UNIVERSITE de NANCY I

U.E.R. de MATHEMATIQUES

ETUDE ET CLASSIFICATION DES BIGRAMMAIRES APPLICATIONS A L'ETUDE DES SYSTEMES TRANSFORMATIONNELS

THESE DE SPECIALITE EN MATHEMATIQUES APPLIQUEES

PRESENTEE PAR

Pierre MARCHAND

SOUTENUE LE 27 NOVEMBRE 1974

JURY :

Président : Monsieur C. PAIR

Examinateurs : Messieurs M. DEPAIX
M. NIVAT
J.L. OVAERT

Monsieur le Professeur M. DEPAIX a accepté de faire partie du Jury de cette Thèse, qu'il soit assuré de ma profonde et respectueuse gratitude.

Que Monsieur le Professeur M. NIVAT soit remercié de l'honneur qu'il me fait en venant juger ce travail.

Monsieur J.L. OVAERT a bien voulu participer à ce jury, qu'il trouve ici l'expression de mon amicale reconnaissance.

Monsieur le Professeur C. PAIR est à l'origine de cette étude, prodiguant avec une patiente bienveillance ses conseils et ses encouragements il n'a cessé de me faire profiter de sa haute compétence, aussi est-ce un grand plaisir pour moi de le remercier aujourd'hui de tout ce qu'il m'a apporté.

J'adresse mes remerciements à Messieurs FINANCE, LESCANNE, MOHR, QUERE et REMY pour les critiques, conseils et suggestions qu'ils m'ont adressés.

Mes remerciements iront aussi à Madame BIEWER, Mademoiselle GALLIER et Mademoiselle TEDESCO qui ont assuré avec conscience et rapidité l'ingrate réalisation matérielle de ce travail ainsi qu'à Monsieur DEBERDT de l'IREM qui a assuré le tirage.

TABLE DES MATIERES

LEXIQUE

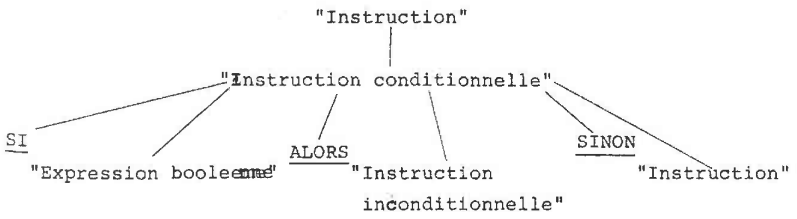
- I N T R O D U C T I O N -

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Le but de ce travail était primitivement de faciliter l'étude de la sémantique des langages de programmation en étudiant des règles d'équivalence permettant de réduire un langage évolué et complexe en un langage noyau.

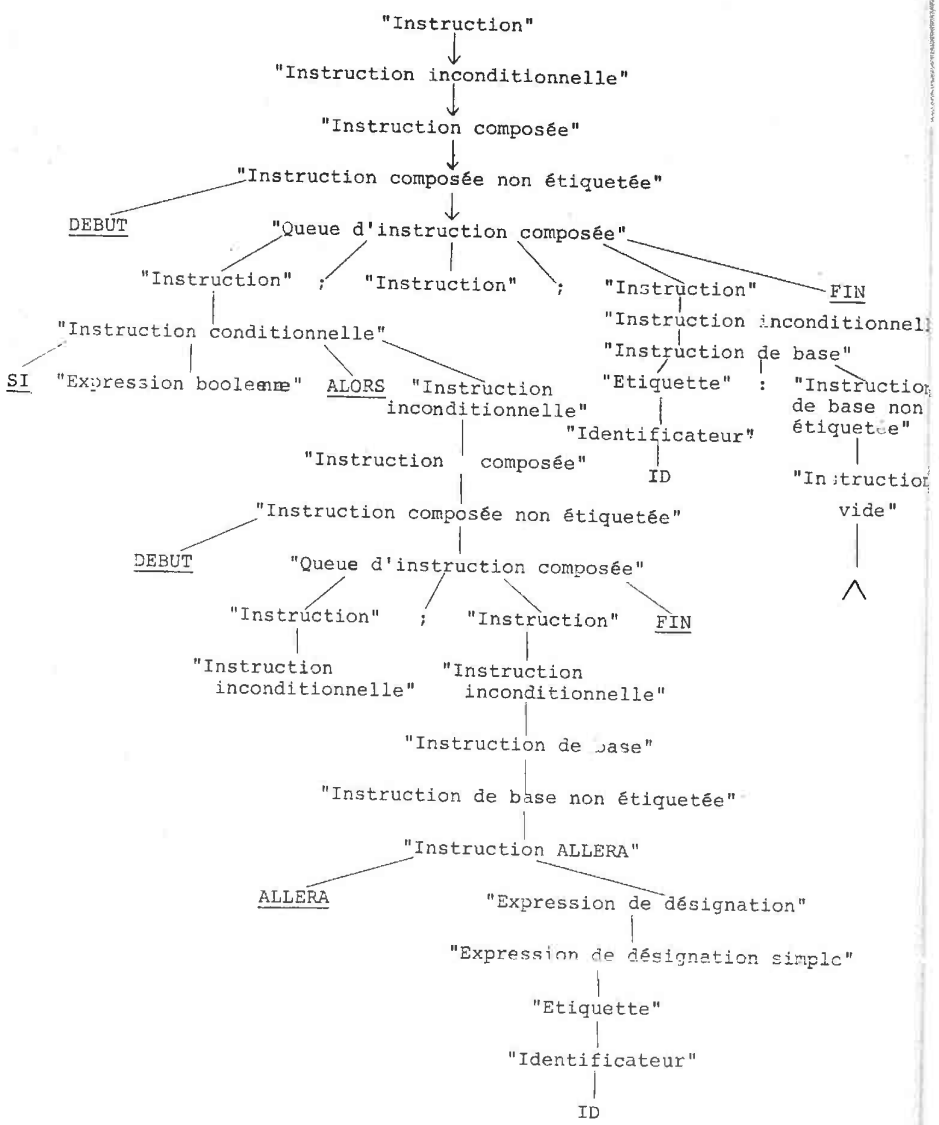
Le problème est donc de traduire un langage dans un autre. Il apparaît rapidement que de tels problèmes sont difficiles si on travaille sur des langages "linéaires" (c'est-à-dire des ensembles de mots sur un certain alphabet). Mais les langages de programmation sont en général décrits en première approximation par des grammaires algébriques et, à chaque mot du langage, est associé un ou plusieurs arbres syntaxiques (ici appelés ramifications). Si on essaie de traduire un langage dans un autre en raisonnant, non plus sur les mots du langage, mais sur les arbres syntaxiques, les problèmes se simplifient notablement. Les règles d'équivalence seront donc des transformations sur des arbres (ou ramifications).

Le problème est maintenant de définir correctement ces transformations. Considérons l'exemple suivant où l'on veut supprimer dans Algal 60, les instructions du type : si... alors... sinon... Chaque fois que l'on rencontrera dans un arbre syntaxique un sous-arbre de la forme :



.../...

On remplacera ce sous-arbre par :



De plus, on greffera en dessous de "expression booleenne", "instruction inconditionnelle" et "instruction" ce qui figurait dans l'arbre syntaxique initial.

.../...

(Sur le langage linéaire, cette transformation s'écrirait : chaque fois que l'on rencontre un sous-mot de la forme

SI α ALORS β SINON γ

où α,β,γ dérivent respectivement de "expression booleenne", "instruction inconditionnelle" et "instruction".
On le remplace par

DEBUT SI α ALORS DEBUT β ; ALLERA ID FIN ; γ ; ID : FIN

Cette façon d'écrire la transformation semble plus simple ; mais, en fait, quand on lira le programme, α par exemple pourra être une expression très longue et même si on fait une lecture à l'avance de k caractères (k fixé), on ne pourra pas savoir dans tous les cas si cette transformation est ou non applicable. En revanche, sur l'arbre syntaxique, cette transformation est purement locale).

A propos de cet exemple, on peut faire plusieurs remarques :

1°) Bien que l'exemple choisi soit assez simple, on remarque que l'écriture de la transformation est longue et pénible. Pour pouvoir énoncer des résultats sur des systèmes utilisant de telles transformations, il est donc nécessaire de formaliser cette notion de transformation.

2°) On a dit que l'on greffait en dessous des non terminaux "expression booleenne", "instruction inconditionnelle" et "instruction" ce qui figurait dans l'arbre initial ; mais si le non terminal "instruction" intervient plus d'une fois, il y a des ambiguïtés dans la reconstruction de l'arbre après l'application d'une transformation. Il faudra donc introduire des marqueurs qui lèvent cette ambiguïté. Nous utiliserons ici les entiers 1,2,... et une transformation sera décrite par un couple d'arbres que l'on peut représenter schématiquement ainsi :

.../...

On utilisera alors cette transformation de la façon suivante : chaque fois que l'on rencontre dans l'arbre t à traduire un sous-arbre égal à r on le remplace par s et ce qui dans t était en dessous de r à la place i_k est greffé en dessous de s à la place j_q telle que $j_q = i_k$. Des arbres de ce type seront utilisés tout au long de cette thèse sous le nom de polynômes.

3°) Il est bien évident qu'il ne suffit pas d'utiliser une fois une transformation pour obtenir la traduction d'un arbre t. Il faut utiliser plusieurs transformations et éventuellement les réitérer. (Par exemple si dans ALGOL 60 on désire supprimer les instructions pour, on construira des transformations qui permettent de diminuer le nombre d'éléments d'une liste de pour et on réitérera cette transformation jusqu'à ce que les listes de pour du programme soient de longueur 1 ; puis une autre transformation permettra de traduire les instructions pour avec un seul élément dans la liste de pour.) Pour traduire un arbre t on applique donc une suite de transformations et on obtient une suite d'arbres t,t1,t2,.... Cette suite d'arbres est analogue à une dérivation dans une grammaire. Nous avons donc étudié sous le nom de bigrammaires des grammaires sur les arbres dont les dérivations correspondent aux suites d'arbres décrites ci-dessus. Cette étude fait l'objet des quatre premiers chapitres de cette thèse, le cinquième étant l'application des résultats obtenus aux problèmes de traduction proprement dits.

4°) On peut faire aussi quelques remarques sur la forme des transformations.

- En général, on utilisera de telles transformations pour traduire un langage complexe en un langage plus simple dont les concepts sont moins élaborés. Donc quand on traduira un mot α on obtiendra un mot α' plus long que α et, pour les transformations, cela signifie que le deuxième arbre utilisé pour décrire une transformation sera "plus grand" que le premier. Nous étudierons cette situation au chapitre 4 à propos des bigrammaires contextuelles et strictement contextuelles.

- On peut envisager de deux façons la traduction d'un langage. Soit les arbres que l'on obtient successivement en appliquant les transformations, gardent tous une "signification" ; soit seuls l'arbre de départ et la traduction de cet arbre ont une "signification", les autres arbres de la suite menant de l'arbre de départ à sa traduction n'étant que des intermédiaires de calcul. Pour pouvoir procéder

.../...

de la première façon, il faut imposer des conditions supplémentaires aux transformations. Ici nous avons traité ce cas en imaginant qu'il existait un sur-langage englobant le langage à traduire et le langage noyau et que les transformations ne font pas sortir de ce sur-langage. Nous étudions de tels systèmes de traduction au chapitre 5 sous le nom de systèmes transformationnels contrôlés et, pour pouvoir démontrer des résultats à propos de ces systèmes transformationnels contrôlés, nous étudions au chapitre 4 des bigrammaires contextuelles et strictement contextuelles particulières dont les productions sont analogues aux transformations de ces systèmes (bigrammaires contextuelles et strictement contextuelles conservant la racine et utilisant dans leurs productions des polynômes injectifs).

Après avoir développé les quelques idées qui sont sousjacentes à ce travail, nous pouvons maintenant indiquer un plan commenté et un résumé des résultats obtenus.

Le chapitre 0 est consacré à des rappels sur les notions de langage, grammaire, bilangage et treillis.

Dans le chapitre 1, on introduit la notion de polynôme qui est fondamentale pour la suite. Ces polynômes serviront à la fois pour décrire les productions d'une bigrammaire et les transformations des systèmes transformationnels.

Dans le chapitre 2, nous donnons les définitions concernant les bigrammaires et des propositions élémentaires et générales à propos de l'utilisation de ces bigrammaires.

Le chapitre 3 contient diverses généralisations aux bigrammaires de la notion de grammaire algébrique. Nous introduisons trois types de bigrammaires : les bigrammaires à contexte libre de degré zéro, les bigrammaires régulières et les bigrammaires à contexte libre de degré un. Dans chacun de ces cas on étudie les bilangages engendrés et les différents langages associés à ces bilangages (langages des mots des racines, langages des mots des feuilles, langages des chemins). Pour ces trois types de bigrammaires, on montre l'existence de systèmes à point fixe associés dont les solutions minimales coïncident avec les bilangages engendrés par les bigrammaires. Dans le cas des bigrammaires régulières, on montre que les bilangages engendrés sont les bilangages réguliers déjà

.../..

étudiés par d'autres méthodes (QUERE {20}). Une nouvelle caractérisation algébrique de ces bilangages réguliers permet de montrer que l'intersection d'un bilangage régulier et d'un bilangage engendré par une bigrammaire à contexte libre est encore engendré par une bigrammaire à contexte libre. Une généralisation aux bilangages engendrés par les bigrammaires à contexte libre de degré un du théorème des paires itérantes donne une idée précise de la puissance de ces bigrammaires et permet en particulier de montrer que ces bigrammaires ne sont pas aussi puissantes que les grammaires contextuelles.

Le chapitre 4 aborde l'étude des bigrammaires contextuelles et strictement contextuelles. Ces bigrammaires sont une généralisation des grammaires contextuelles. Cette étude est surtout faite en vue de son application au chapitre 5. C'est pourquoi on étudie les langages engendrés par ces grammaires en imposant des conditions restrictives aux productions de façon qu'elles puissent être utilisées dans les systèmes transformationnels contrôlés. Malgré cela on montre que les langages engendrés gardent un grand caractère de généralité. En particulier pour les bigrammaires contextuelles on montre que les langages engendrés avec ces restrictions sont tous les langages récursivement énumérables bien que les bilangages engendrés par de telles grammaires restent décidables.

Dans le chapitre 5, on définit le concept de système transformationnel et on applique les résultats obtenus dans les paragraphes précédents. En particulier on utilise le résultat concernant l'intersection d'un bilangage régulier et d'un bilangage engendré par une grammaire à contexte libre, pour démontrer des résultats de décidabilité à propos des systèmes transformationnels dont les transformations sont du même type que les productions des bigrammaires à contexte libre. Les résultats du chapitre 4 nous permettent de démontrer que les problèmes de traduction sont indécidables même si on impose des conditions assez fortes sur les transformations. La fin du chapitre est consacrée à l'introduction de systèmes transformationnels particuliers où l'on peut lever l'indécidabilité des problèmes posés.

CHAPITRE 0 : RAPPELS

0.1.- Langage :

0.1.1.- Définitions.

- Un vocabulaire est un ensemble fini.
- Un mot sur le vocabulaire V est une suite finie d'éléments de V . Si α est un mot sur V , $|\alpha|$ désigne la longueur du mot α .
- L'ensemble des mots sur V est noté V^* . V^* a une structure de monoïde quand on le munit de la loi de composition interne "concaténation". L'élément neutre est le mot vide noté Λ .
- Un langage sur V est un sous-ensemble de V^* .

0.1.2.- Grammaires.

- Une grammaire G est un quadruplet $G = (N,T,P,X)$; N et T sont deux vocabulaires disjoints appelés respectivement vocabulaire auxiliaire et vocabulaire terminal, P est une relation binaire sur $(N \cup T)^*$ et X est un élément de N . Un couple (α,β) en relation modulo P est appelé une production de G . On écrit alors $\alpha \xrightarrow[G]{} \beta$. On impose en général que les productions d'une grammaire soient en nombre fini.
- On écrit $u \xrightarrow[G]{} v$ (u se réécrit v dans G) si $u = u_1 \alpha u_2$, $v = u_1 \beta u_2$ et $\alpha \xrightarrow[G]{} \beta$. La relation $\xrightarrow[G]{*}$ lue de ... dérive ... est la fermeture reflexive et transitive de $\xrightarrow[G]{}$.
- Le langage engendré par G est par définition l'ensemble des mots de T^* dérivant de X .

0.1.3.- Classification des grammaires.

- Les grammaires les plus générales sont appelées semi-thuéiennes. Elles engendrent exactement la classe des langages récursivement énumérables.
- Les grammaires G vérifiant la condition :

$$\alpha \xrightarrow[G]{} \beta \implies \alpha \neq \Lambda \text{ et } |\beta| \geqslant |\alpha|$$

sont appelées grammaires contextuelles. Elles engendrent la classe des

langages contextuels.

- Les grammaires G vérifiant la condition :

$$\alpha \xrightarrow[G]{} \beta \implies \alpha \in N$$

sont appelées grammaires algébriques. Elles engendrent la classe des langages algébriques.

- Les grammaires G vérifiant la condition

$$\alpha \xrightarrow[G]{} \beta \implies \alpha \in N \text{ et } \beta \in T \cup \{\Lambda\}$$

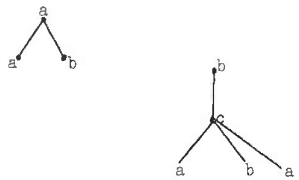
sont appelées grammaires linéaires gauches. Elles engendrent la classe des langages réguliers qui sont aussi les langages reconnus par les automates finis. On définit aussi les grammaires linéaires droites qui engendrent le même type de langages.

0.2.- Ramification :

0.2.1.- Définitions.

On définit intuitivement un arbre sur V comme un graphe sans circuit étiqueté par des éléments de V tel que le demi-degré intérieur de chaque point soit au plus un et tel que les descendants directs d'un point soient totalement ordonnés, ainsi que les points de demi-degré intérieur nul.

Exemple : V = {a,b,c}

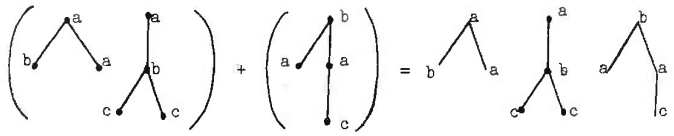


La suite donne une formatisation algébrique de cette notion d'arbre.

Dans l'ensemble des arbres sur V , on peut introduire deux lois.

- Une loi interne qui consiste à justaposer deux arbres et qui correspond à la concaténation dans le monoïde libre V* . On notera + cette loi.

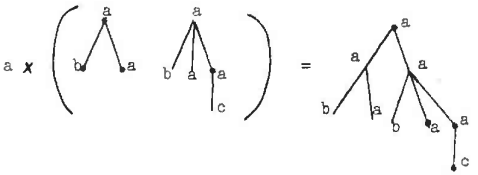
Exemple :



Cette loi est évidemment associative et possède un élément neutre noté Λ qui est l'arbre vide.

- Une loi externe à opérateurs dans V qui consiste à introduire un nouveau point et à relier ce point à tous les points de l'arbre de demi-degré intérieur nul. On notera $\times$ cette loi externe.

Exemple :



Un ensemble B muni de deux telles lois (l'une interne, notée + , associative et possédant un élément neutre et l'autre externe, à opérateur dans V et notée $\times$) s'appelle un V-binoïde.

Etant donnés deux binoïdes B et B' , on appelle homomorphisme de binoïde de B dans B' une application f : B ---> B' telle que

- $f(t+t') = f(t) + f(t')$
- $f(a \times t) = a \times f(t)$
- $f(e) = e'$ (e et e' étant les éléments de B et B' pour la loi +) .

Les V-binoïdes forment une catégorie et dans un article de Pair [16], on montre que cette catégorie possède un objet initial $\hat{V}$ appelé V-binoïde universel et défini à un isomorphisme près par les conditions :

- $\hat{V}$ est un V-binoïde (on notera Λ l'élément neutre de +)
- tout élément r de $\hat{V}$ différent de Λ s'écrit de manière unique $r = (a \times r') + r''$ (a étant un élément de V , r' et r'' des éléments de $\hat{V}$)
- il existe une fonction $\gamma : \hat{V} \longrightarrow \mathbb{N}$ telle que

$$\mathcal{D}(\Lambda) = 0$$

$$\mathcal{D}(a \times r) > \mathcal{D}(r)$$

$$r \neq \Lambda \Longrightarrow \mathcal{D}(r+s) > \mathcal{D}(s)$$

$$s \neq \Lambda \Longrightarrow \mathcal{D}(r+s) > \mathcal{D}(r) \ .$$

On appelle ramification sur V les éléments de $\hat{V}$.

Les arbres sur V définis ci-dessus donnent une réalisation concrète des éléments de $\hat{V}$. En effet, les deux lois définies sur l'ensemble des forêts sur V donnent à cet ensemble une structure de V-binoïde. La fonction "nombre de points du graphe sous-jacent" peut jouer le rôle de la fonction $\mathcal{D}$ ci-dessus et la décomposition d'une forêt t non vide en $t = a \times t' + t''$ est évidente.

Une autre réalisation concrète de $\hat{V}$ est par exemple le langage de Dyck P [6] construit sur $V \cup \bar{V}$ ($\bar{V}$ est un ensemble en bijection avec V par la bijection $a \longmapsto \bar{a}$ et disjoint de V). Pour donner à P une structure de V-binoïde universel, on utilise les deux lois

$$\alpha + \beta = \alpha\beta \quad \text{(concaténation dans } (V \cup \bar{V})^*)$$

$$a \times \alpha = a\, \alpha\, \bar{a} \ .$$

Il est alors facile de montrer que P vérifie les conditions ci-dessus en posant $\mathcal{D}(\alpha) = |\alpha|$ (longueur du mot α).

Le mode principal de raisonnement et de construction dans $\hat{V}$ est la récurrence obtenue grâce aux deux schémas de théorèmes suivants :

(1) $[P(\Lambda)$ et $(\forall r \in \hat{V})\ (\forall s \in \hat{V})\ (\forall a \in V)\ (P(r)$ et $P(s) \Longrightarrow P(a \times r+s))] \Longrightarrow (\forall r \in \hat{V})(P(r))$

P étant une formule quelconque avec ou sans paramètre.

(2) Soient $f : \hat{V}^n \longrightarrow E$ et $g : V \times E \times E \times \hat{V}^{n+2} \longrightarrow E$ deux applications ; il existe une et une seule application $\psi : \hat{V}^{n+1} \longrightarrow E$ telle que

$$\psi(\Lambda, r_1, \ldots, r_n) = f(r_1, \ldots, r_n)$$

$$\psi(a \times r+s, r_1, r_2, \ldots, r_n) = g(a, \psi(r, r_1, \ldots, r_n), \psi(s, r_1, \ldots, r_n), r, s, r_1, \ldots, r_n)$$

cf. Quéré [20].

Définition.- Un bilangage sur V est un sous-ensemble de $\hat{V}$. Dans les travaux de Quéré [20] et de Berlioux [1] on généralise à $\hat{V}$ les notions de langage régulier

et algébrique. Nous donnerons une généralisation à $\hat{V}$ de la notion de grammaire, une classification des grammaires et les propriétés principales des langages et des bilangages engendrés par ces grammaires.

0.2.2.- Fonctions usuelles sur $\hat{V}$.

* Mot des racines : $\rho : \hat{V} \longrightarrow V^*$ définie par $\rho(\Lambda) = \Lambda$ et $\rho(a \times r+s) = a\, \rho(s)$

* Mot des feuilles : $\varphi : \hat{V} \longrightarrow V^*$ définie par $\varphi(\Lambda) = \Lambda$ $\varphi(a \times r+s) =$ si $r = \Lambda$ alors $a\varphi(s)$ sinon $\varphi(r)\ \varphi(s)$

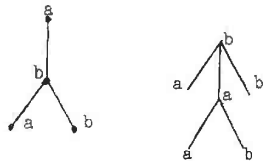
* Familles de prédécesseur a : $F_a : \hat{V} \longrightarrow \mathcal{P}(V^*)$ définie par

$F_a(\Lambda) = \emptyset$ et $F_a(b \times r+s) =$ si $b = a$ alors $\{\rho(r)\} \cup F_a(r) \cup F_a(s)$ sinon $F_a(r) \cup F_a(s)$

* Nombre de points : $n : \hat{V} \longrightarrow \mathbb{N}$ définie par

$$n(\Lambda) = 0 \quad \text{et} \quad n(a \times r+s) = 1 + n(r) + n(s) \ .$$

Exemple 1 : Soit r la ramification



On a $\rho(r) = ab$, $\varphi(r) = abaabb$, $F_a(r) = \{b, \Lambda, ab\}$, $n(r) = 10$.

* Plongement de V^* dans $\hat{V}$: quand il n'y a pas d'ambiguïté, on identifiera le mot de V^* $a_1\, a_2 \ldots a_n$ et la ramification $a_1 \times \Lambda + a_2 \times \Lambda + \ldots + a_n \times \Lambda$.

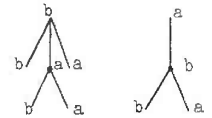
En particulier on identifie a et $a \times \Lambda$.

* Ramification réfléchie : $\sim : \hat{V} \longrightarrow \hat{V}$ définie par

$$\tilde{\Lambda} = \Lambda \quad \text{et} \quad \widetilde{a \times r + s} = \tilde{s} + a \times \tilde{r}$$

($\sim$ donne l'image dans un miroir de r. Par exemple, pour la ramification r de l'exemple 1, on a

$\tilde{r} =$



* Chemins : $ch : \hat{V} \longrightarrow \mathcal{P}_f(V^*)$ définie par

$ch(\Lambda) = \emptyset$ et $ch(axr+s) =$ si $r \neq \Lambda$ alors $a\ ch(r) \cup ch(s)$ sinon $\{a\} \cup ch(s)$

(ch donne les mots lus sur un chemin de la ramification allant d'une racine à une feuille. Par exemple pour la ramification r de l'exemple 1, on a :

$$ch(r) = \{aba\ ,\ abb\ ,\ ba\ ,\ baa\ ,\ bab\ ,\ bb\}.$$

* Hauteur : $h : \hat{V} \longrightarrow \mathbb{N}$ définie par

$$h(\Lambda) = 0 \qquad h(axr+s) = \sup(h(r)+1, h(s))$$

on a la relation $h(r) = \sup(|\alpha| ; \alpha \in ch(r))$.

0.2.3.- Bilangage engendré par une grammaire.

Définition 0.2.3.1.- Soit $G = (N,T,::=,X)$ une grammaire algébrique. On dit qu'une ramification r est engendrée au sens large par G si elle vérifie la condition

$$(\forall \alpha \in (N \cup T)^*)\ (\forall A \in N \cup T)\ (\alpha \in F_A(r) \Longrightarrow A ::= \alpha \text{ ou } \alpha = \Lambda)\ .$$

On dit qu'une ramification r est engendrée par G si elle vérifie les conditions

- $(\forall \alpha \in (N \cup T)^*)\ (\forall A \in N \cup T)\ (\alpha \in F_A(r) \Longrightarrow A ::= \alpha$ ou $(A \in T$ et' $\alpha = \Lambda))$
- $\rho(r) = X$.

On notera $B(G)$ le bilangage engendré par G .

(Comme nous le montrerons dans la suite, une grammaire est un cas particulier de bigrammaire et considérée comme telle, elle engendre un bilangage que l'on notera $BL(G)$. Ces deux bilangages sont distincts et ne doivent pas être confondus. Dans la suite, quand nous parlerons du bilangage engendré par une grammaire algébrique sans autres précisions, il s'agira du langage $B(G)$ et non de $BL(G)$.

Proposition 0.2.3.2.- Les mots des feuilles des ramifications de $B(G)$ sont exactement les mots engendrés par la grammaire G . En effet, les éléments de $B(G)$ ne sont pas autre chose que les arbres syntaxiques associés aux différents mots engendrés par G .

0.3.- Treillis :

Nous aurons besoin dans la suite du théorème du point fixe dans les treillis.

Rappelons que :

Un treillis est un ensemble muni d'une relation d'ordre $\leq$ telle que toute paire $\{a,b\}$ admet une borne supérieure et une borne inférieure notées respectivement $a \vee b$ et $a \wedge b$.

Un treillis T est dit complet si tout sous-ensemble X de T admet une borne supérieure et une borne inférieure notées respectivement $\bigvee_{x \in X} x$ et $\bigwedge_{x \in X} x$.

Une fonction $f : T \longrightarrow T'$ où T et T' sont deux treillis complets est dite continue si pour tout sous-ensemble X non vide de T , on a

$$f\left(\bigvee_{x \in X} x\right) = \bigvee_{x \in X} f(x)\ .$$

Théorème du point fixe [23].- Si $f : T \longrightarrow T$ est continue, f admet un plus petit point fixe $\mu(f)$ et on a

$$\mu(f) = \bigvee_{n=0}^{\infty} f^n(0) \quad \text{où } 0 \text{ est le plus petit élément de } T\ .$$

(On peut trouver dans la littérature des énoncés beaucoup plus forts. Nous n'aurons besoin ici que du théorème assez faible ci-dessus).

Exemple : Si E est un ensemble quelconque, $\mathcal{P}(E)$ est un treillis complet pour la relation d'inclusion. Plus généralement, $(\mathcal{P}(E))^n$ est un treillis complet pour la relation

$$(X_1,X_2,\ldots,X_n) \leq (X'_1,X'_2,\ldots,X'_n) \Longleftrightarrow \forall i \in [1,n]\ (X_i \subset X'_i)\ .$$

Nous appliquerons le théorème du point fixe pour résoudre des systèmes sur $(\mathcal{P}(E))^n$ de la forme

$$X_i = \bigcup_{j=1}^{P_i} G_{ij}(X_1,X_2,\ldots,X_n) \qquad 1 \leq i \leq n$$

où G_{ij} est obtenu par composition à partir d'un certain nombre de fonctions de base $f_\ell^k : (\mathcal{P}(E))^{n_\ell} \longrightarrow \mathcal{P}(E)$ $(1 \leq \ell \leq k)$ que l'on sait être continues. L'application de $(\mathcal{P}(E))^n$ dans $(\mathcal{P}(E))^n$ qui à $(X_1,X_2,\ldots,X_n)$ associe $\left(\bigcup_{j=1}^{P_1} G_{1j}(X_1,\ldots,X_n),\ldots,\bigcup_{j=1}^{P_n} G_{nj}(X_1,\ldots,X_n)\right)$ est continue et admet donc un plus petit point fixe qui est la solution minimale du système. D'autre part, en raisonnant par récurrence sur la complexité des fonctions G_{ij} (c'est-à-dire sur le nombre de fonctions de base utilisées pour construire G_{ij}) , il est facile de montrer qu'il existe un système S' de la forme

$$S' : X_i = \bigcup_{j=1}^{P'_i} G'_{ij}(X_1,\ldots,X_m) \qquad 1 \leq i \leq m$$

et vérifiant les conditions suivantes :

- G'_{ij} est la composée d'une fonction de base et de fonctions projections
- $n \leq m$
- pour toute solution $(A_1,\ldots,A_m)$ de S' , $(A_1,\ldots,A_n)$ est une solution de S
- pour toute solution $(A_1,\ldots,A_n)$ de S , il existe une solution $(A_1,\ldots,A_n,A_{n+1},\ldots,A_n)$ de S'
- on en déduit que les solutions minimales de S et S' coïncident sur les n premières composantes. Par abus de langage, on dira que S et S' sont équivalents, cf. Mohr [14].

Transfert de solutions de système à point fixe.- Soient T_1 et T_2 deux treillis complets et soient $h : T_1 \dashrightarrow T_2$, $f : T_1 \dashrightarrow T_1$, $g : T_2 \dashrightarrow T_2$ trois applications continues telles que le diagramme suivant soit commutatif

$$\begin{array}{ccc} T_1 & \xrightarrow{f} & T_1 \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle h} \\ T_2 & \xrightarrow{g} & T_2 \end{array}$$

On a alors la liaison suivante entre les équations à point fixe

(1) $x = f(x)$ dans T_1 et (2) $y = g(y)$ dans T_2 :

x est solution de (1) $\Longrightarrow$ $h(x)$ est solution de (2).

Si de plus $h(\bot_1) = \bot_2$ où $\bot_1$ et $\bot_2$ désignent les éléments minimaux de T_1 et T_2 , alors on a

x est la solution minimale de (1) $\Longrightarrow$ $h(x)$ est la solution minimale de (2).

Démonstration.-

a) On a $g(h(x)) = g \circ h(x) = h \circ f(x) = h(f(x)) = h(x)$,

donc $h(x)$ est bien un point fixe de g .

b) Comme f et g sont des fonctions continues, leurs plus petits points fixes sont respectivement

$$\mu(f) = \bigvee_{n=0}^{\infty} f^n(\bot_1)$$

$$\mu(g) = \bigvee_{n=0}^{\infty} g^n(\bot_2) \ .$$

Comme h est continue et vérifie $h(\bot_1) = \bot_2$, on obtient

$$h(\mu(f)) = h(\bigvee_{n=0}^{\infty} f^n(\bot_1)) = \bigvee_{n=0}^{\infty} h(f^n(\bot_1)) = \bigvee_{n=0}^{\infty} g^n(h(\bot_1)) = \bigvee_{n=0}^{\infty} g^n(\bot_2) = \mu(g) \ .$$

CHAPITRE 1 : POLYNOMES

1.1.- V-binoïde libre. Polynômes sur V :

Nous allons définir dans ce paragraphe les outils pour construire les règles des grammaires sur les ramifications.

Définition 1.1.1.- Soit B un V-binoïde. On appelle sous-binoïde de B un ensemble B' contenant l'élément neutre de B et stable pour les deux lois définies sur B .

Proposition 1.1.2.- Toute intersection de sous-binoïde B est un sous-binoïde B . Pour tout sous-ensemble A d'un binoïde B , il existe un plus petit sous-binoïde de B contenant A .

Définition 1.1.3.- Un V-binoïde B est dit libre de base T si T est contenue dans B et si toute application de T dans un V-binoïde quelconque B' se prolonge de manière unique en un homorphisme de binoïde de B dans B' .

