ANNALES DE L'I. H. P., SECTION B

JEAN FRIANT

Les langages « Context-Sensitive »

Annales de l'I. H. P., section B, tome 3, n° 1 (1967), p. 35-120

<http://www.numdam.org/item?id=AIHPB_1967__3_1_35_0>

Ann. Inst. Henri Poincaré, Section B :
Vol. III, n° 1, 1967, p. 35-120. *Calcul des Probabilités et Statistique.*

Les langages « Context-Sensitive »

par

Jean FRIANT

AVANT-PROPOS

Cette étude comporte, les définitions de base étant posées, un exposé en forme des principaux résultats connus sur les langages C S (pour « Context-Sensitive » en anglais). Plusieurs résultats importants sur les langages C F (pour « Context-Free ») se trouvent du même coup démontrés. Certaines des démonstrations développées ici sont esquissées dans les ouvrages de S. Y. Kuroda [5] et de P. S. Landweber [6] (cf. bibliographie) qui traitent aussi de ces langages. Les exemples de langages C S décrits permettent d'entrevoir l'utilisation de ce formalisme en théorie des nombres. On termine par l'énoncé d'un résultat (sa démonstration, qui aurait exigé de plus amples développements, a été omise) qui permet de placer dans le cadre des langages C S les langages C F P (pour Context-Free à Peignes) qui constituent une extension importante des langages C F.

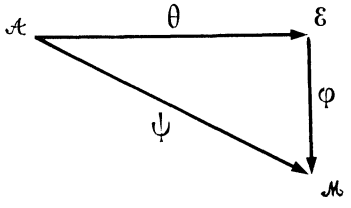
Qu'il me soit permis d'exprimer ici toute ma gratitude à M. le Professeur J. P. Benzécri pour avoir bien voulu diriger cette thèse et m'aider de ses nombreuses suggestions, à MM. les Professeurs D. Dugué et P. Samuel qui ont accepté d'examiner ce travail. Je remercie aussi M. J. Roubaud, qui a dirigé mes premiers pas en linguistique mathématique, ainsi que tous ceux qui m'ont aidé par la lecture et la polycopie du manuscrit.

0. INTRODUCTION. RAPPELS. NOTATIONS

Un texte ou un discours d'une langue naturelle, un message codé, un programme destiné à un ordinateur semblent pouvoir toujours, et parfois de plusieurs façons, être décrits comme une suite d'éléments (lettres, sons, chiffres, instructions) appartenant à un alphabet (ou vocabulaire) fini. Algébriquement on parle de monoïde libre.

0.1. Monoïde libre.

Étant donné un ensemble $\mathcal{A}$ (fini ou non), on appelle monoïde libre sur l'ensemble $\mathcal{A}$, tout monoïde $\mathcal{E}$, muni d'une application θ de $\mathcal{A}$ dans $\mathcal{E}$, tels que pour tout monoïde $\mathcal{M}$ et toute application ψ de $\mathcal{A}$ dans $\mathcal{M}$ il existe un unique homomorphisme de monoïdes φ de $\mathcal{E}$ dans $\mathcal{M}$ rendant commutatif le diagramme suivant :



i. e. : $\psi = \varphi \circ \theta$.

Soit $\mathfrak{L}(\mathcal{A})$ l'ensemble de toutes les suites finies d'éléments, distincts ou non, de $\mathcal{A}$. Nous noterons d'une capitale latine, habituellement affectée d'indice, un élément de $\mathcal{A}$, appelé symbole ou lettre et d'une capitale latine surmontée d'un accent circonflexe un élément de $\mathfrak{L}(\mathcal{A})$, appelé mot. n étant un entier naturel, $n \in \mathbf{N}$, (n) désignera l'ensemble des entiers naturels inférieurs ou égaux à n, et $)n)$ ce même ensemble privé de l'élément 0, i. e. :

$$(n) = \{0, 1, \ldots, n\} = \{i \mid i \in \mathbf{N} \ ; 0 \leqslant i \leqslant n\}$$
$$)n) = \{1, 2, \ldots, n\} = \{i \mid i \in \mathbf{N}^*; 1 \leqslant i \leqslant n\}$$

Il est facile de prouver qu'une solution du problème universel précédent est fournie par l'ensemble $\mathfrak{L}(\mathcal{A})$ muni de l'opération *produit de juxtaposition* qui au couple ordonné de mots $(\widehat{A}, \widehat{B})$ associe le mot $\widehat{C}$ obtenu en écrivant le mot $\widehat{B}$ à droite du mot $\widehat{A}$; formellement si :

$$\widehat{A} = A_1 \ldots A_i \ldots A_n \qquad \forall i \in)n) \ : A_i \in \mathcal{A}$$
$$\widehat{B} = B_1 \ldots B_j \ldots B_m \qquad \forall j \in)m) \ : B_j \in \mathcal{A}$$

alors

$$\widehat{C} = \widehat{A}\widehat{B} = A_1 \ldots A_n B_1 \ldots B_m$$
$$= C_1 \ldots C_k \ldots C_{n+m} \qquad \forall k \in)n) \ : C_k = A_k$$
$$\forall k, n < k \leqslant n + m \ : C_k = B_{k-n}$$

Cette opération est associative mais non commutative. Elle a pour élément neutre, l'unique suite vide, ou mot à zéro lettre, noté $\widehat{\emptyset}$. $\mathfrak{L}(\mathcal{A})$ a donc bien une structure de monoïde.

La longueur d'un mot $\widehat{X}$, notée $|\widehat{X}|$, désignera le nombre de lettres, distinctes ou non, du mot. Ainsi,

$$|\widehat{A}| = n \quad , \quad |\widehat{B}| = m, \qquad |\widehat{C}| = |\widehat{A}\widehat{B}| = n + m ;$$

Ceci est général :

$$\forall \widehat{X}, \widehat{Y} \in \mathfrak{L}(\mathcal{A}) : \qquad |\widehat{X}\widehat{Y}| = |\widehat{X}| + |\widehat{Y}|$$

Le mot vide $\widehat{\emptyset}$ est caractérisé par $|\widehat{\emptyset}| = 0$.

Si $\widehat{X} = \underbrace{A \ \ldots \ A_i \ \ldots \ A_i}_{n \text{ fois}}$, $A_i \in \mathcal{A}$, on écrira plus simplement

$$\widehat{X} = (A_i)^n \qquad \text{et on a évidemment} \qquad |\widehat{X}| = n.$$

L'application θ associée à $\mathfrak{L}(\mathcal{A})$ est celle qui fait correspondre à toute lettre A_i de $\mathcal{A}$ le mot de $\mathfrak{L}(\mathcal{A})$ réduit à cette lettre. Cette application étant évidemment injective, $\mathcal{A}$ peut être identifié à un sous-ensemble de $\mathfrak{L}(\mathcal{A})$, le sous-ensemble des monogrammes ou mots à une seule lettre.

Par abus de langage, $\mathfrak{L}(\mathcal{A})$ s'appellera : le monoïde libre construit sur $\mathcal{A}$. En fait on peut prouver que toutes les solutions du problème universel précédent sont isomorphes entre elles.

0.2. Homomorphisme de monoïdes libres.

Soit le monoïde libre $\mathfrak{L}(\mathcal{A})$ (resp. $\mathfrak{L}(\mathcal{B})$) construit sur l'ensemble de base $\mathcal{A}$ (resp. $\mathcal{B}$). Une application φ de $\mathfrak{L}(\mathcal{A})$ dans $\mathfrak{L}(\mathcal{B})$ est appelée homomorphisme de monoïdes si elle est compatible avec le produit de juxtaposition, i. e. :

$$\forall \widehat{X}, \widehat{Y} \in \mathfrak{L}(A) : \qquad \varphi(\widehat{X}\widehat{Y}) = \varphi(\widehat{X})\varphi(\widehat{Y})$$

L'ensemble $\mathcal{A}$ étant un système de générateurs de $\mathfrak{L}(\mathcal{A})$ l'application φ est parfaitement déterminée par sa restriction à cet ensemble. En effet si :

$$\widehat{A} = A_1 \ \ldots \ A_i \ \ldots \ A_n \qquad , \qquad \forall i \in)n) : A_i \in \mathcal{A}$$

Alors :

$$\varphi(\widehat{A}) = \varphi(A_1) \ \ldots \ \varphi(A_i) \ \ldots \ \varphi(A_n), \qquad \forall i \in)n) : \varphi(A_i) \in \mathfrak{L}(\mathcal{B}).$$

D'autre part, il est facile de prouver que les éléments neutres se correspondent dans un tel homomorphisme :

$$\varphi(\widehat{\emptyset}_{\mathcal{A}}) = \varphi(\widehat{\emptyset}_{\mathcal{B}})$$

Par la suite, le mot vide sera simplement noté $\widehat{\emptyset}$, quel que soit le monoïde libre envisagé, si aucune confusion n'est à craindre.

1. DÉFINITIONS DE BASE

1.1. Point de départ : la linguistique structurale.

A. Martinet [8] définit l'organisation des langues naturelles en une double articulation, distinguant deux plans qualitativement différents : une première articulation constituée d'unités significatives minimales (monèmes) et une deuxième articulation où ces premières unités s'articulent elles-mêmes en une succession de phonèmes (sons) dont le nombre est limité. Après A. Martinet, J. P. Benzécri [1] introduit en linguistique mathématique une distinction semblable en considérant d'une part le niveau de la composition et d'autre part celui de l'expression. Nous reviendrons, plus en détail, sur cette distinction intéressante, au paragraphe 9...

Une grammaire, au sens ordinaire, peut être définie comme une analyse des unités de première articulation. Elle permet notamment de reconnaître les phrases syntaxiquement correctes, grâce à des règles d'assemblage du type : « une phrase peut être formée d'un syntagme nominal suivi d'un syntagme verbal... ». Les grammaires de constituants immédiats de Chomsky ont pour but de formaliser ce point de vue de la composition. Bien qu'aucune langue naturelle n'ait pu, jusqu'à ce jour, être décrite adéquatement par quelque formalisme algébrique que ce soit, la théorie des grammaires formelles mérite d'être étudiée, d'une part les problèmes qu'elle pose présentent un intérêt mathématique certain, d'autre part, elle aide à la description de langages artificiels, notamment les langages de programmation.

1.2. Grammaire de constituants.

Les grammaires de N. Chomsky décrivent la construction des phrases de langages artificiels, par modifications successives portant sur des suites de symboles, c'est-à-dire sur les mots d'un certain monoïde libre. Selon les contraintes imposées à ces modifications, la grammaire sera plus ou moins

puissante; elle permettra de construire des phrases de structure plus ou moins complexe...

D'une façon plus précise, une grammaire de constituants (« phrase structure grammar » en anglais) est la donnée d'un quadruplet :

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

où : 1° $\mathfrak{V}$, appelé *vocabulaire* (ou alphabet), est un ensemble fini de symboles (ou lettres);

2° $\mathfrak{V}_T$ est un sous-ensemble de $\mathfrak{V}$, appelé *vocabulaire terminal.* Si l'on s'intéresse aux langues naturelles, $\mathfrak{V}_T$ pourra être l'ensemble des unités de première articulation appelées, selon les terminologies, monèmes ou morphèmes (en linguistique se pose en fait le difficile problème du choix des unités). On notera $\mathfrak{V}_N$ le complémentaire de $\mathfrak{V}_T$: c'est le *vocabulaire non terminal.* Dans l'exemple choisi un élément de $\mathfrak{V}_N$ désignera une partie syntaxique (nom, verbe, adjectif...) ou d'une façon plus générale un constituant immédiat de la phrase. Un tel symbole n'interviendra pas dans la phrase elle-même mais seulement lors de la description de sa génération. On distinguera donc les symboles terminaux, éléments de $\mathfrak{V}_T$, des symboles non terminaux, éléments de $\mathfrak{V}_N$. Cette distinction s'étendra aux mots de la façon suivante : soit $\widehat{A}$ un mot de $\mathfrak{L}(\mathfrak{V})$:

$$\widehat{A} = A_1 \ldots A_i \ldots A_n$$

$\widehat{A}$ sera dit *terminal* si et seulement si tout symbole A_i est terminal; i. e. si :

$$\widehat{A} \in \mathfrak{L}(\mathfrak{V}_T)$$

Dans le cas contraire (i. e. $\exists i, i \in)n)$: $A_i \in \mathfrak{V}_N$), $\widehat{A}$ sera dit *non terminal.*

3° S est un élément distingué de $\mathfrak{V}_N$, appelé *symbole initial* pour des raisons qui apparaîtront ultérieurement.

4° $\mathfrak{R}$ est un ensemble fini dont les éléments — que nous appellerons *règles de production* ou plus simplement règles — sont des couples ordonnés de mots du monoïde libre $\mathfrak{L}(\mathfrak{V})$, le premier étant toujours un mot sur le vocabulaire non terminal. Ces règles seront désignées par des minuscules latines, éventuellement affectées d'indice. Ainsi :

$$\forall r \in \mathfrak{R} : \qquad r = (\widehat{A}, \widehat{B}) \quad , \quad \widehat{A} \in \mathfrak{L}(\mathfrak{V}_N), \qquad \widehat{B} \in \mathfrak{L}(\mathfrak{V})$$

Si $\widehat{B}$ est un mot terminal (resp. non terminal) on parlera de règle terminale (resp. non terminale).

