

A chaque forêt rationnelle F, on peut associer une relation d'équivalence d'index fini de la façon suivante : la forêt F est reconnue par un automate d'arbres $\mathfrak{A}$ déterministe ascendant. Cet automate définit une application δ de $Q \times F$ dans Q, où Q désigne l'ensemble des états de l'automate, de la manière suivante : étant donné un couple (q, f) de $Q \times F$, on étiquette chaque feuille de l'arbre f par l'état initial q_0 de $\mathfrak{A}$, exceptée la feuille la plus à droite qui, elle, est étiquetée par q. L'automate d'arbre lisant l'arbre f, associe à sa racine un état qui définit $\delta(q, f)$. Ainsi, à l'arbre f, est associée l'application im_f de Q dans Q, donnée par :

$$im_f(q) = \delta(q, f)$$

Deux arbres f_1 et f_2 sont équivalents si ils définissent la même application de Q dans Q. Cette relation d'équivalence, notée $\sim$, jouit de deux propriétés remarquables :

(1) Définissons sur la famille $\mathscr{A}_n$ l'opération notée * qui à deux arbres f_1 et f_2 associe l'arbre $f_3 = f_1 * f_2$ obtenu en confondant la racine de f_2 avec la feuille la plus à droite de f_1. On vérifie aisément que l'équivalence $\sim$ est une congruence relativement à l'opération * :

$$f_2 \sim f_2' \Rightarrow \forall f_1, f_3 \qquad f_1 * f_2 * f_3 \sim f_1 * f_2' * f_3$$

(2) Si, au lieu de greffer l'arbre f_2 à la feuille la plus à droite de f_1, on le greffe à la $i^{\text{ème}}$ feuille de f_1, opération notée $f_1 *_i f_2$, on note que $\sim$ est régulière à droite :

$$f_2 \sim f_2' \Rightarrow \forall f_1, \forall i \qquad f_1 *_i f_2 \sim f_1 * f_2'$$

Nous désignerons (abusivement) l'équivalence $\sim$ comme la congruence syntaxique de la forêt F.

Soit maintenant L un langage de parenthèses et F_L la forêt rationnelle qui lui est canoniquement associée.

F_L définit donc une congruence syntaxique d'index fini. Choisissons dans chaque classe d'équivalence un représentant t et notons cette classe $[t]$.

Muni de l'opération $[t_1][t_2] = [t_1 * t_2]$, l'ensemble des classes d'équivalence reçoit une structure de monoïde. Ce monoïde est, bien entendu, fini.

Nous construisons à présent, en utilisant la congruence syntaxique associée à F_L, une grammaire algébrique engendrant L.

Soit donc $G = \langle \hat{Z}_n, V, P \rangle$ où :

— $\hat{Z}_n$ est l'alphabet de L.

— V est un ensemble en bijection avec l'ensemble des classes de la congruence syntaxique associée à F_L. Nous notons $[\underline{t}]$ la variable de V correspondant à la classe $[t]$.

L'ensemble d'axiomes A est donné par $\{ [\underline{t_{o,i}}], \ldots, [\underline{t_{o,p}}] \}$ tels que pour $i = 1, \ldots, p$, $t_{o,i}$ appartient à F_L et $F_L = \bigcup_{i=1}^{p} [t_{o,i}]$.

— P, l'ensemble des règles, est constitué de productions de trois types différents :

(1) $\forall t, t_1, t_2$ tels que $t \sim t_1 * t_2$, $[\underline{t}] \rightarrow [\underline{t_1}][\underline{t_2}]$

(2) $\forall t, t_1, t_2$ tels que $\exists t' \in [t], t'_1 \in [t_1], t'_2 \in [t_2]$ avec $t' = \underset{t'_1 \quad t'_2}{\overset{z}{\wedge}}$, $[\underline{t}] \to z[\underline{t_1}]\bar{z}[\underline{t_2}]$

(3) $\forall W_t$ mot de $\hat{Z}_n^*$ associé à un arbre de $[t]$ et de longueur minimum $[\underline{t}] \to W_t$.

Nota: La grammaire construite dans [1] comportait 2 sortes de règles supplémentaires

a) $\forall t, t_1$ tels que $\exists t' \in [t]$ et $t'_1 \in [t_1]$ avec $t' = \underset{t_1}{\overset{z}{/}}$ on avait la règle $[\underline{t}] \to z[\underline{t_1}]\bar{z}$

b) $\forall t, t_1$ tels que $\exists t' \in [t]$ et $t'_1 \in [t_1]$ que $t' = \overset{z}{\diagdown}_{t_1}$ on avait la règle $[\underline{t}] \to z\bar{z}[\underline{t_1}]$.

Ces 2 types peuvent être obtenus à l'aide des règles de type (2) et de la règle $[\underline{1}] \to 1$. On peut donc s'en passer.

PROPOSITION *1* [1] : *La grammaire G engendre le langage L à partir de l'ensemble d'axiomes A et est une grammaire N.T.S.*

Dans toute la suite, nous utiliserons cette grammaire pour engendrer un langage de parenthèses, grammaire qui de plus sera supposée réduite. Nous la désignerons sous le nom de grammaire utile de *L*.

Nous montrons tout de suite que cette grammaire possède une propriété qui nous sera utile dans la suite :

LEMME 1 : *Soit* $G = \langle \hat{Z}_n, V, P, A \rangle$ *la grammaire utile de L* ; $\forall \alpha, \beta \in (\hat{Z}_n \cup V)^*$, $\forall v_1, v_2 \in V : \alpha v_1 v_2 \beta \in \hat{L}_G(A) \Rightarrow \exists v \in V : \alpha v \beta \in \hat{L}_G(A)$ *et* $v \to v_1 v_2 \in P$,

Preuve. — Évident, car aux variables v_1 et v_2 correspondent des classes syntaxiques $[t_1]$ et $[t_2]$, où t_1 et t_2 sont des représentants de ces classes. Le produit $[t_1][t_2] = [t_1 * t_2]$ est également une classe syntaxique. Il lui correspond donc une variable v. Or, d'après la définition de l'ensemble des règles P de la grammaire G, on a bien $v \to v_1 v_2$ par définition des règles de type 1. Comme de plus la grammaire est N.T.S., si l'on a $u \xrightarrow{*} \alpha v_1 v_2 \beta$, on a également $u \xrightarrow{*} \alpha v \beta$.