Proposition 1.1.4.- Deux V-binoïdes libres de base T sont canoniquement isomorphes. Dans le cas où V et T sont disjoints, le V-binoïde $\hat{V}(T)$ engendré par T dans le V-binoïde $\widehat{V \cup T}$ est libre sur T . Les éléments de $\hat{V}(T)$ sont exactement les éléments de $\widehat{V \cup T}$ où les éléments de T n'ont d'occurrence qu'aux feuilles.

Démonstration.-

a) Soient B_1 et B_2 deux V-binoïdes libres sur T . L'application Id : T ---> T se prolonge en deux homomorphismes uniques $f_1 : B_1 \dashrightarrow B_2$ et $f_2 : B_2 \dashrightarrow B_1$ qui sont inverses l'un de l'autre.

En effet, $f_2 \circ f_1 : B_1 \dashrightarrow B_1$ est homomorphisme de V-binoïde qui prolonge l'application Id : T ---> T . Or $Id_{B_1} : B_1 \dashrightarrow B_1$ vérifie aussi cette condition. Donc, en vertu de l'unicité d'un tel prolongement, on a $f_2 \circ f_1 = Id_{B_1}$. De même, $f_1 \circ f_2 = Id_{B_2}$.

b) Montrons déjà que $\hat{V}(T)$ est formé des ramifications de $\widehat{V \cup T}$ telles que les éléments de T n'ont d'occurrence qu'aux feuilles : c'est-à-dire qu'il faut montrer que l'ensemble des r de $\widehat{V \cup T}$ vérifiant

$$(\forall a \in T)\ (F_a(r) = \{\Lambda\} \text{ ou } F_a(r) = \emptyset) \qquad (1)$$

est un V-binoïde et que c'est le plus petit V-binoïde contenant T .

De façon évidente, si r et r' vérifient (1), r + r' vérifie aussi (1) et pour tout b de V , b x r vérifie (1) car V et T sont disjoints. De plus, Λ vérifie (1). Donc l'ensemble des r de $\widehat{V \cup T}$ vérifiant (1) est bien un B-binoïde et il contient T . Il suffit alors de montrer que tout V-binoïde B de $\widehat{V \cup T}$ contenant T contient les r de V∪T vérifiant (1). La démonstration se fait par récurrence sur r

$r = \Lambda$ vérifie (1) et on a bien $\Lambda \in B$.

Supposons r vérifie (1) $\Longrightarrow r \in B$

r' vérifie (1) $\Longrightarrow r' \in B$.

Si a x r + r' vérifie (1), on a

- soit $a \in V$ et donc a x r + r' ∈ B , car B est un V-binoïde,
- soit $a \in T$, donc $r = \Lambda$ et donc a + r' ∈ B , car T est contenu dans B .

Donc $\hat{V}(T) = \{r; r \in \widehat{V \cup T}$ et $(\forall a \in T)\ (F_a(r) = \{\Lambda\}$ ou $F_a(r) = \emptyset)\}$.

Montrons maintenant que $\hat{V}(T)$ est bien un V-binoïde libre sur T .

c) Soient B un V-binoïde quelconque et $f : T \dashrightarrow B$ une application quelconque de T dans B . Montrons que f se prolonge de manière unique en un homomorphisme $\tilde{f} : \hat{V}(T) \dashrightarrow B$. On doit avoir nécessairement

$$\tilde{f}(\Lambda) = e \quad \text{(élément neutre de B)}$$

$$(\forall a \in V)\ (\forall r \in \hat{V}(T))\ (\forall r' \in \hat{V}(T))\ (\tilde{f}(a x r + r') = a \times \tilde{f}(r) + \tilde{f}(r')$$

$$(\forall a \in T)\ (\forall r \in \hat{V}(T))\ (\tilde{f}(a + r) = f(a) + \tilde{f}(r)) ,$$

d'où l'unicité de $\tilde{f}$. Il faut vérifier maintenant que $\tilde{f}$ ainsi construit est un homomorphisme de V-binoïde.

$\tilde{f}(\Lambda) = e$ et $\tilde{f}(a x r) = a \times \tilde{f}(r)$ pour $a \in V$ se vérifient immédiatement.

$r \in \hat{V}(T)$ et $r' \in \hat{V}(T) \Longrightarrow \tilde{f}(r + r') = \tilde{f}(r) + \tilde{f}(r')$ se démontrent immédiatement par récurrence sur r .

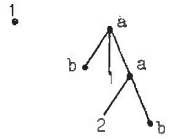
Définition 1.1.5.- Soit V un ensemble disjoint de $\mathbb{N}' = \mathbb{N} - \{0\}$ ($\mathbb{N}$ est l'ensemble des entiers naturels). On appelle polynôme sur V un élément de $\hat{V}(\mathbb{N}')$.

D'après la proposition 1.2.4, un polynôme sur V est une ramification $r \in \widehat{V \cup \mathbb{N}'}$ et vérifiant

$$(\forall i \in \mathbb{N}')\ (F_i(r) = \{\Lambda\} \text{ ou } F_i(r) = \emptyset) .$$

Exemples :

a x (1+2)



1 + a x (b+1+a x (2+b)) .

Remarque : On a de façon évidente $\hat{V}([1,n]) \subset \hat{V}([1,n+1])$ et

$$\hat{V}(\mathbb{N}') = \bigcup_{n=0}^{\infty} \hat{V}([1,n]) .$$

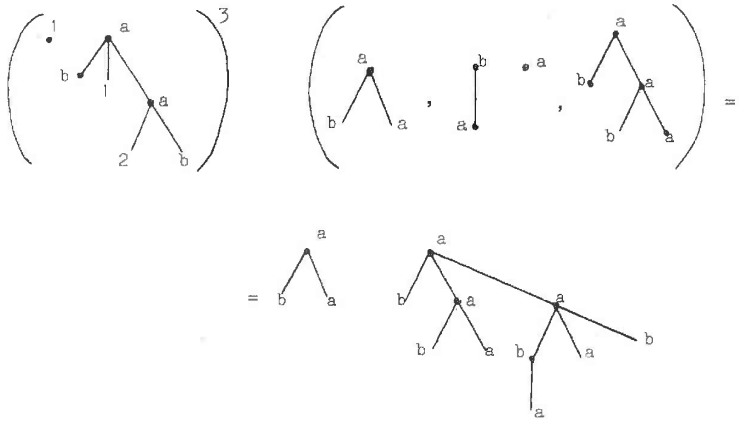
On notera $\hat{V}_n$ l'ensemble $\hat{V}([1,n])$.

Définition 1.1.6.- Pour chaque V-binoïde B , on associe à chaque polynome r de $\hat{V}_n$ une fonction notée r_B^n de B^n dans B de la façon suivante :

Soient $r \in \hat{V}_n$ et b_1 , b_2 , ... , b_n des éléments de B . On considère l'application $f : [1,n] \dashrightarrow B$ définie par $f(i) = b_i$. D'après la proposition 1.1.4, f se prolonge de manière unique en un homomorphisme $\tilde{f} : \hat{V}_n \longrightarrow B$. Par définition, $r_B^n(b_1, b_2, \ldots, b_n) = \tilde{f}(r)$.

Dans le cas où $B = \hat{V}$, on notera r^n au lieu de $r^n_{\hat{V}}$ la fonction de $\hat{V}^n$ dans $\hat{V}$ associé à r .

Exemple :



c'est-à-dire $r_1 + a \times (b+r_1+a\times(r_2+b))$ avec $r_1 = a \times (b+a)$ et $r_2 = b \times a + a$.

A cause de cette proposition, on dira que $r \in \hat{V}_n$ est un polynôme à n variables.

Remarque : En faisant la même démarche que ci-dessus, si on considère le V-binoïde $\hat{V}[T]$ libre de base T et un V-binoïde B quelconque, on peut associer à chaque r de $\hat{V}[T]$ une application de B^T dans B . D'autre part, si on numérote les éléments de T de 1 à n à chaque r de $\hat{V}[T]$, on peut associer une application de B^n dans B . Cette remarque sera utilisée au chapitre 3, proposition 3.1.1.3.

Définition 1.1.7.- Soit $r \in \hat{V}(\mathbb{N}')$. On dit que i est une variable efficace dans r si $F_i(r) = \{\Lambda\}$. (C'est-à-dire que i a une occurrence dans r) . On appelle degré en i la fonction $d_i : \hat{V}(\mathbb{N}') \longrightarrow \mathbb{N}$ définie par

$$d_i(\Lambda) = 0$$

$$d_i(a\times r+s) = \text{si } a = i \text{ alors } 1 + d_i(r) + d_i(s) \text{ sinon } d_i(r) + d_i(s)$$

(d_i donne le nombre d'occurrences de i dans r) .

On appelle degré la fonction $d : \hat{V}(\mathbb{N}') \longrightarrow \mathbb{N}$ définie par

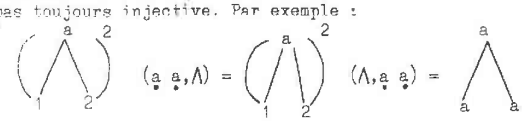
$$d(r) = \sup_{i\in\mathbb{N}'}(d_i(r)) .$$

On appelle corps d'un polynôme $r \in \hat{V}(\mathbb{N}')$ l'image de r par l'homomorphisme $c : \hat{V}(\mathbb{N}') \longrightarrow \hat{V}$ définie par

$$(\forall i \in \mathbb{N}')\ (c(i)=\Lambda)$$

(c(r) est obtenu en supprimant dans r les variables i de $\mathbb{N}'$) .

Remarque : La fonction de $\hat{V}^n$ dans $\hat{V}$ associée à un polynôme r de $\hat{V}_n$ n'est pas toujours injective. Par exemple :



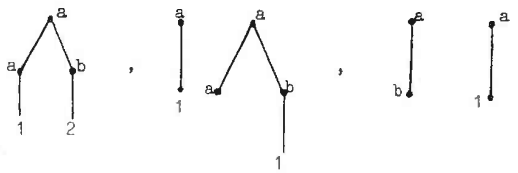
c'est-à-dire $(a\times(1+2))^2\ (a+a,\Lambda) = (a\times(1+2))^2\ (\Lambda,a+a) = a \times (a+a)$.

Dans certaines applications, il est utile de pouvoir inverser les applications associées aux polynômes. Pour cela, on introduit une classe particulière de polynômes : les polynômes injectifs définis comme suit

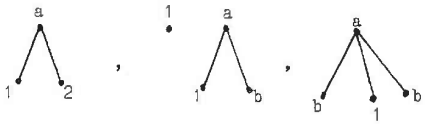
Définition 1.1.8.- On appelle polynômes injectifs sur V les éléments du V-binoïde engendré par $\{a \times i ; a \in T, i \in \mathbb{N}'\}$ dans le V-binoïde $\widehat{V \cup \mathbb{N}'}$. On notera PI(V) l'ensemble des polynômes injectifs sur V .

Comme $\{a \times i ; a \in T, i \in \mathbb{N}'\}$ est contenu dans $\hat{V}(\mathbb{N}')$, PI(V) est contenu dans $\hat{V}(\mathbb{N}')$. Donc tout polynôme injectif est bien un polynôme.

Exemple : Les polynômes suivants sont injectifs



les polynômes suivants ne sont pas injectifs



Intuitivement, un polynôme r est injectif si et seulement si aucun élément de $\mathbb{N}'$ n'apparaît dans la racine de r et si tout i de $\mathbb{N}'$ apparaissant dans r est le seul descendant d'un certain a de V .

C'est-à-dire $r \in PI(V) \Longleftrightarrow r \in \hat{V}(\mathbb{N}')$ et $\rho(r) \in V^*$ et

$(\forall a \in V)\ (\forall \alpha)\ (\alpha \in F_a(r) \Longrightarrow \alpha \in \mathbb{N}'$ ou $\alpha \in V^*)$.

Proposition 1.1.9.- Un polynôme injectif $r \in \hat{V}_n$ admet une fonction associée r^n de $\hat{V}^n$ dans $\hat{V}$ injective sur ses variables efficaces, c'est-à-dire

$$r^n(r_1,r_2,\ldots,r_n) = r^n(r'_1,r'_2,\ldots,r'_n) \Longrightarrow (\forall i \in [1,n])\ (r_i=r'_i \text{ ou } d_i(r)=0) .$$

Démonstration.- On démontre cette proposition par récurrence sur r .

Si $r = a \in V$, c'est immédiat.

Si $r = a \times i$, $r^n(r_1,\ldots,r_n) = a \times r_i$ la proposition est vérifiée.

Supposons la proposition vérifiée pour deux polynômes injectifs s et t et montrons-la pour $a \times s + t = r$

$$r^n(r_1,\ldots,r_n) = a \times s^n(r_1,\ldots,r_n) + t^n(r_1,\ldots,r_n) .$$

Donc $r^n(r_1,\ldots,r_n) = r^n(r'_1,\ldots,r'_n) \Longleftrightarrow [s^n(r_1,\ldots,r_n) = s^n(r'_1,\ldots,r'_n)$ et

$t^n(r_1,\ldots,r_n) = t^n(r'_1,\ldots,r'_n)] \Longleftrightarrow$

$(\forall i \in [1,n])\ (r_i = r'_i$ ou $F_i(s) = F_i(t) = \emptyset) \Longleftrightarrow$

$(\forall i \in [1,n])\ (r_i = r'_i$ ou $F_i(r) = \emptyset)$.

Cette proposition justifie la dénomination d'injectif pour les polynômes de PI(V) .

1.2.- Composition des polynômes :

En utilisant la proposition 1.2.4 et en prenant pour B le V-binoïde $\hat{V}(\mathbb{N}')$, on obtient immédiatement la proposition suivante :

Proposition 1.2.1.- Soient $r \in \hat{V}_n$ et r_1 , r_2 , ... , r_n des éléments de $\hat{V}(\mathbb{N}')$. L'application de $(\hat{V}(\mathbb{N}'))^n$ dans $\hat{V}(\mathbb{N}')$ associée à r applique le n-uple r_1 , r_2 , ... , r_n sur un polynôme r' de $\hat{V}(\mathbb{N}')$. Par définition, r' est appelé le polynôme composé du polynôme r et du n-uple de polynômes $(r_1,r_2,\ldots,r_n)$. De plus, si chaque polynôme r_i appartient à $\hat{V}_k$ (ensemble des polynômes dont les variables sont dans $[1,k]$) , alors r' est aussi dans $\hat{V}_k$ et pour tout V-binoïde B , on a la relation

$$(r')^k_B(b_1,b_2,\ldots,b_k) = r^n_B((r_1)^k_B(b_1,b_2,\ldots,b_k),\ldots,(r_n)^k_B(b_1,\ldots,b_k)) .$$

Le polynôme r' sera noté $r \circ(r_1,r_2,\ldots,r_n)$.

Démonstration.- Seule la dernière égalité n'est pas une conséquence triviale de 1.1.4. Elle se démontre immédiatement par récurrence sur r .

Proposition 1.2.2.- Soient $r \in \hat{V}_n$ et r_1 , r_2 , ... , r_n des éléments de PI(V) , alors

$$r \circ(r_1,r_2,\ldots,r_n) \in PI(V) .$$

Démonstration.- Cette démonstration se fait immédiatement par récurrence sur r .

Proposition 1.2.3.- On a la relation suivante entre les degrés en i de r , r_1 , ... , r_n et $r \circ(r_1,r_2,\ldots,r_n)$

$$d_i(r \circ(r_1,\ldots,r_n)) = \sum_{j=1}^{n} d_j(r)\ d_i(r_j) .$$

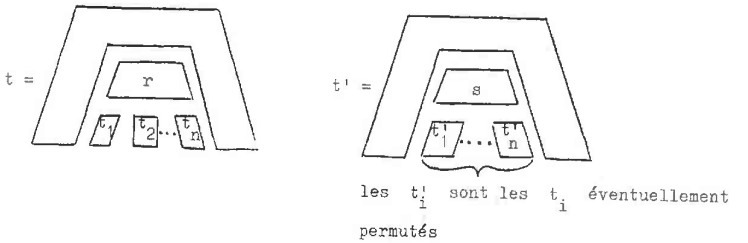
CHAPITRE 2 : BIGRAMMAIRES

La notion de bigrammaire va généraliser la notion de grammaire sur le monoïde libre V^* dans le cas du V-binoïde universel $\hat{V}$. Les polynômes introduits au chapitre précédent vont nous servir à écrire les règles des bigrammaires et seront utilisés pour expliciter les réécritures et les dérivations dans une bigrammaire.

Intuitivement, une bigrammaire travaille de la façon suivante :
On se donne :

- un axiome ou un ensemble d'axiomes,
- des productions qui sont des couples de polynômes ayant les mêmes variables efficaces,
- un vocabulaire terminal.

On passe à l'aide d'une production (r,s) de la grammaire d'une ramification t à une ramification t' en cherchant si une "sous-ramification" de t peut s'écrire $r^n(t_1,t_2,\ldots,t_n)$. Si ce cas se produit, on remplace cette "sous-ramification" par $s^n(t_1,t_2,\ldots,t_n)$. Cela correspond au schéma



les t'_i sont les t_i éventuellement permutés

Le passage de t à t' est une réécriture dans la bigrammaire.

Le bilangage (c'est-à-dire le sous-ensemble de $\hat{V}$) engendré par la bigrammaire est formé des ramifications qui s'obtiennent à partir d'un axiome en utilisant une suite finie de réécritures.

Donnons maintenant des définitions précises.

2.1.- Définition d'une bigrammaire :

Définition 2.1.1.- Soient r et s des polynômes sur V. On dit que r et s sont compatibles s'ils vérifient la condition suivante :

$$(\forall i \in \mathbb{N}')\ (d_i(r) > 0 \iff d_i(s) > 0)$$

(c'est-à-dire que r et s ont les mêmes variables efficaces).

Définition 2.1.2.- Une bigrammaire G est par définition un quadruplet $G = (N,T,P,X)$ tel que

- N et T sont deux ensembles finis disjoints appelés respectivement vocabulaire auxiliaire ou non-terminal et vocabulaire terminal. On note $V = N \cup T$ et on suppose que V est disjoint de $\mathbb{N}'$.
- P est un ensemble fini de couples de polynômes sur V compatibles. Un élément de P est appelé production de G.
- X est un ensemble fini de ramifications sur V appelées axiomes de G. Dans de nombreux cas, X sera réduit à un axiome X_0 et on notera $G = (N,T,P,X_0)$ au lieu de $(N,T,P,\{X_0\})$.

2.2.- Réécritures. Dérivations. Bilangages et langages engendrés :

Dans tout le paragraphe, on considère une bigrammaire $G = (N,T,P,X)$.

Définition 2.2.1.- Soient t et t' deux ramifications de $\hat{V}$. On dit que t se réécrit t' dans G et on note $t \xrightarrow[G]{} t'$ si et seulement si on peut trouver un polynôme u de $\hat{V}_1$ et une production (r,s) de G telle que

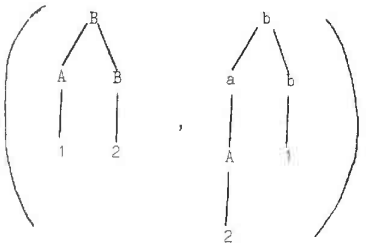
$$t = u^1(r^n(t_1,t_2,\ldots,t_n))$$

et $$t' = u^1(s^n(t_1,t_2,\ldots,t_n))\ .$$

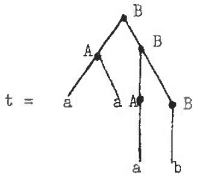
On dit alors que la production (r,s) est applicable à t.

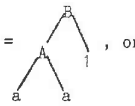
(On notera souvent une réécriture $t \longrightarrow t'$ au lieu de $t \xrightarrow[G]{} t'$ quand il n'y a pas d'ambiguïté sur G).

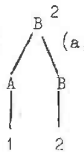
Exemple : Supposons que dans G, il y ait la production

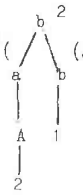
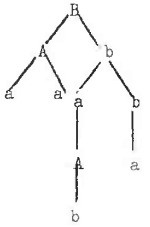


Soit

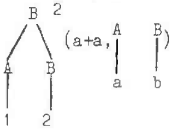


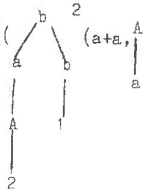
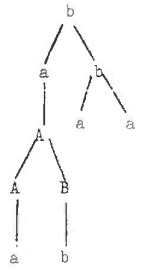
En prenant pour u le polynôme $u =$  , on obtient

$t = u^1($  $(a,b))$, donc $t \xrightarrow[G]{} t'$ avec

$t' = u^1($  $(a,b)) =$ 

En prenant pour u le polynôme $u = 1$, on obtient

$t = u^1($  $(a+a,$ $)$, donc $t \xrightarrow[G]{} t'$ avec

$t' = u^1($  $(a+a,$ $) =$ 

Définition 2.2.2.- La relation $t \overset{*}{\underset{G}{\twoheadrightarrow}} t'$ lue "de t dérive t' dans G" est par définition la fermeture réflexive et transitive de la relation "se réécrit". On notera $t \overset{p}{\underset{G}{\twoheadrightarrow}} t'$ si on utilise p réécritures passer de t à t'.

Proposition 2.2.3.- La relation $\overset{*}{\underset{G}{\twoheadrightarrow}}$ est compatible avec les lois de $\hat{V}$ et avec les fonctions polynômes. C'es-à-dire :

a) $r \overset{*}{\underset{G}{\twoheadrightarrow}} r'$ et $s \overset{*}{\underset{G}{\twoheadrightarrow}} s' \Longrightarrow r + s \overset{*}{\underset{G}{\twoheadrightarrow}} r' + s'$

b) $r \overset{*}{\underset{G}{\twoheadrightarrow}} r'$ et $a \in V \Longrightarrow a \times r \overset{*}{\underset{G}{\twoheadrightarrow}} a \times r'$

c) $r_1 \overset{*}{\underset{G}{\twoheadrightarrow}} r'_1$ et ... et $r_n \overset{*}{\underset{G}{\twoheadrightarrow}} r'_n$ et $r \in \hat{V}_n \Longrightarrow$

$$r^n(r_1,r_2,\ldots,r_n) \overset{*}{\underset{G}{\twoheadrightarrow}} r^n(r'_1,r'_2,\ldots,r'_n) .$$

Démonstration.- a) et b) sont des cas particuliers de c) en prenant respectivement pour polynômes $1 + 2$ et $a \times 1$.

Pour démontrer c), il suffit de montrer que

$$r_i \overset{}{\underset{G}{\twoheadrightarrow}} r'_i \text{ et } r \in \hat{V}([1,n]) \Longrightarrow r^n(r_1,\ldots,r'_i,\ldots,r_n) \overset{}{\underset{G}{\twoheadrightarrow}} r^n(r_1,\ldots,r'_i,\ldots,r_n) \text{ ou}$$

$$r^n(r_1,\ldots,r_i,\ldots,r_n) = r^n(r_1,\ldots,r'_i,\ldots,r_n) .$$

Distinguons les deux cas où i est ou n'est pas une variable efficace de r

- si $d_i(r) = 0$, on a évidemment $r^n(r_1,\ldots,r_i,\ldots,r_n) = r^n(r_1,\ldots r'_i,\ldots,r_n)$
- si $d_i(r) > 0$. Par hypothèse, $r_i \overset{}{\underset{G}{\twoheadrightarrow}} r'_i$, donc il existe $u \in \hat{V}([1])$ et $(s,t) \in P$ tels que

$$r_i = u^1(s^k(s_1,\ldots,s_k))$$

$$r'_i = u^1(t^k(s_1,\ldots,s_k)) .$$

En appliquant la proposition 1.2.1., on obtient alors

$$r^n(r_1,\ldots,r_i,\ldots,r_n) = (r^n(r_1,\ldots,u,\ldots,r_n))^1(s^k(s_1,\ldots,s_k))$$

$$r^n(r_1,\ldots,r'_i,\ldots,r_n) = (r^n(r_1,\ldots,u,\ldots,r_n))^1(t^k(s_1,\ldots,s_k)) ,$$

donc $$r^n(r_1,\ldots,r_i,\ldots,r_n) \overset{}{\underset{G}{\twoheadrightarrow}} r^n(r_1,\ldots,r'_i,\ldots,r_n) .$$

Définition 2.2.4.- On appelle bilangage engendré par G l'ensemble

$$BL(G) = \left\{ r ; (\exists x \in X)\ (x \overset{*}{\underset{G}{\twoheadrightarrow}} r) \text{ et } r \in \hat{T} \right\} .$$

On appelle langage engendré par G l'ensemble

$$L(G) = \left\{ \alpha ; \alpha \in T^* \text{ et } (\exists r \in BL(G))\ (\varphi(r) = \alpha) \right\}$$

(on a donc $L(G) = \varphi(BL(G))$).

Définition 2.2.5.- Soient G et G' deux bigrammaires.

On dit que G et G' sont équivalentes si $BL(G) = BL(G')$.

On dit que G et G' sont faiblement équivalentes si $L(G) = L(G')$.

Ces deux relations sont évidemment des relations d'équivalence.

2.3.- Exemples de bigrammaires :

1°) Considérons la bigrammaire $G = (\{X\},\{a,b,c,d\},P,X)$ telle que $(X,d+d+d) \in P$ et $(dx1+dx1+dx1,dx(a+1)+dx(b+1)+dx(c+1)) \in P$.

Une dérivation dans G est de la forme suivante

$$X \rightarrowtail d + d + d \rightarrowtail d \times a + d \times b + d \times c \rightarrowtail \ldots \rightarrowtail d \times \underbrace{(a+a+\ldots+a)}_{k \text{ fois}} +$$

$$+ d \times \underbrace{(b+\ldots+b)}_{k \text{ fois}} + d \times \underbrace{(c+\ldots+c)}_{k \text{ fois}} .$$

En faisant un raisonnement par récurrence sur la longueur de la dérivation, on obtient

$$BL(G) = \left\{ d \times \underbrace{(a+\ldots+a)}_{k \text{ fois}} + d \times \underbrace{(b+\ldots+b)}_{k \text{ fois}} + d \times \underbrace{(c+\ldots+c)}_{k \text{ fois}} ;\ k \in \mathbb{N} \right\}$$

et $$L(G) = \left\{ a^k b^k c^k ;\ k \in \mathbb{N} \right\}$$

2°) Une grammaire est un cas particulier de bigrammaire où les polynômes intervenant dans les productions sont de degré 0 et sont réduits à leurs racines et où l'axiome appartient à N.

3°) Considérons le bilangage L sur T formé des ramifications binaires, c'est-à-dire telles que

$$r \in L \Longleftrightarrow (\forall a \in T)(\alpha \in F_\alpha(r) \Longrightarrow |\alpha| = 2 \text{ ou } \alpha = \wedge)$$

L est engendré par la bigrammaire $G = (N,T,P,X)$ avec

$$N = \{X,A\}$$
$$P = \{(X,A+X)\ ,\ (X,\wedge)\} \cup \bigcup_{a \in T} \{(A,a)\ ,\ (A,a \times (A+A))\}.$$

Pour démontrer que L est bien engendré par G, on remarque que

$L(G) = \bigcup_{n=0}^{\infty} \underbrace{(L'+L'+\ldots+L')}_{n \text{ fois}}$ où L' est le bilangage des ramifications de $\hat{T}$ dérivant de A dans G. En faisant une récurrence sur la longueur de la dérivation $A \overset{*}{\underset{G}{\gg\!\!-}} r$, on montre facilement que $r \in L$ et $|\rho(r)| = 1$. Réciproquement en faisant une récurrence sur r on montre que

$$r \in L \text{ et } |\rho(r)| = 1 \Longrightarrow A \overset{*}{\underset{G}{\gg\!\!-}} r .$$

2.4.- Degrés d'une bigrammaire.

Définition 2.4.1.- Soit $G = (N,T,P,X)$ une bigrammaire. On pose par définition $d_g(G) = \underset{(r,s)\in P}{\text{Sup}}\ d(r)$ et $d_d(G) = \underset{(r,s)\in P}{\text{Sup}}\ d(s)$. ($d_g$ est le degré à gauche de G et d_d est le degré à droite de G).

Dans le cas où une bigrammaire G est de degré à gauche 1, il est facile de savoir si une règle (r,s) de G est applicable à t. En effet, il suffit de regarder si le corps de r (cf. définition 1.2.7.) apparaît dans t et si les ramifications non vides qui sont "en dessous" de r dans t correspondent à des variables de r. Donc intuitivement le prédicat sur t "(r,s) est applicable à t" peut être reconnu par un automate fini.

En revanche dans le cas où le degré à gauche de G est au moins deux, il est beaucoup plus difficile de savoir si une règle est applicable ou non à une ramification. En effet, pour que la production (r,s) avec $d(r) > 1$ soit applicable à t, il faut non seulement que c(r) apparaisse dans une sous-ramification de t, mais aussi que les ramifications de t en dessous de c(r) correspondant à une même variable i soient égales. Ainsi on trouve intuitivement que le prédicat P(t) "(r,s) est applicable à t" ne peut être reconnu par un automate fini quand $d_g(G) > 1$. Dans toute la suite on se bornera à considérer des bigrammaires telles que $d_g(G) \leqslant 1$. En annexe, on donnera quelques résultats pour $d_g(G) > 1$.

Un cas particulier de bigrammaire est celui des bigrammaires de degré zéro.

2.5.- Bigrammaire conservant la racine.

Définition 2.5.1.- Soit $G = (N,T,P,X)$ une bigrammaire. On dit que G conserve

la racine si G vérifie la condition :

$$(\forall (r,s) \in P)(\rho(r) = \rho(s)).$$

Proposition 2.5.2.- Soit $G = (N,T,P,X)$ une bigrammaire conservant la racine, alors

$$(\forall t \in \hat{V})(\forall t' \in \hat{V})\ (t \overset{*}{\underset{G}{\gg\!\!-}} t' \Longrightarrow \rho(t) = \rho(t')) .$$

Démonstration.- Il suffit de démontrer que $t \underset{G}{\gg\!\!-} t' \Longrightarrow \rho(t) = \rho(t')$.

Si $t \underset{G}{\gg\!\!-} t'$ pour un certain polynôme $u \in \hat{V}([1])$ et une certaine production $G(r,s) \in P$ on a

$$t = u'(r^n(t_1,\ldots,t_n))$$
$$t' = u'(s^n(t_1,\ldots,t_n)) .$$

Si $\rho(u) \in \hat{V}$, alors $\rho(t) = \rho(t') = \rho(u)$.

Si $\rho(u) = \alpha \uparrow \beta$ avec $\alpha \in \hat{V}$ et $\beta \in \hat{V}$ alors

$$\rho(t) = \alpha\, \rho(r^n(t_1,\ldots,t_n))\, \rho \quad \text{et} \quad \rho(t') = \alpha\, \rho(s^n(t_1,\ldots,t_n))\, \beta .$$

Par hypothèse $\rho(r) = \rho(s) = \alpha_1 i_1 \alpha_2 i_2 \ldots \alpha_k i_k \alpha_{k+1}$ avec

$(\forall i \in [1,k+1])(\alpha_i \in \hat{V})$ et $(\forall j \in [1,k])(i_j \in \mathbb{N}')$.

Donc $\rho(r^n(t_1,\ldots,t_n)) = \alpha_1\, \rho(t_{i_1})\, \alpha_2 \rho(t_{i_2}) \ldots \alpha_k \rho(t_{i_k})\, \alpha_{k+1} = \rho(s^n(t_1,\ldots,t_n))$.

Donc $\rho(t) = \rho(t')$.

2.6.- Classification des bigrammaires.

Dans le cas des grammaires sur un monoïde libre, on a classifié les grammaires suivant la complexité des langages qu'elles engendrent
(grammaire linéaire droite (ou gauche) $\Longleftrightarrow$ langage rationnel (ou régulier)
grammaire à contexte libre $\Longleftrightarrow$ langage algébrique
grammaire contextuelle $\Longleftrightarrow$ langage contextuel
grammaire semi-thuéienne $\Longleftrightarrow$ langage récursivement énumérable).

Nous allons, dans le cas des grammaires, donner une classification similaire. Nous examinerons successivement les cas suivants :

1er cas : (chapitre 3. paragraphe 3.1.). Les bigrammaires de degré zéro et dont les productions sont de la forme (A,r), A étant un non-terminal. On obtiendra dans le cas général les C_0-bigrammaires qui généralisent les grammaires à contexte libre. Dans un cas plus particulier, on obtient les bigrammaires régulières qui engendrent les langages réguliers déjà étudiés par Quéré [20] et Berlioux [1].

2ième cas : (chapitre 3. paragraphe 3.2.). Les bigrammaires de degré un et dont

et dont les productions sont de la forme $(A \times 1, r)$, A étant un non-terminal. On obtient dans ce cas les C_1-bigrammaires qui engendrent des langages (cf définition 2.2.4.) plus généraux que les langages algébriques.

Dans ces deux premiers cas, on montrera que les bilangages engendrés sont des solutions de systèmes à point fixe dans le treillis $\mathcal{P}(\hat{T})$ ordonné par inclusion.

3ième cas : (chapitre 4). Les bigrammaires contextuelles et strictement contextuelles qui généralisent la notion de grammaire contextuelle. Les deux types de bigrammaires engendrent des bilangages décidables, mais dans le premier cas les langages engendrés sont des langages récursivement énumérables, alors que dans le second cas, on obtient exactement les langages contextuels.

4ième cas : Les bigrammaires semi-thueiennes qui généralisent la notion de grammaires semi-thueiennes.

Dans ces deux derniers cas, on étudiera des bigrammaires de ces types qui conservent la racine et dont les productions sont formées de polynômes injectifs. On montrera que les types de langages engendrés restent les mêmes malgré ces restrictions. Celles-ci sont introduites en vue de l'étude des systèmes transformationnels (chapitre 5) qui utiliseront des productions de ce type.

CHAPITRE 3 : BIGRAMMAIRE A CONTEXTE LIBRE

3.1.- Bigrammaire de degré zéro et à contexte libre.

3.1.1.- Cas général.

Les productions des bigrammaires de ce type sont de la forme (A,r) où A est dans le vocabulaire auxiliaire et r est une ramification de $\widehat{N \cup T}$. Les polynômes utilisés dans ces bigrammaires sont donc de degré zéro et de plus une réécriture consiste à remplacer un non-terminal par sa définition indépendemment du contexte dans lequel il est placé. D'où la dénomination de bigrammaire de degré zéro et à contexte libre que nous abrégerons en C_0-bigrammaire.
Plus précisément, on pose :

Définition 3.1.1.1.- On appelle bigrammaire à contexte libre de degré zéro ou C_0-bigrammaire, une bigrammaire $G = (N,T,P,X)$ telle que

- G a un seul axiome X
- les productions de G sont de la forme (A,r) où $A \in N$ et $r \in \widehat{N \cup T}$.