Dans l'exemple choisi pour illustrer ces notions on distinguera ces deux sortes de règles :

— les règles de syntaxe (au sens étymologique de règles d'assemblage) du type déjà signalé : « une phrase se compose d'un syntagme nominal et d'un syntagme verbal », ou « un syntagme nominal se compose d'un nom et d'un adjectif », etc.,

— et les règles terminales permettant de passer par particularisation d'une catégorie grammaticale à un certain mot appartenant à cette catégorie.

1.3. Principales classes de grammaires.

1.3.1. Grammaires C F.

En imposant différentes contraintes aux règles, N. Chomsky obtient diverses classes de grammaires. La plus connue est celle des grammaires « context-free » (ou simplement C F). Une grammaire C F est caractérisée par le fait que pour toute règle, $r = (\widehat{A}, \widehat{B})$, le premier membre A est réduit à un seul symbole non terminal, $\widehat{A} \in \mathfrak{V}_N$. Ces grammaires permettent notamment la description (incomplète !) de certains langages de programmation. Mais elles ne peuvent certainement pas rendre entièrement compte de la structure des langues naturelles, d'où plusieurs tentatives pour élargir la notion de grammaire context-free, notamment celle de J. P. Benzécri [*1*] [*2*] qui introduit des symboles et des mots à plusieurs insertions. Mais Chomsky lui-même a introduit des grammaires plus générales que les grammaires C F. Ce sont elles qui vont faire l'objet principal de cette étude.

1.3.2. Grammaires de type 1 et de type 2.

Soit la grammaire :

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

— elle est dite de type 1 si l'axiome (α_1) est vérifié :

(α_1) Pour toute règle de production de $\mathfrak{G}$, définie comme un couple ordonné de mots de $\mathfrak{L}(\mathfrak{V})$, la longueur du premier mot est inférieure ou égale à celle du second, i. e. :

$$\forall r \in \mathfrak{R}, \qquad r = (\widehat{A}, \widehat{B}) : \qquad |\widehat{A}| \leqslant |\widehat{B}|$$

— elle est dite de type 2 si elle satisfait à l'axiome (α_2) :

(α_2) Toute règle de production de $\mathfrak{G}$ est telle que le second mot s'obtient à partir du premier en y remplaçant un seul symbole par un mot non vide, le « contexte » étant de plus conservé, formellement :

$$\forall r \in \mathfrak{R}, \quad r = (\widehat{A}, \widehat{B}) : \quad \widehat{A} = \widehat{X} Y_i \widehat{Z} \quad , \quad \widehat{B} = \widehat{X}\widehat{Y}\widehat{Z}$$

$$\widehat{X}, \widehat{Z} \in \mathfrak{L}(\mathfrak{V}_N) \quad ; \quad Y_i \in \mathfrak{V}_N \quad ; \quad \widehat{Y} \in \mathfrak{L}(\mathfrak{V}) \quad \text{et} \quad \widehat{Y} \neq \widehat{\emptyset}$$

Si pour toute règle $\widehat{X} = \widehat{Z} = \widehat{\emptyset}$, nous sommes en présence d'une grammaire C F. Ainsi toute grammaire C F est une grammaire particulière de type 2. Mais l'on voit aussi que toute grammaire de type 2 est une grammaire de type 1, car :

$$|\widehat{A}| = |\widehat{X}| + |\widehat{Z}| + 1$$

$$|\widehat{B}| = |\widehat{X}| + |\widehat{Z}| + |\widehat{Y}|$$

et comme $\widehat{Y} \neq \widehat{\emptyset}$ on a $|\widehat{Y}| \geqslant 1$ et donc $|\widehat{B}| \geqslant |\widehat{A}|$.

1.4. Dérivations.

1.4.1. Dérivation directe.

Étant donné une grammaire $\mathfrak{G}$

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

et deux mots $\widehat{X}$ et $\widehat{Y}$ de $\mathfrak{L}(\mathfrak{V})$ on dira que $\widehat{Y}$ dérive directement de $\widehat{X}$, selon $\mathfrak{G}$, par la règle de production $r = (\widehat{A}, \widehat{B})$, s'il existe deux mots $\widehat{G}$ et $\widehat{D}$ de $\mathfrak{L}(\mathfrak{V})$ tels que :

$$\widehat{X} = \widehat{G}\widehat{A}\widehat{D} \quad , \quad \widehat{Y} = \widehat{G}\widehat{B}\widehat{D}$$

et on écrira :

$$\widehat{X} \xrightarrow[\mathfrak{G}]{\widehat{G}, r, \widehat{D}} \widehat{Y}$$

ou simplement :

$$\widehat{X} \xrightarrow[\mathfrak{G}]{r} \widehat{Y}$$

ou même :

$$\widehat{X} \xrightarrow[\mathfrak{G}]{} \widehat{Y}$$

lorsqu'il n'est pas utile de préciser la règle de production que l'on applique dans sa dérivation.

Une règle de production, $r = (\widehat{A}, \widehat{B})$, s'applique donc à un mot $\widehat{X}$, si celui-ci peut être décomposé en trois facteurs, celui du milieu étant le premier membre, $\widehat{A}$, de la règle envisagée, auquel on substitue le second membre $\widehat{B}$, pour aboutir au mot $\widehat{Y}$. Ainsi pour toute règle $r = (\widehat{A}, \widehat{B})$, $r \in \mathfrak{R}$, on a :

$$A \xrightarrow[\mathfrak{G}]{\widehat{\emptyset}, r, \widehat{\emptyset}} B$$

Au lieu de règle de production, $r = (\widehat{A}, \widehat{B})$, on parle parfois de règle de réécriture notée $\widehat{A} \rightarrow \widehat{B}$. On utilisera occasionnellement une telle notation.

Il est important de remarquer qu'une même règle de production pourra, parfois, s'appliquer à un mot dans des dérivations différentes, la décomposition de ce mot en trois facteurs n'étant pas toujours définie d'une façon univoque. Dans certains cas, le résultat sera pourtant le même. Ainsi, soit la grammaire $\mathfrak{G}$ ayant pour vocabulaire :

$$\mathfrak{V} = \{ S, A \}$$

et pour règles de production :

$$r_1 = (S, A) \qquad \text{et} \qquad r_2 = (A, AA).$$

Il faut distinguer les dérivations directes suivantes :

$$AA \xrightarrow[\mathfrak{G}]{A, r_2, \widehat{\emptyset}} AAA$$

et

$$AA \xrightarrow[\mathfrak{G}]{\widehat{\emptyset}, r_2, A} AAA$$

C'est la présence du contexte, $\widehat{G} - \widehat{D}$, qui permet de faire cette distinction.

1.4.2. **Dérivation.**

Étant donné deux mots $\widehat{X}$ et $\widehat{Y}$ de $\mathfrak{L}(\mathfrak{V})$, on dira que $\widehat{Y}$ dérive de $\widehat{X}$, selon $\mathfrak{G}$, si l'on peut trouver une suite finie de mots commençant par $\widehat{X}$, se terminant par $\widehat{Y}$ et telle que tout mot de la suite (mis à part le premier $\widehat{X}$) dérive directement du précédent. Soit par exemple la suite :

$$\widehat{X}_0, \widehat{X}_1, \ldots, \widehat{X}_i, \ldots, \widehat{X}_n, \widehat{X}_{n+1}$$

telle que :

$$\widehat{X}_0 = \widehat{X} \quad , \quad \widehat{X}_{n+1} = \widehat{Y},$$

et,

$$\forall i \in (n) \quad : \quad \widehat{X}_i = \widehat{G}_i \widehat{A}_i \widehat{D}_i \quad , \quad \widehat{X}_{i+1} = \widehat{G}_i \widehat{B}_i \widehat{D}_i, \qquad r_i = (\widehat{A}_i, \widehat{B}_i) \in \mathfrak{R}$$

c'est-à-dire :

$$\forall i \in (n) \quad : \quad \widehat{X}_i \xrightarrow[\mathfrak{G}]{\widehat{G}_i, r_i, \widehat{D}_i} \widehat{X}_{i+1}$$

$\widehat{r} = r_0 \ldots r_i \ldots r_n$ étant une suite finie de règles de production de $\mathfrak{G}$, c'est-à-dire un mot du monoïde libre $\mathfrak{L}(\mathfrak{R})$, on écrira :

$$\widehat{X} \xRightarrow[\mathfrak{G}]{\widehat{G}_n, \ldots, \widehat{G}_1, \widehat{G}_0, \widehat{r}, \widehat{D}_0, \widehat{D}_1, \ldots, \widehat{D}_n} \widehat{Y}$$

ou simplement :

$$\widehat{X} \xRightarrow[\mathfrak{G}]{\widehat{r}} \widehat{Y}$$

ou même :

$$\widehat{X} \xRightarrow[\mathfrak{G}]{} \widehat{Y}$$

La suite de mots de $\mathfrak{L}(\mathfrak{V})$:

$$(\widehat{X} = \widehat{X}_0, \widehat{X}_1, \ldots, \widehat{X}_i, \ldots, \widehat{X}_n, \widehat{X}_{n+1}, = \widehat{Y})$$

sera appelée une $\widehat{X}$-dérivation de $\widehat{Y}$, selon $\mathfrak{G}$.

Dans le cas particulier où cette suite est réduite à un seul élément, c'est à-dire où $\widehat{Y} = \widehat{X}$, on écrira par convention :

$$\widehat{X} \xRightarrow[\mathfrak{G}]{\widehat{\varnothing}_{\mathfrak{R}}} \widehat{X},$$

$\widehat{\varnothing}_{\mathfrak{R}}$ étant l'élément neutre du monoïde libre $\mathfrak{L}(\mathfrak{R})$.

Nous venons donc de définir sur $\mathfrak{L}(\mathfrak{V})$ une relation binaire :

— réflexive :

$$\forall \widehat{X} \in \mathfrak{L}(\mathfrak{V}) \qquad \widehat{X} \xRightarrow[\mathfrak{G}]{\widehat{\varnothing}_{\mathfrak{R}}} \widehat{X}$$

— transitive, car si :

$$\widehat{X} \underset{\mathcal{G}}{\overset{\widehat{r_1}}{\Longrightarrow}} \widehat{Y} \qquad \text{et} \qquad \widehat{Y} \underset{\mathcal{G}}{\overset{\widehat{r_2}}{\Longrightarrow}} \widehat{Z}$$

Alors :

$$\widehat{X} \underset{\mathcal{G}}{\overset{\widehat{\widehat{r_1 r_2}}}{\Longrightarrow}} \widehat{Z}$$

— compatible avec le produit de juxtaposition car si :

$$\widehat{X}_1 \underset{\mathcal{G}}{\overset{\widehat{r_1}}{\Longrightarrow}} \widehat{Y}_1 \qquad \text{et} \qquad \widehat{X}_2 \underset{\mathcal{G}}{\overset{\widehat{r_2}}{\Longrightarrow}} \widehat{Y}_2$$

alors :

$$\widehat{X}_1\widehat{X}_2 \underset{\mathcal{G}}{\overset{\widehat{\widehat{r_1 r_2}}}{\Longrightarrow}} \widehat{Y}_1\widehat{Y}_2$$

Le lecteur vérifiera facilement ces deux dernières propriétés, valables même si par exemple $\widehat{r_2} = \widehat{\emptyset}_{\mathcal{R}}$, et constamment utilisées dans les démonstrations à venir. On notera en particulier que si :

$$\widehat{X} \underset{\mathcal{G}}{\overset{\widehat{r}}{\Longrightarrow}} \widehat{Y}$$

alors :

$$\widehat{\forall} G, \widehat{D} \in \mathfrak{L}(\mathcal{V}) : \qquad \widehat{G}\widehat{X}\widehat{D} \underset{\mathcal{G}}{\overset{\widehat{r}}{\Longrightarrow}} \widehat{G}\widehat{Y}\widehat{D}$$

On a vu au paragraphe précédent (1.4.1) qu'un mot pouvait dériver directement d'un autre de plusieurs façons; à plus forte raison, plusieurs dérivations distinctes permettent-elles parfois de passer d'un mot à un autre. Nous nous bornerons ici à illustrer d'un exemple de grammaire C F les problèmes qui se posent à ce propos.