II. IMAGES DE DYCK DANS UN HOMOMORPHISME INVERSE

On sait [6] que, parmi les langages de parenthèses, les langages de Dyck D_n^* jouent un rôle important : ils sont des éléments maximaux car associés aux forêts rationnelles $\mathscr{A}_n$. Nous utiliserons librement les notions de mot irréductible de $\hat{Z}_n^*$, de mot réduit, de mots équivalents, ces divers qualificatifs faisant référence à la congruence de Dyck engendrée par les relations $z\bar{z} \equiv 1$ $\forall z \in Z_n$.

Nous commençons par prouver un lemme concernant les produits de mots irréductibles. Celui-ci permet de montrer facilement que les images homomorphes inverses d'un langage de Dyck sont N.T.S., ce résultat pouvant alors être étendu à tous les langages de parenthèses.

LEMME 2 : *Étant donnés $k+1$ mots irréductibles $u_1, u_2, \ldots, u_k$ et v tels que*
(1) $u_1 u_2 \ldots u_k \equiv v$
(2) $\forall i_i \quad |u_i| \leqslant l \qquad et \qquad |v| \leqslant l$
alors, ou bien $k \leqslant 2$, ou bien il existe un indice i tel que $u_i u_{i+1} \equiv w$ avec $|w| \leqslant l$.

Preuve : La preuve se fait par induction sur l. Si $l=0$, il n'y a rien à prouver car tous les éléments concernés sont vides. Si $l=1$, on constate que si k vaut au moins 3, soit l'un des facteurs u_i est vide, soit l'on trouve un facteur $u_i = z$ et $u_{i+1} = \bar{z}$ ce qui établit le résultat.

Supposons donc maintenant le lemme établi pour $l \leqslant m$. Nous allons établir le lemme pour $l = m+1$ par l'absurde. Si, pour $l = m+1$, le lemme était faux, on pourrait trouver un produit de k mots irréductibles $u_1 u_2 \ldots u_k$ satisfaisant

(1) $u_1 u_2 \ldots u_k \equiv v$
(2) $\forall i \;\; |u_i| \leqslant m+1 \;\;$ et $\;\; |v| \leqslant m+1 \;\;$ et $\;\; k > 2 \;\;$ et $\;\; \forall i \;\; u_i u_{i+1} \equiv w \Rightarrow |w| > l$

On peut toujours choisir ce produit de façon à ce que k soit minimal (et donc, tout produit de moins de termes satisfait le lemme).

Fait 1 : La première lettre de u_1 ne se simplifie pas dans $u_1 u_2 \ldots u_k$. En effet, sinon il existe un indice i tel que u_i contienne la lettre associée à la première lettre de u_1 ; soit

$$u_1 = x u'_1 \qquad u_i = u'_i \bar{x} u''_i \qquad \text{et} \qquad x u'_1 u_2 \ldots u_{i-1} u'_i \bar{x} \equiv 1$$

Ainsi, $u_1 u_2 \ldots u_i \equiv u''_i$.

Notant $\bar{u}'_i$ le mot le plus court tel que $\bar{u}'_i u'_i \equiv 1$, on peut aussi écrire $u_1 u_2 \ldots u_{i-1} \equiv x \bar{u}'_i$. Ainsi, si $i \neq k$, la première équivalence viole-t-elle la minimalité de k et si $i = k$, c'est la seconde qui viole cette minimalité : aucun des produits concernés ne peut contenir de facteur $u_j u_{j+1}$ équivalent à un mot w de longueur au plus $m+1$ sans que le produit initial ne contienne ce même facteur.

Fait 2 : La dernière lettre de u_k ne se simplifie pas dans $u_1 u_2 \ldots u_k$. La preuve de ce second fait est identique à celle du fait 1.

Il résulte de ces deux faits que l'on peut écrire

$$\begin{aligned} u_1 &= x u'_1 \\ u_k &= u'_k y \\ v &= x v' y \qquad \text{et} \qquad u'_1 u_2 \ldots u_{k-1} u'_k \equiv v' \end{aligned}$$

Fait 3 : Pour tout i $1 \leqslant i \leqslant k-1$, la dernière lettre de u_i se simplifie avec la première lettre de u_{i+1}.

En effet, si tel n'est pas le cas

— ou bien la dernière lettre de u_i ne se simplifie pas et l'on a $u_1 u_2 \ldots u_i$ équivalent à un facteur gauche de v. Comme $i<k$, on contredit la minimalité de k

— ou bien la dernière lettre de u_i se simplifie avec une lettre située à sa gauche ; c'est qu'il existe un entier $j \neq 0$ tel que

$$u_{i-j} = u'_{i-j} x u''_{i-j} \qquad u_i = u'_i \bar{x}$$

et

$$u''_{i-j} u_{i-j+1} \ldots u_{i-1} u'_i \equiv 1$$

soit encore $u_{i-j} u_{i-j+1} \ldots u_{i-1} u_i \equiv u'_{i-j}$ ce qui viole la minimalité de k

— ou bien la dernière lettre de u_i se simplifie avec une lettre située à sa droite ; c'est qu'il existe un entier $j \neq 0$ tel que

$$u_{i+j} = u'_{i+j} \bar{x} u''_{i+j} \qquad u_i = u'_i x \qquad \text{et} \qquad u_{i+1} u_{i+2} \ldots u'_{i+j} \equiv 1$$

Il en résulte $u_{i+1} u_{i+2} \ldots u_{i+j} \equiv \bar{x} u''_{i+j}$.