On dira que G est sans $\wedge$ si pour tout (A,r) de P, r est différent de $\wedge$. (Toute grammaire algébrique peut être considérée comme une C_0-bigrammaire).

Remarque : Comme les polynômes utilisés dans une C_0-bigrammaire $G = (N,T,P,X)$ sont de degré zéro, on ne pourra utiliser une production (A,r) de P pour réécrire une ramification t de $\widehat{N \cup T}$ que si A est à une feuille de t. Si la grammaire G est sans $\wedge$, toutes les productions (A,r) telles qu'un certain non-terminal B apparaisse dans r sans être à une feuille de r (c'est-à-dire que $(\exists \alpha \neq \wedge)\ (\alpha \in F_B(r))$ conduisent à des impasses quand on les utilise dans une dérivation. En effet comme la grammaire est sans $\wedge$, le non-terminal B ne pourra jamais être à une feuille et ne pourra jamais être remplacé. Donc dans le cas des C_0-bigrammaires sans $\wedge$, on peut imposer que les règles (A,r) vérifient $(\forall B \in N)(F_B(r) = \emptyset$ ou $F_B(r) = \{\wedge\})$.

En fait cette condition peut aussi être imposée en un certain sens aux C_0-bigrammaires avec $\wedge$. On obtient en effet la proposition :

Proposition 3.1.1.1.- Toute C_0-bigrammaire $G = (N,T,P,X)$ est équivalente (cf. définition 2.2.5) à une C_0-bigrammaire $G' = (N,T,P',X)$ telle que

$$(A,r) \in P' \Longrightarrow (\forall B \in N)(F_B(r) \subset \{\wedge\}).$$

On supposera dans la suite que les C_0-bigrammaires vérifient cette condition.

Démonstration : La démonstration a été faite dans la remarque ci-dessus pour les C_o-bigrammaires sans $\wedge$. Supposons donc que G est avec $\wedge$. On peut déterminer l'ensemble N' des non-terminaux de N dont dérive $\wedge$ dans G en posant

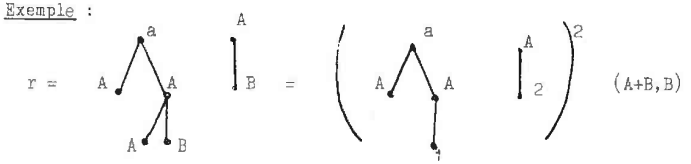
$$N_1 = \{ A \; ; \; A \in N \text{ et } (A,\wedge) \in P \}$$

$$N_{i+1} = N_i \cup \{ A \; ; \; A \in N \text{ et } (\exists r \in \hat{N}_i)(A,r) \in P \}$$

$$N' = \bigcup_{i=0}^{\infty} N_i$$

Considérons une règle (A,r) de G telle que des non-terminaux apparaissent dans r sans être à des feuilles de r . Si en dessous de ces non-terminaux, on ne trouve pas toujours dans r des ramifications de $\hat{N}'$, dans une dérivation utilisant cette production, l'un au moins des non-terminaux provenant de r ne sera jamais à une feuille et ne pourra pas être réécrit. Donc l'utilisation d'une telle production conduit à une impasse, on peut donc enlever ces productions de P sans changer le bilangage engendré par G . En revanche, si dans r on trouve en dessous des non-terminaux des ramifications de $\hat{N}'$, on peut écrire alors $r = s^n(r_1,r_2,\ldots,r_n)$ où s est un polynôme injectif de $\hat{V}_n$ engendré par les ramifications $B \times i$ $(B \in N \; , \; i \in \mathbb{N}')$, $r_1,r_2,\ldots,r_n$ sont dans $\hat{N}'$ et $c(s)$ vérifiant $(\forall B \in N)(F_B(c(s)) \subset \{\wedge\})$.

Exemple :



Lors de l'utilisation de la production (A,r) pour ne pas être conduit à une impasse, il est nécessaire de dériver r_1 , r_2 ,..., r_n en $\wedge$. Donc en remplaçant la production (A,r) par $(A,c(s))$, on obtient les mêmes ramifications engendrées.

Nous allons donner maintenant une caractérisation des bilangages engendrés par les C_o-bigrammaires en terme de système à point fixe.

Lemme 1.- $(\mathcal{P}(\hat{T}))^n$ est un treillis complet pour la relation d'ordre

$$(X_1,X_2,\ldots,X_n) \subset (Y_1,Y_2,\ldots,Y_n) \Longleftrightarrow (\forall i \in [1,n])(X_i \subset Y_i).$$

Démonstration.- Cela est évident car un produit de treillis complet est un treillis complet.

Lemme 2.- $\mathcal{P}(\hat{T})$ est muni d'une structure de T-binoïde quand on le munit de deux lois

$$X + Y = \{ r + s \; ; \; r \in X \text{ et } s \in Y \}$$

$$a \times X = \{ a \times r \; ; \; r \in X \}$$

l'élément neutre de $+$ est $\{\wedge\}$.

Lemme 3.- Soit $\hat{T}[N]$ un T-binoïde libre de base N tel que $N = \{A_1,\ldots,A_n\}$. D'après la remarque suivant la proposition 1.1.6., à chaque r de $\hat{T}[N]$ on peut associer une application $\bar{r} : (\mathcal{P}(T))^n \longrightarrow \mathcal{P}(\hat{T})$. Cette application $\bar{r}$ est continue pour les structures de treillis de $(\mathcal{P}(\hat{T}))^n$ et de $\mathcal{P}(\hat{T})$.

Démonstration.- On peut aussi associer à r une application $\tilde{r} : T^n \longrightarrow T$. Montrons que

$$(1) \qquad \bar{r}(X_1,\ldots,X_n) = \{ \tilde{r}(r_1,\ldots,r_n) \; ; \; r_i \in X_i \} \; .$$

Pour cela, il suffit de revenir aux définitions de $\bar{r}$ et $\tilde{r}$. On a $\bar{r}(X_1,\ldots,X_n) = h(r)$ où h est l'unique homomorphisme de T-binoïde de $\hat{T}[N]$ dans $\mathcal{P}(\hat{T})$ vérifiant $h(A_i) = X_i$. On a $\tilde{r}(r_1,\ldots,r_n) = k(r)$ où k est l'unique homomorphisme de T-binoïde de $\hat{T}[N]$ dans $\mathcal{P}(\hat{T})$ vérifiant $k(A_i) = r_i$. Pour démontrer l'égalité (1) il suffit donc de démontrer que $h' : \hat{T}[N] \longrightarrow \mathcal{P}(\hat{T})$ définie par

$$h'(r) = \{ \tilde{r}(r_1,\ldots,r_n) \; ; \; r_i \in X_i \} \text{ vérifie les deux conditions}$$

- $h'(A_i) = h(A_i)$
- h' est un homomorphisme de T-binoïde,

ces deux propriétés de h' sont trivialement vérifiées

En utilisant l'égalité (1), il est alors évident que $\bar{r}$ est une application continue de $(\mathcal{P}(\hat{T}))^n$ dans $\mathcal{P}(\hat{T})$.

Lemme 4.- Soit $r \in \hat{T}_n$ un polynôme à n variables sur T , chacune d'elles étant efficaces. Soit $G = (N,T,P,X)$ une C_o-bigrammaire. Soit B_1 , B_2 ,..., B_n des éléments de N . On a l'équivalence suivante :

$$(1) \quad r^n(B_1,B_2,\ldots,B_n) \overset{p}{\underset{G}{\twoheadrightarrow}} r' \Longleftrightarrow \exists\, r_1 \, \exists\, r_2 \ldots \exists\, r_n (B_1 \overset{p_1}{\underset{G}{\twoheadrightarrow}} r_1 \text{ et } \ldots$$

$$\text{et } B_n \overset{p_n}{\underset{G}{\twoheadrightarrow}} r_n \text{ et } \sum_{i=1}^{n} p_i = p \text{ et } r' = r^n(r_1,\ldots,r_n)) \quad (2)$$

Démonstration.- Montrons (1) ===> (2) . Pour cela faisons une récurrence sur p . Pour $p = 0$, le résultat est immédiat.
Supposons-le démontré jusqu'au rang p et montrons-le au rang $p + 1$.

On a $r^n(B_1,B_2,\ldots,B_n) \overset{p+1}{\underset{G}{\gg}} r'$, donc $r^n(B_1,\ldots,B_n) \overset{p}{\underset{G}{\gg}} r'' \succ r'$. Par hypothèse de récurrence $r'' = r^n(r'_1,r'_2,\ldots,r'_n)$ et $B_i \overset{p'_i}{\gg} r'_i$ et $\sum_{i=1}^{n} p'_i = p$.

Comme $r \in \hat{T}_n$, la dernière réécriture a agit sur r'_i , pour un certain i de $[1,n]$. Donc $r'_i \underset{G}{\gg} r''_i$ et $r = r^n(r'_1,\ldots,r'_{i-1}, r''_i, r'_{i+1},\ldots,r'_n)$. D'où le résultat. (Ce lemme généralise un lemme bien connu pour les grammaires à contexte libre).

Le fait que (2) ===> (1) découle immédiatement de la proposition 2.2.3.

Proposition 3.1.1.3.- Soit $N = \{A_1,\ldots,A_n\}$ un ensemble disjoint du vocabulaire T . On appelle C_o-système, un système dans $(\mathcal{P}(\hat{T}))^n$ de la forme

$$S : X_i = \bigcup_{j=1}^{p_i} \bar{r}_{ij}(X_1,\ldots,X_n) \quad (1 \leqslant i \leqslant n)$$

où r_{ij} est un élément de $\hat{T}[N]$ et où $\bar{r}_{ij}$ est défini comme au lemme 3.

Tout C_o-système admet des solutions et admet, en particulier, une solution minimale pour la structure de treillis complet de $(\mathcal{P}(\hat{T}))^n$.

A tout C_o-système S , on peut associer la C_o-bigrammaire $G = (N,T,P,X)$ telle que

- $X = A_1$
- $(A_i,r) \in P \iff (\exists j \in [1,p_i])\ (r = r_{ij})$.

Dans ces conditions, si on appelle $(L_1,L_2,\ldots,L_n)$ la solution minimale de S , on a

$$L_i = \{r \ ; \ r \in \hat{T} \text{ et } A_i \overset{*}{\underset{G}{\gg}} r\}$$

En particulier $BL(G) = L_1$.

De façon évidente, on peut associer bijectivement les C_o-bigrammaires et les C_o-systèmes de façon que les bilangages engendrés et les premières composantes des solutions minimales coïncident.

Démonstration.- Le fait que les C_o-systèmes admettent des solutions, résulte du du lemme 3 montrant que les fonctions $\bar{r}$ associées aux éléments de $\hat{T}[N]$ sont continues et du théorème du point fixe.

L'existence d'une solution minimale résulte du théorème du point fixe.

Soient $S : X_i = \bigcup_{j=1}^{p_i} \bar{r}_{ij}(X_1,\ldots,X_n)$ $1 \leqslant i \leqslant n$ un C_o-système et $G = (N,T,P,X)$ la C_o-bigrammaire associée à S . Soit $F : (\mathcal{P}(\hat{T}))^n \longrightarrow \mathcal{P}(\hat{T}))^n$ défini par

$$F(X_1,X_2,\ldots,X_n) = \left(\bigcup_{j=1}^{p_1} \bar{r}_{1j}(X_1,\ldots,X_n) \ , \ldots, \ \bigcup_{j=1}^{p_n} \bar{r}_{nj}(X_1,\ldots,X_n) \right)$$

D'après le théorème du point fixe, la solution minimale de S est $(L_1,L_2,\ldots,L_n) = \bigcup_{n=0}^{\infty} F^n(\phi,\phi,\ldots,\phi)$. Pour démontrer que $L_i = L'_i = \{r \ ; \ r \in \hat{T} \text{ et } A_i \overset{*}{\underset{G}{\gg}} r\}$.

Il suffit donc de démontrer les deux propriétés

(1) $(L'_1,\ldots,L'_i)$ est une solution de S .

(2) $r \in \hat{T}$ et $A_i \overset{*}{\underset{G}{\gg}} r \implies (\exists k \in \mathbb{N})(r \in \pi_i \circ F^k(\phi,\phi,\ldots,\phi))$

(π_i est la $i^{\text{ème}}$ fonction projection de $(\mathcal{P}(\hat{T}))^n$ dans $\mathcal{P}(\hat{T})$ définie par $\pi_i(X_1,\ldots,X_n) = X_i$) .

La propriété (1) résulte immédiatement du lemme 4.

La propriété (2) se démontre facilement en utilisant le lemme 4 et en faisant une récurrence sur la longueur de la dérivation $A_i \overset{*}{\gg} r$.

Nous allons mettre en évidence une forme réduite des C_o-bigrammaires correspondant à la réduite de Chomsky des grammaires à contexte libre [6], et qui nous permettra de donner une autre définition des systèmes à point fixe dont les solutions sont les bilangages engendrés par les C_o-bigrammaires.

Proposition 3.1.1.4.- Pour toute C_o-bigrammaire $G = (N,T,P,X)$, il existe une C_o-bigrammaire équivalente dont les productions sont de la forme $(A,B+C)$ ou $(A,a\times C)$ ou $(A,\wedge)$ $(A \in N, B \in N, a \in T)$.

Démonstration.- Soit $n_G = \sup(n(r) \ ; \ (\exists A \in N)((A,r) \in P)$ n est la fonction nombre de points définie en 0.2.2. On va faire la construction de G' par récurrence sur n_G .

Si $n_G = 1$ ou $n_G = 2$, on peut prendre $G' = G$.

Supposons la construction de G' faite pour $n \leqslant k$.

Soit G une grammaire telle que $n_G = k + 1$ et soit (A,r) une production de G vérifiant $n(r) = k + 1$.

- Si $r = B + r'$ $(A \in N)$, on remplace la production (A,r) par les deux productions $(A,B+B')$, (B',r) où B' est un nouveau non-terminal.
- Si $r = a \times s + t$, on remplace la production (A,r) par les quatre productions $(A,A'+A'')$, $(A',a\times C')$, (C',s) , (A'',t) où A' , A'' et C' sont trois nouveaux non-terminaux.

En effectuant cette transformation sur toutes les productions (A,r) de G vérifiant $n(r) = k + 1$, on obtient une C_o-bigrammaire G_1 qui, de façon évidente, est équivalente à G et vérifie $n_{G_1} \leqslant k$. En appliquant l'hypothèse de récurrence, on en déduit la C_o-bigrammaire G'.

On peut caractériser les bilangages engendrés par les C_o-bigrammaires en terme de solution de système à point fixe dans $\mathcal{P}(\hat{T})$ de la façon suivante :

Proposition 3.1.1.5.- Les applications suivantes sont continues pour la structure de treillis complet de $(\mathcal{P}(\hat{T}))^n$

$$f_1 : (A,B) \longmapsto A \cup B$$

$$f_2 : (A,B) \longmapsto A + B = \{ r + s \ ; \ r \in A \text{ et } s \in B \}$$

$$f_3^a : (A) \longmapsto a \times A = \{ a \times r \ ; \ r \in A \}$$

$$f_4 : A \longmapsto \{\wedge\}$$

Les systèmes de la forme

$$\left.\begin{aligned} X_1 &= F_1(X_1,\ldots,X_n) \\ X_2 &= F_2(X_1,\ldots,X_n) \\ &\vdots \\ X_n &= F_n(X_1,\ldots,X_n) \end{aligned}\right\} \quad S$$

où les F_i sont obtenues par composition des applications précédentes, ont des solutions dans $(\mathcal{P}(\hat{T}))^n$ et les composantes des solutions minimales sont exactement les bilangages engendrés par les C_o-bigrammaires.

Démonstration.- Seule la dernière affirmation n'est pas une trivialité. Remarquons que l'union commute avec les autres applications servant à définir les F_i. Donc un système S peut se mettre sous la forme

$$X_i = \bigcup_{j=1}^{p_i} G'_{ij}(X_1,\ldots,X_m) \qquad 1 \leqslant i \leqslant m$$

où les G_{ij} sont des applications obtenues par composition à partir des applications f_2, f_3^a et f_4. En appliquant le résultat rappelé dans le chapitre 0, on en déduit que S est équivalent à un système S' de la forme

$$X_i = \bigcup_{j=1}^{p_i} G'_{ij}(X_1,\ldots,X_m) \qquad 1 \leqslant i \leqslant m$$

où chaque G'_{ij} est une des fonctions f_2, f_3^a ou f_4. On peut alors associer bijectivement les systèmes de la forme S' et les C_o-bigrammaires sous forme

réduite décrite en 3.1.1.4. grâce à la proposition 3.1.1.3.

Remarque.- De même que pour les grammaires algébriques, on peut pour les C_o-bigrammaires définir les notions de réduites inférieures et supérieures [6].

Pour la réduite supérieure, on supprime dans G les non-terminaux qui n'apparaissent pas dans une dérivation partant de l'axiome et on supprime toutes les productions contenant un de ces non-terminaux.

Pour la réduite inférieure, on supprime dans G les non-terminaux dont ne dérive aucune ramification de $\hat{T}$ ainsi que les productions utilisant un de ces non-terminaux.

Dans la suite nous supposerons que les C_o-bigrammaires utilisées sont réduites inférieurement et supérieurement.

Proposition 3.1.1.6.- Soit L un bilangage engendré par une C_o-bigrammaire G ; alors il existe une C_o-bigrammaire sans $\wedge$ engendrant $L - \{\wedge\}$ et dont les productions sont de la forme $(A,B+C)$, $(A,a\times C)$, (A,a) avec $A \in N$, $B \in N$, $C \in N$ et $a \in T$.

Démonstration.- On peut supposer G de la forme décrite en 3.1.1.4. La démonstration est la même que pour les langages algébriques : on détermine l'ensemble N' des non-terminaux dont dérive $\wedge$

$$N' = \bigcup_{n \geqslant 0} N_n \quad \text{avec} \quad N_{n+1} = \{ A \ ; \ (\exists r \in \hat{N}_n)((A,r) \in P) \} \cup N_n \quad \text{et}$$

$$N_o = \{ A \ ; \ (A,\wedge) \in P \}$$

N_n est stationnaire à partir d'un certain rang).

On supprime les productions du type $(A,\wedge)$ et on ajoute les productions suivantes

- (A,B) si et seulement si $(\exists C \in N')((A,B+C) \in P$ ou $(A,C+B) \in P)$.
- (A,a) si et seulement si $(\exists C \in N')((A,a\times C) \in P)$.

Par la méthode habituelle sur les grammaires, on supprime les règles (A,B)
On obtient ainsi une C_o-bigrammaire du type demandé et engendrant $L - \{\wedge\}$.

Corollaire.- Soit L un langage engendré par une C_o-bigrammaire et ne contenant pas $\wedge$. Il existe un système dans $\mathcal{P}(\hat{T} - \{\wedge\})$ de la forme

$$X_i = F_i(X_1,X_2,\ldots,X_n) \qquad 1 \leqslant i \leqslant n$$

où les F_i sont obtenues par composition à partir des fonctions

$$f_1 : (A,B) \longrightarrow A \cup B$$

$$f_2 : (A,B) \longrightarrow A + B = \{r + s \; ; \; r \in A \text{ et } s \in B\}$$

$$f_3^a : (A) \longrightarrow a \times A = \{a \times r \; ; \; r \in A\}$$

$$f_4^a : A \longrightarrow \{a\}$$

tel que ce système appelé C_o-système sans $\wedge$, admette une plus petite solution pour la structure de treillis de $(\mathcal{P}(\hat{T} - \{\wedge\}))^n$ et que la première composante de cette solution soit L . Réciproquement, un tel système admet des solutions et les composantes de la solution minimale sont engendrées par des C_o-bigrammaires sans $\wedge$.

Démonstration.- Comme L ne contient pas $\wedge$, il existe d'après la proposition 3.1.1.6. une C_o-bigrammaire G sans $\wedge$ engendrant L . Le système sur $\mathcal{P}(\hat{T})$ associé à G grâce à la proposition 3.1.1.3., vérifie alors les conditions imposées dans le corollaire. De plus, les fonctions utilisées laissent stables $\mathcal{P}(\hat{T} - \{\wedge\})$ donc si on restreint à $\mathcal{P}(\hat{T} - \{\wedge\})$ les solutions minimales coïncident.

Proposition 3.1.1.7.- Soit G une C_o-bigrammaire. Soit L le bilangage engendré par G . Alors $\varphi(L)$ et $\rho(L)$ (cf. 0.2.2.) sont des langages algébriques, $Ch(L) = \bigcup_{t \in L} Ch(t)$ est un langage régulier. Réciproquement, tout langage algébrique est l'ensemble des racines (resp. l'ensemble des mots des feuilles) d'un bilangage engendré par une C_o-bigrammaire. De même tout langage régulier est l'ensemble des chemins d'un bilangage engendré par une C_o-bigrammaire.

Démonstration.- Si L contient la ramification $\wedge$, alors $L' = L - \{\wedge\}$ est encore engendré par une C_o-bigrammaire et on a

$$\rho(L) = \rho(L') \cup \{\wedge\}, \; \varphi(L) = \varphi(L') \cup \{\wedge\} \; , \; Ch(L) = Ch(L') \; .$$

Il suffit donc de démontrer la proposition 3.1.1.7. pour un bilangage L ne contenant pas $\wedge$. Dans ce cas L est la première composante de la solution minimale d'un C_o-système S sans $\wedge$:

$$S : X_i = \bigcup_{j=1}^{p_i} \overline{r_{ij}}\,(X_1,\ldots,X_n) \qquad 1 \leq i \leq n \quad .$$

Ce système correspond à la C_o-bigrammaire $G = (N,T,P,X)$ telle que $N = \{X_1, \ldots, X_n\}$, $X = X_1$, $P = \{(X_i, r_{ij}) \; ; \; 1 \leq i \leq n \; , \; 1 \leq j \leq p_i\}$.

Si α est un mot sur $N \cup T$, on peut lui faire,correspondre une application $\bar{\alpha}$ de $(\mathcal{P}(T^* - \{\wedge\}))^n$ dans $\mathcal{P}(T^* - \{\wedge\})$ en posant pour α de la forme

$$\alpha = \alpha_1 X_{i_1} \alpha_2 \ldots X_{i_p} \alpha_p \qquad (\alpha_i \in T^* \text{ et } X_{i_k} \in N)$$

$$\bar{\alpha}(B_1,B_2,\ldots,B_n) = \{\alpha_1 \beta_{i_1} \alpha_2 \ldots \beta_{i_p} \alpha_{i_p} \; ; \; \beta_{i_j} \in B_{i_j}\} \; .$$

Considérons les systèmes sur $\mathcal{P}(T^* - \{\wedge\})$ suivants

$$S_\rho : X_i = \bigcup_{j=1}^{p_i} \overline{\rho(r_{ij})}(X_1,\ldots,X_n) \qquad 1 \leq i \leq n$$

$$S_\varphi : X_i = \bigcup_{j=1}^{p_i} \overline{\varphi(r_{ij})}(X_1,\ldots,X_n) \qquad 1 \leq i \leq n$$

$$S_{ch} : X_i = \bigcup_{j=1}^{p_i} \left(\bigcup_{\alpha \in ch(r_{ij})} \bar{\alpha}(X_1,\ldots,X_n) \right) \qquad 1 \leq i \leq n$$

les systèmes sont du type algébrique et donc leurs solutions minimales sont des n-uplets de langages algébriques. De plus, S_{ch} est de type régulier et donc sa solution minimale est un n-uplet de langages réguliers. Montrons que la première composante de la solution minimale de $S\rho$ (respectivement S_φ , S_{ch}) est $\rho(L)$ (respectivement $\varphi(L)$, $ch(L)$). Prolongeons les applications ρ , φ et ch à $(\mathcal{P}(T^* - \{\wedge\}))^n$ en posant

$$f(B_1,B_2,\ldots,B_n) = (f(B_1,f(B_2),\ldots,f(B_n)) \qquad (f = \rho \text{ , } \varphi \text{ ou } ch) \; .$$

On remarque alors que les diagrammes suivants sont commutatifs

$$\begin{array}{ccc} (\mathcal{P}(\hat{T} - \{\wedge\}))^n & \xrightarrow{F} & (\mathcal{P}(\hat{T} - \{\wedge\}))^n \\ \downarrow \rho(\text{resp. } \varphi, ch) & & \downarrow \rho(\text{resp. } \varphi, ch) \\ (\mathcal{P}(T^* - \{\wedge\}))^n & \xrightarrow{F_\rho \; (\text{resp. } F_\varphi \text{ , } F_{ch})} & (\mathcal{P}(T^* - \{\wedge\}))^n \end{array}$$

où F , F_ρ , F_φ et F_{ch} désignent les fonctions associées aux systèmes S , S_ρ . S_φ et S_{ch} . Le résultat cherché résulte alors du théorème sur le transfert des solutions des systèmes à point fixe démontré au paragraphe 0.5.

La réciproque est évidente. En effet, toute grammaire algébrique peut être considérée comme une C_o-bigrammaire, ce qui résoud le problème pour l'ensemble des mots des racines et l'ensemble des mots des feuilles. D'autre part, si K est un langag régulier engendré par la grammaire linéaire droite $G = (N,T,::=,X)$, K est le des chemins du bilangage L engendré par la bigrammaire $G' = (N,T,P,X)$ telle que

$(A, a \times B) \in P \iff A ::= aB$

$(A, a) \in P \iff A ::= a$

$(A, \wedge) \in P \iff A ::= \wedge$

La proposition 3.1.1.7. signifie que les bilangages engendrés par les C_o-bigrammaires sont les bilangages algébriques pour la structure de binoïde de $\hat{V}$. Dans la thèse de Lescanne [12], on a introduit ces bilangages d'une autre façon sous le nom de langages binoïdaux.

3.1.2.- Un cas particulier : les bigrammaires régulières .

Les C_o-bigrammaires engendrent en général des bilangages non réguliers au sens de Quéré [20]. Dans le cas des grammaires algébriques, un cas particulier est celui des grammaires régulières (ou linéaires droites), dont les règles sont de la forme $(A, \alpha B)$ ou (A, α) avec $\alpha \in T^*$. Ces grammaires engendrent les langages réguliers (ou rationnels). Nous allons définir des bigrammaires qui sont l'analogue des grammaires régulières et montrer en caractérisant les bilangages engendrés en terme de système à point fixe, que les langages engendrés par ces bigrammaires sont bien les bilangages réguliers déjà étudiés par Quéré [20] et Berlioux [1].

Définition 3.1.2.1.- On appelle bigrammaire régulière, une C_o-bigrammaire $G = (N,T,P,X)$ telle que les productions de G sont de la forme

$(A, r+B)$ ou (A, r) avec $\rho(r) \in T^*$, A et $B \in N$.

Remarque.- De même que pour les C_o-bigrammaires on peut supposer sans apporter de restriction que dans les productions $(A, r+B)$ ou (A, r) de la bigrammaire G, les occurences dans r des non-terminaux n'apparaissent qu'aux feuilles, c'est-à-dire que r vérifie

$$(\forall B \in N)(F_B(r) \subset \{\wedge\}) .$$

Nous allons donner une forme réduite des bigrammaires régulières qui permettra de caractériser les bilangages engendrés en terme de système à point fixe.

Proposition 3.1.2.2.- Pour toute bigrammaire régulière G, il existe une bigrammaire G' d'équivalence à G et dont les productions sont de la forme $(A, a \times B+C)$ ou $(A, \wedge)$ avec $A \in N$, $B \in N$, $C \in N$ et $a \in T$.

Démonstration.- Remarquons déjà que l'on peut se ramener au cas où toutes les règles sont de la forme $(A, r+B)$ ou $(A, \wedge)$, quitte à rajouter des non-terminaux et des règles du type $(A, \wedge)$.

a) Soit $G = (N,T,P,X)$ une bigrammaire régulière. Montrons qu'il existe une

bigrammaire régulière G_1 équivalente à G dont les productions sont de la forme $(A, r+B)$ ou $(A, \wedge)$ avec $|\rho(r)| = 1$. En effet, si $(A, a \times s+t+B)$ est une production de G, la grammaire obtenue en remplaçant cette production par les deux productions $(A, a \times s+A')$ et $(A', t+B)$ où A' est un nouveau non-terminal, est équivalente à G. En réitérant cette transformation, on obtient G_1.

b) Montrons qu'il existe une bigrammaire régulière G_2 équivalente à G_1 dont les productions sont de la forme $(A, a \times r+B)$ ou $(A, \wedge)$ avec r réduite à son mot des racines et $\rho(r) \in N^*$. En effet, si $(A, a \times (r'_o + a_1 \times r_1 + r'_1 + \ldots + a_n \times r'_n) + B)$ est une production de G_1 telle que r'_i est une ramification réduite à son mot des racines et $\rho(r'_i) \in N^*$, la grammaire obtenue en remplaçant cette production par les productions $(A, a \times (r'_o + A_1 + r'_1 + \ldots + A_n + r'_n) + B)$, $(A_i, a_i \times r_i + A')$, $(A', \wedge)$ pour $i = 1,2,\ldots,n$ est équivalente à G_1 ($A_1, \ldots, A_n$ et A' sont de nouveaux non-terminaux). En réitérant cette transformation sur G_1, on obtient G_2.

c) On est donc ramené au cas où toutes les productions de G sont de la forme $(A, a \times r+B)$ ou $(A, \wedge)$ avec r réduite à son mot des racines et $\rho(r) \in N^* - \{\wedge\}$. Nous allons construire la bigrammaire G' en raisonnant par récurrence sur l'entier n_G défini par

$$n_G = \mathrm{Sup}(|\rho(r)| ; (\exists A)(\exists B)(\exists a)((A, a \times r+B) \in P)) .$$

Si $n_G = 1$, alors $G' = G$ convient. Supposons G' construite dès que $n_G \leqslant k$ et supposons maintenant que $n_G = k + 1$. Considérons les productions $(A_i, a_i \times r_i + B_i)$, $i = 1,2,\ldots,p$ de G telles que $|\rho(r_i)| = k + 1$. Donc $r_i = A_i^1 + A_i^2 + r'_i$.

Soit P_1 l'ensemble de ces productions et $P_2 = P - P_1$.

Considérons pour chaque i de $[1,p]$ des vocabulaires N_i disjoints de N, disjoints deux à deux et en bijection avec N par l'application $\tau_i : N_i \longrightarrow N$.

Considérons la grammaire $G'' = \left(N \cup \left(\bigcup_{i=1}^{p} N_i\right), T, P'', X\right)$, P'' étant défini par

- $P_2 \subset P''$
- $(A_i, a_i \times (A_i^1 + A_i^2 + r'_i) + B_i) \in P_1 \iff (A_i, a_i \times (\tau_i^{-1}(A_i^1) + r'_i) + B_i) \in P''$

  et $(\tau_i^{-1}(A_i), a_i \times (\tau_i^{-1}(A_i^1) + r'_i) + \tau_i^{-1}(B_i)) \in P''$
- $A \in N_i$, $B \in N_i$, $(\tau_i(A), r + \tau_i(B)) \in P_2 \iff (A, r+B) \in P''$
- $A \in N_i$ et $(\tau_i(A), \wedge) \in P \iff (A, A_i^2) \in P$.

Montrons que G'' est équivalente à G. Pour cela, montrons que

(1) $A \overset{*}{\underset{G''}{\gg}} t \implies h(A) \overset{*}{\underset{G}{\gg}} h(t)$

où h est l'application de $\widehat{N \cup \left(\bigcup_{i=1}^{p} N_i\right) \cup T}$ dans $\widehat{N \cup T}$ définie par

$h(\wedge) = \wedge$

$h(A \times r+s) =$ si $A \in N \cup T$, alors $A \times h(r) + h(s)$ sinon

si $A \in N_i$, alors $\tau_i(A) + A_i^2 \times h(r) + h(s)$.

Remarquons que $h(r+s) = h(r) + h(s)$.

Cette démonstration se fait par récurrence sur la longueur q de la dérivation $A \overset{*}{\underset{G''}{\gg}} t$ et en utilisant le lemme suivant qui est une conséquence immédiate du lemme utilisé dans la démonstration de la proposition 3.1.1.6.

Lemme.- Soit $G = (N,T,P,X)$ une bigrammaire régulière

$a \times (B_1+\ldots+B_n) + B \overset{p}{\underset{G}{\gg}} t \iff (\exists t_1) \ldots (\exists t_n)(\exists s)\ (t = a \times (t_1+\ldots+t_n) + s$ et $B_i \overset{p_i}{\gg} t_i$ et $B \overset{p'}{\gg} s$ et $\left(\sum_{i=1}^{n} p_i\right) + p' = p)$.

Pour $q = 0$, la vérification de (1) est immédiate.

Supposons l'implication (1) établie pour toute dérivation de longueur au plus q , et montrons-là pour une dérivation de longueur $q + 1$.

1er cas : $A \in N$. On a $A \overset{q}{\underset{G''}{\gg}} t' \overset{q}{\underset{G''}{\gg}} t$, avec $(A,t') \in P''$.

- si $t' = \wedge$, alors $t = \wedge$ et le résultat est évident.
- si $t' \in \widehat{N \cup T} - \{\wedge\}$, alors $t' = a \times (B_1+B_2+\ldots+B_n) + B$. En appliquant le lemme ci-dessus et l'hypothèse de récurrence, le résultat est évident.
- si $t' \notin \widehat{N \cup T}$, alors $t' = a \times (B'+B_1+B_2+\ldots+B_n) + B$ et pour un $i \in [1,p]$, on a $A = A_i$ et $B' = \tau_i^{-1}(A_i^1)$. De plus $(A_i, a \times (A_i^1+A_i^2+B_1+\ldots+B_n)+B)$ est une règle de G . En appliquant le lemme ci-dessus, on obtient $t = a \times (s'+t_1+\ldots+t_n) + s$, $B' \overset{q'}{\underset{G''}{\gg}} s'$, $B_j \overset{q_j}{\underset{G''}{\gg}} t_j$, $B \overset{q''}{\underset{G''}{\gg}} s$ et $q' + \sum_{j=1}^{n} q_j + q'' = q$. En appliquant l'hypothèse de récurrence, on obtient $A_i^1 + A_i^2 \overset{*}{\underset{G}{\gg}} h(s')$, $B_j \overset{*}{\underset{G}{\gg}} h(t_j)$ et $B \overset{*}{\underset{G}{\gg}} h(s)$. Donc de $A \overset{*}{\underset{G}{\gg}} a \times (h(s')+h(t_1)+\ldots+h(t_n))+h(s) = h(t)$.