Soit $\mathcal{G}$ la grammaire ayant pour vocabulaire :

$$\mathcal{V} = \{ S, A_1, A_2, A_3 \} \text{ ; avec : } \qquad \mathcal{V}_T = \{ A_1, A_2, A_3 \},$$

et pour ensemble de règles :

$$\mathcal{R} = \{ r_0, r_1, r_2, r'_1, r'_3, r_4 \}$$

définis par :

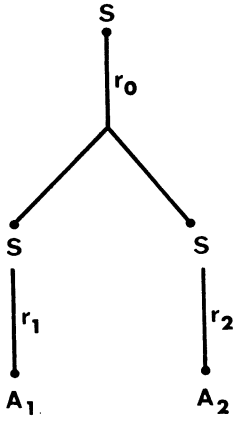
$$\begin{aligned} r_0 &= (S, SS) \\ r_1 &= (S, A_1) \text{ ; } \qquad r_2 = (S, A_2) \\ r'_1 &= (S, A_1S); \qquad r'_3 = (S, SA_3) \\ r_4 &= (S, A_1SA_3). \end{aligned}$$

Il est d'abord facile de donner deux dérivations faisant passer de S au mot A_1A_2; ce sont :

$$S \xRightarrow[\mathcal{G}]{A_1,\widehat{\emptyset},\widehat{\widehat{\emptyset},r_\alpha},\widehat{\emptyset},S,\widehat{\emptyset}} A_1A_2 \quad ; \quad \widehat{r_\alpha} = r_0r_1r_2$$

$$S \xRightarrow[\mathcal{G}]{\widehat{\emptyset},S,\widehat{\widehat{\emptyset},r_\beta},\widehat{\emptyset},\widehat{\emptyset},A_2} A_1A_2 \quad ; \quad \widehat{r_\beta} = r_0r_2r_1$$

Ces deux dérivations ne diffèrent que par l'ordre d'application des règles r_1 et r_2; non, par le point d'application de ces règles. Elles correspondent au même arbre, ou au même schéma parenthétique (intuitivement, dans le schéma parenthétique, on note comme des fonctions les règles... et l'on remonte des terminaux aux symboles S comme par le jeu de fonctions composées; pour un exposé en forme de ces questions, nous renvoyons au § 9, où sera traitée la notion de composition).



$$A_1A_2 = r_0(r_1(A_1), r_2(A_2));$$

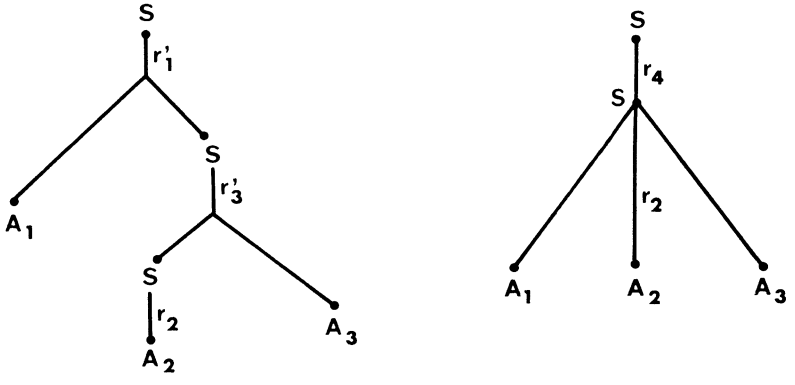
il convient donc de dire que les deux dérivations correspondent à la même structure.

En revanche considérons un deuxième exemple : deux dérivations possibles de $A_1A_2A_3$ à partir de S. On a :

$$S \xRightarrow[\mathcal{G}]{A_1,A_1,\widehat{\emptyset},\widehat{r'},\widehat{\emptyset},\widehat{\emptyset},A_3} A_1A_2A_3 \; ; \qquad \widehat{r'} = r'_1r'_3r_2 \; ;$$

$$S \xRightarrow[\mathcal{G}]{A_1,\widehat{\widehat{\emptyset},r},\widehat{\emptyset},A_3} A_1A_2A_3 \; ; \qquad \widehat{r} = r_4r_2 \; ;$$

Ici il ne s'agit même pas des mêmes règles ; *a fortiori* pas du même arbre ; on a deux arbres et deux structures parenthétiques :



$$A_1A_2A_3 = r'_1(A_1, r'_3(r_2(A_2), A_3))$$
$$A_1A_2A_3 = r_4(A_1, r_2(A_2), A_3).$$

Ici il s'agit donc d'une *ambiguïté* de structure proprement dite, non de deux possibilités différentes de fabriquer, par étapes successives, la même structure.

Dans la première partie de la présente étude, nous nous occuperons très peu de ces questions de structure et d'ambiguïté car dans le cas des grammaires de type 1 et 2 nous ne pouvons associer à toute dérivation un « arbre » du type précédent; en effet les dérivations, au lieu de porter sur un seul symbole, comme dans le cas des grammaires C F, portent sur un mot (grammaires de type 1) ou sur un symbole placé dans un contexte défini (grammaires de type 2). Sans arbre de dérivation, il est fort difficile de définir la notion d'ambiguïté. De ce point de vue les grammaires de type 1 et 2 ne sont pas d'un emploi commode pour décrire la structure des langues naturelles, car s'il est important de savoir si une phrase est syntaxiquement correcte, il est non moins important de lui assigner une structure syntaxique définie pour savoir comment elle doit être comprise. Or les phrases ambiguës sont fréquentes non seulement dans la langue parlée, mais aussi dans la langue écrite, telle cette phrase de Molière tirée des *Fourberies de Scapin* :

« Il me faut aussi un cheval pour monter mon valet qui me coûtera bien trente pistoles. »

A ce propos, nous signalons les grammaires C F à constituants non

connexes de J. P. Benzécri, qui généralisent les grammaires C F ordinaires tout en conservant leur « structure arborescente » (nous y reviendrons au § 9).

1.5. Langages.

1.5.1. Définition 1.

Un langage, défini sur le vocabulaire $\mathfrak{V}$, sera un sous-ensemble du monoïde libre $\mathfrak{L}(\mathfrak{V})$.

Tout mot d'un langage sera appelé phrase.

Nous allons définir ce qu'est le langage engendré par une grammaire de constituants :

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R}).$$

Dans ce cas, nous aurons $\mathfrak{V} = \mathfrak{V}_T$, c'est-à-dire que toute phrase sera un mot terminal. Ainsi le vocabulaire non terminal $\mathfrak{V}_N$ apparaît ici sous son rôle de vocabulaire auxiliaire, ses éléments intervenant seulement dans l'écriture des règles de production et en cours de dérivation; il pourra donc varier d'une grammaire à l'autre, tandis que, dans toute cette étude (à moins d'indication contraire), le vocabulaire terminal $\mathfrak{V}_T$ ne changera pas.

1.5.2. Définition 2.

Le langage engendré par la grammaire

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

est l'ensemble des mots terminaux qui dérivent de S, selon $\mathfrak{G}$.

On le notera $\mathfrak{L}[\mathfrak{G}]$. Ainsi :

$$\mathfrak{L}[\mathfrak{G}] = \left\{ \widehat{X} \mid \widehat{X} \in \mathfrak{L}(\mathfrak{V}_T) \; ; \; \exists \widehat{r} \in \mathfrak{L}(\mathfrak{R}) \quad : \quad S \underset{\mathfrak{G}}{\overset{\widehat{r}}{\Longrightarrow}} \widehat{X} \right\}$$

On saisit pourquoi S (de « sentence » en anglais) est appelé symbole initial.

Exemple : Nous allons décrire une grammaire simplifiée et la génération d'une phrase par cette grammaire.

Le vocabulaire terminal sera réduit à :

$\mathfrak{V}_T = \{$ un, étudiant, problème, très, toujours, doué, difficile, résout, ne, pas $\}$ le vocabulaire non terminal est constitué par :

$$\mathfrak{V}_N = \{ S, S_N, S_V, A_d, A_j, V, \overline{V}, A_r, N \}$$

ces symboles désignant respectivement :

S : phrase		A_r : article
S_N : syntagme nominal		A_d : adverbe
S_V : syntagme verbal		A_j : adjectif
V : verbe	$\overline{V}$: verbe négatif	N : nom.

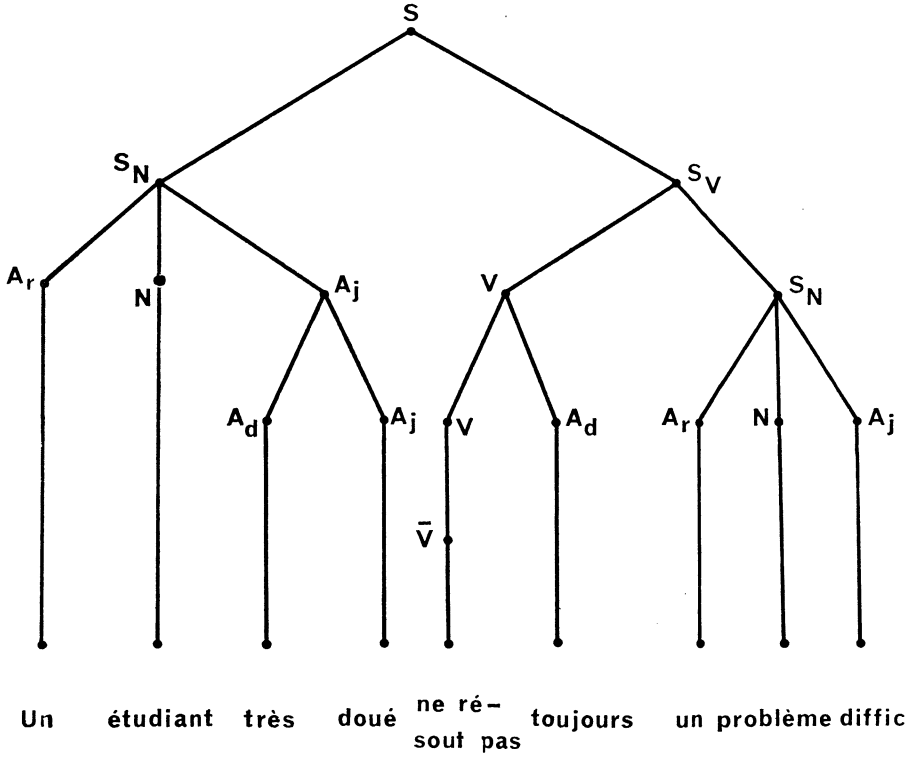
Il y a les règles de syntaxe (le lecteur les traduira facilement)

$S \rightarrow S_N S_V$	$S_V \rightarrow VS_N$
$S_N \rightarrow A_r N A_j$	$V \rightarrow VA_d$
$A_j \rightarrow A_d A_j$	$V \rightarrow \overline{V}$

et les règles terminales

$A_r \rightarrow$ un	$\overline{V} \rightarrow$ ne résout pas
$A_d \rightarrow$ très	$A_d \rightarrow$ toujours
$A_j \rightarrow$ doué	$A_j \rightarrow$ difficile
N $\rightarrow$ étudiant	N $\rightarrow$ problème

Le lecteur pourra écrire la S-dérivation correspondant à l'arbre suivant :



Nous avons donc obtenu la phrase : « Un étudiant très doué ne résout pas toujours un problème difficile ». Nous aurions pu éviter la tournure négative en n'utilisant pas la règle (V, $\overline{V}$). A cause des règles (A_j, A_dA_j) et (V, VA_d) les adverbes peuvent être répétés autant de fois que l'on veut. Pour éviter cela on pourrait introduire des symboles nouveaux par exemple A'_j et la règle (A_j, $A_dA'_j$) remplacerait la règle (A_j, A_dA_j), et on ajouterait (A'_j, doué), etc. A partir de ces simples règles on pourrait engendrer bien d'autres phrases, plusieurs étant dénuées de toute signification car par exemple l'adverbe « très » ne peut être déterminant d'un verbe, pas plus que « beaucoup » ne peut être déterminant d'un adjectif. Cela évidemment peut aussi s'éviter en utilisant suffisamment de règles et une classification rigoureuse. En élargissant le vocabulaire terminal on arriverait à engendrer tout un ensemble de phrases munies de la même structure syntaxique.

1.5.3. **Définition 3. — Principales classes de langages.**

Une partie $\mathfrak{L}$ du monoïde $\mathfrak{L}(\mathfrak{V}_T)$ est un *langage de type 1* (resp. de type 2 ou C F) s'il existe une grammaire $\mathfrak{G}$, de type 1 (resp. de type 2 ou C F) engendrant $\mathfrak{L}$, i. e. :

$$\mathfrak{L} = \mathfrak{L}[\mathfrak{G}].$$

Ainsi à chaque classe de grammaires correspond une classe de langages. L'ensemble des parties de $\mathfrak{L}(\mathfrak{V}_T)$ qui sont des langages de type 1 (resp. 2) sera noté $\mathfrak{L}\{\alpha_1\}$ (resp. $\mathfrak{L}\{\alpha_2\}$).

Si une grammaire $\mathfrak{G}$ est de type 1 nous aurons donc :

$$\mathfrak{L}[\mathfrak{G}] \in \mathfrak{L}\{\alpha_1\}.$$

A toute grammaire de type 1 correspond un langage unique de type 1, éventuellement vide. Mais il importe de remarquer que la correspondance n'est pas injective : plusieurs grammaires pouvant engendrer le même langage... Il faut donc prendre quelques précautions dans la classification des langages. Ainsi pour démontrer qu'un langage n'est pas de type 1, il faudrait démontrer que toute grammaire qui l'engendre n'est jamais de type 1.