Si $j \neq 1$, on contredit la minimalité de k, et si $j=1$, on a $u'_{i+1} \equiv 1$, ce qui implique $u'_{i+1} = 1$ (car u_{i+1} est irréductible) et $\bar{x}$ est la première lettre de u_{i+1}.

En résumé de ces 3 faits, on peut écrire :

$$\begin{aligned} &u_1 = x u'_1 x_1 && \text{(on posera } \bar{x}_0 = x) \\ &u_k = \bar{x}_{k-1} u'_k y && \text{(on posera } x_k = y) \\ &\forall i \quad 2 \leqslant i \leqslant k-1 && u_i = \bar{x}_{i-1} u'_i x_i \end{aligned}$$

avec $v = x v' y$.

On en déduit que $u'_1 u'_2 \ldots u'_k \equiv v'$ avec $|u'_i| \leqslant m-1$ et $|v'| \leqslant m-1$. Par hypothèse de récurrence, il existe un indice i tel que $u'_i u'_{i+1} \equiv w'$ avec $|w'| \leqslant m-1$ et donc $\bar{x}_{i-1} u'_i x_i \bar{x}_i u'_{i+1} x_{i+1} \equiv \bar{x}_{i-1} w' x_{i+1}$ soit $u_i u_{i+1} \equiv w$ avec $|w| \leqslant m+1$ ce qui contredit l'hypothèse que $u_1 u_2 \ldots u_k$ ne satisfait pas le lemme.

Nous allons maintenant utiliser le lemme 2 pour établir deux propositions : l'une affirme que la classe d'un mot quelconque pour la congruence de Dyck est un langage N.T.S., l'autre que l'image homomorphe inverse d'un langage de Dyck est N.T.S.

PROPOSITION 2 [3] : *La classe d'un mot f de $\hat{Z}_n^*$ pour la congruence de Dyck est un langage N.T.S.*

Preuve. — On peut évidemment supposer que f est un mot irréductible non vide de longueur l. On construit alors la grammaire $G=\langle \hat{Z}_n, V, P\rangle$ où $V=\{ [m] \mid m$ est irréductible et $|m|\leqslant l\}$

$$P=\{ [m] \to [m_1][m_2] \mid m\equiv m_1m_2 \} \cup \{ [z] \to z \mid z\in\hat{Z}_n \} \cup \{ [1] \to 1 \}$$

(1) Clairement, le langage engendré par une variable $[m]$ de G ne contient que des mots équivalents à m. Par ailleurs, si $m_1, m_2, \ldots, m_k$ sont des mots irréductibles de longueur au plus l tels que $m_1m_2 \ldots m_k\equiv m$ avec $|m|\leqslant l$, on prouve par induction sur k que $[m]$ engendre $[m_1][m_2] \ldots [m_k]$ dans G : si $k=1$, il n'y a rien à prouver car alors $m=m_1$; si $k=2$, on sait que, par construction de P, $[m] \to [m_1][m_2]$ est une règle de G. Supposons alors le résultat établi pour $k\leqslant n$ et considérons $(n+1)$ mots irréductibles tels que $m_1m_2 \ldots m_{n+1}\equiv m$, avec $\forall i \quad |m_i|, |m|\leqslant l$.

Le lemme 2 garantit que, puisque $n+1\geqslant 3$, il existe un indice i tel que $m_im_{i+1}\equiv p$ avec $|p|\leqslant l$. Ainsi, on sait que $[p] \to [m_i][m_{i+1}]$ (cf. ci-dessus) est une règle de G. On sait aussi que $m\equiv m_1m_2 \ldots m_{i-1}pm_{i+2} \ldots m_{n+1}$ et donc, par induction, $[m] \xrightarrow{*} [m_1][m_2] \ldots [m_{i-1}][p][m_{i+2}] \ldots [m_{n+1}]$. Il en résulte immédiatement $[m] \xrightarrow{*} [m_1][m_2] \ldots [m_{n+1}]$.

Comme tout mot g équivalent à m peut s'écrire comme le produit de ses lettres. On en déduit que $g=x_1x_2 \ldots x_n\equiv m$ implique $[m] \xrightarrow{*} [x_1][x_2] \ldots [x_n]$ et donc $[m] \xrightarrow{*} g$ puisque $[x_i] \to x_i$ est une règle de G. Ainsi est établi le premier fait :

$$\forall [m]\in V \qquad L_G([m])=\{ g \mid g\equiv m \}$$

Il en résulte que $[f]$ engendre le langage cherché.

(2) Nous établissons maintenant que la grammaire G construite ci-dessus est N.T.S.

A un mot $h\in(\hat{Z}_n\cup V)^*$ nous associons les deux mots

$t(h)$ dans $\hat{Z}_n^*$ obtenu en remplaçant chaque variable $[m]$ de h par le mot m

$v(h)$ dans V^* obtenu en remplaçant chaque lettre terminale z de h par la variable $[z]$.

Notons qu'il est prouvé ci-dessus que si $[m]$ engendre h, $[m]$ engendre $v(h)$ et $t(h)$. Supposons donc maintenant que

$$[m] \xrightarrow{*} \alpha u\beta$$
$$[p] \xrightarrow{*} u \qquad \alpha, u, \beta\in(\hat{Z}_n\cup V)^*$$

On sait alors que $[m] \xrightarrow{*} t(\alpha)t(u)t(\beta)$ et que p est équivalent à $t(u)$. Il en résulte que $[m]$ engendre $t(\alpha)pt(\beta)$. Alors, on sait que $[m] \xrightarrow{*} v(\alpha)[p]v(\beta)$ et donc $[m]$ engendre $\alpha[p]\beta$.