2ème cas : $A \notin N$. Donc $A \in N_i$ pour un $i \in [1,p]$. On a $A \overset{q}{\underset{G''}{\gg}} t' \overset{q}{\gg} t$ avec $(A,t') \in P'$. Les différentes possibilités pour t' sont les suivantes :

- $t' = a_i \times (\tau_i^{-1}(A_i^1)+C_1+\ldots+C_n) + \tau_i^{-1}(B_i)$ et $A = \tau_i^{-1}(A_i)$

  Dans ce cas $t = a_i \times (s'+t_1+\ldots+t_n) + s''$ avec

  $\tau_i^{-1}(A_i^1) \overset{q'}{\underset{G''}{\gg}} s'$, $C_j \overset{q_j}{\underset{G''}{\gg}} t_j$, $\tau_i^{-1}(B_i) \overset{q''}{\underset{G''}{\gg}} s''$.

  En appliquant l'hypothèse de récurrence, on obtient

  $A_i^1 + A_i^2 \overset{*}{\underset{G}{\gg}} h(s')$, $C_j \overset{*}{\underset{G}{\gg}} h(t_j)$, $B_i + A_i^2 \overset{*}{\underset{G}{\gg}} h(s'')$.

  Donc $A_i + A_i^2 \overset{}{\underset{G}{\gg}} a_i \times (A_i^1+A_i^2+C_1+\ldots+C_n) + B_i + A_i^2 \overset{*}{\underset{G}{\gg}}$

  $a_i \times (h(s')+h(t_1)+\ldots+h(t_n))+h(s'')$.

  Donc $h(A) \overset{*}{\underset{G}{\gg}} h(t)$.

- $t' = r + B$ avec $B \in N_i$ et $(\tau_i(A), r+\tau_i(B)) \in P_2$.

  Ce cas se traite de la même façon.

- $t' = A_i^2$ et $(\tau_i(A).\wedge) \in P$. Dans ce cas, le résultat est évident.

  En appliquant l'implication (1) avec $A = X$ et $t \in \hat{T}$, on obtient $BL(G'') \subset BL(G)$.

  Pour démontrer l'inclusion réciproque, montrons par récurrence sur p que

  $(\forall t \in \hat{V})(\forall A \in N)\ \left[(A \overset{p}{\underset{G}{\gg}} t \implies A \overset{*}{\underset{G''}{\gg}} t)\right.$ et

  $\left.(A+A_i^2 \overset{p}{\underset{G}{\gg}} t \implies \tau_i^{-1}(A) \overset{p}{\underset{G''}{\gg}} t)\right]$; pour $p = 0$, c'est immédiat. Supposons (2) démontré pour tout entier au plus égal à p , et démontrons (2) pour $p + 1$.

  a) $A \overset{p+1}{\underset{G}{\gg}} t \implies A \overset{}{\underset{G}{\gg}} t' \overset{p}{\underset{G}{\gg}} t$.

- Si $(A,t') \in P_2$, l'application du lemme ci-dessus et de l'hypothèse de récurrence donne le résultat.

- Le cas $A = A_i$ et $t' = a_i \times (A_i^1+A_i^2+r_i') + B_i$ se traite de même

  b) $A + A_1^2 \overset{p+1}{\gg} t \implies A + A_1^2 \overset{}{\gg} t' + A_1^2 \overset{p}{\gg} t$.

- Si $(A,t') \in P_2$, on a - soit $t' = \wedge$ et dans G'' on a $(\tau_i^{-1}(A), A_i^2) \in P''$ d'où le résultat en appliquant l'hypothèse de récurrence

- soit $t' = r + B$ et dans G'' on a $(\tau_i^{-1}(A), r+\tau_i^{-1}(B)) \in P''$ d'où le résultat en appliquant le lemme et l'hypothèse de récurrence.

- Si $A = A_i$ et $t' = a_i \times (A_i^1 + A_i^2 + r_i') + B_i$. Dans G'' on a $(\tau_i^{-1}(A_i), a_i \times (\tau_i^{-1}(A_i^1)+r_i') + \tau_i^{-1}(B_i)) \in P''$. De plus, en appliquant le lemme, on a $t = a_i \times (t_1+t_2) + s$ avec

$A_i^1 + A_i^2 \overset{p'}{\underset{G}{\twoheadrightarrow}} t_1$, $r_i' \overset{p''}{\underset{G}{\twoheadrightarrow}} t_2$, $B_i + A_i^2 \overset{p'''}{\underset{G}{\twoheadrightarrow}} s$ et $p' + p'' + p''' = p$.

D'où en appliquant l'hypothèse de récurrence $\tau_i^{-1}(A_i^1) \overset{*}{\underset{G''}{\twoheadrightarrow}} t_1$, $r_i' \overset{*}{\underset{G''}{\twoheadrightarrow}} t_2$

$\tau_i^{-1}(B_i) \overset{*}{\underset{G''}{\twoheadrightarrow}} s$. Donc $\tau_i^{-1}(A) \overset{*}{\underset{G''}{\twoheadrightarrow}} t$.

Donc (2) est démontré pour tout p . On en déduit immédiatement $BL(G) \subset BL(G'')$. Donc G et G'' sont équivalentes et $n_{G''} < n_G$. D'où en appliquant l'hypothèse de récurrence, la construction de la bigrammaire G' cherchée.

Remarque.- La dernière partie de cette démonstration est inspirée de la construction d'une grammaire régulière engendrant le produit de deux langages réguliers engendrés par deux grammaires régulières.

Proposition 3.1.2.3.- Les applications suivantes sont continues pour la structure de treillis complet de $(\mathfrak{P}(\hat{T}))^n$

$$f_1 : (A,B) \longrightarrow A \cup B$$

$$f_3 : (A,B) \longrightarrow a \times A + B = \{ a \times r + s \ ; \ r \in A \text{ et } s \in B \}$$

$$f_4 : A \longrightarrow \{\Lambda\}$$

Les systèmes de la forme

$$\left.\begin{aligned} X_1 &= F_1(X_1,\ldots,X_n) \\ X_2 &= F_2(X_1,\ldots,X_n) \\ &\cdot \\ &\cdot \\ X_n &= F_n(X_1,\ldots,X_n) \end{aligned}\right\} \quad S$$

où les F_i sont obtenues par composition à partir des applications f_1 , $f_2'^a$, f_4 ont des solutions dans $\mathfrak{P}(\hat{T})$ et les composantes des solutions minimales de tel système sont exactement les bilangages engendrés par les bigrammaires régulières.

Démonstration.- Cela résulte immédiatement de la proposition 3.1.1.3. qui associe bijectivement les C_o-bigrammaires et les S_o-systèmes. Ici il est évident que les C_o-bigrammaires associées aux systèmes décrits dans la proposition 3.1.2.3. sont

régulières.

Les bilangages réguliers introduits par Quéré [20] et définis d'une autre manière que celle-ci, vérifient les mêmes systèmes à point fixe. On trouve donc que les bigrammaires régulières engendrent exactement les bilangages réguliers. Remarquons que la proposition 3.1.2.3. signifie que les langages réguliers sont les langages algébriques pour la structure suivante de $\hat{V}$:

on se donne $card(V)$ opérations binaires f_a telles que

$$f_a(r,s) = a \times r + s$$

et une opération 0-aire Λ .

Annexe au paragraphe 3.1.2. : Une nouvelle caractérisation des bilangages réguliers.

En théorie des langages, on caractérise les langages réguliers à l'aide des classes de contexte à droite. Par définition, si L est un langage sur V et si α est un mot de V^* , la classe de contexte à droite de α est l'ensemble $D_L(\alpha) = \{ \beta \ ; \ \beta \in V^* \text{ et } \alpha\beta \in L \}$. Un théorème classique affirme que L est régulier si et seulement si l'ensemble des $D_L(\alpha)$ est fini. Les outils introduits ici, permettent de généraliser ce théorème aux bilangages réguliers.

Définition 3.1.2.4.- Soit L un bilangage sur V . Soit r une ramification de $\hat{V}$, on appelle classe de contexte de r pour L , l'ensemble $C_L(r)$ défini par

$$C_L(r) = \{ t \ ; \ t \in \hat{V}_1 \text{ et } t^1(r) \in L \} .$$

Proposition 3.1.2.5.- Un bilangage L sur V est régulier si et seulement si l'ensemble des $C_L(r)$ est fini.

Démonstration.- Pour faire cette démonstration, nous utiliserons la caractérisation suivante des bilangages réguliers :
Un bilangage sur V est régulier si et seulement si il existe un V-binoïde fini B et une partie B' de B tels que si h est l'homomorphisme de V-binoïde de $\hat{V}$ dans B on ait $L = h^{-1}(B')$.
(Cette caractérisation a été introduite par Quéré [20] et est aussi rappelée au paragraphe 3.4.).

Supposons que L est un bilangage régulier sur V . Soit B , B' et h comme ci-dessus, tels que $L = h^{-1}(B')$. Montrons que

$$h(r) = h(s) \implies C_L(r) = C_L(s) .$$

De là, on déduira qu'il y a au plus $card(B)$ classes de contexte pour L . Pour démontrer l'implication ci-dessus, il suffit de montrer que

$$h(r) = h(s) \implies (\forall t \in \hat{V}_1)(h(t^1(r)) = h(t^1(s)))$$

et cette implication se démontre immédiatement par récurrence sur t .

Réciproquement, supposons que pour un bilangage L sur V l'ensemble des classes de contexte soit fini, et montrons que L est un bilangage régulier. Soit $\mathcal{C}$ l'ensemble des classes de contexte pour L. Pour obtenir le résultat voulu, il suffit de munir $\mathcal{C}$ d'une structure de V-binoïde telle que l'application $C_L : \hat{V} \longrightarrow \mathcal{C}$ soit l'homomorphisme de binoïde de $\hat{V}$ dans $\mathcal{C}$ muni de cette structure. On aura alors $L = C_L^{-1}(\{C ; C \in \mathcal{C} \text{ et } 1 \in C\})$, et d'après la caractérisation des bilangages réguliers rappelés ci-dessus, on aura bien démontré que L est un bilangage régulier. Pour munir $\mathcal{C}$ d'une structure de V-binoïde telle que $C_L : \hat{V} \longrightarrow \mathcal{C}$ soit un homomorphisme de V-binoïde, il suffit de montrer que C_L est compatible avec les lois de V-binoïde de $\hat{V}$. C'est-à-dire que

- $C_L(r) = C_L(r')$ et $C_L(s) = C_L(s') \Longrightarrow C_L(r+s) = C_L(r'+s')$
- $C_L(r) = C_L(r') \Longrightarrow (\forall a \in V)(C_L(a \times r) = C_L(a \times r'))$.

Supposons que $C_L(r) = C_L(r')$ et $C_L(s) = C_L(s')$.
On obtient :

$$t \in C_L(r+s) \Longleftrightarrow t^1(r+s) \in L \Longleftrightarrow t^1(1+s) \in C_L(r) \Longleftrightarrow t^1(1+s) \in C_L(r') \Longleftrightarrow$$

$$t^1(r'+s) \in L \Longleftrightarrow t^1(r'+1) \in C_L(s) \Longleftrightarrow t^1(r'+1) \in C_L(s') \Longleftrightarrow$$

$$t^1(r'+s') \in L \Longleftrightarrow t \in C_L(r'+s')$$

d'où la première implication.

Supposons que $C_L(r) = C_L(r')$ et soit a un élément quelconque de V.
On obtient :

$$t \in C_L(a \times r) \Longleftrightarrow t^1(a \times r) \in L \Longleftrightarrow t^1(a \times 1) \in C_L(r) \Longleftrightarrow$$

$$t^1(a \times 1) \in C_L(r') \Longleftrightarrow t^1(a \times r') \in L \Longleftrightarrow t \in C_L(a \times r')$$

d'où la deuxième implication. La struxture de V-binoïde de $\mathcal{C}$ est alors définie par

$C + C' = C_L(r+r')$ avec r et r' tels que $C_L(r) = C$ et $C_L(r') = C'$

$a \times C = C_L(a \times r)$ avec r tel que $C_L(r) = C$

3.2.- Bigrammaire de degré un et à contexte libre.

Les productions de ces bigrammaires sont de la forme $(A \times 1, r)$ où r est un polynôme injectif sur $T \cup N$. On remplace donc un non-terminal indépendamment du contexte dans lequel il est placé. Malgré cette remarque, on montrera que les langages engendrés ne sont pas algébriques en général. Nous étudierons une caractérisation de ces bilangages en termes de système à point fixe, puis les langages engendrés.

3.2.1.- Définition d'une C_1-bigrammaire et caractérisation en termes de système à point fixe des bilangages engendrés.

Définition 3.2.1.1.- On appelle bigrammaire à contexte libre de degré un ou C_1-bigrammaire, une bigrammaire $G = (N,T,P,X)$ telle que

- G a un seul axiome X.
- les productions de G sont de la forme $(A \times 1, r)$ où $A \in N$ et $r \in \widehat{N \cup T_1}$ est un polynôme injectif de degré 1.

Nous allons donner une forme réduite des C_1-bigrammaires qui facilitera leur étude. Nous aurons besoin au cours de la démonstration de définir $t \times 1$ où $t \in \widehat{N \cup T}$. $t \times 1$ signifie intuitivement que l'on a greffé 1 en dessous de la première feuille de t utilisant la récurrence $\wedge \times 1 = 1$

$$(a \times r+s) \times 1 = a \times (r \times 1) + s$$

Si t est non vide, $t \times 1$ est bien un polynôme injectif.

Proposition 3.2.1.2.- Pour toute C_1-bigrammaire $G = (N,T,P,X)$, il existe une C_1-bigrammaire équivalente $G' = (N',T,P',X)$ équivalente à G et dont les règles sont de la forme $(A \times 1, B+C \times 1)$ ou $(A \times 1, B \times 1+C)$ ou $(A \times 1, B \times C \times 1)$ ou $(A \times 1, a \times 1)$ où A, B et C sont des non-terminaux et a un terminal.

Démonstration.- On va raisonner par récurrence sur $n_G = \sup(n(r); (\exists A \in N)(A \times 1, r) \in P)$. Pour $n_G = 1,2,3$, la proposition est évidente. Supposons-la démontrée pour $n_G \leq k$ et supposons que $n_G = k + 1$. Soit (A,r) une production de G telle que $n(r) = k + 1$. On a $r = x \times s + t$ avec $x \in N \cup T$. On remplace cette production par les productions suivantes :

- si $t \neq \wedge$ et $x \times s \in \widehat{N \cup T}$ et $t \in \widehat{N \cup T_1}$, alors $(A \times 1, B+C \times 1)$ $(B \times 1, (x \times s) \times 1)$, $(C \times 1, t)$ où B et C sont deux nouveaux terminaux.
- si $t \neq \wedge$ et $x \times s \in \widehat{N \cup T_1}$, alors $(A \times 1, B \times 1+C)$, $(B \times 1, x \times s)$, $(C \times 1, t \times 1)$ où B et C sont deux nouveaux terminaux.
- si $t = \wedge$, alors $(A \times 1, B \times C \times 1)$, $(B \times 1, x \times 1)$, $(C \times 1, s)$ où B et C sont deux nouveaux non-terminaux.

Il est à peu près évident que cette nouvelle grammaire engendre le même bilangage que G. En appliquant l'hypothèse de récurrence, on obtient alors la grammaire G' cherchée.

Remarque.- Quand on écrit dans la démonstration ci-dessus $(x \times s) \times 1$ ou $t \times 1$, on greffe la variable 1 en dessous de la première feuille de la ramification considérée. Ce choix sur la place de cette greffe est un peu arbitraire; cependant quand on utilise des productions où interviennent ces ramifications, on remarque que le non-terminal qui est réécrit, se trouve à une feuille. On utilise donc ces règles comme si elles étaient de degré zéro et la place de la variable 1 n'intervient donc pas.

Proposition 3.2.1.3.- Tout bilangage L ne contenant pas Λ et engendré par une C_0-bigrammaire, est aussi engendré par une C_1-bigrammaire.

Démonstration.- Comme L ne contient pas Λ, il existe une C_0-bigrammaire $G = (N,T,P,X_0)$ sans Λ engendrant L. Considérons la C_1-bigrammaire $G' = (N,T,P',X_0)$ définie par $(A \times 1, r \times 1) \in P' \iff (A,r) \in P$.
Comme dans r les non-terminaux n'apparaissent qu'aux feuilles, dans une dérivation à partir de X_0 dans G', on utilisera les productions comme si elles étaient de degré zéro. Donc G' est équivalente à G. (Il est cependant faux que $r \overset{*}{\underset{G''}{\twoheadrightarrow}} t \iff r \overset{*}{\underset{G}{\twoheadrightarrow}} t$, cela n'est vrai que si r vérifie $(\forall \alpha \in (N \cup T)^*)$ $(\forall A \in N)(\alpha \in F_A(r) \iff \alpha = \Lambda)$.

La réciproque de cette proposition est fausse. En effet, considérons la C_1-bigrammaire $G = (\{X\}, \{a,b,c\}, P, X)$ telle que $P = \{(X \times 1, a \times X \times b \times 1), (X \times 1, c \times 1)\}$. On obtient de façon évidente

$$BL(G) = \{ \underbrace{a \times a \times \ldots \times a}_{n \text{ fois}} \times c \times \underbrace{b \times \ldots \times b}_{n \text{ fois}} \; ; \; n \in \mathbb{N} \} .$$

Donc $ch(BL(G)) = \{a^n c b^n \; ; \; n \in \mathbb{N}\}$ n'est pas un langage régulier, et donc, d'après la proposition 3.1.1.5., $BL(G)$ ne peut pas être engendré par une C_0-bigrammaire.

Pour nous permettre de faire rigoureusement des récurrences sur la longueur d'une dérivation dans une C_1-bigrammaire, on introduit les définitions suivantes :

Définition 3.2.1.4.- On appelle squelette de ramification sur V, une ramification de $\widehat{V \cup \mathbb{N}'}$ telle que tout entier i a au plus une occurrence dans r. C'est-à-dire que r est un squelette sur V si
$(r \in \widehat{V \cup \mathbb{N}'})$ et $(\forall i \in \mathbb{N}')$ $(\text{card}(F_i(r)) \leq 1)$.

Définition 3.2.1.5.- (Fonction associée à un squelette). Soit $r \in \widehat{V \cup [1,n]}$ un squelette de ramification sur V. On associe à r une application encore notée r de $(\hat{V}_1)^n$ dans $\hat{V}$ de la façon suivante : $r(r_1,r_2,\ldots,r_n)$ est obtenu en remplaçant dans r chaque i par r_i, et on utilise la variable 1 de r_i pour greffer ce qui est en dessous de r_i. Formellement, on définit cette application par récurrence sur r en posant

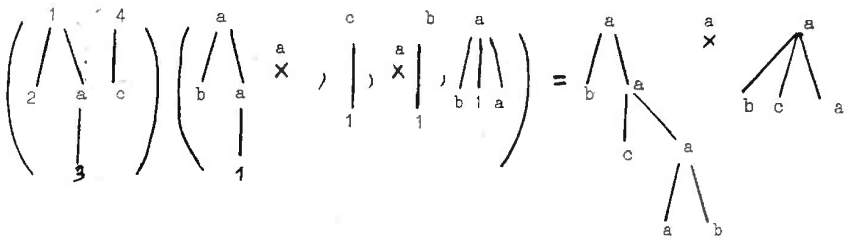
$$\Lambda(r_1,r_2,\ldots,r_n) = \Lambda$$

si $i \in \mathbb{N}'$, alors $(i \times r+s)(r_1,r_2,\ldots,r_n) =$ si $r = \Lambda$ alors $c(r_i)+s(r_1,\ldots,r_n)$
sinon $r_i^1(r(r_1,\ldots,r_n)) + s(r_1,\ldots,r_n)$

si $x \in V$, alors $(x \times r+s)(r_1,\ldots,r_n) = x \times r(r_1,\ldots,r_n) + s(r_1,r_2,\ldots,r_n)$

(quand r_i est réduit à $x \times 1$, on écrira seulement x).

Exemple :



Proposition 3.2.1.6.- Soit $G = (N,T,P,X)$ une C_1-bigrammaire et r un squelette sur T appartenant à $\widehat{T \cup [1,n]}$. Si de $s \in \widehat{T \cup N}$ dérive avec p réécriture s' et si $s = r(r_1,r_2,\ldots,r_n)$, alors s' peut s'écrire $s' = r(s_1,s_2,\ldots,s_n)$, chaque s_i dérivant de r_i avec p_i réécriture. De plus, on peut imposer $\sum_{i=1}^{n} p_i \leq p$. (Si chaque i de $[1,n]$ apparaît effectivement dans r, on a $\sum_{i=1}^{n} p_i = p$).

Réciproquement, si $s = r(r_1,r_2,\ldots,r_n)$ et $s' = r(s_1,\ldots,s_n)$ et si chaque s_i dérive de r_i avec p_i réécritures et enfin si chaque i de $[1,n]$ apparaît effectivement dans r, alors s' dérive de s avec $p = \sum_{i=1}^{n} p_i$ reécritures.

Démonstration.- Pour démontrer cette proposition dans le sens direct, on fait une récurrence sur p et on obtient facilement le résultat. Pour la réciproque, une récurrence sur r donne immédiatement le résultat.

Corollaire.- Soit $G = (N,T,P,X)$ une C_1-bigrammaire.

1°) $r \overset{p}{\twoheadrightarrow} s$ et $r = r_1 + r_2 \iff (\exists s_1,s_2,p_1,p_2)(r_1 \overset{p_1}{\twoheadrightarrow} s_1$ et $r_2 \overset{p_2}{\twoheadrightarrow} s_2$
et $p_1 + p_2 = p$ et $s_1+s_2 = s)$.

2°) Soit r_1 et r_2 deux polynômes sur $\widehat{N \cup T_1}$
$r_1 \circ r_2 \overset{p}{\twoheadrightarrow} s \iff (\exists s_1,s_2,p_1,p_2)(r_1 \overset{p_1}{\twoheadrightarrow} s_1$ et $r_2 \overset{p_2}{\twoheadrightarrow} s_2$ et $p_1+p_2 = p$
et $s_1 \circ s_2 = s)$.

3°) $a \times r' \overset{p}{\twoheadrightarrow} s$ et $\in T \Longleftrightarrow (\exists s')(r' \overset{p}{\twoheadrightarrow} s'$ et $s = a \times s')$.

Ces trois équivalences résultent de 3.2.1.6. en prenant comme squelettes respectivement $1 + 2$, 1×2 , $a \times 1$.

Nous allons maintenant caractériser les bilangages engendrés par les C_1-bigrammaires en terme de système à point fixe. Pour traduire dans un système les règles du type $(A \times 1, B \times C \times 1)$, on aimerait introduire une loi interne dans $\hat{V}$, prolongeant la loi externe $\times$ telle que

$$B \overset{*}{\twoheadrightarrow} r \text{ et } C \overset{*}{\twoheadrightarrow} s \Longleftrightarrow B \times C \overset{*}{\twoheadrightarrow} r \times s \ .$$

Intuitivement, la loi $\times$ doit "greffer" en dessous d'une certaine feuille de r la ramification s . Une étude sur des cas particuliers montre que l'on ne peut pas, à priori, particulariser une feuille de r en dessous de laquelle on greffera s pour obtenir $r \times s$, cela dépend en effet de la forme de la dérivation $B \overset{*}{\twoheadrightarrow} r$.(Dans Quéré [20], on a étudié, sous le nom de bilangage algébrique, les bilangages obtenus en faisant toujours la greffe sous la première feuille de r . Ce cas correspond à des C_1-bigrammaires qui sous forme réduite n'ont pas de règle de la forme $(A \times 1 \ , \ B+C \times 1)$). Pour tourner cette difficulté, on va considérer les bilangages $P(A) = \{r \ ; \ r \in T_1$ et $A \times 1 \overset{*}{\twoheadrightarrow} r\}$; en vertu du corollaire de 3.2.1.6., on a $A \times 1 \overset{*}{\twoheadrightarrow} r$ et $B \times 1 \overset{*}{\twoheadrightarrow} s \Longrightarrow A \times B \times 1 \overset{*}{\twoheadrightarrow} r \circ s$.(Sa variable 1 joue le rôle d'un marqueur indiquant sous quelle feuille de r on doit greffer s , la composition $\circ$ des polynômes donne la loi $\times$ cherchée). Etudions d'abord la liaison entre les bilangages $P(A)$ et les bilangages $L(A) = \{r \ ; \ r \in \hat{T}$ et $A \overset{*}{\twoheadrightarrow} r\}$.

Proposition 3.2.1.7.- Soit $G = (N,T,P,X)$ une C_1-bigrammaire. La fonction $c : \hat{V}_1 \longrightarrow \hat{V}$ est la fonction corps d'un polynôme (cf. 1.1.7.). Soit A un non-terminal de G .

$$A \times 1 \overset{*}{\twoheadrightarrow} r \Longrightarrow r \in \widehat{N \cup T_1} \text{ et } A \overset{*}{\twoheadrightarrow} c(r)$$

$$A \overset{*}{\twoheadrightarrow} s \Longrightarrow (\exists\, r \in \widehat{N \cup T_1})(A \times 1 \overset{*}{\twoheadrightarrow} r \text{ et } c(r) = s).$$

En particulier $L(A) = c(P(A))$ avec $L(A) = \{r \ ; \ r \in \hat{T}$ et $A \overset{*}{\twoheadrightarrow} r\}$ et $P(A) = \{r \ ; \ r \in \hat{T}_1$ et $A \times 1 \overset{*}{\twoheadrightarrow} r\}$.

Démonstration.- Ces deux assertions se démontrent par récurrence sur la longueur des dérivations.

$$- A \times 1 \overset{p}{\twoheadrightarrow} r \Longrightarrow r \in \widehat{N \cup T_1} \text{ et } A \overset{*}{\twoheadrightarrow} c(r) \ .$$

Pour $p = 0$, cette implication est évidente. Supposons-la démontrée jusqu'au rang p . Supposons que $A \times 1 \overset{p+1}{\twoheadrightarrow} r$. Donc $A \times 1 \overset{p}{\twoheadrightarrow} t \twoheadrightarrow r$. En appliquant l'hypothèse de récurrence, on obtient $t \in \widehat{N \cup T_1}$ et $A \overset{p}{\twoheadrightarrow} c(t)$. Pour ne pas

confondre la variable 1 de r et les variables des polynômes utilisés dans les réécritures, nous utiliserons des polynômes u de $\widehat{N \cup T_2}$. De plus u^2 désignera ici la fonction de $\widehat{N \cup T_1}$ dans $\widehat{N \cup T_1}$ associée à la deuxième variable de u et $c(u)$ sera le polynôme obtenu en supprimant la variable 1 si elle apparaît dans u . Comme $t \twoheadrightarrow r$, pour une certaine production $(B \times 1, s)$ de G et un polynôme $u \in \widehat{N \cup T_2}$, on a

$$t = u^2(B \times t') \text{ et } r = u^2(s^1(t')) \ .$$

Donc $r \in \widehat{N \cup T_1}$ et suivant que 1 apparaît dans le mot des feuilles de u ou de t' on a respectivement :

$$c(t) = c(u)^2(B \times t') \text{ et } c(r) = c(u)^2(s^1(t'))$$

$$\text{ou } c(t) = u^2(B \times c(t')) \text{ et } c(r) = u^2(s^1(c(t'))) \ .$$

Dans les deux cas, on obtient $c(t) \twoheadrightarrow c(r)$ et donc $A \overset{p+1}{\twoheadrightarrow} c(r)$.

$$- A \overset{p}{\twoheadrightarrow} s \Longrightarrow (\exists\, r)(A \times 1 \overset{p}{\twoheadrightarrow} r \text{ et } c(r) = s).$$

Pour $p = 0$, on prend $r = A \times 1$. Supposons cette implication démontrée pour p . Supposons que $A \overset{p+1}{\twoheadrightarrow} s$. Donc $A \overset{p}{\twoheadrightarrow} t \twoheadrightarrow s$. En appliquant l'hypothèse de récurrence, on trouve un polynôme $r' \in \widehat{N \cup T_1}$ tel que $A \times 1 \overset{p}{\twoheadrightarrow} r'$ et $c(r') = t$. Avec les mêmes conventions que ci-dessus pour les polynômes u , on obtient pour une certaine production $(B \times 1, v)$ de G , et un polynôme u de $\widehat{N \cup T_2}$

$$t = u^2(B \times t') \text{ et } s = u^2(v^1(t')) \ .$$

Donc $r' = u'^2(B \times t')$ ou $r' = u^2(B \times t'')$ suivant la place de 1 dans r' .
Donc $r = u'^2(v^1(t'))$ ou $r = u^2(v^1(t''))$ est tel que $r' \twoheadrightarrow r$ et $c(r) = s$.

Proposition 3.2.1.8.- (Système à point fixe associé aux C_1-bigrammaires). Considérons l'ensemble des polynômes $\hat{T}_1$ à une variable sur T . $(\mathcal{P}(\hat{T}_1))^n$ est un treillis complet pour la relation d'inclusion et les fonctions suivantes sont continues :

$f_1 : (\mathcal{P}(\hat{T}_1))^2 \longrightarrow \mathcal{P}(\hat{T}_1)$ telle que $f_1(A,B) = A \cup B$

$f_2 : (\mathcal{P}(\hat{T}_1))^2 \longrightarrow \mathcal{P}(\hat{T}_1)$ telle que $f_2(A,B) = c(A) + B = \{c(r) + s \ ; \ r \in A$ et $s \in B\}$

$f_3 : (\mathcal{P}(\hat{T}_1))^2 \longrightarrow \mathcal{P}(\hat{T}_1)$ telle que $f_3(A,B) = A + c(B) = \{r + c(s), \ r \in A$ et $s \in B\}$

$f_4 : (\mathcal{P}(T_1))^2 \longrightarrow \mathcal{P}(\hat{T}_1)$ telle que $f_4(A,B) = A \circ B = \{r \circ s; \ r \in A$ et $s \in B\}$

$P_i : (\mathfrak{P}(T_1))^n \longrightarrow \mathfrak{P}(T_1)$ telle que $P_i(A_1,\ldots,A_n) = A_i$

$f_a : (\mathfrak{P}(T_1))^n \longrightarrow \mathfrak{P}(T_1)$ telle que $f_a(A) = \{a \times 1\}$.

De plus, ces fonctions transforment des ensembles de polynômes injectifs en des ensembles de polynômes injectifs.

Les systèmes S sur $(\mathfrak{P}(\hat{T}_1))^n$ de la forme

$$S : X_i = F_i(X_1,\ldots,X_n) \qquad 1 \leqslant i \leqslant n$$

où les F_i sont obtenus par composition des fonctions ci-dessus sont appelés C_1-systèmes. Tout système de ce type admet des solutions dans $(\mathfrak{P}(\hat{T}_1))^n$, et de plus, la solution minimale est formée d'un n-uple d'ensembles de polynômes injectifs.

Un bilangage L est engendré par une C_1-bigrammaire si et seulement si il existe un tel système dont la solution minimale sont $(L_1,L_2,\ldots,L_n)$ et $c(L_1) = L$.

Démonstration.- La première partie de la proposition est évidente.

Soit S un C_1-système. Soit $\Phi : (\mathfrak{P}(\hat{T}_1))^n \longrightarrow (\mathfrak{P}(\hat{T}_1))^n$ défini par

$$\Phi(X_1,\ldots,X_n) = (F_1(X_1,\ldots,X_n) ,\ldots, F_n(X_1,\ldots,X_n) .$$

La solution minimale de S est $\bigcup_{n \geqslant 0} \Phi^n(\phi,\phi,\ldots,\phi)$. Il est immédiat, par récurrence sur n , que $\Phi^n(\phi,\phi,\ldots,\phi)$ est un n-uple d'ensembles de polynômes injectifs. Donc la solution minimale de S est formée d'ensembles de polynômes injectifs sur T .

En utilisant le résultat rappelé au chapitre 0 , tout C_1-système est équivalent à un C_1-système de la forme

$$S : X_i = \bigcup_{j=1}^{P_i} G_{ij}(X_1,\ldots,X_n) \quad \text{où } G_{ij} \text{ est l'une des fonctions de base autre que l'union.}$$

A tout C_1-système S de cette forme, on peut associer une C_1-bigrammaire $G = (N,T,P,X)$ sous forme réduite et réciproquement en posant

- $N = \{X_1 , X_2 ,\ldots, X_n\}$
- $X = X_1$

- $(X_i \times 1, X_k + X_{k'} \times 1) \in P \iff (\exists j \in [1,p_i])(G_{ij}(X_1,\ldots,X_n) = c(X_k) + X_{k'})$

$(X_i \times 1, X_k \times 1 + X_{k'}) \in P \iff (\exists j \in [1,p_i])(G_{ij}(X_1,\ldots,X_n) = X_k + c(X_{k'}))$

$(X_i \times 1, X_k \times X_{k'} \times 1) \in P \iff (\exists j \in [1,p_i])(G_{ij}(X_1,\ldots,X_k) = X_k \circ X_{k'})$

$(X_i \times 1, a \times 1) \in P \iff (\exists j \in [1,p_i])(G_{ij}(X_1,\ldots,X_n) = \{a \times 1\})$

Posons $P(X_i) = \{r \; ; \; r \in \hat{T}_1 \text{ et } X_i \times 1 \overset{*}{\Longrightarrow} r\}$. En appliquant le corollaire de la proposition 3.2.0.6., on obtient immédiatement que $(p(X_1) ,\ldots, p(X_n))$ est une solution de S . D'autre part, la solution minimale de S est $\bigcup_{n \geqslant 0} \Phi^n(\phi,\phi,\ldots,\phi)$ et on obtient en faisant une récurrence sur p que

$$(\forall p \in \mathbb{N})(X_i \times 1 \overset{p}{\Longrightarrow} r \in \hat{T}_1 \Longrightarrow (\exists n \in \mathbb{N})(r \in p_i(\Phi^n(\phi,\phi,\ldots,\phi)))) .$$

Donc $(p(X_1) ,\ldots, p(X_n))$ est la solution minimale de S . D'après la proposition 3.2.1.7., on a $BL(G) = c(P(X_1))$.

