MATHÉMATIQUES ET SCIENCES HUMAINES

M. C. BARBAULT
J. P. DESCLÉS

Vers une formalisation des grammaires transformationnelles

Mathématiques et sciences humaines, tome 34 (1971), p. 27-41
<http://www.numdam.org/item?id=MSH_1971__34__27_0>

Math. Sci. hum. (9e année, no 34, 1971, p.27-41

VERS UNE FORMALISATION DES GRAMMAIRES TRANSFORMATIONNELLES [1]

par

M. C. BARBAULT [2] et J. P. DESCLÉS [3]

Une transformation formelle est vue dans cet article comme un passage d'un mot accompagné de l'histoire de sa production à un autre mot, lui-même accompagné de sa propre histoire. Étant donné un système formel, une transformation formelle sera donc une règle qui « transforme» la démonstration d'un théorème t *en une démonstration d'un théorème* t'. *Les auteurs étudient ici les transformations « d'arbres en arbres», c'est-à-dire les règles de passages entre démonstrations (ou histoires de production) structurées en arbres. Ces transformations «d'arbres en arbres» sont les seules transformations envisagées par N. Chomsky* [5], [6] *dans les théories transformationnelles (voir l'article précédent dans ce même numéro) qu'il propose.*

1. EXTENSION DU CALCUL DANS UN MONOÏDE LIBRE

1.1. Calculer à l'intérieur d'un monoïde libre, c'est effectuer une série de *transformations entre mots;* la transformation la plus générale est une transformation dite *semi-thuienne* de la forme (voir Gross-Lentin [13]):

$$\hat{g}\,\hat{m}\,\hat{d} \rightarrow \hat{g}\,\hat{\overline{m}}\,\hat{d} \qquad \text{avec} \qquad |\hat{\overline{m}}| \neq 0$$

où $\hat{g}$, $\hat{d}$, $\hat{m}$, $\hat{\overline{m}} \in \mathscr{A}^*$ ($\mathscr{A}^*$ est le monoïde libre sur l'alphabet $\mathscr{A}$).

On dit que le mot $\hat{m}$ se transforme (ou est réécrit) en un autre mot $\overline{m}$ dans le contexte $\hat{g}$, $\hat{d}$.

1.2. Ce calcul ainsi défini ouvre la voie aux *grammaires formelles* (*cf.* [7], [8], [13]): soit V (vocabulaire), l'« alphabet» du système formel; si $\{ V_N, V_T \}$ est une partition de V, on peut apporter des restrictions aux transformations semi-thuiennes entre mots (*cf.* [8], [10], [13]):

1.2.1. *Règles de Kleene* (ou régulières):

$$\begin{cases} m \rightarrow x\,\overline{m} \\ m \rightarrow x \end{cases} \qquad \text{avec} \qquad \begin{cases} m, \overline{m} \neq \emptyset \\ m, \overline{m} \in V_N \\ x \in V_T \end{cases}$$

1. Les auteurs remercient A. Lentin des critiques et améliorations qu'il a proposées.
2. Centre de Documentation en Sciences Humaines, CNRS.
3. Paris VII.

1.2.2. *Règles C.F.* (ou indépendantes du contexte) :

$$m \to \hat{\overline{m}} \qquad \text{avec} \qquad \begin{cases} \hat{m},\ m,\ \neq \emptyset \\ m \in V_N \\ \hat{\overline{m}} \in V^* = (V_N \cup V_T)^* \end{cases}$$

1.2.3. *Règles C.S.* (ou contextuelles) :

$$\begin{aligned} \hat{g}\ m\ \hat{d} &\to \hat{g}\ \hat{\overline{m}}\ d \\ \hat{m} &\to \hat{\overline{m}} \end{aligned} \qquad \text{avec} \qquad \begin{aligned} & m,\ \hat{m},\ \hat{\overline{m}} \neq \emptyset \\ & m \in V_N \\ & \hat{g},\ \hat{d},\ \hat{m},\ \hat{\overline{m}} \in V^* \\ & |\ \hat{\overline{m}}\ | \geqslant |\ \hat{m}\ | . \end{aligned}$$

Une grammaire formelle G est alors la donnée du quadruplet $< V, V_T, P, \mathscr{R} >$ avec P un élément distingué de V_N, dit « axiome» de la grammaire et $\mathscr{R}$ un ensemble *fini* de règles d'un même type: soit de Kleene, soit C.F., soit C.S.; on a alors, selon le type de règles, une K-grammaire, une C.F.-grammaire, une C.S.-grammaire (*cf.* [7], [8], [10], [13]). Le langage produit par une grammaire G est l'ensemble:

$$L(G) = \{\ \hat{m} \in V_T^*;\quad P \Rightarrow \hat{m}\ \}.$$

1.3. Nous allons étendre le calcul à l'intérieur d'un monoïde libre, par un calcul « plus général» non plus sur des mots mais sur des suites finies d'arbres étiquetés sur un vocabulaire ou *dendrons* en définissant le *binoïde libre*, c'est-à-dire un ensemble de séquences finies d'arbres étiquetés — ou mots d'arbres étiquetés — où s'effectueront toutes les *transformations « d'arbres en arbres étiquetés »*. De même que l'on définissait un langage (formel) comme une partie d'un monoïde libre, on définira un *bilangage* comme une partie d'un binoïde libre. Une grammaire transformationnelle GT aura pour but d'*énumérer exactement un bilangage et d'en caractériser les éléments ;* elle sera définie par la donnée d'un quadruplet $< V, \mathscr{B}, \mathscr{T}, \mathscr{C} >$ où V, le vocabulaire, déterminera l'univers linguistique (et formel) dans lequel on se place; $\mathscr{B}$ s'appellera la *base* de la grammaire: en général, un ensemble d'arbres étiquetés, histoires des productions des mots qui appartiennent à un langage produit par une grammaire C.F. ou autre; $\mathscr{T}$ sera un ensemble *fini* de transformations entre arbres ou mots d'arbres et $\mathscr{C}$, un ensemble, éventuellement vide, de conditions restrictives. Une transformation par GT consiste à partir d'un ensemble de la base et à lui appliquer une série de transformations de $\mathscr{T}$; le calcul s'arrête lorsqu'aucune transformation n'est plus applicable.

2. ARBRES ET DENDRONS

2.1. Puisque nous transformons des arbres, il nous faut en donner une définition précise et commode. La structure d'arbre est bien connue et employée souvent, par exemple dans la constitution d'un dictionnaire ou d'un livre; en effet, la suite de symboles *3.7.2.10.5* peut représenter « le chapitre *3*, la section *7*, la sous-section *2*, le paragraphe *10*, le verset *5*» et fournit l'adresse précise d'un renseignement que désire obtenir un lecteur de l'ouvrage; la recherche ne présente aucune difficulté parce que le lecteur exploite la structure d'arbre du dictionnaire.