La seconde proposition annoncée peut s'écrire :

PROPOSITION 3 : *Étant donné un homomorphisme Φ de X^* dans $\hat{Z}_n^*$, le langage $\Phi^{-1}(D_n^*)$ est N.T.S.*

Preuve : Nous commençons par une remarque : si l'image par Φ d'une lettre x n'est pas un mot irréductible, on peut remplacer cette image par le mot irréductible équivalent à $\Phi(x)$. Nous supposerons donc que, quel que soit x, $\Phi(x)$ est un mot irréductible et nous désignons par l la longueur maximale d'une image d'une lettre. On construit alors la grammaire $G=\langle X, V, P\rangle$ où $V=\{\ [m]\mid m$ est un mot irréductible sur $\hat{Z}_n$, $|m|\leqslant l\}$

$$P=\{\ [m]\ \to\ [m_1][m_2]\mid m\equiv m_1m_2\ \}\cup\{\ [m]\ \to\ x\mid \Phi(x)=m\ \}\cup\{\ [1]\ \to\ 1\ \}$$

On notera que, si h est équivalent à m, on sait que $[m]$ engendre $v(h)$ avec les notations de la preuve de la Proposition 2.

(1) Nous prouvons que $L_G([1])=\Phi^{-1}(D_n^*)$. Pour ce faire, il suffit de remarquer d'abord que si $[m]\overset{*}{\to} f\in X^*$, on a $\Phi(f)$ équivalent à m. Comme par ailleurs, à tout mot f de X^*, on peut associer le mot $\theta(f)$ de V^* défini par : si $f=x_1x_2\ \ldots\ x_n$ alors $\theta(f)=[\Phi(x_1)][\Phi(x_2)]\ \ldots\ [\Phi(x_n)]$, il suffit de montrer que si $f\in\Phi^{-1}(D_n^*)$, on a $[1]\overset{*}{\to}\theta(f)$. Cette preuve est immédiate par induction sur la longueur de f (cf. preuve de la proposition 2).

(2) Nous prouvons maintenant que G est une grammaire N.T.S. Cette preuve, à nouveau, est similaire à celle donnée pour la proposition 2 en utilisant pour chaque mot h de $(X\cup V)^*$ le mot $v(h)$ obtenu en remplaçant chaque lettre x de X dans h par $\theta(x)=[\Phi(x)]$.

REMARQUE : La preuve de lemme 2 et donc celle des propositions 2 et 3 sont valides si l'on remplace le langage de Dyck D_n^* par le langage de Dyck bilatère défini comme classe de 1 dans la congruence $x\bar{x}=\bar{x}x=1\ \ x\in Z_n$. Il en résulte en particulier que l'image homomorphe inverse d'un langage de Dyck bilatère est N.T.S.

III. IMAGES D'UN LANGAGE DE PARENTHÈSES DANS UN MORPHISME INVERSE

Une famille de langages est un cylindre si elle est fermée par morphisme inverse et intersection rationnelle. Comme on sait que les langages N.T.S. sont stables par intersection rationnelle, la proposition 3 assure que le cylindre engendré par D_n^* est formé de langages N.T.S. On peut étendre ce résultat :

Nous montrons ici que tout langage de parenthèses engendre un cylindre contenu dans la famille N.T.S.

Nous verrons cependant qu'il existe des langages N.T.S. qui n'appartiennent pas au cylindre engendré par les langages de parenthèses.

Quelques notations :

Étant donné un langage de parenthèses L, nous noterons G_0 la grammaire utile de L. Elle ne comporte que des productions du type

(1) $[\underline{t_i}] \to [\underline{t_j}][\underline{t_k}]$ ou (2) $[\underline{t_i}] \to z[\underline{t_j}]\bar{z}[\underline{t_k}]$ ou (3) $[\underline{1}] \to 1$

Nous noterons G la grammaire suivante de D_n^* : $G = \langle \hat{Z}_n, \{S\}, P \rangle$ P : (1) $S \to SS$, (2) $S \to zS\bar{z}S$, (3) $S \to 1$

Notons θ le morphisme : $\theta : (\hat{Z}_n \cup V_0)^* \to (\hat{Z}_n \cup \{S\})^*$

$$x \in \hat{Z}, \qquad \theta(x) = x$$
$$[\underline{t_i}] \in V_0, \qquad \theta([\underline{t_i}]) = S$$

Notons ψ le morphisme : $\psi : (\hat{Z}_n \cup \{S\})^* \to Z_n^*$

$$x \in \hat{Z}_n, \qquad \psi(x) = x$$
$$S, \qquad \psi(S) = 1$$

Les lemmes 3 et 4 montrent que le fonctionnement de la grammaire G_0 est calqué sur celui de la grammaire G qui à son tour se ramène au fonctionnement de la congruence $\equiv$ sur $\hat{Z}_n^*$.

LEMME 3 : *a) Soient* $f, f' \in (\hat{Z}_n \cup V_0)^*$.

Si $f \overset{*}{\underset{G_0}{\leftrightarrow}} f'$ *alors* $\theta(f) \overset{*}{\underset{G}{\leftrightarrow}} \theta(f')$.

b) Soient $g, g' \in (\hat{Z}_n \cup \{S\})^*$, $f \in (\hat{Z}_n \cup V_0)^*$ *tels que* $\theta(f) = g$ *et* $g \overset{*}{\underset{G}{\leftarrow}} g'$

$\Rightarrow \exists f' \in (\hat{Z}_n \cup V_0)^*$ *tel que* $f \overset{*}{\underset{G_0}{\leftarrow}} f'$ *et* $\theta(f') = g'$.

Démonstration : a) L'image par θ d'une production de G_0 est une production de G. On en déduit facilement la propriété annoncée.

b) Si $m \underset{G}{\leftarrow} S$ est « l'opposée » d'une production de G et si $\theta(m') = m$, il existe $[\underline{t_i}] \in V_0$ telle que : $m' \underset{G_0}{\leftarrow} [\underline{t_i}]$ et $\theta([\underline{t_i}]) = S$.