1.6. **Décidabilité. Récursivité.**

L'un des plus célèbres problèmes indécidables est le *problème de correspondance de Post* (1946) [3]. Nous allons le rappeler ici car les cas d'indécidabilité, que nous rencontrerons, se réduiront à ce problème de Post.

Étant donné deux n-uplets de mots sur un vocabulaire

$$(\widehat{A}_1, \widehat{A}_2, \ldots, \widehat{A}_n), \qquad (\widehat{B}_1, \widehat{B}_2, \ldots, \widehat{B}_n)$$

on se pose le problème suivant : existe-t-il une suite finie d'indices :

$i_1, i_2, \ldots, i_k \qquad \forall j \in]k] : \quad i_j \in]n]$

telle que :

$$\widehat{A}_{i_1} \widehat{A}_{i_2} \ldots \widehat{A}_{i_k} = \widehat{B}_{i_1} \widehat{B}_{i_2} \ldots \widehat{B}_{i_k} \ ?$$

Dans le cas où le vocabulaire $\mathfrak{V}$ contient plus d'un symbole Post a démontré qu'un tel problème est indécidable. Cela signifie donc qu'il n'existe pas d'algorithme général qui, étant donné deux n-uplets quelconques, permette de savoir s'il existe ou non une suite d'indices réalisant la correspondance signalée.

Problème du mot : Quand on considère un langage $\mathfrak{L}$ il se pose naturellement le problème suivant :

— étant donné un mot $\widehat{X}$ quelconque, existe-t-il un procédé effectif pour déterminer si $\widehat{X}$ est une phrase de $\mathfrak{L}$ ou non ? C'est ce que nous appellerons, en bref, le problème du mot (pour ce langage).

Les *langages* pour lesquels ce problème peut être effectivement résolu sont dits *récursifs*. Les langages de type 1 (donc aussi ceux de type 2 et les langages C F) sont tous récursifs [3]. La démonstration de ce résultat repose essentiellement sur le fait que lors d'une dérivation, selon une grammaire de type 1, la longueur des mots ne décroît pas. On n'a donc en fait à envisager pour chaque mot qu'un nombre fini de dérivations qui puissent lui avoir donné naissance à partir de S.

En vue de démontrer l'identité des deux ensembles $\mathfrak{L}\{\alpha_1\}$ et $\mathfrak{L}\{\alpha_2\}$ (proposition fondamentale : § 4) et d'établir quelques propriétés de ces langages (§ 5, 6, 7), nous étudierons auparavant quelques critères de comparaison de grammaires (§ 2) et des simplifications des grammaires de type 1 (§ 3).

2. GRAMMAIRES ÉQUIVALENTES. HOMOMORPHISME DE GRAMMAIRES

2.1. Grammaires équivalentes.

2.1.1. Définition.

Soient deux grammaires de constituants $\mathfrak{G}$ et $\mathfrak{G}'$:

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R}) \qquad \mathfrak{G}' = (\mathfrak{V}', \mathfrak{V}_T, S', \mathfrak{R}');$$

elles seront dites équivalentes si elles engendrent le même langage, i. e. :

$$\mathfrak{L}[\mathfrak{G}] = \mathfrak{L}[\mathfrak{G}'].$$

La notion d'équivalence de grammaires est envisagée ici dans son *sens faible* correspondant au point de vue adopté dans cette étude : point de vue purement ensembliste laissant de côté les questions de description structurale des phrases et d'ambiguïté. Le formalisme du paragraphe 9 nous permettra tout de même d'entrevoir ce dernier point de vue.

Le problème de savoir si deux grammaires $\mathfrak{G}$ et $\mathfrak{G}'$ sont faiblement équivalentes est dans toute sa généralité, indécidable. Il l'est même dans le cas où les grammaires $\mathfrak{G}$ et $\mathfrak{G}'$ sont toutes deux C F [3] et donc *a fortiori* dans les cas plus généraux où elles sont toutes deux de type 2 ou de type 1. Le problème de savoir si la grammaire $\mathfrak{G}$ est « plus puissante » que la grammaire $\mathfrak{G}'$ au sens suivant :

$$\mathfrak{L}[\mathfrak{G}] \subset \mathfrak{L}[\mathfrak{G}'],$$

ne peut lui non plus être décidable, sinon le résultat précédent serait faux... Dans certains cas particuliers, ces problèmes peuvent cependant être résolus.

2.1.2. Un critère d'équivalence.

ÉNONCÉ. — **Soient deux grammaires $\mathfrak{G}$ et $\mathfrak{G}'$:**

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R}) \quad ; \quad \mathfrak{G}' = (\mathfrak{V}', \mathfrak{V}_T, S, \mathfrak{R}')$$

$\mathfrak{V} \subset \mathfrak{V}'$, le symbole initial étant le même.

Une condition suffisante pour que nos deux grammaires soient équivalentes est qu'il existe une application φ de $\mathfrak{R}$ dans $\mathfrak{L}(\mathfrak{R}')$ satisfaisant simultanément aux trois hypothèses suivantes :

($\mathfrak{H}_1$) : Soit $r \in \mathfrak{R}$, $r = (\widehat{A}, \widehat{B})$; notons $\widehat{r'} = \varphi(r)$; on a :

$$\widehat{A} \underset{\mathfrak{G}'}{\overset{\widehat{r'}}{\Longrightarrow}} \widehat{B}$$

($\mathfrak{H}_2$) : Soit $\widehat{E} \in \mathfrak{L}(\mathfrak{V})$ tel que :

$$\widehat{E} \underset{\mathfrak{G}'}{\overset{\widehat{r'}}{\Longrightarrow}} \widehat{F}$$

où $\widehat{r'} = \varphi(r)$, $r \in \mathfrak{R}$, $r = (\widehat{A}, \widehat{B})$. Alors il existe deux mots $\widehat{G}$ et $\widehat{D} \in \mathfrak{L}(\mathfrak{V})$, tels que :

$$\widehat{E} = \widehat{G}\widehat{A}\widehat{D} \quad ; \quad \widehat{F} = \widehat{G}\widehat{B}\widehat{D},$$

i. e. que :

$$\widehat{E} \underset{\mathfrak{G}}{\overset{\widehat{G},r,\widehat{D}}{\longrightarrow}} \widehat{F}$$

($\mathcal{H}_3$) : φ étant, par extension, considéré comme un homomorphisme de $\mathfrak{L}(\mathcal{R})$ dans $\mathfrak{L}(\mathcal{R}')$, pour toute phrase $\widehat{X}$ de $\mathfrak{L}[\mathcal{G}']$ il existe $\widehat{r'}$, $\widehat{r'} \in \varphi(\mathfrak{L}(\mathcal{R}))$ tel que :

$$S \underset{\mathcal{G}'}{\overset{\widehat{r'}}{\Longrightarrow}} \widehat{X}$$

DÉMONSTRATION. — Il s'agit de prouver que $\mathfrak{L}[\mathcal{G}] = \mathfrak{L}[\mathcal{G}']$.

1° $\mathfrak{L}[\mathcal{G}] \subset \mathfrak{L}[\mathcal{G}']$: cela résultera de l'hypothèse ($\mathcal{H}_1$). Nous allons prouver que si $\widehat{X}$ est une phrase du langage $\mathfrak{L}[\mathcal{G}]$, alors $\widehat{X}$ est aussi une phrase du langage $\mathfrak{L}[\mathcal{G}']$, plus précisément si :

$$S \underset{\mathcal{G}}{\overset{\widehat{r}}{\Longrightarrow}} \widehat{X}$$

alors

$$S \underset{\mathcal{G}'}{\overset{\widehat{r'}}{\Longrightarrow}} \widehat{X} \qquad \text{avec} \qquad \widehat{r'} = \varphi(\widehat{r})$$

En effet $\widehat{X}$ étant une phrase de $\mathfrak{L}[\mathcal{G}]$ soit

$$(S = \widehat{X}_0, \widehat{X}_1, \ldots, \widehat{X}_i, \ldots, \widehat{X}_n, \widehat{X})$$

une S-dérivation de $\widehat{X}$, selon $\mathcal{G}$, par $\widehat{r} = r_0 \ldots r_i \ldots r_n$, i. e. :

$$\forall i \in (n) \quad : \quad \widehat{X}_i = \widehat{G}_i \widehat{A}_i \widehat{D}_i \xrightarrow[\mathcal{G}]{\widehat{G}_i, r_i, \widehat{D}_i} \widehat{X}_{i+1} = \widehat{G}_i \widehat{B}_i \widehat{D}_i, \qquad r_i = (\widehat{A}_i, \widehat{B}_i) \in \mathcal{R}$$

Mais, d'après l'hypothèse ($\mathcal{H}_1$), $\widehat{B}_i$ dérive de $\widehat{A}_i$, selon $\mathcal{G}'$, par :

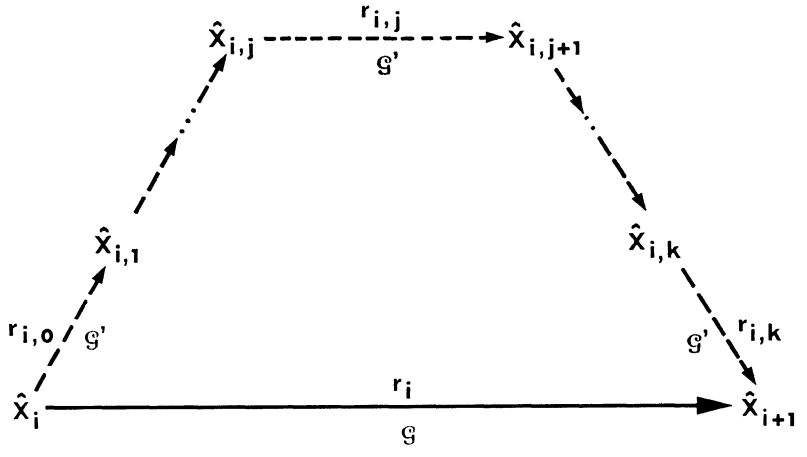
$$\widehat{r'}_i = \varphi(r_i) = r_{i,0} \ldots r_{i,j} \ldots r_{i,k} \qquad \forall j \in)k) : \qquad r_{i,j} \in \mathcal{R}'.$$

$$\widehat{A}_i \underset{\mathcal{G}'}{\overset{\widehat{r'}_i}{\Longrightarrow}} \widehat{B}_i$$

et donc en vertu des propriétés de la relation binaire $\Rightarrow$ (1.4.2) :

$$\widehat{X}_i = \widehat{G}_i \widehat{A}_i \widehat{D}_i \underset{\mathcal{G}'}{\overset{\widehat{r'}_i}{\Longrightarrow}} \widehat{X}_{i+1} = \widehat{G}_i \widehat{B}_i \widehat{D}_i$$

Nous pouvons schématiser ce résultat de la façon suivante :



Il s'ensuit que $\widehat{X}$ est aussi une phrase de $\mathfrak{L}[\mathcal{G}']$ dérivant de S par :

$$\widehat{r'} = \varphi(\widehat{r}) = \widehat{r'}_0 \dots \widehat{r'}_i \dots \widehat{r'}_n$$

2° $\mathfrak{L}[\mathcal{G}'] \subset \mathfrak{L}[\mathcal{G}]$: cela résultera des hypothèses $(\mathcal{H}_2)$ et $(\mathcal{H}_3)$. $\widehat{X}$ étant une phrase de $\mathfrak{L}[\mathcal{G}']$ l'hypothèse $(\mathcal{H}_3)$ nous indique qu'elle peut dériver de S, selon $\mathcal{G}'$, par une suite, $\widehat{r'}$, de règles appartenant à $\varphi(\mathcal{R})$, i. e. :

$$S \overset{\widehat{r'}_0}{\underset{\mathcal{G}'}{\Longrightarrow}} \widehat{X}_1 \Longrightarrow \dots \Longrightarrow \widehat{X}_i \overset{\widehat{r'}_i}{\underset{\mathcal{G}'}{\Longrightarrow}} \widehat{X}_{i+1} \Longrightarrow \dots \Longrightarrow \widehat{X}_n \overset{\widehat{r'}_n}{\underset{\mathcal{G}'}{\Longrightarrow}} \widehat{X}$$

$$\forall i \in (n) \quad : \quad \widehat{r'}_i = \varphi(r_i) \qquad r_i \in \mathcal{R}, \quad \text{par exemple} \quad r_i = (\widehat{A}_i, \widehat{B}_i)$$

S étant un mot de $\mathfrak{L}(\mathcal{V})$, si nous supposons, en raisonnant par récurrence, qu'il en est de même de $\widehat{X}_i$, l'hypothèse $(\mathcal{H}_2)$ nous indique qu'il existe deux mots $\widehat{G}_i$ et $\widehat{D}_i$ de $\mathfrak{L}(\mathcal{V})$ tels que :

$$\widehat{X}_i \xrightarrow[\mathcal{G}]{\widehat{G}_i r_i \widehat{D}_i} \widehat{X}_{i+1}$$

(le schéma ci-dessus reste valable) et $\widehat{X}_{i+1} = \widehat{G}_i \widehat{B} \widehat{D}$ est donc aussi un mot de $\mathfrak{L}(\mathcal{V})$.