3.2.2.- Langages engendrés par une C_1 bigrammaire.

Proposition 3.2.2.1.- Soit $G = (N,T,P,X)$ une C_1- bigrammaire. Le langage $L = \{\alpha; \exists r \in BL(G)$ et $\alpha = \rho(r)\}$ est un langage algébrique.

Démonstration.- On peut supposer G sous forme réduite. Considérons la grammaire à contexte libre $\mathcal{G} = (N,T,::=,X)$ telle que

$$A ::= BC \iff (A\times 1, B\times 1+C) \in P \text{ ou } (A\times 1, B+C\times 1) \in P$$

$$A ::= B \iff (\exists C \in N)\ (A\times 1, B\times C\times 1) \in P)$$

$$A ::= a \iff (A\times 1, a\times 1) \in P\ .$$

On obtient immédiatement

$$r \underset{G}{\gg} s \implies \rho(r) = \rho(s) \text{ ou } \rho(r) \underset{\mathcal{G}}{\succ} \rho(s)$$

$$\alpha \underset{\mathcal{G}}{\succ} \beta \text{ et } \alpha = \rho(r) \implies (\exists s)\ (\beta = \rho(s) \text{ et } r \underset{G}{\gg} s)\ .$$

Donc le langage L est engendré par $\mathcal{G}$.

Proposition 3.2.2.2.- Soit $G = (N,T,P,X)$ une C_1-bigrammaire. Soit L le bilangage engendré par G , alors $Ch(L) = \{\alpha;(\exists r \in L)\ (\alpha \in Ch(r))\}$ est un langage algébrique. Réciproquement, tout langage algébrique ne contenant pas le mot vide est l'ensemble des chemins d'un langage engendré par une C_1-bigrammaire.

Démonstration.- La réciproque est évidente. Pour établir la proposition directe, on peut supposer G sous forme réduite. Modifions légèrement la bigrammaire G pour mettre en évidence les non-terminaux qui apparaissent aux feuilles dans une dérivation à partir de X . Pour cela, considérons la bigrammaire $G' = (N \cup N', T, P', X)$ où N' est un vocabulaire disjoint de $N \cup T$ et en bijection avec N par la bijection $A \longmapsto A'$ et où P' est défini par

$$(A\times 1, B\times 1+C) \in P \iff (A\times 1, B\times 1+C) \in P' \text{ et } (A'\times 1, B'\times 1+C) \in P'$$

$$(A\times 1, B+C\times 1) \in P \iff (A\times 1, B+C\times 1) \in P' \text{ et } (A'\times 1, B+C'\times 1) \in P'$$

$$(A\times 1, a\times 1) \in P \iff (A\times 1, a\times 1) \in P' \text{ et } (A'\times 1, a\times 1) \in P'$$

$$(A\times 1, B\times C\times 1) \in P \iff (A\times 1, B'\times C\times 1) \in P' \text{ et } (A'\times 1, B'\times C'\times 1) \in P'\ .$$

On obtient alors

$$X \underset{G}{\overset{*}{\gg}} r \iff X \underset{G'}{\overset{*}{\gg}} r'$$

où r' est obtenu à partir de r en remplaçant chaque non-terminal de G apparaissant dans r ailleurs qu'à une feuille par le non-terminal de N' correspondant. En particulier $BL(G) = BL(G')$.

Considérons la grammaire algébrique $\mathcal{G} = (N \cup N' \cup N_1, T, ::=, X)$ où N_1 est un vocabulaire disjoint de N , N' et T en bijection avec N et N' par les bijections $A_1 \overset{\longmapsto}{\longleftarrow} A \overset{\longmapsto}{\longleftarrow} A'$ et où $::=$ est défini par

1°) $A ::= B \iff (\exists C)\ ((A\times 1, B\times 1+C) \in P' \text{ ou } (A\times 1, B+C\times 1) \in P')$

2°) $A' ::= B' \iff (\exists C)\ ((A'\times 1, B'\times 1+C) \in P' \text{ ou } (A'\times 1, C+B'\times 1) \in P')$

3°) $A ::= a \iff (A\times 1, a\times 1) \in P'$

4°) $A' ::= a \iff (A'\times 1, a\times 1) \in P'$

5°) $A' ::= B'C' \iff (A'\times 1, B'\times C'\times 1) \in P'$

6°) $A ::= B'C \iff (A\times 1, B'\times C\times 1) \in P'$

7°) $A ::= B_1 \iff (\exists C)\ (A\times 1, B'\times C\times 1) \in P')$

8°) $A_1 ::= B \iff (\exists C)\ [(A'\times 1, C'\times 1+B) \in P' \text{ ou } (A'\times 1, B+C'\times 1) \in P']$

9°) $A_1 ::= B_1 \iff (\exists C)\ [(A'\times 1, B'\times 1+C) \in P' \text{ ou } (A'\times 1, C+B'\times 1) \in P'$
ou $(A'\times 1, B'\times C'\times 1) \in P']$

10°) $A_1 ::= B'C_1 \iff (A'\times 1, B'\times C'\times 1) \in P'$.

La signification des règles 1 à 6 est à peu près évidente. La règle 7 permet de mémoriser qu'à cet endroit on pourra obtenir une "bifurcation" dans les chemins de la ramification si on peut appliquer une production du type $(B'\times 1, C+D'\times 1)$ ou $(B'\times 1, C'\times 1+D)$. La règle 8 est utilisée quand cette "bifurcation" existe. Les règles 9 et 10 conservent un non-terminal de N_1 jusqu'à ce que l'on puisse appliquer une règle 8.

Démontrons que $\mathcal{G}$ engendre $Ch(L)$.

Pour $r \in N \cup N' \cup T$, posons

$$Ch'(r) = \{\alpha;(\exists \beta \in Ch(r))\ (\exists A' \in N')\ (\exists \alpha')\ (\exists \alpha'')\ (\beta = \alpha' A' \alpha'' \text{ et } \alpha = \alpha' A_1)\}\ .$$

Montrons que

$$A \underset{G'}{\overset{*}{\gg}} r \implies (\forall \alpha)\ [(\alpha \in Ch(r) \text{ ou } \alpha \in Ch'(r)) \implies A \underset{\mathcal{G}}{\overset{*}{\succ}} \alpha] \quad (1)\ .$$

La démonstration de (1) se fait par récurrence sur la longueur p de la dérivation $A \underset{G'}{\overset{*}{\gg}} r$.

Si $p = 0$, on a $Ch(r) = \{A\}$ et $Ch'(r) = \emptyset$. Dans $\mathcal{G}$, on a bien $A \underset{\mathcal{G}}{\overset{*}{\succ}} A$.

Supposons (1) démontrée jusqu'à l'ordre p et supposons $A \overset{p+1}{\underset{G'}{\Longrightarrow}} r$. Donc pour un certain r' , on a

$$A \overset{p}{\underset{G'}{\Longrightarrow}} r' \underset{G'}{\longrightarrow} r .$$

Examinons les deux cas où la réécriture $r' \underset{G'}{\longrightarrow} r$ affecte ou non un non-terminal qui est une feuille de r' .

1°) Cette réécriture se fait sur un non-terminal qui est une feuille de r' . Donc $r' = u^1(B)$ où u est un polynôme de $\widehat{N \cup N' \cup T_1}$. On notera α_o le chemin de r' arrivant à la feuille B ($\alpha_o = \alpha'_o B$) . r est de l'une des formes suivantes

- $r = u^1(C+D)$ et $(B \times 1, C \times 1 + D) \in P'$ ou $(B \times 1, C + D \times 1) \in P'$.

  On a :

  $$Ch(r) = (Ch(r') - \{\alpha_o\}) \cup \{\alpha'_o C, \alpha'_o D\}$$
  $$Ch'(r) = Ch'(r') .$$

  On a dans $\mathcal{G}$ les deux règles $B ::= C$ et $B ::= D$. Donc $\alpha_o \underset{\mathcal{G}}{\succ} \alpha'_o C$ et $\alpha_o \underset{\mathcal{G}}{\succ} \alpha_o D$. Donc en appliquant l'hypothèse de récurrence à $A \overset{p}{\underset{G'}{\Longrightarrow}} r'$, on obtient le résultat

- $r = u^1(b)$ et $(B \times 1, b \times 1) \in P'$, un raisonnement similaire donne le résultat voulu,

- $r = u^1(C' \times D)$ et $(B \times 1, C' \times D \times 1) \in P'$.

  On a :

  $$Ch(r) = (Ch(r') - \{\alpha_o\}) \cup \{\alpha'_o C' D\}$$
  $$Ch'(r) = Ch(r) \cup \{\alpha'_o C_1\} .$$

Dans $\mathcal{G}$, on a les deux règles $B ::= C'D$ et $B ::= C_1$. Donc on obtient $\alpha_o \underset{\mathcal{G}}{\succ} \alpha'_o C' D$ et $\alpha'_o \underset{\mathcal{G}}{\succ} \alpha'_o C_1$. D'où le résultat en appliquant l'hypothèse de récurrence.

2°) La réécriture $r' \underset{G'}{\longrightarrow} r$ affecte un non-terminal qui n'est pas une feuille de r . Donc $r' = u^1(B' \times s)$ où u est un polynôme de $\widehat{N \cup N' \cup T_1}$, $B' \in N'$ et s est une ramification non vide.

Soit $\mathcal{E}$ l'ensemble des chemins de r' passant par B' , on a

$$\mathcal{E} = \{\alpha_o B' \beta_i ; 1 \leq i \leq n\} .$$

Soit $\mathcal{E}' = \{\alpha_o B_1\} \cup \{\alpha_o B' \beta_i \beta'_1 E_1 ; (\exists \beta''_1)(\alpha_o B' \beta_i E' \beta''_1 \in \mathcal{E})\}$.

(Intuitivement, $\mathcal{E}'$ est l'ensemble des mots de $Ch'(r')$ passant par B') .

La ramification r est de l'une des formes suivantes :

- $r = u^1(C+D' \times s)$ et $(B' \times 1, C+D' \times 1) \in P$. Dans ce cas, on obtient

  $$Ch(r) = (Ch(r') - \mathcal{E}) \cup \{\alpha_o D' \beta_i ; 1 \leq i \leq n\} \cup \{\alpha_o C\}$$
  $$Ch'(r) = (Ch'(r') - \mathcal{E}') \cup \{\alpha_o D_1\} \cup \{\alpha_o D' \beta_i \beta'_1 E_1 ; \alpha_o B' \beta_i \beta'_1 E_1 \in \mathcal{E}'\} .$$

  Dans $\mathcal{G}$, on a les règles $B' ::= D' | C$, $B_1 ::= D_1 | C$.

  Donc $\alpha_o B' \beta_i \underset{\mathcal{G}}{\succ} \alpha_o D' \beta_o$, $\alpha_o B_1 \underset{\mathcal{G}}{\succ} \alpha_o C$ et comme $\alpha_o B_1 \in Ch'(r')$, on obtient en appliquant l'hypothèse de récurrence que tout α de $Ch(r)$ dérive de A dans $\mathcal{G}$. On obtient de même que tout α de $Ch'(r)$ dérive de A dans $\mathcal{G}$.

- $r = u^1(C' \times s+D)$ et $(B' \times 1, C' \times 1+D) \in P'$. La démonstration est exactement la même que dans le premier cas.

- $r = u^1(C' \times D' \times s)$ et $(B' \times 1, C' \times D' \times 1) \in P'$. Dans ce cas, on obtient

  $$Ch(r) = (Ch(r') - \mathcal{E}) \cup \{\alpha_o C' D' \beta_i ; 1 \leq i \leq n\}$$
  $$Ch'(r) = (Ch'(r') - \mathcal{E}') \cup \{\alpha_o C_1, \alpha_o C' D_1\} \cup \{\alpha_o C' D' \beta'_1 E_1 ; \alpha_o D \beta'_1 E_1 \in \mathcal{E}'\} .$$

  Dans $\mathcal{G}$, on a les règles $B' ::= C'D'$, $B_1 ::= C_1 | C'D_1$. En appliquant l'hypothèse de récurrence, on obtient immédiatement le résultat cherché.

- $r = u^1(b \times s)$ et $(B' \times 1, b \times 1) \in P'$. Dans ce cas, on obtient

  $$Ch(r) = (Ch(r') - \mathcal{E}) \cup \{\alpha_o b \beta_i ; 1 \leq i \leq n\}$$
  $$Ch'(r) = (Ch'(r') - \mathcal{E}') \cup \{\alpha_o a \beta'_1 E_1 ; \alpha_o B' \beta'_1 E_1 \in \mathcal{E}'\} .$$

  Dans $\mathcal{G}$, on a la règle $B' ::= b$, d'où le résultat.

Donc (1) est démontrée par récurrence. On déduit de (1) que

$$r \in BL(G') \text{ et } \alpha \in Ch(r) \Longrightarrow X \overset{*}{\underset{\mathcal{G}}{\succ}} \alpha .$$

Donc $Ch(BL(G')) \subset L(\mathcal{G})$ ($L(\mathcal{G})$ désigne le langage engendré par $\mathcal{G}$) .

Pour démontrer l'inclusion réciproque, montrons que

(2) $$A \overset{*}{\underset{\mathcal{G}}{\succ}} \alpha \in (N \cup N_1 \cup T)^* \Longrightarrow (\exists r)\ (A \overset{*}{\underset{G'}{\Longrightarrow}} r \text{ et } \alpha \in Ch(r)) .$$

La démonstration de (2) se fait par récurrence sur la longueur p de la dérivation $A \overset{*}{\underset{\mathcal{G}}{\succ}} \alpha$. Pour $p = 0$, le résultat est immédiat.

Supposons (2) démontrée jusqu'à l'ordre p et supposons que $A \overset{p+1}{\underset{g}{\succ}} \alpha$. Pour un certain α' , on a donc $A \overset{p}{\underset{g}{\succ}} \alpha' \underset{g}{\succ} \alpha$.

1er cas : $\alpha' \in (N \cup N' \cup T)^*$. En appliquant l'hypothèse de récurrence, on trouve une ramification r' telle que $A \overset{*}{\underset{G'}{\gg}} r'$ et $\alpha' \in Ch(r')$. En examinant chacune des réécritures possibles de α' , on obtient facilement qu'elles correspondent à des réécritures $r' \succ\!\!- r$ dans G' telles que $\alpha \in Ch(r)$.

2ème cas : $\alpha' \notin (N \cup N' \cup T)^*$. Donc un élément de N_1 a une occurrence dans α' . En examinant les règles de g , on obtient immédiatement que cet élément de N_1 est nécessairement la dernière lettre de α' . D'autre part, parmi les dérivations de A à α' dans g , l'une au moins de ces dérivations est de la forme

$$A = \alpha_0 \succ\!\!- \alpha_1 \succ\!\!- \alpha_2 \succ\!\!- \cdots \succ\!\!- \alpha_k \succ\!\!- \alpha_{k+1} \succ\!\!- \cdots \succ\!\!- \alpha_n = \alpha' \succ\!\!- \alpha_{n+1} = \alpha$$

avec $\alpha_k \in (N \cup N' \cup T)^*$ et chaque réécriture $\alpha_j \succ\!\!- \alpha_{j+1}$ pour $j = k , k+1 \ldots n$ opère sur la dernière lettre de α_j en utilisant un non-terminal de N_1 .

Plus exactement, $\alpha_k \underset{g}{\succ} \alpha_{k+1}$ s'obtient en utilisant une règle du type 7 sur la dernière lettre de α_k , pour $j = k+1 , \ldots , n-1$ $\alpha_j \underset{g}{\succ} \alpha_{j+1}$ s'obtient en utilisant des règles du type 9 ou 10 sur la dernière lettre de α_j et $\alpha_n \succ\!\!- \alpha_{n+1}$ s'obtient en utilisant une règle du type 8 sur la dernière lettre de α_n .

Par hypothèse de récurrence, il existe une ramification r_k telle que

$$A \overset{*}{\underset{G'}{\gg}} r_k \quad \text{et} \quad \alpha_k \in Ch(r_k) .$$

On en déduit qu'il existe une ramification r_{k+1} telle que $\alpha_{k+1} \in Ch'(r_{k+1})$ (Ch' est la fonction introduite ci-dessus).

En examinant les règles 9 et 10, on en déduit que

$$\alpha \underset{g}{\succ} \beta \quad \text{et} \quad \alpha \in Ch'(r) \Longrightarrow (\exists s)\ (r \overset{}{\underset{G'}{\gg}} s \quad \text{et} \quad \beta \in Ch'(s))$$

quand $\alpha \succ\!\!- \beta$ est une réécriture obtenue en utilisant une règle du type 9 ou 10 sur la dernière lettre de α .

Donc il existe des ramifications r_j $(k+1 \leq j \leq n)$ telles que

$$r_{k+1} \underset{G'}{\succ} r_{k+2} \succ\!\!- \cdots \succ\!\!- r_n$$

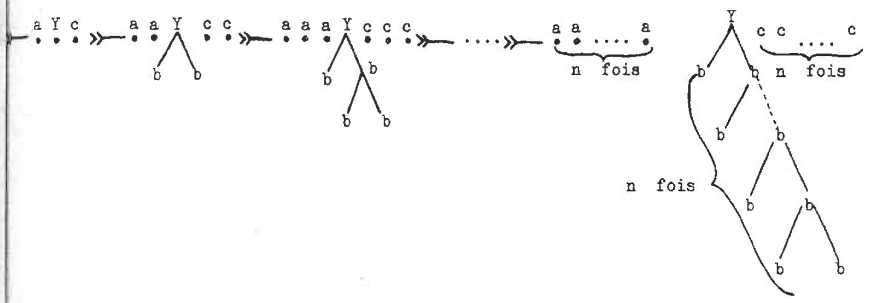
et $$\forall j \in [k+1, n] \ (\alpha_j \in Ch'(r_j)) .$$

La réécriture $\alpha_n \succ\!\!- \alpha_{n+1}$ s'obtient en utilisant une règle du type 8 sur la dernière lettre de α_n . D'après la forme de ces règles, si $\alpha_n \in Ch'(r_n)$, alors il existe une ramification r_{n+1} telle que $\alpha_{n+1} \in Ch(r_{n+1})$ et $r_n \succ\!\!- r_{n+1}$. Donc on a construit une ramification r_{n+1} telle que $A \overset{*}{\underset{G'}{\gg}} r_{n+1}$ et $\alpha = \alpha_{n+1} \in Ch(r_{n+1})$. C. Q. F. D.

Proposition 3.2.2.3.- Les langages engendrés par les C_1-bigrammaires contiennent strictement les langages algébriques.

Démonstration.- Il est évident que tout langage algébrique est engendré par une C_1-bigrammaire (par exemple en utilisant les propositions 3.1.1.5 et 3.2.1.3). Il existe des langages engendrés par des C_1-bigrammaires qui ne sont pas algébriques. Par exemple, considérons la bigrammaire $G = (\{X,Y\},\{a,b,c\},P,X)$ telle que $P = \{(X \times 1, a+(Y \times 1)+c),(Y \times 1, b \times 1),(Y \times 1, a+Y \times (b+b \times 1)+c)\}$.

On obtient immédiatement qu'une dérivation dans G est de la forme



Donc le langage engendré par cette bigrammaire est $\{a^n b^n c^n ; n \geqslant 1\}$ qui n'est pas un langage algébrique.

Proposition 3.2.2.4.- (non démontrée) Les langages engendrés par les C_1-bigrammaires sont des langages contextuels.

Cette proposition n'a pas pu être démontrée jusqu'à présent. La difficulté provient de ce que les règles des C_1-bigrammaires peuvent augmenter le nombre de points d'une ramification sans augmenter la longueur du mot des feuilles. Pour savoir si un mot α sur T est dans le langage engendré par une C_1-bigrammaire G , il faut rechercher dans $BL(G)$ une ramification r telle que $\varphi(r) = \alpha$. A priori, on ne peut pas majorer le nombre de points d'une ramification r de $BL(G)$ susceptible d'avoir α comme mot des feuilles. Pour démontrer que $L(G)$ est un langage contextuel, il suffirait de démontrer un lemme du genre

$$(\forall \alpha \in L(G))\ (\exists r \in BL(G))\ (\varphi(r) = \alpha \quad \text{et} \quad n(r) \leq p|\varphi(r)|)$$

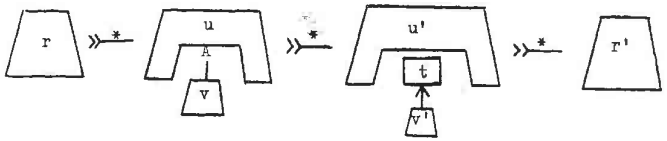
où p est un entier ne dépendant que de G. La démonstration d'un tel lemme doit être similaire aux techniques employées dans la démonstration de la proposition 3.2.2.6 et des lemmes précédant cette proposition.

On peut se demander si l'on obtient avec les C_1-bigrammaires tous les langages contextuels. Il n'en est rien comme le montre la suite.

Pour le voir, nous allons démontrer un lemme analogue au théorème des paires itérantes [6] pour les grammaires à conteste libre (si $\mathcal{G} = (N,T,::=,X)$ est une grammaire algébrique, il existe deux entiers p et q ne dépendant que de G tels que tout mot z engendré par $\mathcal{G}$ et vérifiant $|z| \geqslant p$ s'écrive $z = u\,v\,w\,x\,y$ avec $|v\,w\,x| \leq q$ et $v\,x \neq \Lambda$ et pour tout $n \in \mathbb{N}$, $u\,v^n\,w\,x^n\,y$ est engendré par $\mathcal{G}$).

Définition 3.2.2.5.- Soit $G = (N,T,P,X)$ une C_1-bigrammaire. Soient r et r' deux ramifications de $\widehat{N \cup T}$ telles que $r \overset{*}{\gg} r'$.

On dit que le non-terminal A est auto-imbriqué dans une dérivation menant de r à r' si A apparaît dans cette dérivation et si A réapparaît dans la suite de la dérivation dans une sous-ramification qui dérive strictement de la première occurrence de A. Graphiquement, cela s'écrit



avec $u \overset{*}{\gg} u'$, $v \overset{*}{\gg} v'$, $A \times 1 \overset{*}{\gg} t$, $t \neq A \times 1$ et A apparaît dans t.

Donc formellement, on obtient : A est auto-imbriqué dans une dérivation menant de r à r' si on peut trouver des termes s et s' de cette dérivation et des ramifications $u \in \widehat{T \cup N_1}$, $v \in \widehat{T \cup N}$, $u' \in \widehat{T \cup N_1}$, $v' \in \widehat{T \cup N}$, $t \in \widehat{T \cup N_1}$ telles que :

- $r \overset{*}{\underset{G}{\gg}} s \overset{*}{\underset{G}{\gg}} s' \overset{*}{\underset{G}{\gg}} r'$
- $s = u^1(A \times v)$
- $s' = u'^1(t^1(v'))$
- $u \overset{*}{\underset{G}{\gg}} u'$, $v \overset{*}{\underset{G}{\gg}} v'$, $A \times 1 \overset{*}{\underset{G}{\gg}} t$
- $t \neq A \times 1$ et $F_A(t) \neq \emptyset$.

On dira que A est fortement auto-imbriqué si, de plus, on a $|\varphi(t)| \geqslant 2$.

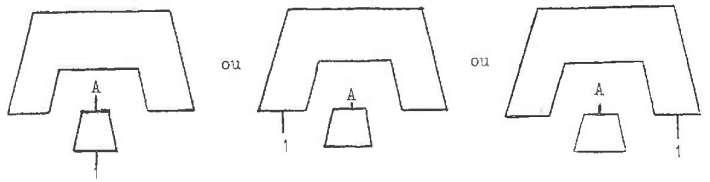
Les démonstrations qui suivent montrent que pour tout r dérivant de X, il existe une dérivation de X à r contenant un non-terminal auto-imbriqué (respectivement fortement auto-imbriqué) dès que $n(r)$ (respectivement $|\varphi(r)|$) est suffisamment grand.

Lemme 1.- Soit $G = (N,T,P,X)$ une C_1-bigrammaire. Soient r et r' tels que r' est dans $\hat{T}$ et r' dérive de r. Si le non-terminal A est auto-imbriqué dans une dérivation menant de r à r', il existe des ramifications s_o et s'_o appartenant à $\widehat{T \cup \{A\}}$, $u_o \in \hat{T}_1$, $v_o \in \hat{T}$, $t_o \in \widehat{T \cup \{A\}}_1$, $w_o \in \hat{T}_2$, $t'_o \in \hat{T}_1$ et $x \in \hat{T}_1$ telles que

1) $r \overset{*}{\underset{G}{\gg}} s_o \overset{*}{\underset{G}{\gg}} s'_o \overset{*}{\underset{G}{\gg}} r'$
2) $s_o = u_o^1(A \times v_o)$
3) $s'_o = u_o^1(t_o^1(v_o))$
4) $t_o = w_o^2(A \times t'_o, 1)$
5) $A \times 1 \overset{*}{\underset{G}{\gg}} t_o$
6) $r' = u_o^1((w_o^2(x^1(t'_o),1))^1(v_o))$
7) $A \times 1 \overset{*}{\gg} x$.

Démonstration.- Le lemme dont l'énoncé est assez lourd procède de l'idée très simple suivante :

En reprenant les notations de la définition 3.2.2.5, on continue dans s et s' les réécritures qui n'affectent pas l'occurrence considérée du non-terminal A pour obtenir des ramifications de $\widehat{T \cup \{A\}}$. L'écriture de t_o (4) met simplement en évidence la présence de A dans t_o, on est contraint d'utiliser un polynôme w_o de $\hat{T}_2$ pour tenir compte des deux cas où la variable 1 de t_o est en-dessous de A et où la variable 1 n'est pas en-dessous de A (graphiquement, t_o peut être d'une des trois formes suivantes :



Remarquons que dans le premier cas, w_o est un polynôme de T_1 et que 2 est une variable inefficace dans w. La démonstration de ce lemme résulte immédiatement de la proposition 3.2.1.6. En reprenant les notations de la définition 3.2.2.5, on obtient $s' = (1 \times (2 \times 3))\ (u',t,v')$. Donc

$$r' = (1 \times (2 \times 3))\ (u_o,\tau,v_o) = u_o^1(\tau^1(v_o)) \quad \text{et} \quad u' \overset{*}{\underset{G}{\gg}} u_o\ ,\ t \overset{*}{\underset{G}{\gg}} \tau\ ,\ v' \overset{*}{\underset{G}{\gg}} v_o\ .$$

Comme t contient une occurrence de A, on peut écrire

$$t = w^2(A \times t',1) \quad \text{avec} \quad w \in \widehat{N \cup T_2} \quad \text{et} \quad t' \in \widehat{N \cup T_1}\ .$$

Donc $\tau = w_o^2(x^1(t'_o),1)$ et $w \overset{*}{\underset{G}{\gg}} w_o$, $A \times 1 \overset{*}{\underset{G}{\gg}} x$, $t' \overset{*}{\underset{G}{\gg}} t'_o$.

Les ramifications $s_o = u_o^1(A \times v_o)$ et $s'_o = u_o^1(t_o^1(v_o))$ avec $t_o^1 = w_o^2(A \times t'_o,1)$ répondent aux conditions.

(Cette démonstration signifie simplement que dans s et s', on a continué jusque dans T les réécritures qui n'affectent pas l'occurrence du non-terminal auto-imbriqué A).

Lemme 2.- Soit $G = (N,T,P,X)$ une C_1-bigrammaire. On suppose G mise sous la forme réduite décrite en 3.2.1.2 et que $\text{card}(N) = k$.

Soit r dérivant de A dans G.

1°) Si $n(r) \geqslant 2^k$, il existe un non-terminal auto-imbriqué dans une dérivation de A à r.

2°) Si $|\varphi(r)| \geqslant 2^k$, il existe un non-terminal fortement auto-imbriqué dans une dérivation de A à r.

Démonstration.- Ces deux assertions se démontrent par récurrence sur k.

1°) Pour $k = 1$, le lemme est évident. Supposons-le démontré pour k et supposons que $\text{card}(N) = k+1$. Dire que A est auto-imbriqué dans la dérivation $A \overset{*}{\underset{G}{\gg}} r$ signifie simplement que A réapparaît dans une réécriture utilisée pour aller de A à r dans G. Si ce cas ne se produit pas, on a, pour un certain r', $A \underset{G}{\gg} r' \overset{*}{\underset{G}{\gg}} r$ et la dérivation $r' \overset{*}{\underset{G}{\gg}} r$ est une dérivation dans la grammaire $G' = (N-\{A\},T,P',X)$ où P' est obtenu en supprimant dans P toutes les productions contenant une occurrence de A.

De plus, on a $r' = B + C$ ou $r' = B \times C$ où B et C sont deux non-terminaux de $N - \{A\}$. ($r' = a$ est incompatible avec l'hypothèse $n(r) \geqslant 2^{k+1}$).

Si $r' = B + C$, on a $r = r_1 + r_2$ et $B \overset{*}{\underset{G'}{\gg}} r_1$ et $C \overset{*}{\underset{G'}{\gg}} r_2$.

Comme $n(r) \geqslant 2^{k+1}$, on obtient $n(r_1) \geqslant 2^k$ ou $n(r_2) \geqslant 2^k$. Le résultat cherché s'obtient alors en appliquant l'hypothèse de récurrence à celles des deux dérivations de G' vérifiant $n(r_i) \geqslant 2^k$.

Si $r' = B \times C$, on a $r = r_1^1(r_2)$ et $B \times A \overset{*}{\underset{G'}{\gg}} r_1$ et $C \overset{*}{\underset{G'}{\gg}} r_2$.

Donc $n(r) = n(r_1) + n(r_2) - 1$ et comme $n(r) \geqslant 2^{k+1}$, on obtient $n(r_1) \geqslant 2^k + 1$ ou $n(r_2) \geqslant 2^k$. De plus, $B \times 1 \overset{*}{\underset{G'}{\gg}} r_1$ implique que $B \overset{*}{\underset{G'}{\gg}} c(r_1)$ et $n(c(r_1)) = n(r_1) - 1$. On obtient alors le résultat en appliquant l'hypothèse de récurrence.

2°) Pour $k = 1$, l'hypothèse $|\varphi(r)| \geqslant 2$ implique que la grammaire G contient l'une des deux productions $(A \times 1, A + A \times 1)$ ou $(A \times 1, A \times 1, A)$ et que l'une de ces deux productions a été utilisée dans la dérivation $A \overset{*}{\underset{G}{\gg}} r$. Donc A est fortement auto-imbriqué dans cette dérivation.

Supposons le lemme démontré pour k et supposons que $\text{card}(N) = k+1$. Si A est fortement auto-imbriqué dans la dérivation $A \overset{*}{\underset{G}{\gg}} r$, le lemme est vérifié. Supposons que ce cas ne se produise pas. Considérons une dérivation $A = r_0 \underset{G}{\gg} r_1 \underset{G}{\gg} \dots \underset{G}{\gg} r_p = r$ de A à r et soit q le plus grand entier de $[0,p]$ tel que A ait une occurrence dans r_q. Donc $r_q = a^1(A \times b)$ où $a \in \widehat{N \cup T_1}$ et $b \in \widehat{N \cup T}$. Donc $r = a'^1(r'^1(b'))$ avec $a' \in \widehat{N \cup T_1}$ et $a \overset{*}{\underset{G}{\gg}} a'$, $A \times 1 \overset{*}{\underset{G}{\gg}} r'$, $b \overset{*}{\underset{G}{\gg}} b'$. On obtient donc

$$A \overset{*}{\underset{G}{\gg}} a'^1(A \times b') \overset{*}{\underset{G}{\gg}} a'^1(r'^1(b')) = r\ .$$

Comme A n'est pas fortement auto-imbriqué dans la dérivation $A \overset{*}{\underset{G}{\gg}} r$, on en déduit que $|\varphi(a'^1(A \times b'))| = 1$ et donc $|\varphi(r')| = |\varphi(r)| \geqslant 2^{k+1}$.

Mais, dans la dérivation $A \times 1 \overset{*}{\underset{G}{\gg}} r'$, A ne réapparaît pas dans les diverses réécritures utilisées pour aller de $A \times 1$ à r' dans G. Pour un certain r'', on a $A \underset{G}{\gg} r'' \overset{*}{\underset{G}{\gg}} c(r')$ et la dérivation $r'' \underset{G}{\gg} r'$ est une dérivation dans la grammaire $G' = (N-\{A\},T,P',X)$ définie comme au cas 1°). La fin de la démonstration est alors identique au 1°).

Proposition 3.2.2.6.- Soit $G = (N,T,P,X)$ une C_1-bigrammaire. Il existe un entier p ne dépendant que de G et vérifiant les deux propriétés suivantes :

1°) Si $r \in BL(G)$ et $n(r) \geqslant p$, alors il existe $u \in \hat{T}_1$, $v \in \hat{T}_2$, $w \in \hat{T}_1$, $x \in \hat{T}_1$, $y \in \hat{T}$ tels que

$$r = u^1((v^2(w^1(x),1))^1(y)) \quad \text{avec} \quad v^2(c(x),\Lambda) \neq \Lambda$$

et si on pose $s_o = w$

$$s_{n+1} = v^2(s_n^1(x),1) ,$$

alors, pour tout $n \in \mathbb{N}$, $r_n = u^1(s_n^1(y))$ est dans $BL(G)$

(voir le graphique de la page 3.35)

2°) Si $z \in L(G)$ et $|z| \geqslant p$, une des propriétés suivantes est vérifiée.

- il existe $u_1, u_2, w_1, v_2, w_1, w_2, x_1, x_2, y'$ dans T^* tels que $z = u_1 v_1 w_1 x_1 y' x_2 w_2 v_2 u_2$ avec $v_1 x_1 x_2 v_2 \neq \Lambda$ et pour tout $n \in \mathbb{N}$, $z_n = u_1 v_1^n w_1 x_1^n y' x_2^n w_2 v_2^n u_2$ est dans $L(G)$.