En effet,

(1) si $m = SS$ et $m' = [\underline{t_j}][\underline{t_k}]$ alors $[\underline{t_i}] = [\underline{t_j * t_k}]$ convient

(2) si $m = zS\bar{z}S$ et $m' = z[\underline{t_j}]\bar{z}[\underline{t_k}]$ alors $[\underline{t_i}] = [\underline{t_j \overset{z}{\wedge} t_k}]$ convient

(3) si $m = 1$ et $m' = 1$ alors $[\underline{t_i}] = [\underline{1}]$ convient

On en déduit que :

si $g \underset{G}{\leftarrow} g'$ et $\theta(f) = g$ alors $\exists f' \in (\hat{Z}_n \cup V_0)^*$ tel que $f \underset{G_0}{\leftarrow} f'$ et $\theta(f') = g'$

On peut donc affirmer que :

si $g \underset{G}{\rightarrow} g'$ et $\theta(f)=g$ alors $\exists f' \in (\hat{Z}_n \cup V_0)^*$, $f \underset{G_0}{\rightarrow} f'$ et $\theta(f')=g'$

Autrement dit, toute chaîne de longueur 1 pour $\overset{*}{\underset{G}{\leftarrow}}$ se relève par θ en une chaîne de longueur 1 pour $\overset{*}{\underset{G_0}{\leftarrow}}$:

$$\forall f, g, g' \text{ tels que } \begin{array}{c} f \\ \theta \downarrow \\ g \underset{G}{\leftarrow} g', \end{array} \qquad \exists f' \text{ tel que } \begin{tikzcd} f & f' \arrow[l, "G_0"'] \\ g \arrow[u, leftarrow, "\theta"'] & g' \arrow[l, "G"] \arrow[u, leftarrow, "\theta"] \end{tikzcd}$$

On en déduit par récurrence que toute chaîne de longueur n pour $\overset{*}{\underset{G}{\leftarrow}}$ se relève par θ en une chaîne de longueur n pour $\overset{*}{\underset{G_0}{\leftarrow}}$:

$$\forall f, g, g', \quad \begin{array}{c} f \\ \theta \downarrow \\ g \overset{*}{\underset{G}{\longleftarrow}} g' \end{array} \qquad \exists f', \quad \begin{tikzcd} f & f' \arrow[l, "n", "G_0"'] \\ g \arrow[u, leftarrow, "\theta"'] & g' \arrow[l, "n", "G"'] \arrow[u, leftarrow, "\theta"] \end{tikzcd}$$

LEMME 4 : *a) Soient* $g, g' \in (\hat{Z}_n \cup \{S\})^*$ *si* $g \overset{*}{\underset{G}{\leftrightarrow}} g'$ *alors* $\psi(g) \equiv \psi(g')$.

b) Soient $h, h' \in \hat{Z}_n^*$ *et* $g \in (\hat{Z}_n \cup S)^*$ *tels que* $\psi(g)=h \equiv h'$, *alors* $\exists g' \in (\hat{Z}_n \cup S)^*$ *tel que* $g \overset{*}{\underset{G}{\rightarrow}} g'$ *et* $\psi(g')=h'$.

Démonstration : Notons $\mathcal{S} = \{(z\bar{z}, 1)\}_{z \in Z_n}$, $\mathcal{S}^{-1} = \{(1, z\bar{z})\}_{z \in Z_n}$ (engendre la congruence $\equiv$).

(a) Il suffit de vérifier que ψ envoie les productions (1) et (3) sur $(1, 1)$ et la production (2) sur $(1, z\bar{z})$.

(b) Il suffit de montrer la propriété b pour $(h, h') \in \mathcal{S} \cup \mathcal{S}^{-1}$.

* $(h, h')=(z\bar{z}, 1)$ g est nécessairement de la forme $g=S^p z S^q \bar{z} S^r (p, q, r) \in N^3$. Posons $g'=1$. On a bien :

$$\begin{tikzcd} g \arrow[r, leftrightarrow, "*", "G"'] \arrow[d, "\psi"'] & g' \arrow[d, "\psi"] \\ z\bar{z} \arrow[r, "\mathcal{S}"'] & 1 \end{tikzcd}$$

* $(h, h') = (1, z\bar{z})$ alors $g=S^p$. Posons $g'=z\bar{z}$.

On a bien :

$$\begin{tikzcd} g \arrow[r, leftrightarrow, "*", "G"'] \arrow[d, "\psi"'] & g' \arrow[d, "\psi"] \\ 1 \arrow[r, leftrightarrow, "\mathcal{S}"'] & z\bar{z} \end{tikzcd}$$

* La propriété b se déduit par récurrence sur la longueur d'une chaîne pour le système $\mathscr{S}$ allant de h à h'.

Notons μ le morphisme : $(\hat{Z}_n \cup V_0)^* \to N$ (monoïde additif)

$$x \in \hat{Z}_n \Rightarrow \mu(x) = 1$$
$$x = [\underline{t_i}] \in V_0 \Rightarrow \mu(x) = 0$$

LEMME 5 : *Étant donnés* $k+1$ *mots* $f_1, f_2, \ldots, f_k$ *et* m *tels que*

(1) $f_1 f_2 \ldots f_k \overset{*}{\underset{G_0}{\leftrightarrow}} m$

(2) $\forall i \quad \mu(f_i) \leqslant l \quad$ *et* $\quad \mu(m) \leq l$

alors : $k=1$ *ou bien il existe un indice* $i \leqslant k-1$ *tel que* $f_i f_{i+1} \overset{*}{\underset{G_0}{\leftrightarrow}} f$ *avec* $\mu(f) \leqslant l$.

Démonstration : Ce lemme découle immédiatement du lemme 2 via un aller-retour du morphisme $\psi \circ \theta$.

En effet : puisque $f_1 f_2 \ldots f_k \overset{*}{\underset{G_0}{\leftrightarrow}} m$, si l'on note $\begin{cases} u_i = \psi \circ \theta(f_i), \\ v = \psi \circ \theta(m) \end{cases}$ par les lemmes $3a$, $4a$ on obtient : $u_1 u_2 \ldots u_k \equiv v$.

D'autre part $|u_i| = \mu(f_i) \leq l$ et $|v| = \mu(m) \leqslant l$.