Il en résulte que :

$$S \xrightarrow[\mathcal{G}]{r_0} \widehat{X}_1 \longrightarrow \dots \longrightarrow \widehat{X}_i \xrightarrow[\mathcal{G}]{r_i} \widehat{X}_{i+1} \longrightarrow \dots \longrightarrow \widehat{X}_n \xrightarrow[\mathcal{G}]{r_n} \widehat{X}$$

et $\widehat{X}$ est bien aussi une phrase de $\mathfrak{L}[\mathfrak{G}]$. Elle dérive de S, selon $\mathfrak{G}$, par $\widehat{r} \in \mathfrak{L}(\mathcal{R})$ tel que $\varphi(\widehat{r}) = \widehat{r'} = \widehat{r'}_0 \ldots \widehat{r'} \ldots \widehat{r'}_n$.

Ainsi s'achève la démonstration d'un critère d'équivalence que nous utiliserons par la suite.

2.2. Homomorphisme de grammaires.

2.2.1. Définition.

Étant donné deux grammaires $\mathfrak{G}$ et $\mathfrak{G}'$

$$\mathfrak{G} = (\mathcal{V}, \mathcal{V}_T, S, \mathcal{R}) \qquad \mathfrak{G}' = (\mathcal{V}', \mathcal{V}_T, S', \mathcal{R}')$$

un homomorphisme de monoïdes, φ, de $\mathfrak{L}(\mathcal{V})$ dans $\mathfrak{L}(\mathcal{V}')$ sera appelé homomorphisme de grammaires si les trois conditions suivantes sont simultanément satisfaites :

1° La restriction de φ à $\mathcal{V}_T$ (donc à $\mathfrak{L}(\mathcal{V}_T)$) est l'application identique;
2° $\varphi(S) = S'$.
3° Pour toute règle de $\mathfrak{G}$, par exemple $r = (\widehat{A}, \widehat{B})$, nous avons :

— ou bien $r' = (\varphi(\widehat{A}), \varphi(\widehat{B}))$ est une règle de $\mathfrak{G}'$;
— ou bien $\varphi(\widehat{A}) = \varphi(\widehat{B})$.

Dans le premier cas, on posera $\varphi(r) = r'$, dans le second $\varphi(r) = \widehat{\mathcal{O}}_{\mathcal{R}'}$, $\mathcal{O}_{\mathcal{R}'}$, désignant l'élément neutre de $\mathfrak{L}(\mathcal{R}')$. Cela nous permet, par extension, de considérer φ comme un homomorphisme du monoïde $\mathfrak{L}(\mathcal{R})$ dans le monoïde $\mathfrak{L}(\mathcal{R}')$.

Le résultat important découlant immédiatement de cette définition est le suivant :

2.2.2. Conséquence.

La grammaire $\mathfrak{G}'$ est au moins « aussi puissante » que la grammaire $\mathfrak{G}$, i. e. :

$$\mathfrak{L}[\mathfrak{G}] \subset \mathfrak{L}[\mathfrak{G}']$$

En effet à toute dérivation selon $\mathfrak{G}$ correspond, par φ, une dérivation selon $\mathfrak{G}'$. Soit la dérivation directe :

$$\widehat{X} \xrightarrow[\mathfrak{G}]{\widehat{G}, r, \widehat{D}} \widehat{Y} \qquad r = (\widehat{A}, \widehat{B}) \in \mathcal{R}$$

— Si $\varphi(r) = r' = (\varphi(\widehat{A}), \varphi(\widehat{B})) \in \mathcal{R}'$ alors :

$$\varphi(\widehat{X}) \xrightarrow[\mathcal{G}']{\varphi(\widehat{G}),r',\varphi(\widehat{D})} \varphi(\widehat{Y})$$

— Si $\varphi(r) = \widehat{\varnothing}_{\mathcal{R}'}$, alors $\varphi(\widehat{A}) = \varphi(\widehat{B})$ et donc $\varphi(\widehat{X}) = \varphi(\widehat{Y})$.

Dans tous les cas $\varphi(\widehat{Y})$ dérive de $\varphi(\widehat{X})$ selon $\mathcal{G}'$. Ce résultat s'étend immédiatement à toute dérivation qui n'est qu'une succession de dérivations directes. En particulier, $\widehat{X}$ étant une phrase de $\mathcal{L}[\mathcal{G}]$, si :

$$S \xRightarrow[\mathcal{G}]{\widehat{r}} \widehat{X}$$

alors :

$$\varphi(S) = S' \xRightarrow{\varphi(\widehat{r})=\widehat{r'}} \varphi(\widehat{X}) = \widehat{X} \qquad \text{car} \qquad \widehat{X} \in \mathcal{L}(\mathcal{V}_T)$$

Ainsi $\widehat{X}$ est aussi une phrase de $\mathcal{L}[\mathcal{G}']$, ce qui établit notre résultat.

3. RÉDUCTIONS DES GRAMMAIRES DE TYPE 1

Nous rappelons qu'une grammaire est dite de type 1, si pour toute règle de production, $r = (\widehat{A}, \widehat{B})$, la longueur du second membre, $\widehat{B}$, est supérieure ou égale à celle du premier membre, $\widehat{A}$.

3.1. Grammaires syntaxiques.

3.1.1. Définition.

Une grammaire

$$\mathcal{G} = (\mathcal{V}, \mathcal{V}_T, S, \mathcal{R})$$

sera dite syntaxique si toute règle de production, $r = (\widehat{A}_{(r)}, \widehat{B}_{(r)})$, a un second membre dont tous les symboles appartiennent exclusivement soit à $\mathcal{V}_T$ soit à $\mathcal{V}_N$:

$$\forall r \in \mathcal{R} \quad : \quad (\widehat{B}_{(r)} \in \mathcal{L}(\mathcal{V}_T)) \vee (\widehat{B}_{(r)} \in \mathcal{L}(\mathcal{V}_N))$$

Si une grammaire est syntaxique, on peut réaliser une partition de $\mathcal{R}$ en

deux sous-ensembles $\mathcal{R}_N$ (règles non terminales) et $\mathcal{R}_T$ (règles terminales), définis par :

$$\mathcal{R}_T = \{ r \mid r \in \mathcal{R} ; \quad \widehat{B}_{(r)} \in \mathfrak{L}(\mathcal{V}_T) \}$$
$$\mathcal{R}_N = \{ r \mid r \in \mathcal{R} ; \quad \widehat{B}_{(r)} \in \mathfrak{L}(\mathcal{V}_N) \}$$
$$\mathcal{R} = \mathcal{R}_T \cup \mathcal{R}_N$$

La grammaire de l'exemple 15-2, où l'on a distingué les règles de la syntaxe de celles relatives au lexique est évidemment une grammaire syntaxique.

3.1.2. **Lemme.**

Toute grammaire de type 1 est équivalente à une grammaire syntaxique de type 1.

DÉMONSTRATION. — Soit $\mathcal{G}$ une grammaire de type 1

$$\mathcal{G} = (\mathcal{V}, \mathcal{V}_T, S, \mathcal{R}).$$

Définissons la grammaire

$$\mathcal{G}' = (\mathcal{V}', \mathcal{V}_T, S, \mathcal{R}')$$

de la façon suivante :

— $\mathcal{V}' = \mathcal{V} \cup \mathcal{V}_1$, $\mathcal{V}_1$ étant tel que :

. $\mathcal{V}_1$ et $\mathcal{V}$ n'ont aucun symbole commun, $\mathcal{V} \cap \mathcal{V}_1 = \emptyset$;

. il existe une bijection ψ de $\mathcal{V}_T$ sur $\mathcal{V}_1$. On posera :

$$\forall A_i \in \mathcal{V}_T \quad : \quad A'_i = \psi(A_i).$$

En posant de plus :

$$\forall A_j \in \mathcal{V}_N \quad : \quad A_j = \psi(A_j)$$

ψ pourra, par une extension naturelle, être considéré comme un homomorphisme de $\mathfrak{L}(\mathcal{V})$ dans $\mathfrak{L}(\mathcal{V}')$ et on écrira :

$$\forall \widehat{X} \in \mathfrak{L}(\mathcal{V}) \quad : \quad \widehat{X}' = \psi(\widehat{X})$$

Un mot $\widehat{X}$ de $\mathfrak{L}(\mathcal{V})$ se trouve ainsi transformé en un mot non terminal $\widehat{X}'$ de $\mathfrak{L}(\mathcal{V}')$, tout symbole terminal de $\widehat{X}$ étant remplacé par son symbole correspondant de $\mathcal{V}_1$, or $\mathcal{V}_1 \subset \mathcal{V}'_N$.

— $\mathcal{R}'$ est alors défini comme suit, il comprend :

. des règles de production non terminales en bijection avec les règles de $\mathcal{G}$:

$$\forall r = (\widehat{A}, \widehat{B}) \in \mathcal{R} : \qquad r' = (\psi(\widehat{A}), \psi(\widehat{B})) = (\widehat{A}', B') \in \mathcal{R}'_N.$$

En posant $r' = \psi(r)$, ψ peut aussi être considéré comme un homomorphisme (plus précisément un isomorphisme) de $\mathfrak{L}(\mathcal{R})$ sur $\mathfrak{L}(\mathcal{R}'_N)$.

. des règles de production terminales :

$$\forall A_i \in \mathfrak{V}_T \quad : \quad t_i = (A'_i, A_i) \in \mathfrak{R}'_T$$
$$\mathfrak{R}' = \mathfrak{R}'_N \cup \mathfrak{R}'_T \qquad \text{et} \qquad \mathfrak{R}'_N \cap \mathfrak{R}'_T = \emptyset$$

$\mathfrak{G}'$ est donc une grammaire syntaxique de type 1 : prouvons qu'elle est équivalente à la grammaire $\mathfrak{G}$.

1° $\mathfrak{L}[\mathfrak{G}] \subset \mathfrak{L}[\mathfrak{G}']$.

L'application ψ transformant un mot terminal $\widehat{X}$ en le mot $\widehat{X}'$ de $\mathfrak{L}(\mathfrak{V}_1)$ et vérifiant les conditions 2° et 3° d'un homomorphisme de grammaires il est facile de prouver, en raisonnant comme au paragraphe 2.2.2, que si $\widehat{X}$ est une phrase de $\mathfrak{L}[\mathfrak{G}]$ telle que :

$$S \underset{\mathfrak{G}}{\overset{\widehat{r}}{\Longrightarrow}} \widehat{X}$$

Alors

$$\psi(S) = S \underset{\mathfrak{G}'}{\overset{\psi(\widehat{r}) = \widehat{r'}}{\Longrightarrow}} \psi(\widehat{X}) = \widehat{X}', \qquad \widehat{r'} \in \mathfrak{L}(\mathfrak{R}'_N) ;$$

$\widehat{X}$ étant un mot terminal nous avons par exemple :

$$\widehat{X} = A_{i_1} \ldots A_{i_j} \ldots A_{i_r} \quad , \quad \forall j \in)r) \quad : \quad A_{i_j} \in \mathfrak{V}_T$$

et donc :

$$\widehat{X}' = A'_{i_1} \ldots A'_{i_j} \ldots A'_{i_r} \quad , \quad \forall j \in)r) \quad : \quad A'_{i_j} \in \mathfrak{V}_1 ;$$

il s'ensuit que :

$$\widehat{X}' \underset{\mathfrak{G}'}{\overset{\widehat{t'}}{\Longrightarrow}} \widehat{X}, \qquad \widehat{t'} = t_{i_1} \ldots t_{i_j} \ldots t_{i_r} \in \mathfrak{L}(\mathfrak{R}'_T)$$

$\widehat{X}$ est donc aussi une phrase de $\mathfrak{L}[\mathfrak{G}']$ dérivant de S selon $\mathfrak{G}'$ par $\widehat{r'}\widehat{s'}$: $\widehat{r'} \in \mathfrak{L}(\mathfrak{R}'_N)$ et $\widehat{s'} \in \mathfrak{L}(\mathfrak{R}'_T)$ (propriété de transitivité de la relation $\Rightarrow$).

N. B. — On notera que la phrase $\widehat{X}$ est engendrée, selon $\mathfrak{G}'$, par une S-dérivation où les règles terminales s'appliquent regroupées en fin de dérivation.

2° $\mathfrak{L}[\mathfrak{G}'] \subset \mathfrak{L}[\mathfrak{G}]$.

Considérons pour cela l'homomorphisme φ de $\mathfrak{L}(\mathfrak{V}')$ dans $\mathfrak{L}(\mathfrak{V})$ dont la restriction à $\mathfrak{V}$ est l'application identique et la restriction à $\mathfrak{V}_1$ l'application réciproque de ψ, i. e. :

. $\forall A_i \in \mathfrak{V} \subset \mathfrak{V}'$: $\varphi(A_i) = A_i$, en particulier $\varphi(S) = S$,
. et $\forall A'_i \in \mathfrak{V}_1$: $\varphi(A'_i) = A_i$.