- il existe $u_1 , u_2 , v_1 , v_2 , v_3 , w_1 , w_2 , x' , y'$ dans T^* tels que $z = u_1 v_1 w_1 x' w_2 v_2 y' v_3 u_2$ avec $v_1 v_2 v_3 x' \neq \Lambda$ et pour tout $n \in \mathbb{N}$, $z_n = u_1 v_1^n w_1 x' w_2 (v_2 x' v_3)^{n-1} v_2 y' v_3 u_2$ est dans $L(G)$.

- il existe $u_1 , u_2 , v_1 , v_2 , v_3 , w_1 , w_2 , x' , y'$ dans T^* tels que $z = u_1 v_1 y' v_2 w_1 x' w_2 v_3 u_2$ avec $v_1 v_2 v_3 x' \neq \Lambda$ et pour tout $n \in \mathbb{N}$, $z_n = u_1 v_1 y' v_2 (v_1 x' v_2)^{n-1} w_1 x' w_2 v_3^n u_2$ est dans $L(G)$.

Démonstration.- On peut supposer dans les deux cas que la grammaire G est mise sous forme réduite. Soient $k = \text{card}(N)$ et $p = 2^k$.

1°) Si $r \in BL(G)$ et $n(r) \geqslant p$, on a $X \overset{*}{\Longrightarrow} r$ et en appliquant le lemme 2, on obtient qu'un non-terminal A de N est auto-imbriqué pour cette dérivation. En appliquant le lemme 1, on obtient

- $r = u^1((v^2(w^1(x),1))^1(y))$
- $X \overset{*}{\Longrightarrow} u^1(A \times y)$
- $A \times 1 \overset{*}{\Longrightarrow} v^2(A \times x,1)$ et $v^2(c(x),\Lambda) \neq \Lambda$
- $A \times 1 \overset{*}{\Longrightarrow} w$.

On en déduit immédiatement par récurrence que les ramifications s_n appartenant à $\hat{T}_1$ et définies par $s_o = w$, $s_{n+1} = v^2(s_n^1(x),1)$ dérivent toutes de $A \times 1$. Comme on a de plus $X \overset{*}{\Longrightarrow} u^1(A \times y)$, on en déduit le résultat annoncé.

Si on analyse le résultat, on s'aperçoit qu'il y a deux cas de figure.

a) La variable 1 apparaît dans x . Dans ce cas, $v \in \hat{T}_1$ et les formules se simplifient en

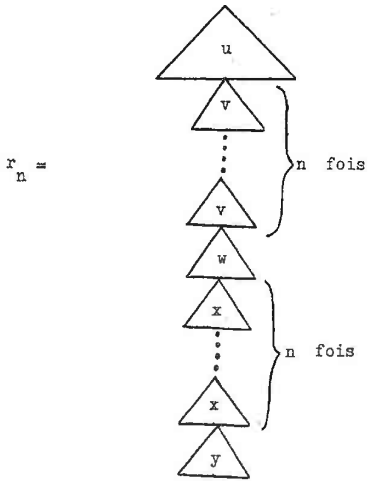
$$r = u^1((v^1(w^1(x)))^1(y))$$

$$s_o = w$$

$$s_{n+1} = v^1(s_n^1(x))$$

et schématiquement, on a

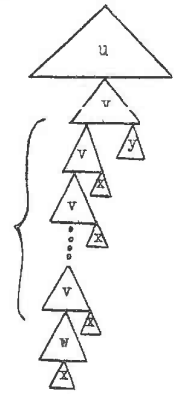
$r_n =$



Ce cas correspond exactement à une généralisation sur les ramifications du théorème $u\ v\ w\ x\ y$ sur les grammaires algébriques.

b) La variable 2 apparaît dans v . Dans ce cas, $x \in \hat{T}$. On obtient alors schématiquement

$r_n =$



2°) Si $z \in L(G)$ et $|z| \geqslant p$, pour un certain r, on a

$$X \overset{*}{\underset{G}{\Longrightarrow}} r \quad \text{et} \quad \varphi(r) = z \quad \text{et} \quad |\varphi(r)| \geqslant p .$$

D'après le lemme 2, on en déduit qu'un non-terminal A de N est fortement auto-imbriqué dans la dérivation $X \overset{*}{\underset{G}{\Longrightarrow}} r$.

Reprenons les notations de la 1ère partie. Le fait que A soit fortement auto-imbriqué ajoute la condition supplémentaire $|\varphi(v^2(A \times x,1))| \geqslant 2$.

Si on est dans le 1er cas de figure cité ci-dessus, on obtient

$$\varphi(u) = u_1 \; 1 \; u_2$$
$$\varphi(v) = v_1 \; 1 \; v_2$$
$$\varphi(w) = w_1 \; 1 \; w_2$$
$$\varphi(x) = x_1 \; 1 \; x_2$$
$$\varphi(y) = y' .$$

Donc $\varphi(r) = u_1 \, v_1 \, w_1 \, x_1 \, y' \, x_2 \, w_2 \, v_2 \, u_2 = z$.

La condition $|\varphi(v^2(A \; x,1))| \geqslant 2$ implique

$$v_1 \, x_1 \, x_2 \, v_2 \neq \Lambda .$$

De plus

$$\varphi(r_n) \in L(G) \quad \text{et}$$
$$\varphi(r_n) = u_1 \, v_1^n \, w_1 \, x_1^n \, y' \, x_2^n \, w_2 \, v_2^n \, u_2 .$$

Si on est dans le 2ème cas de figure, on obtient

$$\varphi(u) = u_1 \; 1 \; u_2$$
$$\varphi(v) = v_1 \; 1 \; v_2 \; 2 \; v_3$$

(ou $\varphi(v) = v_1 \; 2 \; v_2 \; 1 \; v_3$, cas qui se traite de manière similaire et qui correspond à la 3ème possibilité de la proposition 3.2.2.6, 2°))

$$\varphi(w) = w_1 \; 1 \; w_2$$
$$\varphi(x) = x'$$
$$\varphi(y) = y' .$$

On a alors

$$\varphi(r_n) = u_1 \, v_1^n \, w_1 \, x' \, w_2 (v_2 x' v_3)^{n-1} \, v_2 \, y' \, v_3 \, u_2$$
$$\varphi(r) = u_1 \, v_1 \, w_1 \, x' \, w_2 \, v_2 \, y' \, v_3 \, u_2 .$$

De plus $|\varphi(v^2(A \; x,1))| \geqslant 2$ implique $|u_1 v_2 v_3 x'| \neq \Lambda$.

Corollaire.- Il existe des langages contextuels qui ne sont pas engendrés par les C_1-bigrammaires.

Démonstration.- D'après la proposition ci-dessus, si on ordonne par longueur croissante les mots d'un langage engendré par une C_1-bigrammaire, on obtient une suite α_n de mots sur T vérifiant $|\alpha_{n+1}| - |\alpha_n| \leqslant d$ où d ne dépend que de G.

Un langage comme $L = \{a^{n^2} ; n \geqslant 0\}$ ne peut donc être engendré par une C_1-bigrammaire. Il est facile de voir que $L = \{a^n b^n a^n d^n e^n ; n \geqslant 0\}$ ne peut pas non plus être engendré par une C_1-bigrammaire.

Remarque 1 : On pourrait certainement comme dans le cas des grammaires à contexte libre affiner la proposition 3.2.2.6 en montrant qu'il existe un entier q ne dépendant que de G tel qu'on puisse imposer en reprenant les notations de la proposition 3.2.2.6 dans le 1er cas $n(v^2(w^1(x),1)) \leqslant q$ et dans le 2ème cas $|v_1 w_1 x_1 y' x_2 w_2 v_2| \leqslant q$ ou $|v_1 w_1 x' w_2 v_2 y' v_3| \leqslant q$.

Remarque 2 : Des langages et bilangages d'un type approchant ceux étudiés dans ce paragraphe ont été introduits par Avawind K Joshi, Léon S Levy et Masako Takahashi [9]. Dans cet article est introduite la notion de "tree adjunct grammar" qui est un cas particulier de C_1-bigrammaire de la forme suivante :

- l'axiome X est une ramification de $\widehat{N \cup T}$ telle que les éléments de T n'apparaissent qu'aux feuilles, $\varphi(X)$ est dans T^* et $\rho(X)$ est réduit à une lettre de N
- les productions sont de la forme

  $(A \times i, r)$ avec $\rho(r) = A$, $\varphi(r) = \alpha \; i \; \beta$, $\alpha \in T^*$, $\beta \in T^*$ et $i \in F_A(r)$.

De plus, les ramifications utilisées dans cet article ont toutes une racine réduite à une lettre, sont ramifiées à chaque niveau et de plus, le mot des feuilles est dans T^*, c'est-à-dire qu'elles vérifient

- $|\rho(r)| = 1$
- $(\forall A \in N) \in (\alpha \in F_A(r) \Longrightarrow |\alpha| \geq 2)$
- $(\forall a \in T)\ (F_a(r) \in \{\Lambda\})$.

On définit alors le langage engendré comme étant l'ensemble des mots des feuilles des ramifications dérivant de X . On obtient des résultats analogues à ceux énoncés dans ce paragraphe, en particulier les langages engendrés ne sont pas en général algébriques. Cependant, un langage comme $\{a^n b^n c^n ; n \geq 1\}$ ne peut pas être engendré par une "tree adjunct grammar", alors que ce langage est obtenu avec des C_1-bigrammaires. De plus, on ne trouve pas pour ce type de C_1-bigrammaire de proposition analogue à la proposition 3.2.2.6 qui permet de montrer facilement qu'un langage ne peut pas être engendré par une C_1-bigrammaire. D'ailleurs, dans cet article, seule la structure des langages engendrés est étudiée, la structure des bilangages est passée sous silence.

3.3.- Intersection des bilangages engendrés par les C_0-bigrammaires et les C_1-bigrammaires avec des bilangages réguliers.

Il est bien évident que le classe des bilangages engendrée par les C_0-bigrammaires (resp. par les C_1-bigrammaires), n'est pas stable par intersection. En effet, dans les deux cas, si on considère la sous-classe formée par les bilangages ne contenant que des ramifications réduites à leurs racines, on obtient exactement la classe des langages algébriques qui n'est pas stable par intersection.

Cependant, les bilangages réguliers jouent un rôle analogue aux langages réguliers, et on peut espérer que l'intersection d'un bilangage régulier et d'un bilangage engendré par une C_0-bigrammaire (resp. par une C_1-bigrammaire) peut être engendré par une C_0-bigrammaire (resp. par une C_1-bigrammaire).

3.3.1.- Intersection d'un bilangage régulier et d'un bilangage engendré par une C_0-bigrammaire.

Rappel : Quéré [20] a donné la caractérisation suivante des bilangages réguliers :

L est un bilangage régulier dans $\hat{T}$ si et seulement si il existe un T-binoïde fini $\mathcal{B}$ et un sous-ensemble fini $\mathcal{B}'$ tel que si ψ est l'homomorphisme de $\hat{T}$ dans $\mathcal{B}$, alors $L = \psi^{-1}(\mathcal{B}')$.

En utilisant cette caractérisation des bilangages réguliers, on va montrer la proposition suivante :

Proposition 3.3.1.1.- Soit L un bilangage sur T engendré par une C_0-bigrammaire et L' un bilangage régulier sur T ; alors $L \cap L'$ est engendré par une C_0-bigrammaire.

Démonstration.- Soit $G = (N,T,P,X)$ une C_0-bigrammaire réduite engendrant L . Soit $\mathcal{B}$ un V-binoïde fini et $\mathcal{B}'$ un sous-ensemble de $\mathcal{B}$ tel que $L' = \psi^{-1}(\mathcal{B}')$ où ψ est l'homomorphisme de T dans $\mathcal{B}$. Considérons la C_0-bigrammaire $G' = (N \times B \cup \{X'\}, T, P', X')$ telle que

$((A,s)\ ,\ (B,s') + (C,s'')) \in P' \Longleftrightarrow s' + s'' = s$ et $(A, B+C) \in P$

$((A,s)\ ,\ a \times (B,s')) \in P' \Longleftrightarrow a \times s' = s$ et $(A, a \times B) \in P$

$((A,s)\ ,\ \Lambda) \in P' \Longleftrightarrow s$ est l'élément neutre de $\mathcal{B}$ et $(A,\Lambda) \in P'$

$(X'\ ,\ (X,s)) \in P' \Longleftrightarrow s \in \mathcal{B}'$.

De façon évidente, on obtient en raisonnant par récurrence sur la longueur des dérivations

$$(\forall r\in\hat{T})(\forall A\in N)(\forall s\in B)((A,s)\overset{*}{\gg}_{G'} r \Longleftrightarrow A\overset{*}{\gg}_{G} r \text{ et } \psi(r) = s) .$$

Donc G' engendre $L\cap L'$.

3.3.2.- Intersection d'un bilangage régulier et d'un bilangage engendré par une C_1-bigrammaire.

Pour pouvoir démontrer un résultat analogue au précédent à propos des C_1-bigrammaires, il est utile de donner une caractérisation des bilangages réguliers à l'aide d'une structure adaptée au problème. La structure de dioïde introduite par Quéré [20] n'est pas utilisable ici. En effet, les structures de dioïde gauche et droit, introduites dans la thèse de Quéré, correspondent respectivement à des greffes sous la première ou la dernière feuille d'une ramification. Dans les C_1-bigrammaires, la greffe en dessous d'une ramification est effectuée en dessous d'une quelconque des feuilles de la ramification. Pour tenir compte de ce fait, on introduit la notion de dioïde généralisé (en abrégé DG), qui recouvre les notions de dioïde droit et gauche.

Définition 3.3.2.1.- On appelle dioïde généralisé (DG), un ensemble $\mathcal{D}$ muni de trois lois de composition internes notées $\times$, $+_G$, $+_D$ vérifiant les conditions suivantes :

1°) ces trois lois sont associatives

2°) $(s+_G t)\times r = s\times r +_G t$

$(s+_D t)\times r = s +_D t\times r$

3°) il existe dans $\mathcal{D}$ un élément e tel que

e est neutre pour $\times$

$(\forall r\in\mathcal{D})(r+_G e = r)$

$(\forall r\in\mathcal{D})(e+_D r = r)$.

Exemple : L'ensemble $\hat{V}[1]$ des polynômes à une variable 1 sur un vocabulaire V est un DG pour les lois suivantes :

- $\times$ est la composition des polynômes
- $r +_G t = r + c(t)$ (c est la fonction corps)
- $r +_D t = c(r) + t$

(ici l'élément neutre de $\times$ est le polynôme réduit à 1) .

Définition 3.3.2.2.- Soient $\mathcal{D}$ et $\mathcal{D}'$ deux dioïdes généralisés. On appelle homomorphisme de $\mathcal{D}$ dans $\mathcal{D}'$, une application h de $\mathcal{D}$ dans $\mathcal{D}'$ telle que

$h(r\times s) = h(r)\times h(s)$

$h(s+_G t) = h(s) +_G h(t)$

$h(s+_D t) = h(s) +_D h(t)$

$h(e) = e'$ où e et e' sont les éléments neutres respectifs de $\mathcal{D}$ et $\mathcal{D}'$ pour $\times$.

Proposition 3.3.2.3.- Une condition nécessaire et suffisante pour qu'un bilangage L sur V soit régulier est qu'il existe un dioïde généralisé fini D , un homomorphisme de dioïde généralisé $h : \hat{V}[1]\longrightarrow D$ et une partie D' de D telle que $L = c(h^{-1}(D'))$.

Démonstration.- Supposons que L soit un bilangage régulier sur V . Donc, il existe un V-binoïde fini B et une partie B' de B tels que pour l'homomorphisme de binoïde $f : \hat{V}\longrightarrow B$, on ait $L = f^{-1}(B')$. Soit e l'élément neutre de B pour la loi + . Considérons l'ensemble D des applications de B dans B . On munit cet ensemble d'une structure de dioïde généralisé en posant :

$u +_G v = u + v(e)$

$u +_D v = u(e) + v$

$u\times v = u\circ v$

L'application identique Id est élément neutre pour $\times$, et on a bien $u +_G Id = u$, $Id +_D v = v$.
Les relations $(u+_G v)\times w = u\times w +_G v$ et $(u+_D v)\times w = u +_D v\times w$ se vérifient immédiatement.

D'après la proposition 1.1.6., on peut associer à tout élément r de $\hat{V}[1]$, un élément $h(r)$ de D , et d'après la construction de cette application, il est immédiat que h est un homomorphisme de dioïde généralisé. D'autre part, on a la relation $f(r) = h(r')(e)$ pour chaque r' de $\hat{V}[1]$ tel que $c(r') = r$.
Soit $D' = \{u ; u\in D \text{ et } u(e)\in B'\}$. On a donc $L' = c(h^{-1}(D'))$.

Réciproquement, supposons que L soit un bilangage tel qu'il existe un dioïde généralisé fini D , un homomorphisme $h : \hat{V}[1]\longrightarrow D$ et une partie D' de D tels que $L = c(h^{-1}(D'))$. Montrons que L est un bilangage régulier. Considérons sur $\hat{V}[1]$, la relation d'équivalence $r\sim s$ si et seulement si $c(r) = c(s)$. On obtient ainsi une bijection $\bar{c} : \hat{V}[1]/_\sim \longrightarrow \hat{V}$ qui permet de donner par transport de structure, une structure de V-binoïde à $\hat{V}[1]/_\sim$. Si on note $\bar{r}$ la classe d'équivalence de r dans $\hat{V}[1]$, la structure de V-binoïde de $\hat{V}[1]/_\sim$ défini

ci-dessus est obtenue aussi en considérant les deux lois

$$\bar{r} + \bar{s} = \{ r' +_G s' \ ; \ r' \in \bar{r} \text{ et } s' \in \bar{s} \} \cup \{ r' +_D s' \ ; \ r' \in \bar{r} \text{ et } s' \in \bar{s} \}$$

$$a \times \bar{r} = \{ a \times r' \ ; \ r' \in \bar{r} \} \cup \{ 1 +_G a \times r \} \cup \{ a \times r +_D 1 \} .$$

Considérons maintenant l'application $f : \hat{V}[1]/_\sim \to \mathcal{P}(D)$ définie par $f(\bar{r}) = \{ h(r') \ ; \ r' \in \bar{r} \}$. Soit $B = f(\hat{V}[1]/_\sim)$; on peut munir B d'une structure de V-binoïde en posant

$$b + c = f(\bar{r}+\bar{s}) \quad \text{avec} \quad f(\bar{r}) = b \quad \text{et} \quad f(\bar{s}) = c$$

$$a \times b = f(a \times \bar{r}) \quad \text{avec} \quad f(\bar{r}) = b$$

$$e = f(\bar{1}) \ .$$

Le fait que h soit un homomorphisme de dioïde généralisé et que les définitions de $+$ et $\times$ dans $\hat{V}[1]/_\sim$ assurent la cohérence de ces définitions, f devient donc un homorphisme de binoïdes entre $\hat{V}[1]/_\sim$ et B. De plus, on a

$$u \in L \iff (\exists\, r \in \hat{V}[1]/_\sim)(c(r) = u \text{ et } h(r) \in D') .$$

On en déduit donc

$$u \in L \iff f(\bar{c}^{-1}(u)) \cap D' \neq \emptyset$$

Donc L est l'image réciproque par l'homomorphisme de binoïdes $f \circ \bar{c}^{-1}$ de $\hat{V}$ dans B de la partie B' de B formée des b tel que $b \cap D'$ soit non vide. Donc L est un bilangage régulier.

Proposition 3.3.2.4.- L'intersection d'un bilangage régulier et d'un bilangage engendré par une C_1-bigrammaire, est encore engendrée par une C_1-bigrammaire.

Démonstration.- Soit L un bilangage régulier et soit L' un bilangage engendré par la C_1-bigrammaire $G = (N,T,P,X)$. On peut supposer G sous la forme réduite définie à la proposition 3.2.1.2. Comme L est un bilangage régulier, il existe un dioïde généralisé fini D, une partie D' de D et un homomorphisme $h : \hat{V}[1] \to D$ tel que $L = c(h^{-1}(D'))$. Considérons la bigrammaire $G' = (\{X_o\} \cup N \times D \ , \ T, \ P', \ X_o)$ telle que :

- X_o est un symbole qui n'est pas dans N
- $(X_o \times 1 \ , \ (X,d) \times 1) \in P' \iff d \in D'$
- $((A,d) \times 1, (B,d') \times (C,d'') \times 1) \in P' \iff (A \times 1, B \times C \times 1) \in P$ et $d = d' \times d''$
- $((A,d) \times 1 \ , \ (B,d') + (C,d'') \times 1) \in P' \iff (A \times 1, B+C \times 1) \in P$ et $d = d' +_D d''$

- $((A,d) \times 1 \ , \ (B,d') \times 1 + (C,d'')) \in P' \iff (A \times 1, B \times 1 + C) \in P$ et $d = d' +_G d''$
- $((A,d) \times 1, \ a \times 1) \in P' \iff (A \times 1 \ , \ a \times 1) \in P$ et $h(a \times 1) = d$

On obtient, pour cette bigrammaire

$$(\forall A \in N)(\forall d \in D)(\forall r \in \hat{T}[1])((A,d) \times 1 \overset{*}{\twoheadrightarrow}_{G'} r \iff A \times 1 \overset{*}{\twoheadrightarrow}_{G} r \text{ et } h(r) = d).$$

On en déduit donc que G' engendre $L \cap L'$.

Chapitre 4 : Bigrammaires de type contextuel

4.1.- Introduction.

Les bigrammaires que nous allons étudier dans ce chapitre sont des généralisations des grammaires contextuelles définies sur le monoïde libre.

Nous étudierons d'abord les bigrammaires contextuelles dont les productions sont de la forme (r,s) avec $n(r) \leq n(s)$. Il est facile de montrer que les bilangages engendrés par ces bigrammaires sont décidables. En revanche, les langages engendrés par ces bigrammaires ne le sont pas. En vue de l'étude des systèmes transformationnels, nous étudierons les langages engendrés par ces bigrammaires quand on impose des conditions supplémentaires sur la forme des productions (bigrammaire conservant la racine et dont les productions sont formées de polynômes injectifs). Nous montrerons que malgré ces restrictions on ne restreint pas la généralité des langages engendrés.

Nous étudierons ensuite les bigrammaires strictement contextuelles dont les productions sont de la forme (r,s) avec $n(r) \leq n(s)$, $|\varphi(r)| \leq |\varphi(s)|$ et $|\varphi(r)| = |\varphi(s)| \Longrightarrow n(r) = n(s)$.
L'introduction de ces bigrammaires est justifiée par l'étude de système transformationnel permettant de traduire un langage évolué L en un langage conceptuellement plus simple L' . Dans un tel cas, la traduction d'un mot α de L donnera un mot α' de L' plus long que α et cela se traduira localement par l'utilisation de productions (r,s) telles que $n(r) \leq n(s)$ et $|\varphi(r)| \leq |\varphi(s)|$. D'autre part, les productions (r,s) telles que $|\varphi(r)| = |\varphi(s)|$ correspondront simplement à un réarrangement des non-terminaux et dans ce cas, on aura $n(r) = n(s)$.

Nous montrerons que les langages engendrés par de telles bigrammaires sont exactement les langages contextuels, et de même que pour les bigrammaires contextuelles, nous montrerons que l'on ne restreint pas la généralité des langages engendrés en imposant des conditions supplémentaires sur les productions du même type que celles indiquées ci-dessus, pour les bigrammaires contextuelles.

4.2.- Bigrammaires contextuelles.

Définition 4.2.1.- Soit $G = (N,T,P,X)$ une bigrammaire. On dit que G est une

bigrammaire contextuelle si elle vérifie la condition

$(\forall (r,s) \in P)(n(s) \geqslant n(r))$ (n désignant la fonction nombre de points).

Proposition 4.2.2.- Soit G une bigrammaire contextuelle.

$$r \underset{G}{\twoheadrightarrow} t \implies n(r) \leqslant n(t) \quad .$$

Cela est immédiat. Remarquons cependant que si on considérait des bigrammaires de degré à gauche plus grand que 1, alors cette proposition deviendrait fausse. On pourrait, en effet, avoir des règles de la forme $\begin{pmatrix} A & A & A \\ | & | & \vdots \\ 1 & 1 & A \\ & & | \\ & & 1 \end{pmatrix}$ qui peuvent diminuer le nombre de points d'une ramification. (Voir le paragraphe 4.4).

Proposition 4.2.3.- Les bilangages engendrés par les bigrammaires contextuelles sont décidables.

Démonstration.- Il s'agit de trouver un algorithme permettant de savoir si une ramification $r \in \hat{T}$ appartient ou non à $BL(G)$. Considérons la suite de bilangages $(L_n)_{n \in \mathbb{N}}$ définie par

$$L_0 = X \cap \{s \ ; \ s \in \widehat{T \cup N} \text{ et } n(s) \leqslant n(r)\}$$

$$L_{n+1} = L_n \cup \left(\{s \ ; \ (\exists t \in L_n)(t \underset{G}{\twoheadrightarrow} s)\} \cap \{s \ ; \ s \in \widehat{T \cup N} \text{ et } n(s) \leqslant n(r)\} \right)$$

On obtient $(\forall n \in \mathbb{N})(L_n \subset L_{n+1})$

$$(\forall n \in \mathbb{N})(L_n \subset \{s \ ; \ s \in \widehat{T \cup N} \text{ et } n(s) \leqslant n(r)\})$$

De plus $L_n = L_{n+1} \implies L_{n+1} = L_{n+2}$.

Donc la suite L_n est stationnaire à partir d'un certain rang n_o ($n_o \leqslant \text{card}(\{s \ ; \ s \in \widehat{T \cup N} \text{ et } n(s) \leqslant n(r)\})$) , et une condition nécessaire et suffisante pour que r appartienne à $BL(G)$ est que r appartienne à L_{n_o} .

Théorème 4.2.4.- Tout langage récursivement énumérable est engendré par une bigrammaire contextuelle G ; de plus, on peut imposer à G de vérifier les conditions suivantes :

1°) G conserve la racine.

2°) Toute production de G est formée de polynômes injectifs.

Démonstration.- Soit L un langage récursivement énumérable sur le vocabulaire V . Il existe donc une grammaire semi-thueienne [8] $\mathcal{G} = (V,P,x)$ engendrant L . Si on impose seulement aux bigrammaires considérées d'être contextuelles et de conserver la racine, il est facile d'exhiber une bigrammaire contextuelle engendrant le même langage que $\mathcal{G}$. En effet, considérons la bigrammaire $G = (\phi, V \cup \{A\}, P', A)$ telle que :

- A est un symbole qui n'est pas dans V .
- $(\alpha, \beta) \in P \iff (A \times (1+\alpha+2) \ , \ A^k \times (1+\beta+2) \in P'$

  A^k est la ramification $\underbrace{A \times A \times \ldots \times A}_{k \text{ fois}}$ et on choisit comme valeur de k

  $k = 1$ si $|\alpha| \leqslant |\beta|$ et $k = |\alpha| - |\beta| + 1$ si $|\beta| < |\alpha|$.
- $(A, A \times x) \in P'$.

Cette bigrammaire est bien contextuelle grâce au choix de k . D'autre part, en faisant un raisonnement par récurrence sur la longueur des dérivations, on obtient immédiatement

$$A \underset{G}{\overset{*}{\twoheadrightarrow}} r \implies (\exists n > 0)(\exists \gamma \in T^*)(r = A^n \times \gamma \text{ et } x \underset{\mathcal{G}}{\overset{*}{\succ}} \gamma)$$

$$x \underset{\mathcal{G}}{\succ} \gamma \implies (\exists n > 0)(A \underset{G}{\overset{*}{\twoheadrightarrow}} A^n \times \gamma) \quad .$$

Donc les langages engendrés par $\mathcal{G}$ et G sont égaux.

Si on impose à la bigrammaire G de vérifier les deux conditions du théorème 4.2.4. , G est plus difficile à construire, bien que l'idée de la démonstration soit la même.

Les définitions et notations que nous allons introduire ici, seront utilisées uniquement dans le cadre de cette démonstration.

Définition 1.- L_o est le bilangage engendré par la grammaire algébrique $\mathcal{G}_o = (\{A\}, V, ::=, A)$ avec

$$A ::= A \mid AA$$

et $$A ::= a \text{ pour tout } a \text{ de } V$$

- L_1 est le bilangage engendré par la grammaire algébrique $\mathcal{G}_1 = (\{A,B,C\}, V, ::=, B)$ avec

$$B ::= B \mid CA \mid a \text{ pour tout } a \text{ de } V$$

$$A ::= CA \mid a \text{ pour tout } a \text{ de } V$$

$$C ::= a \text{ pour tout } a \text{ de } V$$

- L_2 est le bilangage image de L_1 par l'application

$t : \widehat{V \cup \{A,B,C\}} \longrightarrow \widehat{V \cup \{A\}}$ défini par

$$t(r) = \wedge$$

$$t(A \times r+s) = A \times t(r) + t(s)$$

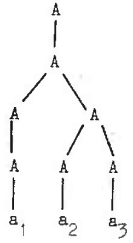
$$t(B \times r+s) = A \times t(r) + t(s)$$

$$t(C \times r+s) = A \times t(r) + t(s)$$

$$t(a \times r+s) = a \times t(r) + t(s)$$

Exemple :

r_o =

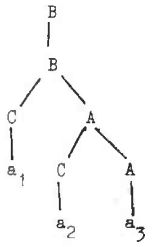


r_o est une ramification de L_o mais n'est pas dans L_2

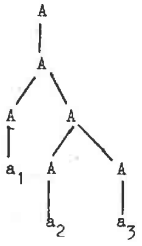
r_1 est une ramification de L_1

r_2 est une ramification de L_2

r_1 =



$r_2 = h(r_1)$ =



Lemme 1.- L_2 est contenu dans L_o .

Lemme 2.- L'application t définie ci-dessus vérifie

$$r \in L_1 \text{ et } r' \in L_1 \text{ et } t(r) = t(r') \Longrightarrow r = r' \quad .$$

En effet, si on appelle L_B , L_A , L_C les bilangages obtenus en utilisant $\mathcal{G}_1$ et en prenant comme axiome respectivement B , A et C , ces trois bilangages vérifient le système d'équation

$$L_B = B \times L_B \cup B \times (L_B + L_A) \cup \bigcup_{a \in V} B \times \{a\}$$

$$L_A = A \times (L_C + L_A) \cup \bigcup_{a \in V} A \times \{a\}$$

$$L_C = \bigcup_{a \in V} C \times \{a\}$$

Démontrons par récurrence sur $n(r)$ que

$$r \in L_X \text{ et } r' \in L_X \text{ et } t(r) = t(r') \Longrightarrow r = r'$$

pour $X = A$, B ou C . Cette implication est évidente pour $n(r) = 0$.
Supposons-la démontrée pour $n(r) \leq k$ et supposons que $n(r) = k + 1$

$$r \in L_A \text{ et } r' \in L_A \text{ et } t(r) = t(r') \Longrightarrow (r \in A \times (L_C + L_A) \text{ et } r' \in A \times (L_C + L_A))$$

$$\text{ou } \left(r \in \bigcup_{a \in V} A \times \{a\} \text{ et } r' \in \bigcup_{a \in V} A \times \{a\} \right) \quad .$$

D'où en appliquant l'hypothèse de récurrence on en déduit $r = r'$. Les deux autres cas se traitent de même.

Définition 2.- Soit $\alpha \in V^*$, on dit que $r \in \widehat{V \cup \{A\}}$ est associé à α si et seulement si r est un élément de L_o et si $\varphi(r) = \alpha$.

Lemme 3.- Soit $\alpha \in V^* - \{\wedge\}$. Pour chaque $p \geqslant 3|\alpha| - 1$, il existe une et une seule ramification r de L_2 associée à α telle que $n(r) = p$.

D'après le lemme 2 , il suffit de démontrer que pour tout $p \geqslant 3|\alpha| - 1$ il existe une et une seule ramification r' de L_1 telle que $n(r') = p$ et $\varphi(r') = \alpha$.
Remarquons qu'une ramification r' de L_1 est de la forme

$$r' = \underbrace{B \times B \times \ldots \times B}_{q \text{ fois}} \times r'_o \text{ avec } q \geqslant 1 \text{ et } \rho(r'_o) = CA \text{ ou } \rho(r'_o) \in V$$

De plus $B \times r'_o$ est dans L_1 .

Faisons une récurrence sur $|\alpha|$. Si $|\alpha| = 1$ alors $\alpha = a \in V$ et les seules ramifications r' de L_1 telles que $\varphi(r') = \alpha$ sont de la forme

$$r' = \underbrace{B \times B \times \ldots \times B}_{q \text{ fois}} \times a \text{ avec } q \geqslant 1$$

Donc, pour chaque $p \geqslant 3|\alpha| - 1 = 2$, on trouve bien une unique ramification de L_1 dont le mot des feuilles est α et ayant un nombre de points égal à p .

Supposons ce résultat établi pour $|\alpha| \leq k$ et supposons que $|\alpha| = k + 1$. On a $\alpha = a\alpha'$ avec $\alpha' \in V^* - \{\wedge\}$. Soit $p \geqslant 3|\alpha| - 1$. Par hypothèse de récurrence, il existe une unique ramification r_1 de L_1 telle que $\varphi(r_1) = \alpha'$ et

$n(r_1) = p - 3$. r_1 est de la forme

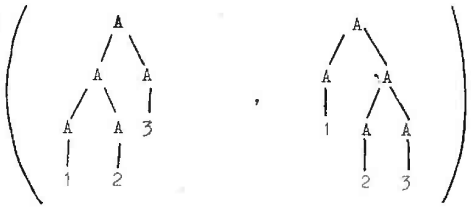
$$r_1 = \underbrace{B \times B \times \ldots \times B}_{q \text{ fois}} \times r_0 \quad \text{avec} \quad q \geqslant 1 \quad \text{et} \quad \rho(r_0) = CA \quad \text{ou} \quad \rho(r_0) \in V$$

Donc $r' = \underbrace{B \times B \times \ldots \times B}_{q \text{ fois}} \times (C \times a + A \times r_0)$ est dans L_1 et vérifie $\varphi(r') = a\alpha' = \alpha$ et $n(r') = n(r_1) + 3 = p$. D'où l'existence de la ramification r' . De plus, toute ramification r' vérifiant ces conditions, doit être de la forme ci-dessus. D'où l'unicité.