Donc, par le lemme 2, ou bien $k=1$, ou bien il existe un indice $i \leqslant k-1$ tel que $u_i u_{i+1} \equiv w$ avec $|w| \leqslant l$; ainsi

* Si $k=1$, le lemme est vérifié.

* Sinon, d'après les lemmes 3-b, 4-b, il existe $f \in (\hat{Z}_n \cup V_0)^*$ tel que

$$\begin{array}{ccc} f_i f_{i+1} & \overset{*}{\underset{G_0}{\longleftarrow}} & f \\ {\scriptstyle \psi\circ\theta}\downarrow & & \downarrow{\scriptstyle \psi\circ\theta} \\ u_i u_{i+1} & \overset{*}{\underset{G}{\longleftarrow}} & w \end{array}$$

Comme $\mu(f) = |w|$ on a bien : $f_i f_{i+1} \overset{*}{\underset{G_0}{\leftarrow}} f$ et $\mu(f) \leqslant l$.

Remarques : 1) Notons $\mathscr{C}$ le système de congruence formé :

. des inverses des productions de G_0 de type (1) et (2) :

$$[\underline{t_j}][\underline{t_h}] \underset{\mathscr{C}}{\to} [\underline{t_i}]$$
$$z[\underline{t_j}]\bar{z}[\underline{t_h}] \underset{\mathscr{C}}{\to} [\underline{t_i}]$$

. de la production de type (3) :

$$[\underline{1}] \underset{\mathscr{C}}{\to} 1$$

$\mathscr{C}$ engendre la même congruence que P donc : $\overset{*}{\underset{\mathscr{C}}{\leftrightarrow}} = \overset{*}{\underset{G_0}{\leftrightarrow}}$.

2) Pour tout mot $f \in (\hat{Z}_n \cup V_0)^*$, $\mathscr{C}$-irréductible, on a l'inégalité : $|f| \leqslant 2\mu(f)+1$.

En effet, f ne peut contenir de facteur de la forme $[\underline{t_j}][\underline{t_h}]$ puisque

$$[\underline{t_j}][\underline{t_h}] \underset{\mathscr{C}}{\to} [\underline{t_j * t_h}]$$

(voir préliminaire lemme 1). Par conséquent, l'ensemble $\{ [f]_{\overset{*}{\underset{G_0}{\leftrightarrow}}} \mid \mu(f) \leqslant l \}$ est fini.

PROPOSITION 4: *Toute classe d'un mot* $f \in (\hat{Z}_n \cup V_0)^*$ *pour la congruence* $\overset{*}{\underset{G_0}{\leftrightarrow}}$ *est N.T.S. En effet* $[f]_{\overset{*}{\underset{\mathscr{G}_0}{\leftrightarrow}}} = L_{G_0}(A)$ *où A est l'ensemble fini* $\{ g \mid g \in [f]_{G_0} ; \mu(g) \leq l \}$.

REMARQUE : Si l'on note $\overset{*}{\underset{\mathscr{G}_0}{\leftrightarrow}}$ la restriction de $\overset{*}{\underset{G_0}{\leftrightarrow}}$ à $\hat{Z}_n^*$, $[f]_{\overset{*}{\underset{G_0}{\leftrightarrow}}} = [f]_{\overset{*}{\underset{\mathscr{G}_0}{\leftrightarrow}}} \cap (Z_n)^*$, donc $[f]_{\overset{*}{\underset{G_0}{\leftrightarrow}}}$ est aussi un langage N.T.S.

PROPOSITION 5 : *Étant donné un homomorphisme* ϕ *de* X^* *dans* $(\hat{Z}_n)^*$ *et un langage de parenthèses* $L \subset (\hat{Z}_n)^*$, $\phi^{-1}(L)$ *est N.T.S.*

Démonstration : Il suffit de recopier la démonstration des propositions correspondantes à propos de D_n^* en ayant soin de remplacer :

. la congruence $\equiv$ par la congruence $\overset{*}{\underset{G_0}{\leftrightarrow}} (= \overset{*}{\underset{\mathscr{C}}{\leftrightarrow}})$

. la notion de $\mathscr{S}$-irréductible par la notion de $\mathscr{C}$-irréductible

. la notion de longueur d'un mot f, par celle d'image par la fonction μ de f.

COROLLAIRE : *Si L est un langage de parenthèses, le cylindre* $\mathscr{C}(L)$ *engendré par L est formé de langages N.T.S.*

Démonstration : En effet, la famille N.T.S. est fermée par intersection rationnelle [2].

IV. LES LANGAGES N.T.S. CONTIENNENT STRICTEMENT LE CYLINDRE ENGENDRÉ PAR LES LANGAGES DE PARENTHÈSES

Nous exhibons ici un langage N.T.S. ($\# S_< = \{ \# a^p b^q \mid 1 \leqslant p \leqslant q \}$) qui n'est pas dans le cylindre engendré par des langages de parenthèses. Pour établir ce fait, nous montrons que si L est un langage de parenthèses et $\alpha \in \hat{Z}_n^*$ alors $\alpha^{-1} L$ est union finie de classes pour la congruence $\overset{*}{\underset{G_0}{\leftrightarrow}}$ (proposition 6).

Nous en déduirons que si $L' \in X^*$ est dans le cylindre engendré par un langage de parenthèses, pour tout $\alpha' \in X^*$, $\alpha'^{-1} L'$ est N.T.S. (proposition 7). Cette propriété est violée par $\# S_<$; qui n'est donc pas de la forme $\phi^{-1}(L) \cap K$.

Dans ce qui suit, nous fixons L langage de parenthèses sur $\hat{Z}_n^*$ et un mot $\alpha \in (\hat{Z}_n)^+$.

G_0 est la grammaire utile de L, A_0 l'ensemble des variables telles que $L_{G_0}(A_0) = L$.

Si T est un arbre de dérivation dans la grammaire G_0, nous notons $r(T)$ l'étiquette de sa racine, $fr(T)$ le mot formé par ses feuilles (le produit ou frontière de cet arbre de dérivation), et $\| T \|$ le nombre de nœuds de T ($\| T \|$ est la taille de l'arbre T).