2.2. Soit N^+ l'ensemble des entiers strictement positifs; on construit le monoïde libre $\mathscr{N}$ de générateur N^+, dont le neutre est 0, puis on munit $\mathscr{N}$ d'une *relation d'ordre* H, qui se lit « $\hat{n}$ dépend hiérarchiquement de $\hat{m}$»:

$$\forall\ \hat{m}, \hat{n} \in \mathscr{N} : H(\hat{m}, \hat{n}) \Leftrightarrow \exists\ \hat{x} \in \mathscr{N} : \hat{m} \,.\, \hat{x} = \hat{n}.$$

Autrement dit, sur le monoïde libre $\mathcal{N}$, entre $\hat{m}$ et $\hat{n}$, il existe la relation H si et seulement si, « $\hat{m}$ est un *facteur gauche* de $\hat{n}$» ou« $\hat{n}$ admet $\hat{m}$ comme facteur gauche».

Toute partie *finie* D de $\mathcal{N}$ telle que:

$$(a_1) \quad \hat{n} \in D, \qquad H(\hat{m}, \hat{n}) \Rightarrow m \in D$$

$$(a_2) \quad \hat{m} \cdot j \in D, i < j \Rightarrow \hat{m} \cdot i \in D$$

$$(a_3) \quad 0 \in D$$

est un *domaine structuré en arbre* que, par la suite et par abus de langage, nous appellerons simplement *arbre*. Notons que dans la « théorie des graphes» [3], la partie D définit ce que nous appelons ailleurs « arborescence doublement orientée» [1]; nous avons en effet deux ordres: un ordre hiérarchique ou de dépendance et un ordre sur les successeurs immédiats de chaque sommet; parce que nous avons ce deuxième ordre (séquentiel), nous sommes autorisés à tracer sur une feuille de papier ou sur un« listing» l'arborescence doublement orientée D où l'on distingue la gauche de la droite. Il résulte de l'axiome (a_1) que si D contient $\hat{n}$, D contient nécessairement tous les facteurs gauches stricts de $\hat{n}$. L'adresse d'un élément de D, appelée *sommet* ou *nœud*, par exemple l'adresse *3.7.2.10.5.*, indique très clairement le *chemin* qu'il faut suivre pour atteindre ce nœud: c'est un algorithme de recherche, car on démontre qu'entre la *racine* 0 et un sommet quelconque $\hat{n}$ d'un arbre D, il existe un chemin unique. Ce concept d'arbres formalise l'analyse en constituants immédiats [5] ou encore les graphes de dépendance de Tesnière [6]. Dans ces analyses linguistiques, chaque nœud porte une information, c'est-à-dire une étiquette: on introduit alors le concept d'arbre étiqueté sur un vocabulaire d'étiquettes V; c'est le concept de *dendron sur V* que l'on appellera, si aucune ambiguïté n'apparaît, simplement dendron.

2.3. Soit V le vocabulaire, disjoint de $\mathcal{N}$ dans la plupart des cas; un *dendron sur V* est la donnée d'une application α d'un domaine structuré en arbre D dans V. On notera D (α), le domaine de α et $\mathcal{Y}$ l'ensemble de tous les dendrons sur V, ou *forêt* étiquetée sur V.

Proposition 1. A tout dendron sur V, on associe canoniquement un mot du monoïde libre $(V \cup \{\,], [\,\})^*$.

En effet, un dendron sur V s'exprime sous forme d'une séquence linéaire en « ouvrant» et en « fermant» des parenthèses ( ], [ ) sous chaque nœud (*cf. infra* 8.3.). Nous supposons que cette association canonique est connue du lecteur. La notation α désignera aussi bien l'application de D dans V que le mot associé canoniquement. Désignons également par $\mathcal{Y}$ le *langage* construit à l'aide des règles:

(*i*) $V \subset \mathcal{Y}$

(*ii*) $l \in V; n > 0; \alpha_1, \ldots, \alpha_n \in \mathcal{Y} \Rightarrow l] \, \alpha_1 \ldots \alpha_n \, [\in \mathcal{Y}$

(*iii*) tout mot de $\mathcal{Y}$ est construit avec les règles (*i*) et (*ii*) et elles seulement.

On se définit deux applications *racine* et *terminaux*:

$$RAC: \quad \mathcal{Y} \to V \qquad \text{avec} \qquad \alpha \in V^D$$
$$\alpha \mapsto l = \alpha(0)$$

$$TER: \quad \mathcal{Y} \to V^* \qquad \text{avec} \qquad l_i = \alpha(\hat{m})$$
$$\alpha \mapsto l_1 \ldots l_i \ldots l_p$$

où $\hat{m}$ n'est facteur gauche strict d'aucun autre élément, soit:

$$\{\hat{n}; H(\hat{m}, \hat{n})\} = \emptyset.$$

3. OPÉRATIONS ÉLÉMENTAIRES SUR LES DENDRONS

Ces opérations (*cf.* [1], [4], [17]) agissent soit sur les arbres, ce sont alors des transformations structurelles, soit directement sur les dendrons.

3.1. Coupe au nœud a

Soit $a \in D$; l'opération « coupe en a», notée $/a$, définit un autre domaine à partir de D:

$$D \xrightarrow{/a} D' = \{ b'; \exists\, b \in D: b = a \,.\, b' \}.$$

On notera les nœuds b', tels que $b = a \,.\, b'$, par b/a.

3.2. Sous-dendron de α au nœud a

Soit $a \in D(\alpha)$; l'opération « sous-dendron», notée aussi $/a$ est une transformation:

$$\alpha \xrightarrow{/a} \alpha/a = \{ (b, l); (a \,.\, b, l) \in \alpha \}.$$

3.3. Remplacement de sous-dendron α/a par le dendron β

Soit $a \in D(\alpha)$; la transformation « remplacement» substitue au sous-dendron α/a, le dendron β accroché au nœud a et l'on note cette opération: $a \leftarrow \beta$:

$$\alpha \xrightarrow{a \leftarrow \beta} \alpha\,[a \longleftarrow \beta] = \{ (b, l); (b \,/\, a, l) \in \alpha \} \cup \beta.$$

3.4. Le « remplacement» de α/a par un dendron réduit à un élément $(b, \alpha(a))$ est le « sur-dendron» restant après la coupe au nœud a. Nous noterons $\alpha\text{-}\alpha/a$ l'effacement de α/a dans α.

4. BINOÏDE SUR V

Introduisons, à la suite de C. Pair [14], le *binoïde sur V;* un binoïde B sur V est la donnée d'un quadruplet $< S(B), \frown, \Delta, \Lambda >$ où S (B) est un ensemble (l'ensemble sous-jacent à B) muni de deux lois:

— la *concaténation* $\frown$, opération interne, associative dont l'élément neutre est Λ (suite vide, de longueur nulle);

— l'*enracinement* Δ, opération externe à opérateur dans V:

$$\forall\, l \in V, \forall\, \rho \in S(B): l \,\Delta\, \rho = \rho' \in S(B).$$

Tout élément de B s'appelle *polydendron* [1]: c'est un mot, c'est-à-dire une séquence ordonnée et finie, composée d'éléments de S (B). Toute partie de B s'appelle un *bilangage* ([14]).