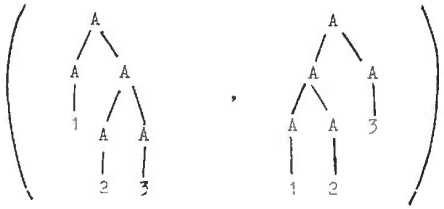
Lemme 4.- Il existe une bigrammaire contextuelle G conservant la racine et dont les règles sont formées de polynômes injectifs tels que
$(\forall r)(\forall s)(r \in L_o \text{ et } s \in L_o \text{ et } \varphi(r) = \varphi(s) \text{ et } n(r) \leq n(s) \Longrightarrow r \overset{*}{\underset{G}{\twoheadrightarrow}} s)$,
(c'est-à-dire que s dérive de r dans G dès que r et s sont associés à un même mot et vérifient $n(r) \leq n(s)$).

Considérons une bigrammaire G de vocabulaire terminal $V \cup \{A\}$ et dont les productions sont :

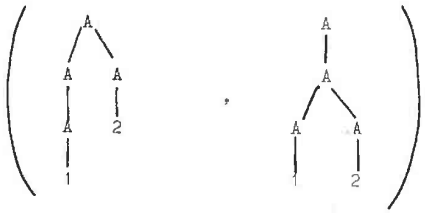
production ①



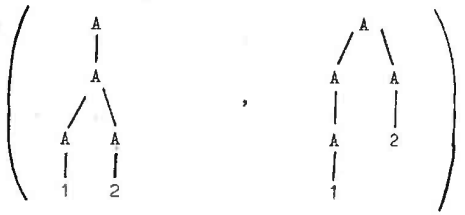
production ②



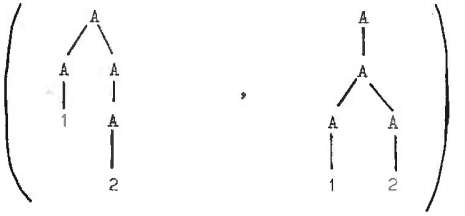
production ③



production ④



production ⑤



production ⑥

production ⑦

La bigrammaire G possède les propriétés immédiates suivantes :

- $r \underset{G}{\twoheadrightarrow} s \Longrightarrow \varphi(r) = \varphi(s)$
- $r \in L_o$ et $r \underset{G}{\twoheadrightarrow} s \Longrightarrow s \in L_o$
- si on n'utilise pas la production 7 , alors

$$r \underset{G}{\twoheadrightarrow} s \Longrightarrow n(r) = n(s)$$

- sauf pour la production 7 , $(t,t') \in P \Longrightarrow (t',t) \in P$.

Pour démontrer le lemme 4 , il suffit alors de démontrer les deux assertions suivantes :

i) $(\forall r \in L_o)(\exists s \in L_2)(\varphi(r)=\varphi(s)$ et $r \overset{*}{\gg}_G s)$ (la dérivation $r \overset{*}{\gg}_G s$ étant obtenue en utilisant seulement les productions 1 à 6).

ii) $(\forall r \in L_2)(\forall s \in L_2)(\varphi(r)=\varphi(s)$ et $n(s) \geqslant n(r) \Longrightarrow r \overset{*}{\gg}_G s)$ (la dérivation $r \overset{*}{\gg}_G s$ étant obtenue en utilisant seulement la production 7).

Démonstration de i).- On procède par récurrence sur $n(r)$.

Si $n(r) = 2$ et $r \in L_o$, alors $r =$, donc $r \in L_2$, et il suffit de prendre $s = r$.

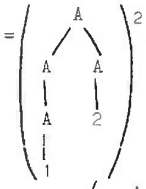
Supposons la propriété i) démontrée pour $n(r) \leqslant k$ et démontrons i) pour $n(r) = k + 1$.

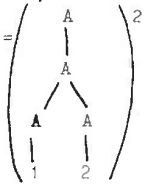
On a soit $r = A \times r'$ et $r' \in L_o$

soit $r = A \times (r_1 + r_2)$ et $r_1 \in L_o$ et $r_2 \in L_o$.

1°) si $r = A \times r'$ et $r' \in L_o$, par hypothèse de récurrence, il existe un $s' \in L_2$ tel que $\varphi(s') = \varphi(r')$, $n(s') = n(r')$ et $r' \overset{*}{\gg}_G s'$. Donc $r \overset{*}{\gg}_G A \times s'$ et comme $s' \in L_2$ $A \times s' \in L_2$. Donc $s = A \times s'$ répond à la question.

2°) si $r = A \times (r_1 + r_2)$ et $r_1 \in L_o$ et $r_2 \in L_o$, par hypothèse de récurrence, il existe s_1 et s_2 de L_2 tels que $\varphi(s_1) = \varphi(r_1)$, $\varphi(s_2) = \varphi(r_2)$, $n(s_1) = n(r_1)$, $n(s_2) = n(r_2)$, $r_1 \overset{*}{\gtrdot}_G s_1$ et $r_2 \overset{*}{\gtrdot}_G s_2$. Donc $r \overset{*}{\gtrdot} A \times (s_1 + s_2) = r'$. Distinguons plusieurs cas

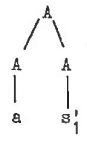
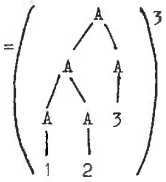
α) $s_1 = A^2 \times s_1'$, alors $r' =$  (s_1' , s_2') et grâce à

à la production 3 , $r' \gtrdot_G r'' =$  (s_1' , s_2') et on est

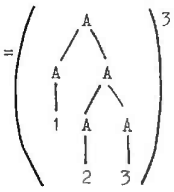
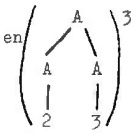
ramené au cas 1°) traité ci-dessus.

β) $s_1 = A \times a$, alors $r' \in L_2$ et il suffit de prendre $s = r'$.

γ) s_1 n'est pas d'une des deux formes ci-dessus, alors

$s_1 =$  et $r' =$  (a,s_1',s_2')

en appliquant la production 1 , on obtient

$r' \gtrdot_G r'' =$  (a,s_1',s_2') . On applique de nouveau l'hypothèse de récurrence en  $(\wedge,s_1',s_2')$ et on est ramené au cas β .

Démonstration de ii).- Si r et s sont deux ramifications de L_2 telles que $\varphi(r) = \varphi(s)$ et $n(r) \leqslant n(s)$, on a $s = \underbrace{A \times A \times \ldots \times A}_{q \text{ fois}} \times r$ avec $q = n(s) - n(r)$.

En appliquant q fois la production 7 de G , on obtient $r \overset{*}{\gg}_G s$.

Fin de la démonstration du théorème 4.2.4.- Tout langage récursivement énumérable L est engendré par une grammaire semi-thueienne. Soit $\mathcal{G} = (V,P,x)$ une grammaire semi-thueienne d'axiome x engendrant le langage récursivement énumérable L Construisons une bigrammaire contextuelle $G' = (N,T,P',X)$ qui engendre le même langage que $\mathcal{G}$. On définit G' par :

- $N = \emptyset$
- $T = V \cup \{A\}$ (A est une lettre qui n'est pas dans V)
- X est une ramification de L_2 associée à x .
- Pour chaque production $\alpha \longrightarrow \beta$ de $\mathcal{G}$, on introduit dans P' une production (s,s') telle que $s \in L_2$, $s' \in L_2$, $\varphi(s) = \alpha$, $\varphi(s') = \beta$ et $n(s') \geqslant n(s)$, (ce qui est possible d'après le lemme 3). On introduit de

plus dans P' les 7 productions de la grammaire G définie ci-dessus.

Le théorème résulte alors de la proposition suivante :

Proposition.- $s \overset{*}{\underset{\mathcal{G}}{\rightarrow}} \gamma \iff (\exists s \in L_o)(X \overset{*}{\underset{\mathcal{G}'}{\rightarrow}} s \text{ et } \varphi(s) = \gamma)$.

Démonstration.- $(\exists s \in L_o)(r \overset{*}{\underset{\mathcal{G}}{\rightarrow}} s \text{ et } \varphi(s) = \gamma) \Longrightarrow x \overset{*}{\underset{\mathcal{G}}{\rightarrow}} \gamma$ est immédiat par récurrence sur la longueur de la dérivation $r \overset{*}{\underset{G'}{\rightarrow}} s$. Pour démontrer l'implication dans l'autre sens, il suffit de démontrer que

$\gamma \underset{\mathcal{G}}{\rightarrow} \delta$ et $s \in L_o$ et $\varphi(s) = \gamma \Longrightarrow (\exists s' \in L_o)\ (\varphi(s') = \delta$ et $s \underset{G'}{\rightarrow} s')$.

Si $\gamma \underset{\mathcal{G}}{\rightarrow} \delta$, alors $\gamma = \gamma' \alpha \gamma''$, $\delta = \gamma' \beta \gamma''$ et $\alpha \rightarrow \beta$ est une production de $\mathcal{G}$. Soit $(t,t') \in P'$ la production de $\mathcal{G}'$ associée à $\alpha \rightarrow \beta$.
Soient r' et r'' deux ramifications de L_o associées à γ' et γ'' . Pour tout entier k , la ramification $r_k = A^k \times (A \times (r' + A \times (t + r''))$ est une ramification de L_o associée à $\gamma = \gamma' \alpha \gamma''$. Soit $s \in L_o$ associé à γ ; pour un k suffisamment grand $n(r_k) \geqslant n(s)$. Donc $s \overset{*}{\underset{G'}{\rightarrow}} r_k$ puisque G' contient toutes les productions de G . En appliquant la production (t,t') , on obtient

$$r_k = A^k \times (A \times (r' + A \times (t + r'))) \underset{}{\rightarrow} A^k \times (A \times (r' + A \times (t' + r'))) = s' .$$

On a $s' \in L_o$, $\varphi'(s') = \gamma' \beta \gamma''$ et $s \overset{*}{\underset{\mathcal{G}'}{\rightarrow}} s'$. C.Q.F.D.

4.3.- Bigrammaire strictement contextuelle.

Dans le paragraphe précédent, nous avons vu que, même si on impose aux bigrammaires d'augmenter le nombre de points d'une ramification au cours d'une dérivation, les langages engendrés restent très généraux. Cela tient au fait que dans une telle bigrammaire, une production (s,t) peut vérifier $n(s) \leqslant n(t)$ et $|\varphi(s)| > |\varphi(t)|$. Nous allons envisager un nouveau type de bigrammaire où ce cas ne se produit pas.

Définition 4.3.1.- On dit qu'une bigrammaire $G = (N,T,P,X)$ est strictement contextuelle si elle vérifie la condition suivante

$$(\forall (r,s) \in P)(n(r) \leqslant n(s) \text{ et } |\varphi(r)| \leqslant |\varphi(s)| \text{ et } (|\varphi(r)| = |\varphi(s)| \Longrightarrow n(r) = n(s))) .$$

Intuitivement dans une grammaire strictement contextuelle, l'application d'une production à une ramification augmente le nombre de points et la longueur du mot des feuilles et de plus, le nombre de points ne peut augmenter strictement que s'il en est de même pour la longueur du mot des feuilles.

Proposition 4.3.2.- Les langages et les bilangages engendrés par une bigrammaire

strictement contextuelle, sont décidables.

Démonstration.- La proposition 4.2.3. résoud le problème pour les bilangages.
Soit $G = (N,T,P,X)$ une bigrammaire strictement contextuelle engendrant un langage L . Posons $k = \underset{(r,s) \in P}{\text{Sup}} (n(s) - n(r))$ et $h = \underset{\substack{(r,s) \in P \\ |\varphi(s)| > |\varphi(r)|}}{\text{Inf}} (|\varphi(s)| - |\varphi(r)|)$.

Supposons que $t \overset{*}{\underset{G}{\rightarrow}} t'$ et que $n(t') - n(t) \geqslant pk$ $(p \in \mathbb{N})$.

Donc au cours de la dérivation $t \overset{*}{\underset{G}{\rightarrow}} t'$, on a utilisé au moins p productions augmentant strictement le nombre de points. A chaque utilisation d'une telle production, la longueur du mot des feuilles a augmenté d'au moins h . Donc, on obtient

$$t \overset{*}{\underset{G}{\rightarrow}} t' \text{ et } n(t') - n(t) \geqslant pk \Longrightarrow |\varphi(t')| - |\varphi(t)| \geqslant p\,h .$$

Soit $\alpha \in T^*$ et cherchons si $\alpha \in L$.

$$\alpha \in L \iff (\exists x \in X)(\exists r \in \hat{T})(x \overset{*}{\underset{G}{\rightarrow}} r \text{ et } \varphi(r) = \alpha) .$$

Considérons un $x \in X$ et $r \in T^*$ vérifiant cette formule.

On a $|\varphi(r)| - |\varphi(x)| < p\,h$ avec $p = \underset{x \in X}{\text{Sup}} \left(\dfrac{|\alpha| - |\varphi(x)|}{h} + 1 \right)$.

On en déduit $n(r) - n(x) < pk$.

Donc $n(r) < pk + \underset{x \in X}{\text{Sup}} (|\varphi(x)|) = q(|\alpha|)$.

Donc pour savoir si $\alpha \in L$, il suffit de regarder si parmi les ramifications r dérivant d'un axiome x dans G et vérifiant $n(r) \leqslant q(|\alpha|)$, l'une d'entre elles est dans $\hat{T}$ et admet α comme mot des feuilles. Comme $q(|\alpha|)$ est une fonction calculable de $|\alpha|$ et que l'application d'une production de G ne fait qu'augmenter le nombre de points d'une ramification, ce problème est décidable.

Théorème 4.3.3.- Les bigrammaires strictement contextuelles conservant la racine et dont les productions sont formées de polynômes injectifs, engendrent tous les langages contextuels.

Démonstration.- Si on impose seulement aux bigrammaires strictement contextuelles de conserver la racine, il est facile de simuler une grammaire contextuelle $\mathcal{G}$ avec une bigrammaire G de ce type. En effet, à toute production $\alpha \rightarrow \beta$ de $\mathcal{G}$, on associe une production de G de la forme $(A \times (1 + \alpha + 2)\ ,\ A \times (1 + \beta + 2))$ et on prend comme axiome dans G $A \times x$ où x est l'axiome de $\mathcal{G}$. On obtient alors immédiatement :

$$(\forall \gamma)(x \overset{*}{\underset{\mathcal{G}}{\rightarrow}} \gamma \iff A \times x \overset{*}{\underset{G}{\rightarrow}} A \times \gamma) .$$

Si on impose de plus que les productions de G soient formées de polynômes injectifs, cette association est moins facile à expliciter. Les définitions et propositions que nous allons énoncer maintenant seront utilisées seulement dans le cadre de la démonstration ci-dessous, sauf mention expresse du contraire.

Définition 1.- On appelle bilangage L_1 , le bilangage engendré par la grammaire $G_1 = (\{A\}, V, ::=, A)$ avec $A ::= AA$ et $A ::= a$ pour chaque a de V .

Définition 2.- On appelle bilangage L'_2 , le bilangage engendré par la grammaire $G'_2 = (\{A,B,C\}, V, ::=, B)$ avec $B ::= a \mid CA$, $A ::= a \mid CA$ et $C ::= a$. L_2 est obtenu à partir de L'_2 par la transcription τ : $A \longmapsto A$, $B \longmapsto A$, $C \longmapsto A$, $a \longmapsto a$.

Lemme 1.- τ est une bijection de L'_2 sur L_2 et $L_2 \subset L_1$.

Définition 3.- Soit $\alpha \in V^*$, on dit que $r \in \overbrace{V \cup \{A\}}$ est associé à α si $r \in L_1$ et $\varphi(r) = \alpha$.

Lemme 2.- Pour tout $\alpha \in V^* - \{\wedge\}$, il existe une unique ramification r_α de L_2 associée à α . Toute ramification r associée à α vérifie $n(r) = 3|\alpha| - 1$.

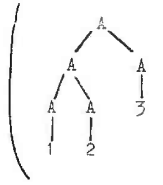
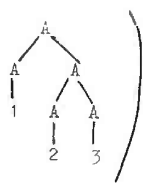
Démonstration.- Il est évident que la grammaire G'_2 engendre $V^* - \{\wedge\}$ et qu'elle n'est pas ambigüe. D'où l'existence et l'unicité de r_α . La deuxième assertion se démontre par récurrence sur $|\alpha|$. Si $|\alpha| = 1$, alors $\alpha = a$ et $r = A \times a$, donc $n(r) = 2 = 3 \times 1 - 1$. Supposons ce résultat démontré pour $|\alpha| \leq k$ et supposons que $|\alpha| = k + 1$. Soit r associé à α , $r = A \times (r_1 + r_2)$ avec r_1 associé à α_1 , r_2 associé à α_2 et $\alpha_1 \alpha_2 = \alpha$. Donc

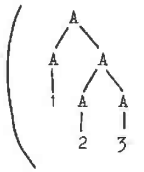
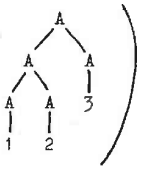
$$n(r) = 1 + n(r_1) + n(r_2) = 1 + 3|\alpha_1| - 1 + 3|\alpha_2| - 1 = 3(|\alpha_1| + |\alpha_2|) - 1 = 3|\alpha| - 1 .$$

Lemme 3.- Il existe une bigrammaire $G_0 = (\emptyset, V \cup \{A\}, P_0, X)$ telle que pour tout $\alpha \in V^* - \{\wedge\}$ et tout $r \in L_1$ associé à α , on ait

$$(\forall s \in \overbrace{V \cup \{A\}})(r \overset{*}{\gg}_{G_0} s \iff s \text{ associé à } \alpha).$$

Démonstration.- Mettons dans P_0 les deux productions suivantes :

Production 1 $\left(\right.$  ,  $\left. \right) = (r,s)$

production 2 $\left(\right.$  ,  $\left. \right) = (s,r)$

l'implication $r \overset{*}{\gg}_G s \implies s$ associé à α est évidente.

Pour démontrer la réciproque, il suffit, en vertu de la symétrie des deux productions de G , de montrer qu'en utilisant seulement la production 1 , on a $r \overset{*}{\gg}_G r_\alpha$ où r_α est l'unique ramification de L_2 associée à α . Nous allons faire la démonstration par récurrence sur $n(r)$.

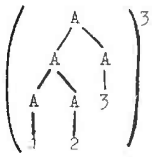
Si $n(r) = 2$ $\quad r =$  et donc $r_\alpha = r$.

Supposons la propriété démontrée pour $n(r) \leq k$ et supposons que $n(r) = k + 1$ $r = A \times (r_1 + r_2)$, $r_1 \in L_1$ et $r_2 \in L_1$. En appliquant l'hypothèse de récurrence, on en déduit que $r_1 \overset{*}{\gg}_G r'_1$ et $r_2 \overset{*}{\gg}_G r'_2$ avec $r'_1 \in L_2$ et $r'_2 \in L_2$.

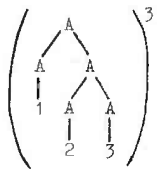
Donc $r \overset{*}{\gg}_G A \times (r'_1 + r'_2) = r'$.

α) Si $r'_1 =$  , alors $r' \in L_2$ et $r_\alpha = r'$.

β) Si $r'_1 \neq$  , alors $r'_1 =$ 

Donc $r' =$  (a , r''_1 , r''_2)

En appliquant la production 1 , on obtient

$r' \gg_G$  (a , r''_1 , r''_2)

En appliquant l'hypothèse de récurrence à $\left(\begin{array}{c} A \\ A \quad A \\ 2 \quad 3 \end{array}\right)^3$ (Λ, r_1'', r_2'')

on est ramené au cas α) .

Fin de la démonstration du théorème 4.3.3.- Soit $\mathcal{G} = (N,T,P,X)$ une grammaire contextuelle engendrant un langage L . Construisons une bigrammaire $\mathcal{G}'$ strictement contextuelle conservant la racine engendrant L . $\mathcal{G}'$ est définie par

$\mathcal{G}' = (N, T \cup \{A\}, P', X')$ (A étant supposé ne pas appartenir à $N \cup T$).

Les éléments de X' sont obtenus en choisissant pour chaque axiome x de $\mathcal{G}$ une ramification associée à x .

Les productions de P sont les deux productions de P_0 et pour chaque production $\alpha \longrightarrow \beta$ de $\mathcal{G}$, une production (r,s) où r et s sont associés respectivement à α et β . Comme $\mathcal{G}$ est une grammaire contextuelle, si $\alpha \longrightarrow \beta$ on a $|\alpha| \leq |\beta|$, donc en vertu de la relation démontré au lemme 3 entre $|\alpha|$ et $n(r)$ si r est associé à α , on en déduit que $\mathcal{G}''$ est une bigrammaire strictement contextuelle conservant la racine.

Pour finir la démonstration, il suffit alors de montrer que

$$(\forall t \in L_1)(\forall t' \in L_1)(\varphi(t) \overset{*}{\underset{\mathcal{G}}{\gg}} \varphi(t') \Longleftrightarrow t \overset{*}{\underset{\mathcal{G}'}{\gg}} t') .$$

La démonstration est similaire à celle faite pour le théorème 3.2.4.

Théorème 3.3.4.- Tout langage engendré par une bigrammaire strictement contextuelle, est un langage contextuel.

Démonstration.- La classe des langages contextuels est exactement la classes des langages reconnus par les automates linéairement bornés (cf. [8]) . Il s'agit donc étant donnée une grammaire strictement contextuelle $G = (N,T,P,X)$ de construire un automate linéairement borné $\mathcal{A}$ reconnaissant le langage L engendré par G . La construction que nous allons donner n'est pas entièrement formalisée, mais on se persuadera facilement qu'un automate linéaire borné est capable de réaliser les tâches que l'on va décrire.

1°) On suppose que l'on utilise la représentation de $\hat{V}$ par le langage de Dyck décrit en 0.2.1. On notera $|r|$ la longueur du mot de $(V \cup \bar{V})^*$ représentant r . On a évidemment $|r| = 2n(r)$.

2°) Classiquement, la longueur du ruban d'entrée alloué à un automate linéairement borné pour reconnaître un mot α est $|\alpha|$. Mais on n'augmente pas la puissance d'un tel automate en supposant que cette longueur est une fonction linéaire affine de $|\alpha|$. En effet, si $\mathcal{A}$ est un automate utilisant, pour reconnaître un

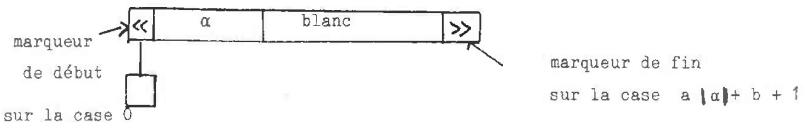
mot α , une longueur de ruban égale à $a|\alpha| + b$, on peut simuler $\mathcal{A}$ à l'aide d'un automate $\mathcal{A}'$ n'utilisant qu'une longueur de ruban égale à $|\alpha|$ en supposant que le vocabulaire de $\mathcal{A}'$ est formé de a-uplets du vocabulaire de $\mathcal{A}$ et en augmentant la capacité mémoire de $\mathcal{A}$.

Comme les ramifications engendrées par la bigrammaire G et susceptibles d'avoir α comme mot des feuilles ont un nombre de points majoré par une fonction linéaire affine de $|\alpha|$ (cf. la démonstration de la proposition 3.2.2.), il existe une fonction linéaire $|\alpha| \mapsto a|\alpha| + b$ ne dépendant que de G telle que

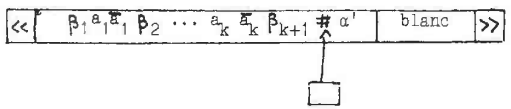
$$(\forall \alpha \in V^*)(\forall r \in \hat{V})(\forall r' \in BL(G)) \left[(X \overset{*}{\underset{G}{\gg}} r \overset{*}{\underset{G}{\gg}} r' \text{ et } \varphi(r') = \alpha) \Longrightarrow |r| \leq a|\alpha| + b \right]$$

3°) Il existe un automate linéairement borné $\mathcal{A}_0$ tel que, si on écrit sur son ruban un mot α de T^* et si on lui alloue une longueur de ruban égale à $a|\alpha| + b$ ($a \geq 1$) , à la fin d'un calcul réussi une quelconque des ramifications t vérifiant $|t| \leq a|\alpha| + b$ et $\varphi(t) = \alpha$ est écrite sur le ruban et que réciproquement pour chaque ramification t vérifiant $|t| \leq a|\alpha| + b$ et $\varphi(t) = \alpha$, il existe un calcul réussi de $\mathcal{A}_0$ de donnée α et avec t écrit sur le ruban à la fin du calcul. Un tel automate est évidemment indéterministe et on peut facilement justifier son existence en raisonnant de la façon suivante :

- Etat initial de $\mathcal{A}_0$



- Etape intermédiaire de calcul



avec $a_1\, a_2 \cdots a_k\, \alpha' = \alpha$ et β_i quelconque dans $(T \cup \bar{T})^*$ vérifiant pour tout a de T $a\bar{a}$ n'est pas un facteur de β_i .

Après cette phase du calcul, on obtient un mot t de $(T \cup \bar{T})^*$ sur le ruban. On fait alors vérifier à l'automate que t est dans le langage de Dyck. La façon dont on a construit t implique $\varphi(t) = \alpha$.

4°) Soit (r,s) une production de G, il existe un automate linéairement borné $\mathcal{A}_{(r,s)}$ tel que si une ramification t est écrite sur le ruban de $\mathcal{A}_{(r,s)}$, à la fin d'un calcul réussi une quelconque des ramifications t' telles que $t' \twoheadrightarrow t$ en utilisant la règle (r,s) se trouve écrite sur le ruban et que réciproquement pour toute ramification t' se réécrivant t en utilisant (r,s) soit le contenu du ruban à la fin d'un calcul réussi de $\mathcal{A}_{(r,s)}$ de donnée t. (S'il n'existe pas de tel t', $\mathcal{A}_{(r,s)}$ s'arrête dans un état de non satisfaction).

5°) X est un ensemble fini de ramification, donc il existe un automate $\mathcal{A}_X$ linéairement borné reconnaissant X.

6°) On peut supposer que les conditions suivantes sont vérifiées :

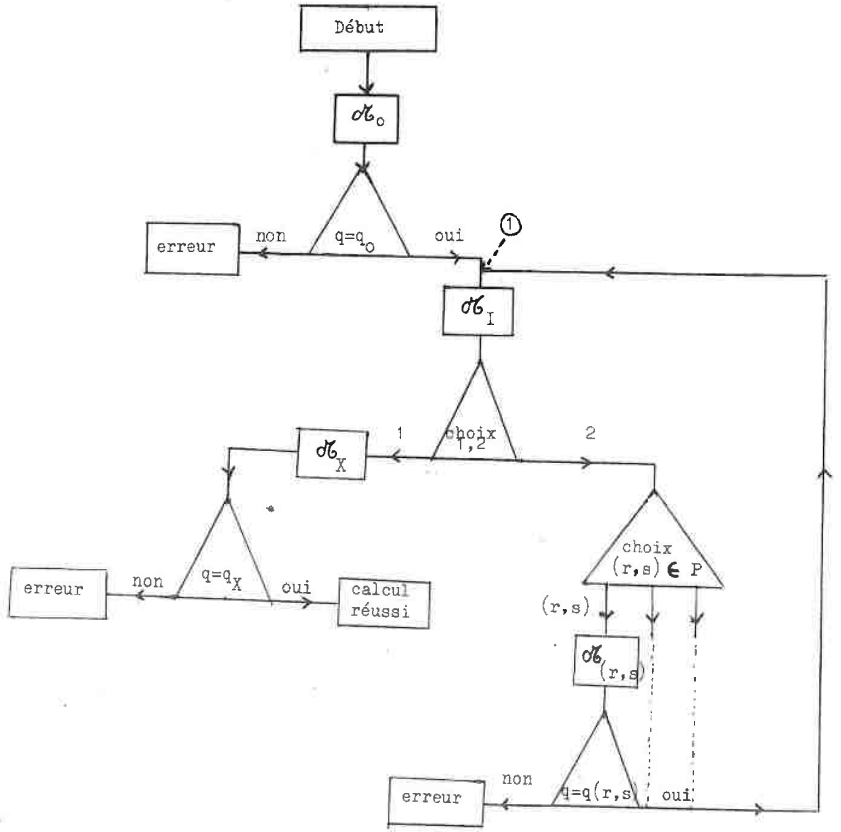
- $\mathcal{A}_0$ a un ensemble d'état disjoint des ensembles d'états des différents automates $\mathcal{A}_{(r,s)}$ et $\mathcal{A}_X$.
- Les automates $\mathcal{A}_{(r,s)}$ et $\mathcal{A}_X$ ont un même état initial q_1 qui n'est plus utilisé dès qu'un calcul est commencé.
- Deux quelconques des automates $\mathcal{A}_{(r,s)}$ et $\mathcal{A}_X$ ont des ensembles d'états d'intersection réduite à $\{q_1\}$.
- Chacun des automates $\mathcal{A}_0$, $\mathcal{A}_{(r,s)}$ et $\mathcal{A}_X$ a un et un seul état de satisfaction noté respectivement q_0, $q_{(r,s)}$ et q_X.

7°) On appelle $\mathcal{A}_I$ l'automate qui partant d'un état final de $\mathcal{A}_0$ ou d'un des automates $\mathcal{A}_{(r,s)}$ termine son calcul dans l'état q_1, la tête de lecture écriture étant positionnée sur le marqueur de début "$\ll$" et le contenu du ruban n'ayant pas été modifié. On peut supposer que les états intermédiaires de $\mathcal{A}_I$ ne sont pas utilisés dans les autres automates décrits ci-dessus.

8°) L'automate $\mathcal{A}$ cherché peut alors se construire de la façon suivante :

- L'ensemble des états de $\mathcal{A}$ est l'union de l'ensemble des états des automates $\mathcal{A}_0$, $\mathcal{A}_X$, $\mathcal{A}_I$, $\mathcal{A}_{(r,s)}$ $((r,s) \in P)$.
- L'état initial de $\mathcal{A}$ est celui de $\mathcal{A}_0$.
- Le seul état de satisfaction de $\mathcal{A}$ est celui de $\mathcal{A}_X$ (noté q_X)
- L'ensemble des transitions de $\mathcal{A}$ est l'union des ensembles des transitions des automates $\mathcal{A}_0$, $\mathcal{A}_{(r,s)}$ $((r,s) \in P)$, $\mathcal{A}_X$, $\mathcal{A}_I$.

Les différentes conditions imposées sur les ensembles d'états de ces différents automates font que le fonctionnement de $\mathcal{A}$ peut être représenté par l'organigramme suivant (on notera q l'état de l'automate à un instant donné du calcul).



- La pseudo-instruction choix u , v ... signifie que l'on choisit de manière indéterministe de poursuivre les actions en suivant un des arcs étiquetés par u , v ,..

- Les cases | erreur | signifient que dans ce cas l'automate se bloque dans un état de non satisfaction.

Il est facile de se persuader que si le mot α est engendré par G, alors il existe un calcul réussi de $\mathcal{A}$ de donnée α. En effet, un choix convenable des actions à effectuer fait "remonter" par $\mathcal{A}$ une dérivation de $x \in X$ à r telle que

$\varphi(r) = \alpha$. Réciproquement, si α est la donnée d'un calcul réussi de $\mathcal{A}$, la suite des contenus du ruban de $\mathcal{A}$ à chaque passage au point ① de l'organigramme donne une dérivation d'un axiome à une ramification de mot des feuilles α .

Donc $\mathcal{A}$ reconnait bien le langage engendré par G .

4.4.- Bigrammaire de degré gauche strictement plus grand que un.

Rappel : Soit $G = (N,T,P,X)$ une bigrammaire. On pose

$$d_g(G) = \mathrm{Sup}(d(r) \ ; \ (\exists s)(r,s) \in P) \ .$$

On suppose ici que $d_g(G) > 1$.

Théorème 4.4.1.- Pour les bigrammaires de degré gauche strictement plus grand que 1, le problème suivant est récursivement indécidable même si on impose à ces bigrammaires d'avoir des productions (r,s) vérifiant $n(r) \leq n(s)$.

Une ramification r appartient-elle ou non au bilangage engendré par G ?

Démonstration.- Nous allons montrer que la résolution du problème ci-dessus permet de résoudre le problème de Post.
Soit $(\alpha_1, \beta_1) \ldots (\alpha_n, \beta_n)$ des couples de mots de T^* . Le problème de Post consiste à savoir s'il existe une suite finie d'indices i_1 , i_2 ,..., i_k tels que

$$\alpha_{i_1} \alpha_{i_2} \cdots \alpha_{i_k} = \beta_{i_1} \beta_{i_2} \cdots \beta_{i_k} \ .$$

Ce problème est équivalent à celui de l'arrêt des machines de Turing et ne se résout donc pas par algorithme. cf [8] .
Etant donné $\alpha \in T^*$, on notera $\bar{\alpha}$ la ramification de $\hat{T}$ dont le seul chemin est α . C'est-à-dire

$$\bar{\Lambda} = \Lambda$$
$$\overline{a\alpha} = a \times \bar{\alpha} \ .$$

Considérons la bigrammaire $G = (N, T \cup \{*\}, X)$ telle que $N = \{X,A,B\}$

$$P = P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5 \cup P_6 \cup P_7 \quad \text{avec}$$

$$P_1 = \{(X, A \times (\bar{\alpha}_i + \bar{\beta}_i)) \ ; \ 1 \leq i \leq n\}$$
$$P_2 = \{(A \times (1+2) \ , \ A \times (\bar{\alpha}_i \times 1 + \bar{\beta}_i \times 2)) \ ; \ 1 \leq i \leq n\}$$
$$P_3 = \{(A \times (1+1), \ A \times A \times 1)\}$$
$$P_4 = \{(A \times A \times 1 \ , \ A \times (1+B)) \ , \ (A \times a \times 1 \ , \ A \times (1+B)) \ ; \ a \in T\}$$
$$P_5 = \{(A \times B \times 1 \ , \ B \times 1 + B) \ , \ (B \times 1 + B \times 2), 1 + B \times B \times 2) \ , \ (B \times 1 \ , \ B \times B \times 1)\}$$
$$P_6 = \{(B \times 1 + B \times 1 \ , \ B \times B \times B \times 1)\}$$
$$P_7 = \{(B \times 1 \ , \ * \times 1)\}$$

Pour obtenir une ramification de $BL(G)$, on doit utiliser la production de P_1 , puis un certain nombre de fois des productions de P_2 , puis la production de P_3 qui ne peut être utilisée que si le problème de Post pour les couples $(\alpha_i \ , \beta_i)$

admet une solution. A ce niveau, on obtient une ramification de la forme

$A \times A \times \overline{\alpha}_{i_1} \times \overline{\alpha}_{i_2} \times \ldots \times \overline{\alpha}_{i_k}$.