DÉFINITION 7 : Étant donné un arbre de dérivation T' dont le mot $fr(T)$ s'écrit αf, nous appellerons arbre *marqué à gauche* T par rapport à α l'arbre obtenu en marquant dans T' les feuilles d'étiquette non vide qui forment cette occurrence de α, (f est un mot de $(\hat{Z}_n \cup V_0)^*$).

Nous notons $\underline{d(T)}$ l'unique mot tel que $fr(T)=\alpha d(T)$. Nous appellerons *arête de* T, la branche joignant la racine à la feuille marquée la plus à droite.

DÉFINITION 8 : Un arbre marqué à gauche sera dit *minimal* si et seulement si (1) $d(T)$ ne contient pas de facteur $u \in (Z_n \cup V_0)^*$ tel que $\exists S \in V_0$, $S_i \overset{*}{\underset{G_0}{\to}} u$ et $|u| \geqslant 2$.

(2) Si T' est un arbre de dérivation, $fr(T')=fr(T) \Rightarrow \| T' \| \geqslant \| T \|$.

Nous montrons que de tels arbres minimaux existent bien :

LEMME 6 : *$\forall f \in (\hat{Z} \cup V_0)^*$ $\forall \alpha \in \hat{Z}_n$, $\forall S \in V_0$ si $S \overset{*}{\underset{G_0}{\to}} \alpha f$, alors $\exists T$, arbre minimal tel que $d(T) \overset{*}{\underset{G_0}{\to}} f$ et $r(T)=S$.*

Démonstration : On effectue une récurrence sur $|f|$.

a) Si $|f|=0$.

Soit T un arbre de dérivation de S en α, dont la taille est minimale. T est minimal $d(T)=1$ et $d(T) \overset{*}{\underset{G_0}{\to}} 1=f$.

b) Si $|f|=n+1$.

1^er^ cas : Si f ne contient aucun facteur $u \in (\hat{Z}_n \cup V_0)^*$ tel que, $\exists S_i \in V_0$, $S_i \overset{*}{\underset{G_0}{\to}} u$ et $|u| \geqslant 2$, alors tout arbre de dérivation T de S en αf, (dont on marque les feuilles composant α), de taille minimale, est minimal.

Pour un tel arbre, on a $d(T)=f$ et $r(T)=S$.

2^e^ cas : $f=f_1 u f_2$ avec $S_i \overset{*}{\underset{G_0}{\to}} u$ et $|u| \geqslant 2$, posons $f'=f_1 S_i f_2$.

Comme G_0 est N.T.S., $S \overset{*}{\to} \alpha f'$ or $|f'| < |f|$.

Par hypothèse de récurrence, il existe un arbre minimal T, tel que $d(T) \overset{*}{\underset{G_0}{\to}} f'$.

D'où : $d(T) \overset{*}{\to} f$.

LEMME 7 : *Si T est un arbre minimal,* $|d(T)| \leqslant 3|\alpha|+1$.

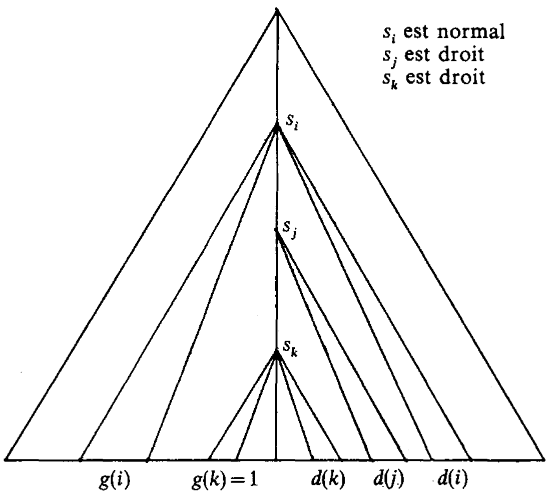
Démonstration : Notons $s_0, s_1, \ldots, s_i, s_{i+1}, \ldots, s_m$ l'arête de T (nous les appellerons *sommets* de T). Nous noterons :

. $d(i)$ le mot formé par les feuilles non marquées, qui descendent de s_i et ne descendent pas de s_{i+1}

$g(i)$ le mot formé par les feuilles marquées, qui descendent de s_i et ne descendent pas de s_{i+1}.

Un sommet s_i est *normal* si $g(i) \neq 1$
droit si $g(i) = 1$

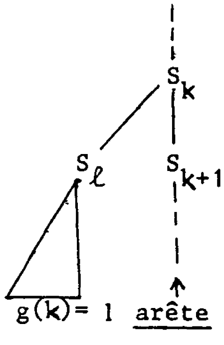
Schéma :



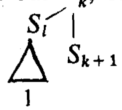
. Notons $s_{i_1}, s_{i_2}, \ldots, s_{i_p}$ les sommets normaux. Comme $\alpha = g(i_1)g(i_2)\ldots g(i_p)$ $p \leqslant |\alpha|$.

* Montrons que l'arête ne peut contenir deux sommets droits consécutifs :
Soit s_k un sommet droit, d'étiquette S_k.

* 1) On ne peut avoir :

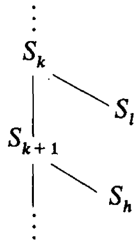


Car on aurait $S_k \xrightarrow{*} S_{k+1}$, donc $S_k = S_{k+1}$. Par conséquent, en supprimant dans T le facteur S_k, on obtiendrait T' tel que $\| T' \| < \| T \|$, et $fr(T') = fr(T)$



ce qui contredit la condition 2).

* 2) Si s_k, s_{k+1} sont deux sommets droits consécutifs, d'étiquettes S_k, S_{k+1}, leur occurrence dans T est du type :



Donc $S_l \xrightarrow[G_0]{*} d(k)$ et $S_h \xrightarrow[G_0]{*} d(k+1)$.