1. Ce terme est dû à A. Lentin.

4.1. Remarques

1) C. Pair [14] (*cf.* article de C. Pair dans ce numéro) appelle *ramification* ce que nous nommons polydendron.

2) La concaténation $\frown$ dans la suite de cet article ne recevra, comme il est fréquent, aucune marque spécifique dans une expression où elle occurre; ainsi, ce que l'on devrait écrire:

$$\forall\ \rho, \rho' \in S\ (B): \rho \frown \rho' = \rho''$$

s'écrira:

$$\forall\ \rho, \rho' \in S\ (B): \rho\ \rho' = \rho''.$$

4.2. Proposition 2. Le monoïde libre $\mathscr{V}^*$, dont le générateur est $\mathscr{V}$, est un binoïde libre sur V.

La preuve est laissée à l'initiative du lecteur. On démontre que $\mathscr{V}^*$ est effectivement un objet libre. Remarquons que la forêt $\mathscr{V}$ est un ensemble de dendrons et que $\mathscr{V}^*$ est un ensemble de mots $\alpha_1, \alpha_2 \ldots \alpha_n$, c'est-à-dire de polydendrons. On peut toujours « enraciner » deux polydendrons pour former un nouveau dendron, élément du binoïde libre, à condition de s'imposer les identifications naturelles: $V \subset \mathscr{V} \subset \mathscr{V}^*$.

Ainsi:

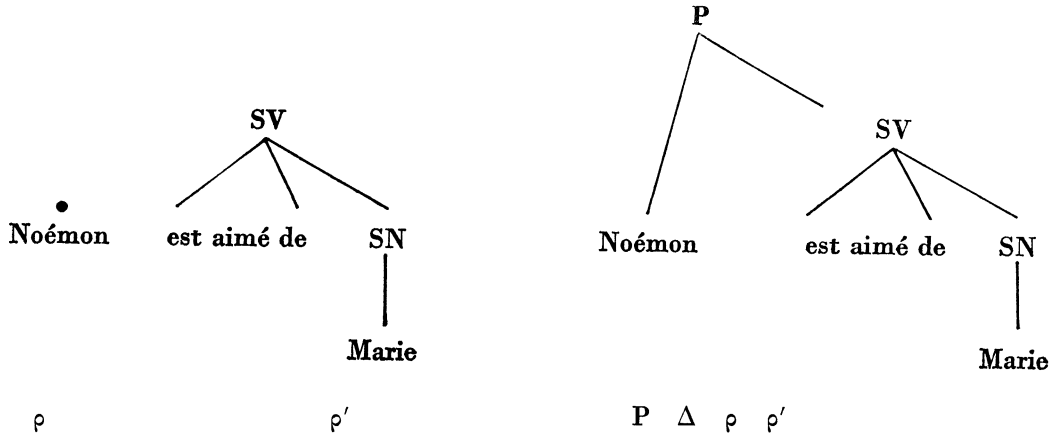

Fig. 1

Étendons les applications *RAC* et *TER* à $\mathscr{V}^*$:

$$\begin{aligned} RAC:\ & \mathscr{V}^* \to V^* \\ & \alpha_1 \ldots \alpha_n \mapsto RAC\ (\alpha_1) \ldots RAC\ (\alpha_n) \\ TER:\ & \mathscr{V}^* \to V^* \\ & \alpha_1 \ldots \alpha_n \mapsto TER\ (\alpha_1) \ldots TER\ (\alpha_n). \end{aligned}$$

Ainsi, dans l'exemple ci-dessus:

$RAC\ (\rho) = TER\ (\rho) =$ Noémon

$RAC\ (P\ \Delta\ \rho\ \rho') = P$

$TER\ (P\ \Delta\ \rho\ \rho') =$ Noémon est aimé de Marie.

5. CALCUL DANS LE BINOÏDE LIBRE $\mathscr{V}^*$

On va établir un calcul « plus général » que le calcul des mots, en calculant sur des polydendrons.

5.1. Soit ρ un polydendron de $\mathscr{V}^*$; on dit que ρ' *occurre* dans ρ si et seulement si:

$$\rho' \in \mathscr{V}^*, \exists\, \gamma, \delta \in \mathscr{V}^* : \rho = \gamma\, \rho'\, \delta.$$

Une transformation T qui opère dans le binoïde libre $\mathscr{V}^*$ est une donnée d'un couple (i (T), t (T)) avec:

$$\mathbf{i}\,(T) \in \mathscr{V}^* \quad \text{source (ou initial) de T}$$
$$\mathbf{t}\,(T) \in \mathscr{V}^* \quad \text{but (ou terminal) de T,}$$

on note alors:

$$\mathbf{i}\,(T) \xrightarrow{T} \mathbf{t}\,(T) \qquad \text{ou encore:} \qquad \mathbf{i}\,(T) \longrightarrow \mathbf{t}\,(T).$$

5.2. Une transformation T est *applicable* au polydendron ρ pour donner le résultat ρ' si et seulement si:

$$\exists\, \gamma, \delta \in \mathscr{V}^* : \rho = \gamma\, \mathbf{i}\,(T)\, \delta;$$

on note:

$$\rho = \gamma\, \mathbf{i}\,(T)\, \delta \xrightarrow{T} \gamma\, \mathbf{t}\,(T)\, \delta = \rho',$$

ou encore:

$$\rho \xrightarrow{T} \rho'.$$

Appliquer T au polydendron ρ, c'est donc vérifier que i (T) occurre dans ρ puis substituer à i (T) le polydendron t (T) qui occurre alors dans ρ', polydendron « *transformé de* » ρ par la transformation T. Le calcul dans un binoïde libre est « plus riche » que le calcul dans un monoïde libre; pour s'en convaincre, il suffit d'identifier toute lettre l de V avec un dendron sur V, d'où l'inclusion: $V^* \subset \mathscr{V}^*$; toute transformation entre mots de V^* sera une transformation entre polydendrons de $\mathscr{V}^*$.

5.3. Soit $\mathscr{S} = \{ T_1, \ldots, T_n \}$ un ensemble de transformations entre polydendrons. Le polydendron ρ', est dit *transformé du polydendron* ρ par l'ensemble $\mathscr{S}$ en *p pas de calcul*, si et seulement si:

$$\exists\, \sigma_0, \sigma_1, \ldots, \sigma_p \in \mathscr{V}^* :$$

(*i*) $\sigma_0 = \rho$

(*ii*) $\forall\, i \in]\, p\,], \exists\, T_{j_i} \in \mathscr{T}, j \in]\, n\,]$:

$$\sigma_{i-i} \xrightarrow{T_{j_i}} \sigma_i$$

(*iii*) $\sigma_p = \rho'$.