En utilisant les productions de P_4 , on obtient une ramification de la forme

$A \times B \times B^p$ avec $p = \sum_{j=1}^{k} |\alpha_{ij}|$.

En utilisant les productions de P_5 , on peut obtenir une ramification de la forme $B^k + B^k$ avec $2k \leq p + 1$, puis en appliquant la production de P_6 , on obtient la ramification B^{k+2} . En réitérant ce procédé, on arrive à la ramification B^4 . En appliquant la production de P_7 , on obtient $*\times*\times*\times*$.

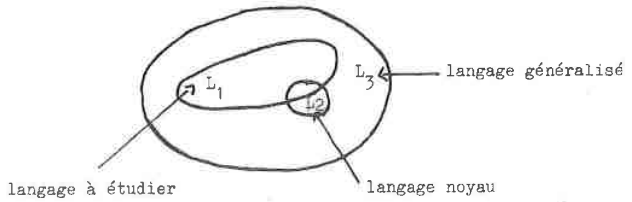
En résumé, on a obtenu :

- $BL(G)$ est non vide si et seulement si le problème de Post associé au couple (α_i , β_i) admet une solution.
- $*^4$ est une ramification de $BL(G)$ si et seulement si le problème de Post associé au couple (α_i , β_i) admet une solution.

Chapitre 5 : Systèmes transformationnels

Introduction.- L'idée de ce chapitre est d'utiliser les productions des bigrammaires pour traduire un langage décrit à l'aide d'une grammaire algébrique dans un autre langage. Dans l'étude des langages de programmation, on est souvent dans la situation suivante :

on veut étudier un langage L_1 engendré par une grammaire G_1, mais la complexité de ce langage rend difficile une étude directe. On désire donc traduire toutes les phrases du langage L_1 en des phrases d'un langage L_2 dont les concepts de base sont moins nombreux et moins complexes. On pourrait être tenté de choisir pour L_2 un sous-langage de L_1 , mais sur des exemples pratiques, on s'aperçoit rapidement que certains concepts de L_1 ne peuvent pas se décrire de façon plus simple dans le cadre de L_1 lui-même. On est alors obligé d'introduire dans L_2 de nouveaux concepts. Souvent, on imposera l'existence d'un langage L_3 contenant L_1 et L_2 et tel que l'application des transformations ne fasse pas sortir de L_3 (intuitivement, on impose ainsi que chacune des étapes intermédiaires de la traduction garde une "signification"). On obtient alors la configuration suivante :



Nous allons décrire dans la suite, une façon de traduire L_1 en L_2 . Dans la pratique, cette traduction devra conserver le "sens" des phrases de L_1 , ce qui permettra l'étude de la sémantique de L_1 en ne considérant que la sémantique de L_2 .

5.1.- Définition et utilisation d'un système transformationnel.

Définition 5.1.1.- Soit $G = (N,T,::=,X)$ et $G' = (N',T',::=',X')$ deux grammaires algébriques. On dit que G est contenue dans G' si

- $N \subset N'$
- $T \subset T'$
- $::= \subset ::='$
- $X = X'$

Si G est contenue dans G' , les propriétés suivantes sont alors trivialement vérifiées :

- Toute dérivation dans G est une dérivation dans G'
- Le langage engendré par G est contenu dans le langage engendré par G'
- Le bilangage engendré par G est contenu dans le bilangage engendré par G' .

Définition 5.1.2.- On appelle système transformationnel un triplet (G_1,G_2,P) tel que G_1 et G_2 sont des grammaires algébriques $(G_i=(N_i,T_i,::=_i,X_i)$, $i = 1 , 2)$ et P est un ensemble fini de couples de polynômes compatibles sur un vocabulaire V contenant $N_1 \cup N_2 \cup T_1 \cup T_2$.
Un système transformationnel sert à traduire le langage engendré par la grammaire G_1 dans le langage engendré par la grammaire G_2 en utilisant les productions de P sur les ramifications engendrées par G_1 .

Définition 5.1.3.- On appelle système transformationnel contrôlé, un quadruplet (G_1,G_2,G_3,P) tel que

- G_1 , G_2 , G_3 sont trois grammaires algébriques vérifiant $G_1 \subset G_3$ et $G_2 \subset G_3$. On notera $G_i = (N_i,T_i,::=_i,X_i)$ pour $i = 1,2,3$.
- P est un ensemble fini de couples (r,s) de polynômes injectifs compatibles sur $N_3 \cup T_3$ et vérifiant les conditions :

  1°) $\rho(r) = \rho(s)$

  2°) $c(r)$ et $c(s)$ sont engendrés au sens large par G_3 .

  3°) $(\forall i \in \mathbb{N}')(\forall A,B)(i \in F_A(r)$ et $i \in F_B(r)$ $\Longrightarrow A = B)$

(c'est-à-dire que la lettre qui est au dessus de la variable i dans r se retrouve au dessus de la variable i dans s).

Donc si (G_1,G_2,G_3,P) est un système transformationnel contrôlé, (G_1,G_2,P) est un système transformationnel, mais de plus, grâce aux conditions 1°), 2°) et 3°) imposées aux productions de P , on est sûr que l'application d'une production de P à une ramification engendrée par G_3 donne une ramification engendrée par G_3 .

Exemple : $G_1 = (\{A\} , \{a,b\} ,::=_1,A)$ avec $A::=_1 abA \mid ab$

$G_2 = (\{A\} , \{a,b\} ,::=_2,A)$ avec $A::=_2 aAb \mid ab$

$G_3 = (\{A\} , \{a,b\} ,::=_3,A)$ avec $A::=_3 abA \mid aAb \mid ab$

$P = \{(A \times(a+b+A\times 1) , A \times(a+A\times 1+b))\}$.

De façon évidente, le système transformationnel contrôlé $\mathcal{C} = (G_1,G_2,G_3,P)$ ci-dessus

permet la traduction du langage $L_1 = \{(a\ b)^n \ ; \ n > 0\}$ en le langage $L_2 = \{a^n\ b^n \ ; \ n > 0\}$.

Remarque 1.- Les conditions 1°) et 3°) imposées aux systèmes transformationnels contrôlés peuvent paraître restrictives. En fait, la condition $\rho(r) = \rho(s)$ sera facile à remplir dans la pratique à la condition de commencer la traduction suffisamment "haut" dans la ramification pour remonter jusqu'à un concept commun aux deux grammaires. Dans le cas des langages de programmation, on aura par exemple :

$\rho(r) = \rho(s) =$ < programme >

$\rho(r) = \rho(s) =$ < instruction >

$\rho(r) = \rho(s) =$ < expression arithmétique >

La condition 3°) est indispensable si on veut que le système transformationnel contrôlé ne fasse pas sortir des ramifications engendrées par G_3 .

Remarque 2.- On a imposé aux couples (r,s) de P d'être formés de polynômes injectifs. On peut imaginer des systèmes transformationnels contrôlés où cette condition n'est pas vérifiée. Les conditions 2°) et 3°) doivent être alors modifiées et sont plus longues à écrire. En fait, cette généralisation n'apporte pas grand_chose car, dans la pratique, on peut se ramener à des polynômes injectifs en allant plus "bas" dans la ramification à traduire et en augmentant éventuellement le nombre de variables.

5.2.-Problème fondamental lié à un système transformationnel.

5.2.1.- Introduction du problème.

Dans la pratique, on est confronté avec le problème suivant : un langage L_1 engendré par une grammaire G_1 est donné. On construit un langage L_2 engendré par une grammaire G_2 dans lequel on espère pouvoir traduire L_1 . On commence alors à imaginer des productions (r,s) qui permettent de commencer la traduction de L_1 en L_2 . Au bout d'un certain temps, on est persuadé d'avoir envisagé tous les cas possibles ; on obtient alors un système transformationnel $\mathcal{C} = (G_1,G_2,P)$ pour lequel on voudrait montrer qu'il vérifie la condition : pour toute ramification r engendrée par G_1 , il existe une ramification s engendrée par G_2 dérivant de r en utilisant les productions de P. Démontrer que cette condition est vérifiée, c'est résoudre le problème fondamental lié au système transformationnel $\mathcal{C}$. Cette condition s'écrit formellement :

$$(\forall r \in B(G_1))(\exists\, s \in B(G_2))(r \overset{*}{\underset{P}{\gg}} s) \quad .$$

Dans certains cas, on veut résoudre le problème fondamental sur des particuliers, c'est-à-dire en n'envisageant que certaines ramifications r de $B(G_1)$. Il s'agit alors de démontrer que le prédicat $\mathcal{C}(r)$ défini par

$$\mathcal{C}(r) = (\exists s \in B(G_2))(r \overset{*}{\underset{P}{\gg}} s)$$

est vrai pour les ramifications envisagées.

Si on introduit la bigrammaire $G(r)=(V-(N_2 \cup T_2), N_2 \cup T_2, P, r)$, démontrer que le prédicat $\mathcal{C}(r)$ est vrai, revient exactement à démontrer que le bilangage $BL(G(r))$ a une intersection non vide avec $B(G_2)$. Comme $B(G_2)$ est le bilangage engendré par une grammaire algébrique, $B(G_2)$ est en particulier un bilangage régulier (cf Quéré [20]). Cette remarque nous permettra de montrer que dans des cas particuliers, le prédicat $\mathcal{C}(r)$ est décidable.

5.2.2.- Indécidabilité du problème fondamental dans le cas général.

Si on examine les productions qui ont été utilisées pour démontrer le théorème 4.3.3. on s'aperçoit que ces productions vérifient les conditions imposées aux productions d'un système transformationnel contrôlé. Soit $\mathcal{G} = (N,T,P,X)$ une grammaire contextuelle. Considérons le système transformationnel contrôlé (G_1,G_2,G_3,P') tel que

$G_1 = (\{A\}, \{X\}, ::=, A)$ et $A::=X$

$G_2 = (\{A\}, T, ::=, A)$ avec $A::= AA$ et $A::=a$ pour chaque élément a de T

$G_3 = G_1 \cup G_2$

P' est l'ensemble des productions de la bigrammaire qui permet de simuler $\mathcal{G}$ et qui ont été définies dans la démonstration du théorème 4.3.3.

Résoudre le problème fondamental pour ce système transformationnel contrôlé, c'est exactement montrer que le langage engendré par $\mathcal{G}$ est non vide. Or, on sait que le problème de savoir si un langage contextuel est vide ou non, est récursivement indécidable (cf [8]).
On a donc obtenu la proposition :

Proposition 5.2.2.1.- Le problème fondamental pour les systèmes transformationnels est récursivement indécidable, même si on se restreint au cas particulier des systèmes transformationnels contrôlés dont les productions sont du type de celles employées dans les bigrammaires strictement contextuelles.

L'indécidabilité du problème fondamental nous amène à examiner des cas particuliers où on saura résoudre ce problème par algorithme. Nous allons étudier dans les deux

paragraphes suivants, deux cas où l'on sait résoudre le problème fondamental. Le premier cas correspond aux systèmes transformationnels dont les productions sont du type "contexte libre", le deuxième cas correspond aux systèmes transformationnels où l'on sait mesurer en un certain sens, l'éloignement d'une ramification engendrée par G_3 du bilangage engendré par G_2.

5.3.- Système transformationnel à contexte libre.

Définition 5.3.1.- On dit qu'un système transformationnel est à contexte libre, si ses productions sont du type de celles utilisées dans les bigrammaires à contexte libre. On parlera donc de C_0-systèmes transformationnels et de C_1-systèmes transformationnels suivant que les productions de ces systèmes transformationnels sont du type de celles utilisées dans les C_0-bigrammaires ou dans les C_1-bigrammaires. (Dans le cas des systèmes transformationnels, on peut avoir quelques difficultés à faire la distinction entre vocabulaire terminal et non terminal. Cette distinction a été introduite pour des raisons techniques, mais de même que pour les grammaires algébriques, la théorie ne change pas si ón ne suppose pas que le vocabulaire terminal et le vocabulaire non-terminal sont disjoints).
Dans le cas des systèmes transformationnels à contexte libre, les bilangages et les langages obtenus par traduction, ont les propriétés suivantes :

cas d'un C_0-système transformationnel (G_1,G_2,P) :
Si $r \in B(G_1)$, la bigrammaire $G(r)$ est une C_0-bigrammaire. L'ensemble des traductions possibles de r est $BL(G(r)) \cap B(G_2)$. Donc d'après la proposition 3.3.1.1 ce bilangage est engendré par une C_0-bigrammaire. Donc si α est un mot engendré par G_1, l'ensemble des traductions possibles de α forme en général un langage algébrique.
De même, la traduction du bilangage $B(G_1)$ est engendrée par une C_0-bigrammaire et l'ensemble des traductions des mots engendrés par G_1 forment un langage algébrique.

cas d'un C_1-système transformationnel (G_1,G_2,P) :
Si $r \in B(G_1)$, la bigrammaire $G(r)$ est une C_1-bigrammaire. L'ensemble des traductions possibles de r est $BL(G(r)) \cap B(G_2)$. Donc, d'après la proposition 3.3.2.1 ce bilangage est engendré par une C_1-bigrammaire. Contrairement au cas précédent, si α est un mot engendré par G_1, l'ensemble des traductions possibles de α peut ne pas être un langage algébrique.
La traduction du bilangage $B(G_1)$ est engendrée par une C_1-bigrammaire et l'ensemble des traductions des mots engendrés par G_1 ne forme pas en général un langage algébrique.

Remarque.-Le cas des systèmes transformationnels contrôlés et à contexte libre est

sans intérêt. En effet, vu les conditions que l'on impose aux productions de tels systèmes transformationnels, on s'aperçoit facilement qu'un système transformationnel contrôlé et à contexte libre (G_1,G_2,G_3,P) permettra la traduction du langage engendré par G_1 dans le langage engendré par G_2 si et seulement si la grammaire G_1 est contenue dans la grammaire G_2 .
On reprend les notations introduites au paragraphe 5.2.1.

Proposition 5.3.2.- Dans le cas d'un système transformationnel $\mathcal{C}$ à contexte libre, le prédicat $\mathcal{C}(r)$ est calculable.

Démonstration.- Pour chaque r , la bigrammaire $G(r)$ est à contexte libre. Donc, d'après les propositions 3.3.1.1. et 3.4.1.1., le bilangage $BL(G(r)) \cap B(G_2)$ est engendré par une bigrammaire à contexte libre (C_0-bigrammaire ou C_1-bigrammaire suivant que $\mathcal{C}$ est un C_0-système transformationnel ou un C_1-système transformationnel). Or dans le cas des bigrammaires à contexte libre, le problème de savoir si le bilangage engendré est vide ou nonest trivialement récursivement décidable.

Proposition 5.3.3.- Dans le cas des systèmes transformationnels à contexte libre, le problème fondamental est récursivement décidable.

Démonstration.- Il suffit de traiter le cas des C_1-systèmes transformationnels qui contient le cas des C_0-systèmes transformationnels. Soit $\mathcal{C} = (G_1,G_2,P)$ un C_1-système transformationnel.

- On peut supposer que les productions de P sont sous la forme réduite décrite en 3.2.1.2.
- Le bilangage $B(G_1)$ engendré par la grammaire G_1 est un bilangage régulier. Donc, il existe une bigrammaire régulière $G' = (N,N_1 \cup T_1,P',X)$ engendrant $B(G_1)$. On peut supposer que N est disjoint du vocabulaire V sur lequel travaille $\mathcal{C}$ et que les productions de G' sont du type de celles décrites en 3.1.2.2.
- Le bilangage $B(G_2)$ engendré par la grammaire G_2 est un bilangage régulier. Donc, d'après 3.3.2.3., il existe un dioïde généralisé fini D , une partie D' de D et un homomorphisme h de $\hat{V}_1$ dans D tel que $B(G_2) = c(h^{-1}(D'))$.
- Considérons la grammaire $G'' = (\{X_0\} \cup N \times D , (N_1 \cup T_1) \times D , P'', X_0)$ telle que
  - X_0 est un nouveau symbole
  - $((A,d),(a,d') \times (B,d'') + (C,d'')) \in P'' \Longleftrightarrow (A,a \times B+C) \in P'$ et $d' \times d'' +_G d'' = d$
  - $((A,d),\wedge) \in P'' \Longleftrightarrow (A,\wedge) \in P'$ et d est l'élément neutre de $\times$ dans D .

  - $(X_0,(X,d)) \in P'' \Longleftrightarrow d \in D'$.

Cette bigrammaire engendre un bilangage L_1 qui est un bilangage régulier.

Si f est la transcription de $(N_1 \cup T_1) \times D$ dans $N_1 \cup T_1$ définie par $f((a,d)) = a$, on a $f(L_1) = B(G_1)$.

Si g est l'application de $(N_1 \cup T_1) \times D$ dans D définie par

$$g(\wedge) = e \quad (e \text{ est l'élément neutre de } X \text{ dans } D)$$

$$g((a,d) \times r+s) = d \times g(r) +_G g(s) .$$

On obtient $(\forall r \in L_1)(g(r) \in D')$.

Réciproquement, il est facile de montrer que

$$r \in (N_1 \cup T_1) \times D \text{ et } f(r) \in B(G_1) \text{ et } g(r) \in D' \Longrightarrow r \in L_1 .$$

Intuitivement G'' engendre les ramificiations de $B(G_1)$ étiquetées par des éléments de D de façon que le calcul de g sur ces ramifications donne un élément de D' .

- Considérons l'ensemble P''' de productions défini par

$$((a,d)\times 1 , (b,d')\times 1 + (c,d'')) \in P''' \Longleftrightarrow (a \times 1 , b \times 1+c) \in P \text{ et } d = d' +_G d''$$

$$((a,d)\times 1 , (b,d') + (c,d'')\times 1) \in P''' \Longleftrightarrow (a \times 1, b + c \times 1) \in P \text{ et } d = d' +_D d''$$

$$((a,d)\times 1 , (b,d') \times (c,d'') \times 1) \in P''' \Longleftrightarrow (a \times 1 , b \times c \times 1) \in P \text{ et } d = d' \times d''$$

$$((a,d)\times 1 , (b,d') \times 1) \in P''' \Longleftrightarrow (a \times 1 , b \times 1) \in P \text{ et } d = d'$$

$$((a,d) \times 1 , b \times 1) \in P''' \Longleftrightarrow b \in N_2 \cup T_2 \text{ et } h(b \times 1) = d .$$

- Considérons la bigrammaire $G''' = (\{X_0\} \cup N \times D \cup V \times D , N_2 \cup T_2 , P'' \cup P''' , X_0)$.
  Il est facile de montrer que l'on a l'équivalence suivante

$$(\forall r \in B(G_1))((\exists s \in B(G_2))(r \overset{*}{\gg}_P s) \Longleftrightarrow (X_0 \overset{*}{\gg}_{G'''} s \text{ et } (\exists r' \in (N_1 \cup T_1) \times D)(X_0 \overset{*}{\gg}_{G'''} r' \text{ et } r' \overset{*}{\gg}_{G''} s \text{ et } f(r') = r))).$$

Considérons la bigrammaire $G^{IV} = (\{X_0\} \cup N \times D \cup V \times D , N_2 \cup T_2 , P^{IV}, X_0)$ obtenue à partir de G''' en effectuant la réduite inférieure et supérieure de G''' et soit G^V la bigrammaire

$$G^V = (\{X_0\} \cup N \times D , (N_1 \cup T_1) \times D , P'' \cap P^{IV} , X_0) .$$

Le problème fondamental associé à $\mathcal{C} = (G_1,G_2,P)$ revient alors à démontrer que $f(BL(G^V)) = B(G_1)$, donc à comparer deux bilangages réguliers, ce qui se fait de manière effective.
Il est facile de vérifier que chacune des étapes ci-dessus est effective. Donc, le problème fondamental associé à $\mathcal{C} = (G_1,G_2,P)$ est décidable.

5.4.- Système transformationnel séquentiel et quasi-séquentiel.

Dans ce paragraphe, nous n'envisagerons que des systèmes transformationnels contrôlés.

5.4.1.- Condition nécessaire pour que le problème fondamental associé à un système transformationnel ait une réponse positive.

Il n'est pas très évident de mettre en évidence de telles conditions nécessaires valables dans le cas général. Nous nous bornerons à donner deux conditions particulièrement simples.

Proposition 5.4.1.1.- Soit $\mathcal{C} = (G_1, G_2, G_3, P)$ un système transformationnel contrôlé tel que G_1 soit une grammaire réduite. Des conditions nécessaires pour que le problème fondamental associé à $\mathcal{C}$ ait une réponse positive sont que :

1°) pour tout non-terminal A de N_1 qui n'est pas dans N_2, il existe un couple (r,s) de P tel que A apparaisse dans r, sans apparaître dans s.

2°) pour toute production $A ::=_1 \alpha$ de G_1 qui n'est pas dans G_2, il existe un couple (r,s) de P tel que

$$\alpha \in F_A(r) \quad \text{et} \quad \alpha \notin F_A(s) .$$

5.4.2.- Condition suffisante pour que le problème fondamental associé à un système transformationnel ait une réponse positive.

Définition 5.4.2.1.- Soit $\mathcal{C} = (G_1, G_2, G_3, P)$ un système transformationnel contrôlé. Soit E l'ensemble des productions de G_3 qui sont des productions de G_1 sans être des productions de G_2, ou qui apparaissent dans des polynômes qui figurent en deuxième composante dans un élément de P. On dit que $\mathcal{C}$ est un système transformationnel séquentiel, si on peut totalement ordonner E de façon que $E = \{A_1 ::= \alpha_1, \ldots, A_n ::= \alpha_n\}$ et que les conditions suivantes soient vérifiées : Pour chaque i, soit $G^{(i)}$ la grammaire obtenue en adjoignant à G_2 les règles $A_j :: = \alpha_j$ $(j \geqslant i)$. Pour chaque i, il existe des productions (r,s) de P telles que :

- $\alpha_i \in F_{A_i}(r)$ et $c(s)$ sont engendrés au sens large par $G^{(i+1)}$
- pour chaque ramification t engendrée par $G^{(i)}$ et contenant la règle $A_i ::= \alpha_i$, une telle production est applicable à t.

Proposition 5.4.2.2.- Une condition suffisante pour que le problème fondamental associé à un système transformationnel contrôlé $\mathcal{C} = (G_1, G_2, G_3, P)$ ait une réponse positive est que $\mathcal{C}$ soit un système transformationnel séquentiel.

Cette proposition est évidente. En effet, dans un système transformationnel séquentiel, si une production $A ::= \alpha$ apparaît lors de la traduction d'une ramification de $B(G_1)$ et si elle n'est pas une production de G_2, une production (r,s) de P permet de faire disparaître la production $A ::= \alpha$ et les conditions imposées font qu'elle ne réapparaît pas dans la suite de la traduction.
Cependant les conditions imposées aux systèmes transformationnels séquentiels sont très restrictives. En effet dans un tel système, la traduction d'une production qui n'est pas dans G_2 doit se faire en une seule fois, ce qui exclut toute possibilité de récursivité dans la traduction.
Nous allons introduire un nouveau type de système transformationnel plus général où l'on saura aussi résoudre le problème fondamental.

Définition 5.4.2.3.- Soit $A ::= \alpha$ une production d'une grammaire algébrique G. On appelle fonction d'évaluation pour la production $A ::= \alpha$ une application e de l'ensemble des ramifications engendrées au sens large par G dans $\mathbb{N}$ qui vérifie les conditions suivantes :

1°) $e(r) = 0 \iff \alpha \notin F_A(r)$

2°) $e(r+r') \geqslant e(r) + e(r')$

3°) $B \neq A$ ou $\rho(r) \neq \alpha \implies e(B \times r) = e(r)$
$\rho(r) = \alpha \implies e(A \times r) > e(r)$.

Exemple : La fonction e donnant le nombre d'utilisations de la règle $A ::= \alpha$ dans une ramification est une fonction d'évaluation.

Définition 5.4.2.4.- Soit $\mathcal{C} = (G_1, G_2, G_3, P)$ un système transformationnel contrôlé. Soit E l'ensemble des productions de G_3 qui sont des productions de G_1 sans être des productions de G_2 ou qui apparaissent dans des polynômes qui figurent en deuxième composante dans un élément de P. On dit que $\mathcal{C}$ est un système transformationnel quasi-séquentiel si on peut totalement ordonner E de façon que $E = \{A_1 ::= \alpha_1, \ldots, A_n ::= \alpha_n\}$ et si pour chaque production $A_i ::= \alpha_i$, on peut trouver une fonction d'évaluation e_i et que les conditions suivantes soient vérifiées :

- les grammaires $G^{(i)}$ sont définies comme dans la définition 5.4.2.1.
- pour chaque i, il existe des productions (r,s) de P telles que $c(r)$ et $c(s)$ sont engendrées au sens large par $G^{(i)}$ et, $t \rightarrow t'$ en utilisant cette production implique $e_i(t) > e_i(t')$.
- pour chaque ramification t engendrée par $G^{(i)}$ et contenant la production

$A_i ::= \alpha_i$, une production (r,s) de P vérifiant la condition ci-dessus est applicable à t .

Proposition 5.4.2.5.- Une condition suffisante pour que l'on puisse résoudre le problème fondamental associé à un système transformationnel contrôle est qu'il soit quasi-séquentiel. Cette proposition est à peu près évidente.

Remarque et rappel.- On rappelle qu'un ensemble F est dit muni d'une relation d'ordre bien fondé s'il est muni d'une relation d'ordre (en général partiel) telle qu'il n'existe pas de suite infinie strictement décroissante dans F . Voir par exemple l'article de Jullien [10] où ces ensembles sont étudiés sous le nom d'ensembles bel-ordonnés. Ces ensembles munis de relation d'ordre bien fondé sont utilisés par exemple (cf. Vuillemin [13]) dans des techniques permettant de démontrer l'arrêt de programme. On utilise des fonctions u à valeur dans F qui sont strictement décroissantes quand on parcourt un programme. L'introduction de la notion de système transformationnel quasi-séquentiel procède de la même idée. En effet, considérons sur $\mathbb{N}^n$ l'ordre lexicographique qui est un ordre bien fondé et l'application de $u : B(G_3) \longrightarrow \mathbb{N}$, définie par $u(r) = (e_1(r) , e_2(r),..., e_n(r))$; dire que le système transformationnel $\mathcal{T} = (G_1,G_2,G_3,P)$ est quasi-séquentiel, implique en particulier que la fonction u est décroissante au cours de la traduction d'une ramification r (tout du moins si on traite les règles $A_1 ::= \alpha_1$,..., $A_n ::= \alpha_n$ dans l'ordre 1 , 2 ,..., n). On pourrait donc envisager des systèmes transformationnels pour lesquels le problème fondamental se résoudrait grâce à la théorie des ordres bien fondés. Il nous semble que la définition des systèmes transformationnels quasi-séquentiels est suffisamment large pour englober la plupart des cas rencontrés dans la pratique.

CONCLUSION

La première partie de ce travail montre que les résultats bien connus sur les langages se généralisent assez facilement aux bilangages. Certains résultats sont surprenants à priori ; en particulier, il faut imposer des conditions très fortes aux bigrammaires pour ne pas engendrer des langages trop généraux (récursivement énumérables).

Un certain nombre de problèmes n'ont pas été résolus ici :

- Caractériser les C_1-bigrammaires qui engendrent des bilangages réguliers. Ce problème est évidemment indécidable dans le cas général, puisqu'il contient le problème de la caractérisation des grammaires algébriques engendrant des langages réguliers. Cependant, nous avons donné une façon d'engendrer les bilangages réguliers avec des C_0-bigrammaires dont les règles sont de la forme $(A, a \times B+C)$ ou (A, Λ) , ce qui correspond à la notion de grammaire linéaire gauche. Il serait intéressant de trouver dans les C_1-bigrammaires une classe de bigrammaires engendrant tous les bilangages réguliers et correspondant à la notion de grammaire linéaire droite.
- Les C_1-bigrammaires engendrent des langages d'un type assez particulier. On pourrait chercher un type de grammaire engendrant exactement ces langages.
- Nous avons donné un type de bigrammaires qui engendre exactement les langages contextuels. On pourrait chercher des conditions plus générales sur les productions (r,s) d'une bigrammaire pour que les langages engendrés soient des langages contextuels.

La deuxième partie sur les systèmes transformationnels n'a été qu'ébauchée. Vu les résultats que l'on a obtenu sur les bigrammaires, il est bien évident que la plupart des problèmes que l'on peut se poser, sont indécidables dans le cas général. Il serait intéressant, cependant, de trouver des conditions nécessaires et des conditions suffisantes pour que le problème fondamental associé à un système transformationnel soit décidable. L'introduction des systèmes transformationnels séquentiels et quasi-séquentiels va dans ce sens.

On pourrait d'autre part, faire une étude analogue à la théorie des schémas de programmes récursifs pour les problèmes de traductions. Dans un cas particulier, ce type de problème pourrait s'énoncer ainsi :

On considère un langage L engendré par une grammaire G et on veut le traduire en un langage L' engendré par une grammaire G' . Soient A , B ,... les non-terminaux de G qui ne sont pas des non-terminaux de G' et qui nécessitent donc une traduction. Pour chacun de ces non-terminaux, on se donne une définition

récursive du type

A <=== si p alors r sinon s

où r et s sont des polynômes injectifs de degré 1 à une variable 1 et p est un prédicat sur des ramifications. Pour traduire un mot α engendré par G , on effectue l'analyse syntaxique de α , ce qui donne une ou plusieurs ramifications t engendrées par G . Ensuite, pour traduire t , on parcourt cette ramification, et chaque fois que l'on rencontre A par exemple, on le remplace par sa définition en tenant compte de la valeur de p sur la ramification qui est en dessous de A dans t . Ce genre de systèmes de traduction récursifs va évidemment être lié à la théorie des C_1-bigrammaires. La théorie devrait être analogue à la théorie des schémas récursifs monadiques . On peut aussi envisager des systèmes de traduction récursifs utilisant la théorie des bigrammaires contextuelles ou strictement contextuelles. Les définitions récursifs utilisées seraient alors du type

r <=== si p(1,2,...,n) alors s sinon s'

où r , s , s' sont des polynômes à n variables et p est un prédicat à n variables sur des ramifications.

Ce genre de problèmes sera étudié dans un travail ultérieur.

BIBLIOGRAPHIE

[1] P. BERLIOUX, Etude d'automates et de grammaires de ramifications, Thèse de Spécialité, Grenoble (1972).

[2] W. BRAINERD, Tree generating regular systems, Information and control 14, p. 217-231 (1969).

[3] N. CARBONELL, Rôle des fonctions récursives primitives de ramifications dans la définition d'une langue naturelle, Thèse de Spécialité, Nancy (1973).

[4] N. CHOMSKY, Formal properties of grammars, Hand book of mathematical psychologie, (Luce, Bush and Galanter editors), Wiley and Sons, p. 323-418 (1963).

[5] M. DAVIS, Computability and unsolvability, Mac Graw-Hill New-York (1958).

[6] S. GINSBURG, The mathematical theory of context-free languages. Mac Graw Hill, New-York (1966).

[7] GROSS et LENTIN, Notions sur les grammaires formelles, Gauthier-Villars Paris (1967).

[8] J. E. HOPCROFT and J. D. ULLMAN, Formal languages and their relation to automata, Addison-Wesley, Reading, mars 1969.

[9] A. JOSHI, L. LEVY and M. TAKAHASHI, A tree generating system, Dans automata, languages and programming, (NIVAT editor), North Holland Amsterdam, p. 453 à 465 (1973).

[10] P. JULLIEN, Notion de prébelordre dans ordres totaux finis, Gauthier-Villars Paris (1971).

[11] C. LEMAIRE, Système général d'analyse syntaxique, Thèse de spécialité, Nancy (1971).

[12] P. LESCANNE, Etude de quelques théories des langages et généralisation du théorème de Kleene, Thèse de Spécialité, Nancy (1971).

[13] Z. MANNA, S. NESS and J. VUILLEMIN, Inductive methods for proving properties of programs, CACM 18, p. 491-502 (1973).

[14] R. MOHR, Modèle algébrique pour l'analyse syntaxique de figure, Thèse de Spécialité, Nancy (1973).

[15] M. NIVAT, Langages algébriques sur le magma libre et Sémantique des schémas de programme, Dans automata languages and programming (NIVAT M. editor), p. 293-357, North Holland Amsterdam (1973).

[16] C. PAIR, Préliminaires à l'étude algébrique des ramifications, Communication soumise au Congrès 1968 de l'IFIP.

[17] C. PAIR, Sur les notions algébriques liées à l'analyse syntaxique, Exposé fait au Centre d'Automatique de l'Ecole des Mines, Fontainebleau, mars 1969.

[18] C. PAIR et A. QUERE, Définition et étude des bilangages réguliers, Information and control 13, p. 565-593 (1968).

[19] C. PAIR et A. QUERE, Sur les fonctions récursives primitives de ramifications (1969).

[20] A. QUERE, Etude des ramifications et des bilangages, Thèse de Spécialité, Nancy (1969).

[21] B. ROSEN, Subtree replacement systems, Center for research in computing technology, Harward University.

[22] W. C. ROUNDS, Mappings and grammars on trees, Math. systems theory 4, p. 257-287 (1970).

[23] SZASZ, Théorie des treillis, Dunod France (1971).

[24] J. N. THATCHER, Characterizing derivation tries of context free grammars through generalized finite autamata theory, IBM Research Note NC 719.

-=o0o=-

NOM DE L'ETUDIANT : MARCHAND Pierre

NATURE DE LA THESE : Doctorat de Spécialité en Mathématiques Appliquées

VU, APPROUVE

& PERMIS D'IMPRIMER

NANCY, le 1 11 74

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

UNIVERSITÉ DE NANCY I
Le Président

J.R. HELLUY