Nécessairement $d(k) \neq 1$ et $d(k+1) \neq 1$ (sinon par le même raisonnement que dans *1), T ne vérifie pas la condition 2).

Il existe une variable S, $S \xrightarrow[G_0]{} S_h S_l$.

Donc, $S \xrightarrow[G_0]{*} d(k+1)d(k)$ et $| d(k+1)d(k) | \geqslant 2$, ce qui contredit la condition 1. (Fin de *).

** Le nombre de sommets normaux est au plus $| \alpha |$. En vertu de *, le nombre de sommets droits est au plus $| \alpha | + 1$. Or, un sommet normal engendre au plus 2 lettres de $d(T)$ et un sommet droit en engendre au plus 1. Ainsi

$$| d(T) | \leqslant 2 | \alpha | + | \alpha | + 1 = 3 | \alpha | + 1 .$$

PROPOSITION 6 : *Il existe un ensemble fini de mots* $\{ f_i \}_{i \in [1,n]}$ *tels que*

$$\alpha^{-1} L = \bigcup_{i=1}^{n} [f_i]_{\overset{*}{\underset{\mathscr{G}_0}{\leftrightarrow}}}$$

Démonstration : Posons $H = \{ h \in (\hat{Z} \cup V_0)^* \mid |h| \leqslant |\alpha| + 1 \text{ et } h \in \alpha^{-1} L_G(A_0) \}$.

. D'après les lemmes 6 et 7, $L_{G_0}(H) = \alpha^{-1} L$.

. Notons $\{ h_i \}_{i \in [1,n]}$ l'ensemble H. Soit f_i un mot de $L_{G_0}(h_i)$ $L_{G_0}(h_i) \subset [f_i]_{\overset{*}{\underset{G_0}{\leftrightarrow}}}$,

donc, $\alpha^{-1} L \subset \bigcup_{i=1}^{n} [f_i]_{\overset{*}{\underset{\mathscr{G}_0}{\leftrightarrow}}}$ $f_i \in \alpha^{-1} L$ et $\alpha^{-1} L$ est saturé par $\overset{*}{\underset{\mathscr{G}_0}{\leftrightarrow}}$, et donc

$$\bigcup_{i=1}^{n} [f_i]_{\overset{*}{\underset{G_0}{\leftrightarrow}}} \subset \alpha^{-1} L.$$

LEMME 8 : *Soit* X *un alphabet fini*, ϕ *un morphisme* $X^* \to \hat{Z}_n^*$, $\alpha \in X^*$, $K \in Rat(X^*)$ *et* $L \subset \hat{Z}_n^*$.

$$\alpha^{-1}(\phi^{-1}(L) \cap K) = \phi^{-1}((\phi(\alpha))^{-1}L) \cap \alpha^{-1}K$$

La preuve est laissée au lecteur.

PROPOSITION 7 : *Si* $A \subset X^*$ *est dans* $\mathscr{C}(\mathscr{T})$, *le cylindre engendré par les langages de parenthèses, alors pour tout* $\alpha \in X^*$, $\alpha^{-1}A$ *est N.T.S.*

Démonstration : $A \in \mathscr{C}(\mathscr{T}) \Leftrightarrow A = \phi^{-1}(L) \cap K$ (pour ϕ morphisme, $K \in Rat\, X^*$, L langage de parenthèses)

$$\alpha^{-1}A = \phi^{-1}[(\phi(\alpha))^{-1}L] \cap \alpha^{-1}K.$$

D'après la proposition 6, $(\phi(\alpha))^{-1}L = \bigcup_{i=1}^{n} [f_i]_{\overset{*}{\underset{G_0}{\leftrightarrow}}}$.

La proposition 5 s'étend sans difficulté aux unions finies de classes mod $\overset{*}{\underset{\mathscr{G}_0}{\leftrightarrow}}$.

Donc, $\phi^{-1}\left[\bigcup_{i=1}^{n} [f_i]_{\overset{*}{\underset{G_0}{\leftrightarrow}}}\right]$ est N.T.S.

Comme $\alpha^{-1}K \in \mathrm{Rat}\, X^*$, $\alpha^{-1}A$ est N.T.S.

COROLLAIRE : $\# S_<$ *est N.T.S. et* $\# S_< \notin \mathscr{C}(\mathscr{T})$.

Démonstration : $\# S_<$ est engendré par la grammaire N.T.S.

$$G = \langle \{\#, a, b\}, \{\sigma, S\}, P, \{\sigma\} \rangle$$

$$P : \begin{bmatrix} \sigma \to \sigma b + \# S \\ S \to aSb + ab \end{bmatrix}$$

$$\#^{-1}(\# S_<) = \{ a^p b^q \mid 1 \leqslant p \leqslant q \} = S_<$$

Or $S_<$ se décompose en une infinité de classes syntaxiques, donc $S_<$ n'est pas N.T.S.

BIBLIOGRAPHIE

1. J.-M. AUTEBERT, J. BEANQUIER, L. BOASSON et G. Sénizergues · *Remarques sur les langages de parenthèses*. Theoretical Computer Science, vol. 31, 1984, p. 337-349.
2. L. BOASSON, *Grammaires à non-terminaux séparés*, 7e ICALP, Lecture Notes in Computer Science, vol. 85, 1980, p. 105-118.
3. R. V. BOOK, *N.T.S. Grammars and Church-Rosser systems*, Information Processing Letters, vol. 13, 1981, p. 73-76.
4. R. MCNAUGHTON, *Parenthesis grammars*, Journal of the Association for Computing Machinery, vol. 14, 1967, p. 490-500.
5. G. SÉNIZERGUES, *Décidabilité de l'équivalence des grammaires N.T.S.*, Thèse de 3e cycle de l'Université Paris 7, 1981.
6. M. TAKAHASHI, *Generalisations of regular sets and their application to a study of context-free languages*, Information and Control, vol. 27, 1975, p. 1-36.
7. M. TAKAHASHI, *Nets-sets and relativized closure properties*, Theoretical Computer Science, vol. 22, 1983, p. 253-264.