On écrit alors:

$$\rho \overset{\mathscr{T}}{\Rightarrow} \rho' \qquad \text{ou encore} \qquad \rho \Rightarrow \rho'.$$

La notation $]\, p\,]$ est la donnée de l'ensemble $\{ 1, 2, \ldots, p \}$; par contre $[\, p\,]$ est la donnée de l'ensemble $\{ 0, 1, 2, \ldots, p \}$.

Nous avons défini une transformation de ρ en ρ' en appliquant une transformation T_{j_1} de $\mathscr{T}$, transformation qui a donné pour résultat σ_1; appliquant une transformation T_{j_2} de $\mathscr{T}$ à σ_1, on obtient σ_2; finalement, en appliquant T_{j_p} à σ_{p-1}, on obtient ρ'.

6. GRAMMAIRE TRANSFORMATIONNELLE

Soit GT = < V, $\mathscr{B}$, $\mathscr{T}$, $\mathscr{C}$ > une grammaire transformationnelle (*cf. supra* 1.3.); nous savons maintenant ce que signifient les transformations successives opérées par la grammaire G T. Le vocabulaire V définit l'univers linguistique $\mathscr{V}^*$: c'est le binoïde libre sur V; $\mathscr{B}$ est un bilangage contenu dans $\mathscr{V}^*$: c'est le *bilangage de base; $\mathscr{T}$* est un ensemble *fini* de transformations entre polydendrons de $\mathscr{V}^*$; $\mathscr{C}$ est un ensemble de conditions restrictives.

6.1. Le bilangage transformé par GT à partir du bilangage de base $\mathscr{B}$ est:

$$\mathrm{LL\ (GT)} = \{\, \rho \in \mathscr{V}^*;\ \exists\ \rho_0 \in \mathscr{B}\colon \rho_0 \overset{\mathscr{T}}{\Rightarrow} \rho \,\}.$$

6.2. Le *langage produit par G T* est:

$$\mathrm{L\ (GT)} = \{\, \hat{m} \in \mathrm{V}^*;\ \hat{m} = \mathit{TER}\ (\rho)\colon \rho \in \mathrm{LL\ (G\ T)} \,\}.$$

7. PROBLÈME CENTRAL

Des motivations linguistiques nous ont amenés à définir les *transformations formelles.* Nous pouvons énoncer un problème central de la théorie des grammaires transformationnelles formelles, problème qui aura, nous le verrons à la fin de cet article, des conséquences théoriques pour la « théorie linguistique naïve» des grammaires transformationnelles.

7.1. Soit G une grammaire formelle, de Kleene ou C.F. qui produit un langage L (G). A tout mot $\hat{m}$ de L (G) est associée une « histoire de production», c'est-à-dire un dendron; on associe ainsi au langage L (G) un bilangage LL (G), dit *bilangage induit par G.* Soit maintenant une grammaire transformationnelle GT, dont la base $\mathscr{B}$ est LL (G); la grammaire GT transforme le bilangage LL (G) en un bilangage LL (GT) et le langage L (GT) est dit *langage transformé* du langage L (G) par GT. S'il existe une grammaire G′ qui produit *exactement* le même langage, on dit alors que *GT induit G′ à partir de G.* On résume ceci par le schéma:

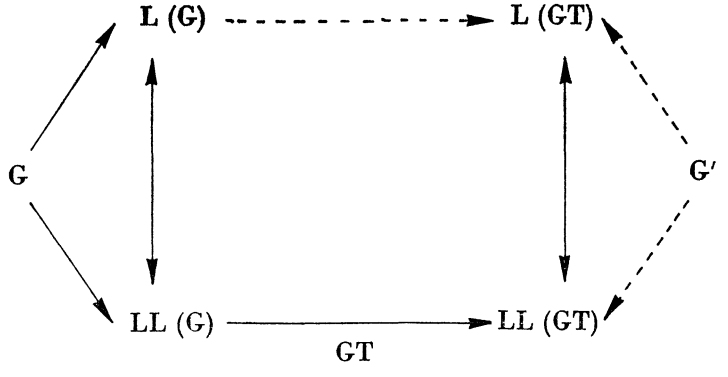

Fig. 2

7.2. PROBLÈME

Étant donné une classe $\mathscr{K}$ de grammaire G, trouver une grammaire transformationnelle GT, telle qu'il existe dans la classe $\mathscr{K}$ au moins une grammaire G à partir de laquelle GT induise une grammaire G' qui n'appartienne plus à la classe $\mathscr{K}$. On dit que GT *augmente la puissance* ou, si l'on veut préciser, *fait sortir* de la classe $\mathscr{K}$.

7.3. Deux grammaires appartiennent à la classe $\mathscr{K}$ (classe des K.-grammaires, des C.F.-grammaires, des C.S.-grammaires...) si toutes les règles sont du même type (de Kleene, C.F., C.S. ..., *cf. supra* 1.1.). Une grammaire G n'appartient pas à une classe $\mathscr{K}$ si le langage L (G) produit par G, ne peut être produit par *aucune* grammaire G' de la classe $\mathscr{K}$: G est dite «plus puissante» que toute grammaire G' de $\mathscr{K}$; il en résulte que les règles de G ne sont pas du type requis pour la classe $\mathscr{K}$. Ainsi, toute grammaire qui produit $L_{nn} = \{ a^n b^n ; n \geqslant 1 \}$ n'appartient pas à la classe des K-grammaires ; la grammaire G_{nnn} qui produit $L_{nnn} = \{ a^n b^n c^n ; n \geqslant 1 \}$ n'appartient pas à la classe des C.F. grammaires et elle est plus puissante que la grammaire G_{nn} qui produit L_{nn}.

On parlera, par extension, de la classe des langages, des bilangages associés à une classe de grammaires.

8. GRAMMAIRES TRANSFORMATIONNELLES SIMPLES

Ce type de grammaires transformationnelles que l'on appelle aussi *régulières* (*cf.* [4], [14], [16]) ainsi que les résultats qui suivent ont été introduits à peu près simultanément par Brainerd [4], Thatcher [16] Pair [14] et Barbault-Desclés [1].

8.1. UNE GRAMMAIRE TRANSFORMATIONNELLE SIMPLE

GTS est un triplet $< V, \mathscr{B}, \mathscr{T} >$ où $\mathscr{B}$ est sous-ensemble de la forêt $\mathscr{V}$ et $\mathscr{T}$ un ensemble *fini* de transformations entre dendrons dites transformations *simples*, définies ci-dessous :

8.1.1. On a une *transformation sous le nœud a :*

$$\alpha \xrightarrow{a} \beta \Leftrightarrow \begin{cases} (i)\ a \in D(\alpha) \\ (ii)\ \exists\, T \in \mathscr{T} : \alpha / a = i(T),\ \beta = \alpha\,[a \leftarrow t(T)]. \end{cases}$$

8.1.2. On a une *transformation directe entre dendrons :*

$$\alpha \to \beta \Leftrightarrow \exists\, a \in D(\alpha) : \alpha \xrightarrow{a} \beta.$$

8.1.3. On a une *transformation entre dendrons :*

$$\alpha \Rightarrow \beta \Leftrightarrow \begin{cases} \exists\, \alpha_0, \alpha_1, \ldots, \alpha_p ; \\ (i)\ \alpha_0 = \alpha \\ (ii)\ \forall\, i \in]\, p\,] : \alpha_{i-1} \to \alpha_i \\ (iii)\ \alpha_p = \beta. \end{cases}$$

Une GTS transforme un dendron en un dendron par remplacement d'un dendron qui « occurre sous un nœud ».