Il en résulte que :

. pour toute règle non terminale de $\mathfrak{G}'$, $r' = (\widehat{A}', \widehat{B}') \in \mathfrak{R}'_N$:

$$\varphi(r') = (\varphi(\widehat{A}'), \varphi(\widehat{B}')) = (\widehat{A}, \widehat{B}) = r \in \mathfrak{R}$$

. et pour toute règle terminale de $\mathfrak{G}'$, $t_i = (A'_i, A_i) \in \mathfrak{R}'_T$:

$$\varphi(t_i) = (\varphi(A'_i), \varphi(A_i)) = (A_i, A_i)$$

Les trois conditions de la définition 2.2.1 étant réalisées, φ est un homomorphisme de grammaires et notre résultat se trouve donc démontré.

3.1.3. Généralisation.

La démonstration précédente ne tient aucunement compte de l'axiome (α_1). Elle peut s'appliquer en particulier à une grammaire de type 2 ou à une grammaire C F, d'où la généralisation suivante du lemme précédent :

— A toute grammaire de type 1, de type 2 ou C F on peut associer une grammaire syntaxique équivalente de même type.

Dans la définition des règles, $r = (\widehat{A}, \widehat{B})$, d'une grammaire de constituants, nous avons posé que le premier membre $\widehat{A}$ est un mot du vocabulaire non terminal, $\widehat{A} \in \mathfrak{L}(\mathfrak{V}_N)$. Cette restriction n'est pas toujours faite, mais comme on le voit cela ne modifie en rien la capacité générative des classes de grammaires ainsi définies.

3.2. Grammaires d'ordre *n* [5].

3.2.1. Définitions.

— Une règle $r = (\widehat{A}, \widehat{B})$ d'une grammaire $\mathfrak{G}$, de type 1, sera dite d'*ordre p* si le second membre $\widehat{B}$ est de longueur p, i. e. : $|\widehat{B}| = p$.

— La grammaire $\mathfrak{G}$ elle-même sera dite d'ordre n si toutes ses règles sont d'ordre au plus égal à n.

3.2.2. Lemme.

Toute grammaire, de type 1, est équivalente à une grammaire de type 1, d'ordre 2.

DÉMONSTRATION. — Supposons, en vertu du lemme 3.1.2, que la grammaire de type 1

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

est syntaxique. Il en résulte en particulier, d'après ce même lemme, que l'on peut supposer que toute règle terminale est d'ordre 1. Nous allons prouver que la grammaire $\mathfrak{G}$, supposée d'ordre n, $n \geqslant 3$, est équivalente à une grammaire $\mathfrak{G}'$ d'ordre $n-1$, ce qui permettra bien de se ramener à une grammaire d'ordre 2.

Cette nouvelle grammaire

$$\mathfrak{G}' = (\mathfrak{V}', \mathfrak{V}_T, S, \mathfrak{R}')$$

nous allons la définir à partir de la grammaire initiale $\mathfrak{G}$. Ainsi les règles de $\mathfrak{G}$ d'ordre au plus égal à 2 vont être conservées (on aura donc en particulier $\mathfrak{R}'_T = \mathfrak{R}_T$); celles d'ordre n, $n \geqslant 3$, seront remplacées par une suite de règles d'ordre au plus égal à $n-1$ et construites en s'aidant de symboles nouveaux. Les symboles de $\mathfrak{V}$ étant tous conservés nous aurons donc $\mathfrak{V} \subset \mathfrak{V}'$; le symbole initial est le même pour les deux grammaires. Le tableau suivant va préciser cette construction et définir une application φ de $\mathfrak{R}$ dans $\mathfrak{L}(\mathfrak{R}')$ qui nous mettra dans la situation du paragraphe 2.1.2.

$$\begin{array}{c} \mathfrak{G} \qquad\qquad\qquad \mathfrak{G}' \\ r = (\widehat{A}, \widehat{B}) \in \mathfrak{R} \longrightarrow \varphi(r) \in \mathfrak{L}(\mathfrak{R}') \end{array}$$

$\lvert\widehat{B}\rvert \leqslant 2$	$\varphi(r) = r \in \mathfrak{R}'$
$\lvert\widehat{B}\rvert \geqslant 3$ et $\widehat{A} \in \mathfrak{V}$ e. g. $\widehat{A} = A_1$ et $\widehat{B} = B_1B_2B_3\widehat{B}'$	$\varphi(r) = r_1r_2$ $r_1 = (A_1, B_1A'_r)$, $r_2 = (A'_r, B_2B_3\widehat{B}')$, $A'_r \notin \mathfrak{V}$
$\lvert\widehat{B}\rvert \geqslant 3$ et $\lvert\widehat{A}\rvert > 1$ e. g. $\widehat{A} = A_1A_2\widehat{A}'$ et $\widehat{B} = B_1B_2B_3\widehat{B}'$	$\varphi(r) = r'_1r'_2$ $r'_1 = (A_1A_2, B_1A''_r)$, $r'_2 = (A''_r\widehat{A}', B_2B_3\widehat{B}')$, $A''_r \notin \mathfrak{V}$

Le symbole nouveau A'_r ou A''_r introduit dans le 2e ou 3e cas est différent pour chaque règle, comme il apparaît à l'indice r, dont nous l'avons pourvu. La grammaire $\mathfrak{G}'$ ainsi construite est bien d'ordre strictement inférieur à n, $n > 2$. Pour prouver l'équivalence des deux grammaires $\mathfrak{G}$ et $\mathfrak{G}'$, il suffit

de démontrer que les hypothèses du critère d'équivalence 2.1.2 sont satisfaites.

— ($\mathcal{H}_1$) : Il est immédiat, d'après la construction des règles de $\mathcal{G}'$, que :

$$\forall r \in \mathcal{R},\ r = (\widehat{A}, \widehat{B}) \quad : \quad \widehat{A} \xRightarrow[\mathcal{G}']{\varphi(r)} \widehat{B}.$$

— ($\mathcal{H}_2$) : Il s'agit de prouver que si :

$$\widehat{E} \in \mathfrak{L}(\mathcal{U}) \quad \text{et} \quad \widehat{E} \xRightarrow[\mathcal{G}']{r'} \widehat{F}, \qquad r' = \varphi(r) \quad , \quad r = (\widehat{A}, \widehat{B}) \in \mathcal{R}$$

alors il existe $\widehat{G}$ et $\widehat{D}$ tels que :

$$\widehat{E} = \widehat{G}\widehat{A}\widehat{D} \qquad \text{et} \qquad \widehat{F} = \widehat{G}\widehat{B}\widehat{D}.$$

. Dans le 1er cas, $\varphi(r) = r$, c'est trivial.

. Dans le 2e cas, on vérifie que toute dérivation selon $\mathcal{G}'$ par $\varphi(r) = r_1 r_2$, partant d'un mot $\widehat{E}$ de $\mathfrak{L}(\mathcal{U})$ se présente sous la forme

$$\widehat{E} = \widehat{G}\widehat{A}\widehat{D} \xRightarrow[\mathcal{G}']{\widehat{G}B_1,\widehat{G},r_1r_2,\widehat{D},\widehat{D}} \widehat{F} = \widehat{G}\widehat{B}\widehat{D}$$

La règle $r_2 = (A'_r,\ B_2B_3\widehat{B}')$ ne peut en effet s'appliquer ni à $\widehat{G}$ ni à $\widehat{D}$ car ce sont là des mots de $\mathfrak{L}(\mathcal{U})$ alors que A'_r est un symbole nouveau.

. Dans le 3e cas, $\widehat{E}$ étant un mot de $\mathfrak{L}(\mathcal{U})$ toute $\widehat{E}$-dérivation de $\widehat{F}$ selon $\mathcal{G}'$ par $\varphi(r) = r'_1 r'_2$, se décompose de la façon suivante :

$$\widehat{E} = \widehat{G}A_1A_2\widehat{D} \xRightarrow[\mathcal{G}']{\widehat{G},r'_1,\widehat{D}} \widehat{G}B_1A''_r\widehat{D}$$

$$= \widehat{G}B_1\widehat{A}''_r\widehat{A}'\widehat{D}' \xRightarrow[\mathcal{G}']{\widehat{G}B_1,r'_2,\widehat{D}'} \widehat{G}B_1B_2B_3\widehat{B}'\widehat{D}' = \widehat{F}$$

car $\widehat{G}$ et $\widehat{D}$ mots de $\mathfrak{L}(\mathcal{U})$ ne peuvent contenir le symbole A''_r, $A''_r \in \mathcal{U}'$, d'où $\widehat{D} = \widehat{A}'\widehat{D}'$ et

$$\widehat{E} = \widehat{G}A_1A_2\widehat{A}'\widehat{D}' = \widehat{G}\widehat{A}\widehat{D}' \qquad \widehat{F} = \widehat{G}B_1B_2B_3\widehat{B}'\widehat{D}' = \widehat{G}\widehat{B}\widehat{D}', \quad \text{C. Q. F. D.}$$

— ($\mathcal{H}_3$) : Il s'agit de prouver que toute phrase $\widehat{X}$ de $\mathfrak{L}[\mathcal{G}']$ peut dériver de S par une suite de règles $\widehat{r'}$ telle que $\widehat{r'} \in \varphi(\mathfrak{L}(\mathcal{R}))$.

En effet $\widehat{X}$ étant une phrase de $\mathfrak{L}[\mathcal{G}']$ soit :

$$(S = \widehat{X}_0, \widehat{X}_1, \ldots, \widehat{X}_i, \ldots, \widehat{X}_n, \widehat{X})$$

une S-dérivation de $\widehat{X}$, selon $\mathcal{G}'$.

$\widehat{X}_{i+1}$ dérive directement de $\widehat{X}_i$ ($i \in (n)$) selon $\mathfrak{G}'$:

— par une règle r, commune à $\mathfrak{R}$ et $\mathfrak{R}'$,
— par une règle de type r_i ($i \in)2)$),
— ou par une règle de type r'_i ($i \in)2)$).

Remarquons, tout d'abord, que toute application de la règle r_2 (resp. r'_2) doit être précédée par celle de la règle r_1 (resp. r'_1). Réciproquement, si l'on applique la règle r_1 (resp. r'_1) il faudra aussi appliquer la règle r_2 (resp. r'_2) pour aboutir à une phrase qui est toujours un mot terminal.

De façon précise, étant donné une paire de règles (r_1, r_2) (resp. (r'_1, r'_2)) associée à une règle r de la grammaire $\mathfrak{G}$, on peut montrer que toute S-dérivation d'une phrase de $\mathfrak{L}[\mathfrak{G}']$ comportera l'application de r_1 et r_2 (resp. r'_1 et r'_2) un nombre égal de fois; et de plus à toute application de r_1 (resp. r'_1) sera associée *de manière unique*, une application ultérieure de la règle r_2 (resp. r'_2) où disparaît le symbole A'_r (resp. A''_r) introduit par l'application de r_1 (resp. r'_1). On va maintenant prouver que l'on peut donner de toute phrase de $\mathfrak{L}[\mathfrak{G}']$ une S-dérivation où des paires de règles associées sont *immédiatement* consécutives (r_2 ou r'_2 est appliquée immédiatement après r_1 ou r'_1; un A'_r ou un A''_r disparaît sitôt après qu'il a apparu...).

· Si $\quad \widehat{X}_i \xrightarrow[\mathfrak{G}']{\widehat{G},r_1,\widehat{D}} \widehat{X}_{i+1} \qquad r_1 = (A_1, B_1A'_r)$

nous pouvons distinguer dans la S-dérivation de $\widehat{X}$ les étapes suivantes :

$$\widehat{X}_i = \widehat{G}A_1\widehat{D} \xrightarrow[\mathfrak{G}']{\widehat{G},r_1,\widehat{D}} \widehat{X}_{i+1} = \widehat{G}B_1A'_r\widehat{D}$$
$$\widehat{s} \Big\Vert \mathfrak{G}'$$
$$\widehat{X}_{i+h} = \widehat{G}'A'_r\widehat{D}' \xrightarrow[\mathfrak{G}']{\widehat{G}',r_2,\widehat{D}'} \widehat{X}_{i+h+1} = \widehat{G}'B_2B_3\widehat{B}'\widehat{D}'$$

$h \geqslant 1$, si $h = 1$ on aura $\widehat{s} = \widehat{\emptyset}_{\mathfrak{R}'}$. Il est facile de constater que $\widehat{X}_{i+h+1}$ peut dériver de $\widehat{X}_i$, selon $\mathfrak{G}'$, par $r_1r_2\widehat{s} = \varphi(r)\widehat{s}$.

· Si $\quad \widehat{X}_i \xrightarrow{\widehat{G},r'_1,\widehat{D}} \widehat{X}_{i+1} \qquad r'_1 = (A_1A_2, B_1A''_r)$

nous avons alors :

$$\widehat{X}_i = \widehat{G}A_1A_2\widehat{D} \xrightarrow{\widehat{G},r'_1,\widehat{D}} \widehat{X}_{i+1} = \widehat{G}B_1A''_r\widehat{D}$$
$$\widehat{v} \Big\Vert \mathfrak{G}'$$
$$\widehat{X}_{i+h} = \widehat{G}'A''_r\widehat{A}'\widehat{D}' \xrightarrow{\widehat{G}',r'_2,\widehat{D}'} \widehat{X}_{i+h+1} = \widehat{G}'B_2B_3\widehat{B}'\widehat{D}'$$

$h \geqslant 1$, avec $\widehat{v} = \widehat{\emptyset}_{\mathfrak{R}'}$, si $h = 1$.