8.2. *Un bilangage $\mathscr{R}$ est simple* ou régulier si et seulement si, il existe une GTS $= \prec \mathscr{V}, \mathscr{B}, \mathscr{T} >$ telle que:

$$\mathscr{R} = \text{LL (GTS)} = \{ \beta \in \mathscr{V}; \exists\, \alpha_0 \in \mathscr{B}: \alpha_0 \Rightarrow \beta \}.$$

8.3 Exemple

Soit le vocabulaire V = { S_1, SN, Noémon, ... }; on donne le dendron α sous la représentation linéaire (*cf.* proposition 1) (on introduit diverses parenthèses pour une meilleure compréhension). Voici trois transformations simples:

T_1: P] Noémon [→ P] Marie [
T_2: SV] est aimé de SN [→ SV] aime SN [
T_3: SN] Marie [→ SN] Noémon [.

Si:

$$\text{GTS} = < \text{V}, \{ \alpha \}, \{ T_1, T_2, T_3 \} >,$$

alors:

P { Marie SV] aime SN (Noémon) [} ∈ LL (GTS).

8.4. Le lecteur remarquera que dans l'exemple précédent, les transformations T_1 ($i = 1, 2, 3$) n'ont pas d'interprétation linguistique. La transformation T_2 est non-C.S. puisque le mot **i** (T_2) construit sur V ∪ {], [} est *plus long* que le mot **t** (T_2), ce qui est contraire à la définition (*cf. supra* 1.2.3.). Les transformations simples entre dendrons n'ont *a priori* aucune restriction, et on pourrait penser que les GTS feront sortir de la classe $\mathscr{B}$. Or, le théorème auquel nous amènent les résultats obtenus par W. S. Brainerd [4], C. Pair et A. Quère [14] et J. W. Thatcher [15], répond par la négative à notre espoir: sortir de la classe des langages C.F. en employant une grammaire transformationnelle simple.

8.4.1. *Proposition 3.* Soit une GTS $= < \text{V}, \mathscr{B}, \mathscr{T} >$ quelconque; l'ensemble des mots, construits sur V ∪ {], [} associés canoniquement aux dendrons sur V de LL (GTS) peut être produit par une C.F.-grammaire.

8.4.2. *Proposition 4.* Soit $\mathscr{R}$ un bilangage simple; le langage L ($\mathscr{R}$) = { $\hat{m} \in \text{V}^*; \exists\, \alpha \in \mathscr{R}: \hat{m} = TER(\alpha)$ } est un langage qui peut être produit par une C.F.-grammaire.

8.4.3. *Proposition 5.* Soit $\mathscr{L}$ un langage produit par une C.F.-grammaire quelconque; il existe un bilangage simple $\mathscr{R}$ tel que L ($\mathscr{R}$) = $\mathscr{L}$.

8.4.4. *Proposition 6.* Si $\mathscr{B}$ la base d'une GTS est un bilangage simple, alors LL (GTS) est aussi un bilangage simple.

Nous obtenons ainsi le:

8.4.5. *Théorème 1.* Les grammaires transformationnelles simples ne font pas sortir de la classe des C.G.-grammaires.

8.5. Revenons à la proposition 3; bien que les règles semblent de type quelconque, en particulier non-C.S., nous leur imposons cependant une condition restrictive: les deux membres de la règle sont des dendrons. Le théorème 1 se comprend intuitivement en remarquant qu'effectuer une transformation simple revient à appliquer successivement toute une série de règles de type C.F. au nœud *a* étiqueté *l* puisque l'on substitue le dendron **t** (T) au dendron î (T); on remplace ainsi tout un ensemble de règles C.F. par un autre ensemble de règles toujours C.F. et il est normal que l'on ne sorte pas de la classe C.F. Ce type de grammaires transformationnelles simples est inadéquat pour traiter toutes les transformations linguis-

tiques. Par le théorème 1, on sait en effet que l'on ne peut pas énumérer certaines phrases produites uniquement à l'aide de grammaires non-C.F.; ainsi, le langage L_{nnn} ne pourra jamais être produit à partir d'une composante de base qui serait C.F. et transformée ensuite par une certaine GTS. Il est nécessaire de trouver des grammaires transformationnelles qui soient non seulement *effectives*, au sens de calculables par une machine de Turing, mais encore *opérationnelles*, c'est-à-dire susceptibles de « passer» en machine réelle (*cf.* [2]).

9. GRAMMAIRES TRANSFORMATIONNELLES AVEC PILE

Nous avons remarqué qu'une transformation linguistique chez N. Chomsky ne porte pas sur toute l'histoire de la production, mais sur une partie seulement:

(*e.g.*: P] SN SV (SN V) [ou: P] SN SV] P] SN SV [[[ ).

Pour utiliser la partie « utile», nous avons imaginé le concept de *souche* qui, outre l'interprétation linguistique qu'il pourra avoir, rendra économique, donc plus opérationnelle notre grammaire (*cf. infra* 10.2).

9.1. Définition 1

Une *souche* est la donnée d'un couple d'applications (x, σ), où x est une *injection* de D (σ) dans $\mathbf{N}^*$ (entiers), telles que le diagramme suivant commute:

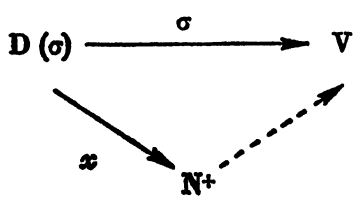

Fig. 3

La souche est un dendron où l'on porte une information supplémentaire: un numéro qui va permettre de conserver la mémoire, après transformation, de l'origine d'un nœud étiqueté (*cf. infra* 9.4.).

Remarque: Pour éviter quelques difficultés dans les définitions suivantes, on supposera que D (σ) a un cardinal strictement supérieur à un; sinon il faudrait prendre des précautions supplémentaires.

9.2. Définition 2

Un dendron α a une souche (x, σ), ou (x, σ) est une souche de α, si et seulement si:

(S_1) D $(\sigma) \subset$ D (α)

(S_2) [Restriction à D (σ)] $(\alpha) = \sigma$

(S_3) $\forall\, i \in \mathbf{N}^*, \forall\, j \in \mathbf{N}^+, \forall\, \hat{m} \in$ D (σ):

$$[\hat{m} \,.\, i \in \text{D}\,(\sigma)] \wedge [\hat{m} \,.\, j \notin \text{D}\,(\sigma)] \wedge [j > i] \Rightarrow [\hat{m} \,.\, j \notin \text{D}\,(\alpha)]$$

9.3. DÉFINITION 3

La transformation T_S entre souches (x, σ) et (x', σ') *induit la transformation* T entre dendrons α et α' si et seulement si:

(T_1) α a une souche (x, σ), α' a une souche (x', σ'),

(T_2) pour tout i de $\mathbf{N}^+$ qui occurre à la fois dans $TER(x)$ et $TER(x')$ avec $x(a) = i$ et $x'(a') = i$, on a: $\alpha' / a' = \alpha / a$.

Ainsi, le sous-dendron α/a « accroché» au nœud a, étiqueté i par l'application x, est transporté par T au nœud a', étiqueté i par x'. Nous avons introduit l'application x pour traiter des transformations du type suivant:

9.4. $P\{SN\ (a)\ SV\ ]\ M\ V\ (b)\ SN\ (c)\ [\} \xrightarrow{T}$

$$P\ \{\ SN\ (c)\ SV\ ]\ M'\ V\ (b)\ d\ SN\ (a)\ [\}$$

induite par la transformation entre souches:

$$\begin{pmatrix} P\ \{\ SN\ SV\ ]\ M\ V\ SN\ [\} \\ 1\ \{\ \ 2\ \ 3\ ]\ 4\ 5\ \ 6\ [\} \end{pmatrix} \xrightarrow{T_S} \begin{pmatrix} P\ \{\ SN\ SV\ ]\ M'\ V\ d\ SN\ [\} \\ 1\ \{\ 6\ \ \ 3\ ]\ 4\ \ 5\ 7\ \ 2\ \ [\} \end{pmatrix}$$

9.5. *Une grammaire transformationnelle avec souches* $< V, \mathscr{B}, \mathscr{S}, \mathscr{T}_S >$ sera bien définie si on se donne un ensemble $\mathscr{S}$ restreint de souches, en général fini, et un ensemble *fini* $\mathscr{T}_S$ de transformations entre souches; le couple $(\mathscr{S}, \mathscr{T}_S)$, même fini peut cependant induire un nombre potentiellement infini de transformations entre dendrons et ainsi formaliser les relations susceptibles d'exister entre deux séquences munies de leurs histoires respectives.

9.6. DÉFINITION 4

Une pile de dendrons sur V est un ensemble Π de mots π construits sur la forêt $\mathscr{V}$ considérée comme vocabulaire, à l'aide de la « concaténation à droite», notée $\circ$, définie par:

(i) $\Lambda \in \Pi$

(ii) $\pi \in \Pi, \alpha \in \mathscr{V} \Rightarrow \pi \circ \alpha \in \Pi$

(iii) $\pi \circ \alpha \in \Pi, \alpha \in \mathscr{V} \Rightarrow \pi \in \Pi$.

Si $\pi = \alpha_1 \circ \alpha_2 \circ \ldots \circ \alpha_n$[1], le « dessus de pile» $d(\pi) = \alpha_n$ et le « dessous de pile» $g(\pi) = \alpha_1 \circ \ldots \circ \alpha_{n-1}$ sont bien définis.

9.7. Soit A une partie de V et $\dot{A}$ un sous-ensemble disjoint de V en *bijection* avec A $(\forall\ l \in A: l \mapsto \dot{l} \in \dot{A})$; notons $\dot{\mathscr{V}}$ la forêt des *dendrons pointés* sur $\dot{V} = V \cup \dot{A}$. Si $\mathscr{S}$ désigne un ensemble de souches, $P_{\mathscr{S}}$ est un ensemble d'applications ou *pointages* P_α:

$$P_\alpha: \dot{\alpha} \to \alpha: (a, l) \mapsto \begin{cases} (a, \dot{l}) \text{ si } \exists\ (x, \sigma) \in \mathscr{S}: \alpha / a \text{ a pour souche } (x, \sigma), \\ (a, l), \text{ sinon.} \end{cases}$$

On introduit de plus l'application de *dépointage:*

$$\overline{P}: \dot{\mathscr{V}} \to \mathscr{V}: \beta \mapsto \begin{cases} \alpha \text{ si } \beta = \dot{\alpha} \\ \beta, \text{ sinon.} \end{cases}$$

1. $\forall\ i \in]\ n\ ]$ $\qquad \alpha_i \in \mathscr{V}$.

9.8. Soit $\dot{\Pi}$ une pile de dendrons sur $\dot{V}$; on se donne deux opérations:

9.8.1. *L'élagage (ELA)*: c'est une opération définie sur $\dot{\Pi} \times \dot{\mathscr{V}}$ par l'application:

$$ELA: \dot{\Pi} \times \dot{\mathscr{V}} \to \dot{\Pi} \times \dot{\mathscr{V}}$$
$$(\dot{\pi}, \dot{\alpha}^a) \mapsto (\dot{\pi} \circ \{ \alpha - \dot{\alpha} / a \}^a, \alpha/a)$$

où $\dot{\alpha}^a$ désigne un dendron pointé au nœud a, *i.e.*, ayant un nœud a étiqueté par une lettre de $\dot{A}$; $\{ \alpha - \dot{\alpha} / a \}^a$ désigne le dendron α où l'on a effacé le dendron α / a sous le nœud a étiqueté par une lettre pointée $\dot{l}$: c'est le complémentaire du dendron α/a par rapport à α, complémentaire dont un des terminaux est nécessairement étiqueté par l.

9.8.2. *L'écussonnage (ECU)*: c'est une relation entre $\dot{\Pi} \times \dot{\mathscr{V}}$ et $\dot{\Pi} \times \dot{\mathscr{V}}$ définie par l'application:

$$ECU: \dot{\Pi} \times \dot{\mathscr{V}} \to \dot{\Pi} \times \mathscr{V}$$

$$(\dot{\pi}, \beta) \to \begin{cases} (g(\dot{\pi}), \overline{P}(\alpha)) \text{ si } \textit{condition Ecu}, \text{ avec } \alpha \text{ défini ci-dessous.} \\ (\dot{\pi}, \beta), \text{ sinon.} \end{cases}$$

Condition Ecu:

$$\exists\, l \in A: \{ (l \in RAC(\beta)) \vee (\dot{l} \in RAC(\beta)) \} \wedge \{ \dot{l} \in TER(d(\dot{\pi})) \}$$

si:

$$(a_1, \dot{l}), \ldots, (a_n, \dot{l}) \in d(\dot{\pi}) \qquad \text{avec} \qquad \dot{l} \in TER(d(\dot{\pi})).$$

Alors on a:

$$d(\dot{\pi}) \to d(\dot{\pi})\,[a_1 \leftarrow \beta] \to \ldots \beta_i\,[a_i \leftarrow \beta] \to \ldots \beta_n\,[a_n \leftarrow \beta] = \alpha\,^1.$$

On a par conséquent défini $\alpha(d(\dot{\pi}) \Rightarrow \alpha)$ en « écussonnant» β sur chacun des terminaux $(a_i, \dot{l})$ de $d(\dot{\pi})$ à condition que la racine de β porte la même étiquette, à un pointage près, que ces terminaux.