La règle r'_2 permettant de dériver $\widehat{X}_{i+h+1}$ de $\widehat{X}_{i+h}$ étant associée à la règle r'_1 dérivant $\widehat{X}_{i+1}$ de $\widehat{X}_i$, le symbole A''_r n'intervient pas dans les règles de la suite $\widehat{v}$ et il existe donc $\widehat{v}_1$ et $\widehat{v}_2$ de $\mathfrak{L}(\mathcal{R}')$ tels que :

$$\widehat{G}B_1 \underset{\mathfrak{G}'}{\overset{\widehat{v}_1}{\Longrightarrow}} \widehat{G}' \qquad \text{et} \qquad \widehat{D} \underset{\mathfrak{G}'}{\overset{\widehat{v}_2}{\Longrightarrow}} \widehat{A}'\widehat{D}'$$

On peut donc dériver $\widehat{X}_{i+h+1}$ de $\widehat{X}_i$, selon $\mathfrak{G}'$, par :

$$\widehat{v}_2 r'_1 r'_2 \widehat{v}_1 = \widehat{v}_2 \varphi(r) \widehat{v}_1.$$

Par de telles modifications dans l'ordre des dérivations directes, on peut toujours se ramener à une S-dérivation de $\widehat{X}$, selon $\mathfrak{G}'$, vérifiant l'hypothèse ($\mathcal{H}_3$). En effet, on vérifierait facilement que tout regroupement de règles r_1 et r_2, r'_1 et r'_2 reste acquis, ne disparaissant pas dans les modifications ultérieures.

Ainsi s'achève la démonstration du lemme 3.2.2.

3.2.3. **Grammaires normales.**

Étant donné une grammaire $\mathfrak{G}$, de type 1, nous pouvons d'après les lemmes précédents la supposer syntaxique et d'ordre 2. Il en résulte que toute règle de production est de l'un des types suivants :

$$(A_i,\ A_k)$$

$$(A_iA_j,\ A_kA_l)$$

ou :

$$(A_i,\ A_kA_l)$$

Toute règle terminale est d'ordre 1 et on peut démontrer que les règles non terminales de ce type, par exemple (A_i, A_k), peuvent être éliminées sans diminuer la capacité générative de la grammaire à condition d'ajouter à toute règle contenant $A_i^!$ dans son second membre la règle qui s'en déduit en remplaçant ce symbole A_i par A_k.

Nous appellerons grammaire normale de type 1 toute grammaire de type 1 où les réductions précédentes auront été faites; ses règles terminales seront donc toutes de la forme (A_i, A_k) et ses règles non terminales de l'une des deux formes (A_iA_j, A_kA_l), (A_i, A_kA_l)... Dans le cas particulier où la grammaire est C F les lemmes précédents s'appliquent toujours (les démonstrations étant quelque peu simplifiées) et nous obtenons une grammaire normale au sens de Chomsky [3] où toutes les règles non terminales sont de la forme (A_i, A_kA_l). Nous résumons ces résultats dans le scholie suivant :

Scholie. — **Toute grammaire de type 1 (resp. C F) est équivalente à une grammaire normale de type 1 (resp. normale C F).**

On pourrait maintenant démontrer la proposition fondamentale (§ 4) en se basant sur le lemme précédent. Mais auparavant nous allons établir un autre résultat qui nous servira dans l'étude des propriétés de clôture (§ 6 et 7).

3.3. Grammaires préservant la longueur. Grammaires linéaires bornées.

3.3.1. Définitions.

— On dira qu'une grammaire de type 1, *préserve la longueur*, si pour toute règle de $\mathfrak{G}$, $r = (\widehat{A}, \widehat{B})$:

. $\widehat{A}$ est le symbole initial,

. ou $\widehat{B}$ ne contient pas le symbole initial et $\widehat{A}$ et $\widehat{B}$ ont même longueur, i. e. :

$\forall r \in \mathfrak{R}$, $r = (\widehat{A}, \widehat{B})$: $\widehat{A} \neq S$ implique $|\widehat{A}| = |\widehat{B}|$ et $\widehat{B}$ ne contient pas S.

Si donc $\mathfrak{G}$ est une grammaire de type 1, normale (lemme 3.2.3) préservant la longueur : les règles de production seront de l'un des types suivants :

$$(A_1, B_1) \quad ; \quad (A_1A_2, B_1B_2) \quad ; \quad (S, C_1C_2)$$

où B_1 et B_2 sont différents de S.

Dans une dérivation, selon une telle grammaire $\mathfrak{G}$, la longueur des mots est conservée par application des règles, sauf celles du type (S, C_1C_2).

— Une grammaire de type 1 sera dite *linéaire bornée* si elle est d'ordre 2, préserve la longueur et si de plus pour toute règle du type (S, C_1C_2) :

$$C_1 = S, \qquad C_2 \neq S.$$

3.3.2. Lemme.

Toute grammaire de type 1 est équivalente à une grammaire de type 1 linéaire bornée.

DÉMONSTRATION. — D'après les résultats précédents nous pouvons supposer que la grammaire de type 1 en question

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

est normale. Il en résulte que toute règle terminale est du type

$$r_1 = (A_i, A_k)$$

et toute règle non terminale de l'un des deux types :

$$r_2 = (A_iA_j, A_kA_l)$$

$$r_3 = (A_i, A_kA_l)$$

On pourra supposer que $S = A_0$.

Soit la grammaire $\mathfrak{G}'$ définie comme suit :

$$\mathfrak{G}' = (\mathfrak{V}', \mathfrak{V}_T, S', \mathcal{R}')$$

— $\mathfrak{V}' = \mathfrak{V} \cup \{ S', W \}$ S' et W étant deux symboles nouveaux.
— S' est le symbole initial de $\mathfrak{G}'$.
— $\mathcal{R}'$ comprend :

— des règles spéciales : $s' = (S', S'W)$

$$s = (S', S)$$

$$\forall A_i \in \mathfrak{V}_N \quad : \quad w_i = (A_iW, WA_i)$$

— et des règles correspondant à celles de $\mathcal{R}$:

. toute règle de $\mathfrak{G}$ de type r_1 ou r_2 sera aussi une règle de $\mathfrak{G}'$. On aura donc

$$\mathcal{R}'_T = \mathcal{R}_T$$

. toute règle de $\mathfrak{G}$ de type r_3, $r_3 = (A_i, A_kA_l)$, sera remplacée, dans $\mathfrak{G}'$, par la règle de type r'_3, $r'_3 = (A_iW, A_kA_l)$.

Il est facile de constater que $\mathfrak{G}'$ est bien une grammaire linéaire bornée (on se rappellera que son symbole initial est S'). Il s'agit de montrer qu'elle est équivalente à $\mathfrak{G}$, i. e. :

$$\mathfrak{L}[\mathfrak{G}] = \mathfrak{L}[\mathfrak{G}'].$$

1° $\mathfrak{L}[\mathfrak{G}] \subset \mathfrak{L}[\mathfrak{G}']$.

$\widehat{X}$ étant une phrase de $\mathfrak{L}[\mathfrak{G}]$, soit :

$$(S = \widehat{X}_0, \widehat{X}_1, \ldots, \widehat{X}_i, \ldots, \widehat{X}', \ldots, \widehat{X})$$

une S-dérivation de $\widehat{X}$ selon $\mathfrak{G}$, où $\widehat{X}$ dérive de $\widehat{X}'$ par une suite, $\widehat{t}$, de règles terminales communes à $\mathcal{R}$ et $\mathcal{R}'$ (dans une grammaire syntaxique les règles terminales peuvent être appliquées en dernier lieu : § 3.1.2). Pour que $\widehat{X}$

puisse aussi être considéré comme une phrase de $\mathfrak{L}[\mathfrak{G}']$ il nous reste donc à prouver que $\widehat{X}'$ peut dériver de S' selon $\mathfrak{G}'$.

Soit k le nombre de fois où des règles de type r_3 sont appliquées dans la dérivation :

$$S \underset{\mathfrak{G}'}{\Longrightarrow} \widehat{X}'$$

nous en déduisons la S'-dérivation de $\widehat{X}'$ selon $\mathfrak{G}'$:

$$S' \underset{\mathfrak{G}'}{\overset{s'^k}{\Longrightarrow}} S'W^k \underset{\mathfrak{G}'}{\overset{s}{\longrightarrow}} SW^k \Longrightarrow \cdots \Longrightarrow \widehat{X}_i W^h \Longrightarrow \cdots \Longrightarrow \widehat{X}'.$$

$h = |\widehat{X}| - |\widehat{X}_i|$, bien entendu si $h = 0$ on a $W^0 = \widehat{\emptyset}$. Donc si :

$$\widehat{X}_i \underset{\mathfrak{G}}{\overset{r_3}{\longrightarrow}} \widehat{X}_{i+1}$$

alors :

$$\widehat{X}_i W^h \underset{\mathfrak{G}'}{\overset{r_3}{\longrightarrow}} \widehat{X}_{i+1} W^h$$

Mais si :

$$\widehat{X}_i \underset{\mathfrak{G}}{\overset{\widehat{G}, r_3, \widehat{D}}{\longrightarrow}} \widehat{X}_{i+1} \quad , \quad r_3 = (A_i, A_k A_l)$$

avec, par exemple :

$$\widehat{D} = A_{i_1} \cdots A_{i_j} \cdots A_{i_m} \quad , \quad \forall j \in) \, m) : A_{i_j} \in \mathfrak{V}_N$$

alors :

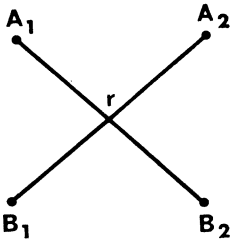
$$\widehat{X}_i W^h \underset{\mathfrak{G}'}{\overset{w_{i_m} \cdots w_{i_j} \cdots w_{i_1} r'_3}{\Longrightarrow}} \widehat{X}_{i+1} W^{h-1}$$

Des règles de type r'_3 intervenant k fois, tous les symboles W disparaissent et $\widehat{X}'$ dérive donc de S' selon $\mathfrak{G}'$. C. Q. F. D.

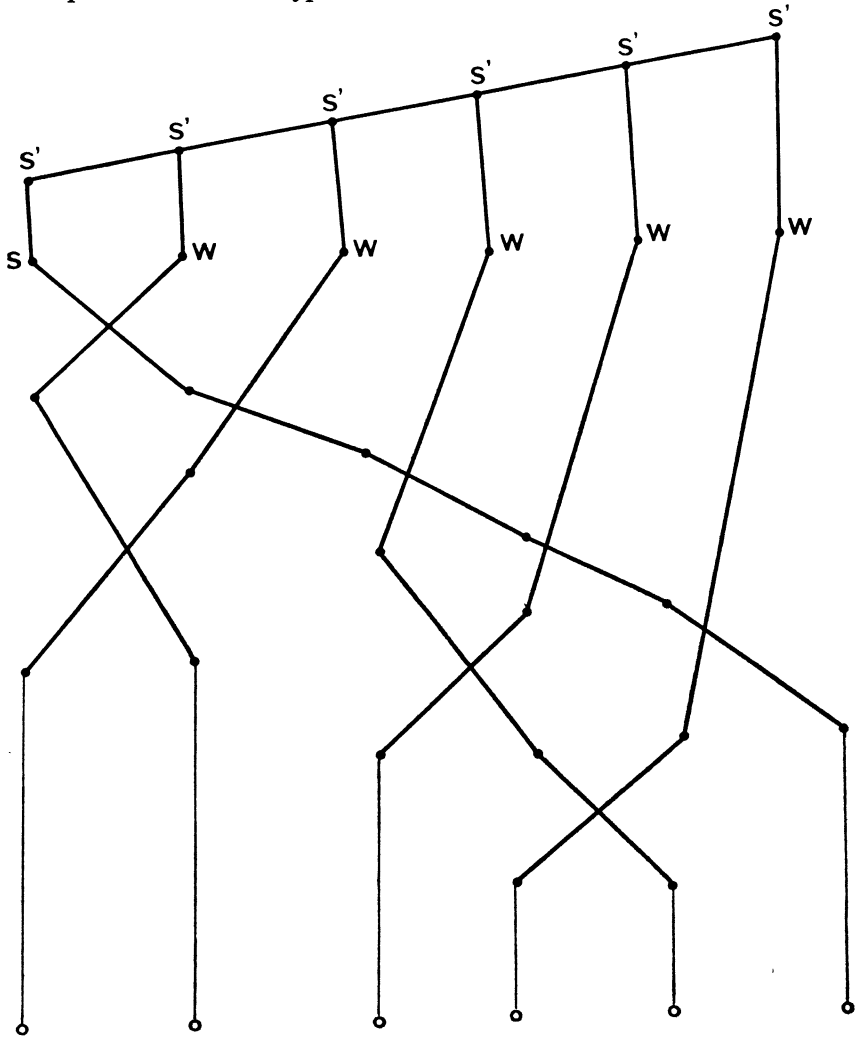
Remarque. Notations. — Il est intéressant de noter que dans une grammaire linéaire bornée, les mots d'une dérivation ne s'allongent que par application de la règle $s' = (S', S'W)$. Ainsi dans le cas ci-dessus où cette règle est appliquée k fois : $|\widehat{X}| = k + 1$. D'autre part, avant l'application de la règle $s = (S', S)$ tout mot de la dérivation commence à gauche par le symbole initial S' et ce symbole ne peut occuper que cette position dans l'écriture d'un mot d'une S'-dérivation. En convenant de représenter la règle :

$$r = (A_1 A_2, B_1 B_2)$$

sous la forme :



une S'-dérivation selon la grammaire $\mathfrak{G}'$ précédente, linéaire bornée, sera représentée non par un arbre (comme dans le cas d'une grammaire C F) mais par un schéma du type suivant :



Les traits plus fins correspondent à l'application de règles terminales.