Par élagage, il est alors possible de dégager une souche de $\mathscr{S}$; on stocke sur la pile $\dot{\Pi}$ le dendron $\{ \alpha - \dot{\alpha}/a \}^a$ en conservant, grâce au pointage, la mémoire du nœud terminal d'où l'on a élagué le dendron α/a qui, nécessairement, a une souche dans $\mathscr{S}$. L'écussonnage permet ensuite de raccrocher le dendron α/a au nœud terminal d'où l'on était parti.

9.9. On définit maintenant une grammaire transformationnelle avec pile GTP par la donnée du 12-uple:

$$< (V, A, V_a), (\mathscr{B}, \mathscr{J}), (\mathscr{S}, \mathscr{T}_S); \circ, P_{\mathscr{S}}, \overline{P}, ELA, ECU) >$$

où:

L'opération « concaténation à droite » sur $\dot{\mathscr{V}}$, les applications $P_{\mathscr{S}}$, $\overline{P}$, *ELA*, *ECU* ont été définies ci-dessus. Appelons Ω l'ensemble de ces opérations et applications. La « concaténation à droite» permet de construire une pile $\dot{\Pi}$ de dendrons sur $\dot{V} = V \cup \dot{A}$; grâce à $P_{\mathscr{S}}$, on pointera des racines de souches dégagées ensuite par élagage; l'écussonnage accrochera, après transformations éventuelles, les dendrons aux terminaux des dessus de pile, terminaux dont on a gardé la mémoire par le pointage.

---

1. On peut aussi écrire :

$$(\ldots (d(\dot{\pi})\,[a_1 \leftarrow \beta])\,[a_2 \leftarrow \beta] \ldots)\,[a_n \leftarrow \beta] = \alpha.$$

V_a (vocabulaire d'arrêt) $\subset V$; $A \subset V$; V définit la forêt $\mathscr{V}$.

$\mathscr{B}$ est la *base* de la grammaire: c'est un bilangage contenu dans $\mathscr{V}$.

$\mathscr{J}$ est une application de $\mathscr{B}$ dans $\dot{\Pi}$ qui détermine des états initiaux de pile:

$$\forall \alpha \in \mathscr{B}: \alpha \mapsto \mathscr{J}(\alpha) \in \dot{\Pi}.$$

$\mathscr{S}$ est un ensemble de souches, en général fini; $\mathscr{T}_S$ est un ensemble *fini* de transformations entre souches de $\mathscr{S}$, qui induisent des transformations entre dendrons de $\mathscr{V}$.

Si l'on connaît un état de pile $\mathscr{J}(\alpha)$, associé à un dendron α de $\mathscr{B}$, on cherche à transformer α par les transformations induites de $\mathscr{T}_S$, en utilisant tour à tour pointages et pile. Les calculs s'arrêtent orsque:

— soit aucune transformation n'est applicable: il y a alors bloquage;

— soit le dendron obtenu, après transformations, est entièrement étiqueté sur V_a et de plus, le contenu de la pile est réduit au mot vide Λ.

9.10. Soit $\mathscr{T}$ l'ensemble, éventuellement infini, des transformations induites par l'ensemble fini $\mathscr{T}_S$; on note $\overline{\mathscr{T}}$ la relation sur $\dot{\Pi} \times \dot{\mathscr{V}}$, associée à $\mathscr{T}$ et Ω:

$$\overline{T} \in \overline{\mathscr{T}}: (\dot{\pi}, \dot{\alpha}) \overset{\overline{T}}{\to} (\dot{\pi}l, \dot{\alpha}l) \Leftrightarrow \begin{cases} \exists\, T \in \mathscr{T}: \dot{\alpha} \overset{T}{\to} \dot{\alpha}' \quad \text{et} \quad \dot{\pi}' = \dot{\pi}, \\ \text{ou:} \\ \exists\, w \in \Omega: (\dot{\pi}, \dot{\alpha}) \mapsto (\dot{\pi}l, \dot{\alpha}l). \end{cases}$$

Si $\mathscr{V}_a$ est la forêt étiquetée sur V_a, alors:

$$LL(GTP) = \{ \alpha \in \mathscr{V}_a; \exists\, (\mathscr{J}(\alpha_0), \alpha_0) \in \dot{\Pi} \times \mathscr{B}: (\mathscr{J}(\alpha_0), \alpha_0) \overset{\overline{\mathscr{T}}}{\Rightarrow} (V, \alpha) \}.$$

9.11.1. *Proposition 7.* La grammaire G′ induite par une GTP à partir d'une K-grammaire peut appartenir à une clause plus élevée dans la hiérarchie des grammaires.

On peut transformer le langage de Kleene $L_n = \{ a^n b; n \geqslant 1 \}$ en $L_{nn} = \{ a^n b^n; n \geqslant 1 \}$ qui ne peut être produit par aucune K-grammaire.

9.11.2. *Proposition 8.* La grammaire G′ induite par une GTP à partir d'une C.F.-grammaire peut appartenir à une classe plus élevée dans la hiérarchie des grammaires.

Au dendron α_{nn} histoire du mot $a^n b^n$ de L_{nn}, on associe par une GTP un dendron α_{nnn}; seuls ces dendrons α_{nnn} appartiennent au bilangage produit par la GTP de base L_{nn} et:

$$L_{nnn} = \{ a^n b^n c^n = TER(\alpha_{nnn}); \alpha_{nnn} \in LL(GTP) \}.$$

9.11.3. *Théorème 2.* Les grammaires transformationnelles à pile augmentent la puissance en faisant monter dans la hiérarchie des grammaires à partir des classes de Kleene et « Context-free ».

10. GÉNÉRALISATION DES GRAMMAIRES TRANSFORMATIONNELLES AVEC PILE

10.1. Pour formaliser les transformations linguistiques, nous avions comme objectif (*cf.* 8.6. *supra*) de calculer non pas sur des dendrons mais sur des polydendrons. On peut généraliser les grammaires trans-

formationnelles avec pile en introduisant des souches de polydendrons et transformations entre souches de polydendrons qui induiront des transformations entre polydendrons (*cf.* [1]).

10.2. On peut introduire certaines restrictions supplémentaires en se donnant des contraintes sur l'ordre d'application des transformations; on introduit pour cela un automate fini qui contrôle le déroulement du calcul (voir Ginsburg-Partee [11]). Moyennant quelques complications mineures dans les données, notre démarche semble être une approche d'une formalisation adéquate aux grammaires transformationnelles « naïves» de N. Chomsky [5], [6] (et peut-être de Z. S. Harris [12]), bien qu'elle soit moins fidèle à la lettre qu'une formalisation comme celle de Ginsburg-Partee [11]. Ces derniers ne font appel qu'aux concepts directement suggérés par le modèle et l'on pourrait penser *a priori* que notre formalisation est moins économique et plus abstraite puisque nous avons introduit des concepts nouveaux comme la souche, les transformations entre souches qui induisent des transformations entre polydendrons. Il se pose en effet le problème d'adéquation du système formel de la théorie naïve. Ce sont « des bonnes raisons» issues de l'expérimentation du modèle transformationnel (*cf. supra* 9) qui nous ont fait introduire ces nouveaux concepts qui ne se trouvent pas explicitement chez N. Chomsky. Les concepts de souches et transformations entre souches ont-ils une interprétation linguistique ? Et doit-on les considérer comme des opérateurs linguistiques ? Si oui, il s'agit alors de caractériser à la suite d'analyses linguistiques, un ensemble de souches $\mathscr{S}$, un ensemble de transformations $\mathscr{T}_S$, puis de procéder à une réduction de ces ensembles en vue d'optimiser le système en prenant pour critère [1] la commodité des algorithmes de production et de reconnaissance. Il est évident que les critères sont hiérarchisés; les critères linguistiques ont la préséance sur les critères de commodité; cependant, après réduction, on peut s'interroger sur la signification linguistique des ensembles réduits et envisager d'examiner si ces ensembles réduits n'appartiennent pas à un niveau plus profond et plus abstrait que l'ensemble déterminé par la première analyse linguistique. C'est par un balancement dialectique entre différentes exigences que l'on déterminera les « bons» ensembles. C'est justement à cause de la plus grande abstraction des grammaires transformationnelles à pile que nous pensons qu'elles contiennent une partie isomorphe aux grammaires transformationnelles formelles de Ginsburg-Partee et les dépassent, tout en étant peut-être plus opérationnelles.

Nous devions nous assurer que les grammaires transformationnelles avec pile nous sortaient de la classe C.F. (*cf.* théorème 2), ce que les grammaires transformationnelles simples ne permettraient pas (*cf.* théorème 1); ces dernières étaient un système formel inadéquat pour toute théorie naïve des langues naturelles mais fournissaient un résultat négatif intéressant:

(C) *les transformations simples ne sont pas suffisantes pour caractériser à partir d'une base C.F. toutes les phrases d'une langue naturelle.*

Il nous fallait par conséquent compliquer la nature des transformations; la pile a introduit cette complication en opérant non plus de simples substitutions aux nœuds des dendrons mais en opérant localement à l'intérieur d'un dendron grâce au couple de concepts (pile, souche).

10.3. Plusieurs problèmes intéressants et ouverts se posent.

10.3.1. Interpréter d'une manière satisfaisante la souche et examiner sa pertinence linguistique (*cf.* discussion en 10.2).

10.3.2. Lors d'un calcul transformationnel, peut-on dire que la pile ait une interprétation linguistique?

---

1. Pour réduire, voire augmenter les ensembles $\mathscr{S}$ et $\mathscr{T}_S$, d'autres critères devront intervenir : par exemple, le critère fourni par la génétique... C'est un faisceau de critères qui amènent à choisir les « bons» ensembles.

10.3.3. Peut-on décrire toutes les phrases d'une langue donnée avec un nombre raisonnable de souches et transformations entre souches ? Y a-t-il des relations entre phrases dont les GTP ne rendront pas compte ?

10.3.4. Guinsburg-Partee annoncent, dans une remarque, que le modèle formel qu'ils proposent [11] engendre tous les ensembles récursivement énumérables. Il serait intéressant de savoir si les grammaires transformationnelles avec pile produisent les mêmes ensembles; il faudrait de plus caractériser des restrictions sur la nature des transformations pour énoncer des résultats soit positifs, soit négatifs (ce qui est beaucoup plus facile) du genre: *les restrictions du type... sont nécessaires pour caractériser une langue* ou *les restrictions du type... ne sont pas suffisantes pour caractériser une langue.*

10.4. Devant des problèmes linguistiques comme les anaphoriques, les diverses relatives (restrictives ou qualificatives...) certains enchaînements d'énoncés dans le discours, certains problèmes de subordination, nous songeons à utiliser des grammaires transformationnelles « plus puissantes » que les grammaires transformationnelles à pile qui y seront contenues. Pour cela, on pense travailler par une approche catégorielle qui s'inspire des travaux d'Eilenberg et Wright [9] sur des structures plus complexes que le dendron et la pile: la treille et la cascade (*cf.* [1], [2]).

BIBLIOGRAPHIE

[1] Barbault, M. C. et Desclés, J. P., "*Approches structurelle et catégorielle du système de transition avec application à la science du calcul et à la linguistique mathématique*", thèses 3e cycle, Faculté des Sciences, Paris, 1970. [2] "Vers un traitement automatique des textes", à paraître 1972, *Revue de linguistique appliquée.*

[3] Berge, C., *Théorie des graphes et ses applications*, Paris, Dunod, 1958.

[4] Brainerd, W. S., "Tree generating regular systems", *Information and control*, 14, 1969, pp. 217-231.

[5] Chomsky, N., *Syntactic structures*, La Haye, Mouton, 1957 (traduit en français : *Structures syntaxiques*, Paris, Seuil, 1969).

[6] — *Aspects of the theory of syntax*, Cambridge, Mass., MIT Press, 1965.

[7] "Formal properties of grammars", *Handbook of mathematical psychology*, D. Luce, E. Bush. E. Galanter (eds.), New York, John Wiley, 1963.

[8] Chomsky, N. et Miller, G. *L'analyse formelle des langues naturelles*, Paris, Gauthier-Villars/ Mouton, 1968.

[9] Eilenberg, S. et Wright, J. B., "Automata in general algebras", *Information and Control*, 11, 1967, pp. 452-470.

[10] Friant, J., "Les langages 'Context-Sensitive' ", *Ann. Inst. Henri Poincaré*, Vol. III, no 1, 1967, pp. 35-180.

[11] Ginsburg, S. et Partee, B., "A mathematical model of transformational grammars", *Information and Control*, 15, 1969, pp. 262-334.

[12] Harris, Z. S., "Mathematical structure of language", *Inter-science*, New York, John Wiley, 1968.

[13] Gross, M. et Lentin, A., *Notions sur les grammaires formelles*, Paris, Gauthier-Villars, 1970.

[14] Pair, C. et Quère, A., "Définition et étude des bilangages réguliers", *Information and Control*, 13, 1968, pp. 565-593.

[15] Tesnière, L., *Éléments de syntaxe structurale*, Paris, Klincksieck, 1966.

[16] Thatcher, J.W., "Characterising derivation trees of Context-Free grammars through a generalization of finite automata theory", *Journal of Computer and System Sciences*, 1, 1967, pp. 317-322.

[17] Veillon, G., Veruynes et Vauquois, B., "Un métalangage de grammaires transformationnelles", *document CETA*, CNRS, Grenoble, janvier 1967.