2° $\mathfrak{L}[\mathfrak{G}'] \subset \mathfrak{L}[\mathfrak{G}]$.

Cela résulte de ce que l'homomorphisme φ de $\mathfrak{L}(\mathfrak{V}')$ dans $\mathfrak{L}(\mathfrak{V})$ tel que :

. $\varphi(S') = S$,

. $\varphi(W) = \widehat{\emptyset}$,

· et $\forall A_i \in \mathfrak{V} \quad \varphi(A_i) = A_i$

est un homomorphisme de grammaires, ce que l'on vérifiera facilement.

La définition et le lemme sur les grammaires syntaxiques (§ 3.1) étaient appliqués aux grammaires de type 1 et de type 2. Les définitions et les lemmes concernant les grammaires d'ordre 2 (§ 3.2) et les grammaires linéaires bornées (§ 3.3) s'appliquaient aux grammaires de type 1. Ils s'étendront également aux grammaires de type 2 par la démonstration de la proposition fondamentale.

4. PROPOSITION FONDAMENTALE

ÉNONCÉ. — **L'ensemble des langages de type 1 est identique à l'ensemble des langages de type 2, i. e. :**

$$\mathfrak{L}\{\alpha_1\} = \mathfrak{L}\{\alpha_2\}$$

(cf. définitions des § 1.3 et 1.5).

DÉMONSTRATION. — 1° $\mathfrak{L}\{\alpha_2\} \subset \mathfrak{L}\{\alpha_1\}$, c'est-à-dire : tout langage de type 2 est un langage de type 1.

Cela résulte immédiatement des définitions, toute grammaire de type 2 étant de type 1.

2° $\mathfrak{L}\{\alpha_1\} \subset \mathfrak{L}\{\alpha_2\}$, c'est-à-dire : tout langage de type 1 est un langage de type 2.

Soit $\mathfrak{L}$ un langage de type 1, $\mathfrak{L} \in \mathfrak{L}\{\alpha_1\}$; il existe par définition, une grammaire $\mathfrak{G}$:

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

de type 1, engendrant $\mathfrak{L}$, $\mathfrak{L} = \mathfrak{L}[\mathfrak{G}]$.

En vertu des lemmes précédents, nous pouvons supposer que la gram-

maire $\mathfrak{G}$ est syntaxique (§ 3.1) et d'ordre 2 (§ 3.2). Il en résulte que toute règle de production de $\mathfrak{G}$ est de l'un des types suivants :

$$r_1 = (A_i, A_k)$$
$$r_2 = (A_iA_j, A_kA_l)$$
$$r_3 = (A_i, A_kA_l)$$

Nous allons construire une grammaire $\mathfrak{G}'$, de type 2, équivalente à la grammaire $\mathfrak{G}$:

$$\mathfrak{G}' = (\mathfrak{V}', \mathfrak{V}_T, S, \mathfrak{R}')$$

avec $\mathfrak{V} \subset \mathfrak{V}'$, le symbole initial étant conservé. Les règles de $\mathfrak{G}$, de type r_1 ou de type r_3, satisfaisant à l'axiome (α_2) seront conservées. Et pour toute règle de type r_2, on introduira deux symboles nouveaux, à chaque fois différents, et dans $\mathfrak{G}'$, la règle r_2 sera remplacée par une suite de quatre règles, respectant l'axiome (α_2), les deux premières introduisant chacune un symbole nouveau, les deux autres les faisant disparaître. D'autre part, on définira une application φ de $\mathfrak{R}$ dans $\mathfrak{L}(\mathfrak{R}')$ permettant d'appliquer le critère d'équivalence du paragraphe 2.1.2. Plus précisément :

$$\forall r \in \mathfrak{R} \quad , \quad r = (\widehat{A}, \widehat{B}) \quad :$$

. si r est du type r_1 ou du type r_3, alors : $\varphi(r) = r \in \mathfrak{R}'$,

. mais si r est du type r_2, par exemple $r_2 = (A_iA_j, A_kA_l)$ alors $\varphi(r_2) = s_1s_2s_3s_4$ avec :

$$s_1 = (A_iA_j, A'_iA_j) ; \qquad s_2 = (A'_iA_j, A'_iA'_j) ;$$
$$s_3 = (A'_iA'_j, A_kA'_j) ; \qquad s_4 = (A_kA'_j, A_kA_l)$$

A'_i, $A'_j \notin \mathfrak{V}$: ces symboles interviennent uniquement dans l'écriture des règles s_h, $h \in)4)$.

Il est facile de constater que la grammaire $\mathfrak{G}'$ est bien de type 2. Prouvons qu'elle est équivalente à la grammaire $\mathfrak{G}$ en démontrant que les hypothèses du § 2.1.2 sont satisfaites.

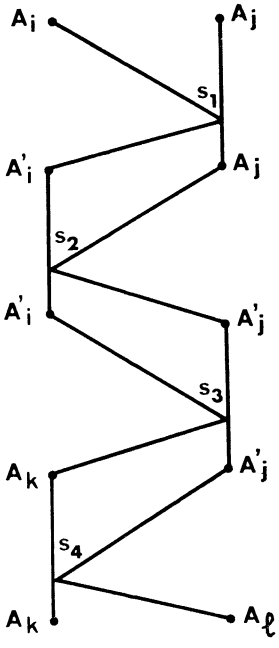
— $(\mathcal{H}_1)$: C'est immédiat, d'après la construction des règles de $\mathfrak{G}'$

$$\forall r \quad , \quad r = (\widehat{A}, \widehat{B}) \quad : \quad \widehat{A} \overset{\varphi(r)}{\underset{\mathfrak{G}'}{\Longrightarrow}} \widehat{B}$$

Ainsi dans le cas où $r = (A_iA_j, A_kA_l)$ nous avons :

$$A_iA_j \overset{\widehat{\varnothing},\widehat{\varnothing},\widehat{\varnothing},\widehat{\varnothing},s_1s_2s_3s_4,\widehat{\varnothing},\widehat{\varnothing},\widehat{\varnothing},\widehat{\varnothing}}{\underset{\mathfrak{G}'}{\Longrightarrow}} A_kA_l$$

En utilisant les notations du § 3.3.2, cette dérivation se représente comme suit :



— ($\mathcal{H}_2$) : Dans le cas où r est de type r_1 ou r_3, c'est trivial, car $\varphi(r) = r$.

. Dans le cas où r est de type r_2, par exemple $r_2 = (A_iA_j, A_kA_l)$

$\widehat{E}$ étant un mot de $\mathfrak{L}(\mathfrak{V})$ toute $\widehat{E}$-dérivation de $\widehat{F}$, selon $\mathfrak{G}'$, par $\varphi(r_2) = s_1s_2s_3s_4$ se présente sous la forme suivante :

$$\widehat{E} = \widehat{G}A_iA_j\widehat{D} \xRightarrow[\mathfrak{G}']{\widehat{G},\widehat{G},\widehat{G},\widehat{G},s_1s_2s_3s_4,\widehat{D},\widehat{D},\widehat{D},\widehat{D}} \widehat{F} = \widehat{G}A_kA_l\widehat{D}$$

les règles s_2, s_3, s_4 ne pouvant s'appliquer ni à $\widehat{G}$ ni à $\widehat{D}$, car ce sont là des mots de $\mathfrak{L}(\mathfrak{V})$ qui ne contiennent donc aucun des deux symboles A'_i et A'_j.

— ($\mathcal{H}_3$) : $\widehat{X}$ étant une phrase de $\mathfrak{L}[\mathfrak{G}']$ soit :

$$(S = \widehat{X}_0, \widehat{X}_1, \ldots, \widehat{X}_m, \ldots, \widehat{X}_n, \widehat{X})$$

une S-dérivation de $\widehat{X}$ selon $\mathfrak{G}'$.

Toute dérivation directe s'opère soit par une règle de production r, de type r_1 ou r_3, commune à $\mathcal{R}$ et $\mathcal{R}'$, soit par une règle spéciale à $\mathcal{R}'$, du

type s_h ($h \in)4$)). Dans ce dernier cas, étant donné leur nature, il est évident que les trois autres règles s_j interviennent, leur ordre d'application devant de plus être respecté... Supposons par exemple que :

$$\widehat{X}_m \underset{\mathfrak{G}'}{\overset{s_1}{\Longrightarrow}} \widehat{X}_{m+1}$$

les dérivations directes précédentes n'appliquant que des règles de type r_1 ou r_3. Nous pouvons toujours distinguer les étapes suivantes dans notre S-dérivation :

$$\widehat{X}_m = \widehat{G}_1 A_i A_j \widehat{D}_1$$

$$\downarrow s_1$$

$$\widehat{X}_{m+1} = \widehat{G}_1 A'_i A_j \widehat{D}_1 \underset{\mathfrak{G}'}{\overset{\widehat{a}_1}{\Longrightarrow}} \widehat{G}_2 A'_i A_j \widehat{D}_2 = \widehat{X}_{m+h_1}$$

$$\downarrow s_2$$

$$\widehat{X}_{m+h_1+1} = \widehat{G}_2 A'_i A'_j \widehat{D}_2 \underset{\mathfrak{G}'}{\overset{\widehat{a}_2}{\Longrightarrow}} \widehat{G}_3 A'_i A'_j \widehat{D}_3 = \widehat{X}_{m+h_2}$$

$$\downarrow s_3$$

$$\widehat{X}_{m+h_2+1} = \widehat{G}_3 A_k A'_j \widehat{D}_3 \underset{\mathfrak{G}'}{\overset{\widehat{a}_3}{\Longrightarrow}} \widehat{G}_4 A_k A'_j \widehat{D}_4 = \widehat{X}_{m+h_3}$$

$$\downarrow s_4$$

$$\widehat{X}_{m+h^3+1} = \widehat{G}_4 A_k A_l \widehat{D}_4$$

avec $1 \leqslant h_1 < h_2 < h_3$; les suites de règles $\widehat{a}_1$, $\widehat{a}_2$, $\widehat{a}_3$ pouvant éventuellement être réduites à $\widehat{\varnothing}_{\mathfrak{R}'}$.

Il est important de signaler que (comme dans la démonstration du lemme 3.2.2) le symbole A'_i introduit par s_1, dans l'écriture du mot $\widehat{X}_{m+1}$, est celui qui intervient dans le premier membre de la règle s_2 introduisant le symbole A'_j, et que ce sont ces symboles qui disparaissent par application des règles s_3 et s_4, mises en évidence ci-dessus. D'autres règles s_h ($h \in)4$)) pourront donc exister dans les suites de règles $\widehat{a}_1$, $\widehat{a}_2$ ou $\widehat{a}_3$, mais elles concerneront d'autres occurrences ou disparitions des symboles A'_i et A'_j. Il en résulte que la dérivation décrite ci-dessus :

$$\widehat{X}_m = \widehat{G}_1 A_i A_j \widehat{D}_1 \underset{\mathfrak{G}'}{\overset{s_1\widehat{a}_1 s_2 \widehat{a}_2 s_3 \widehat{a}_3 s_4}{\Longrightarrow}} \widehat{G}_4 A_k A_l \widehat{D}_4 = \widehat{X}_{m+h_3+1}$$

peut se mettre sous la forme suivante, l'ordre des dérivations directes étant modifié :

$$\widehat{X}_m = \widehat{Y}_m = \widehat{G}_1 A_i A_j \widehat{D}_1 \xrightarrow[\mathfrak{G}']{\widehat{a_1}\widehat{a_2}} \widehat{Y}_{m+h_2-2} = \widehat{G}_3 A_i A_j \widehat{D}_3 \xrightarrow[\mathfrak{G}']{\varphi(r_3)} \widehat{Y}_{m+h_3+2}$$
$$= \widehat{G}_3 A_k A_l \widehat{D}_3 \xrightarrow[\mathfrak{G}']{\widehat{a_3}} \widehat{Y}_{m+h_3+1} = \widehat{X}_{m+h_3+1} = \widehat{G}_4 A_k A_l \widehat{D}_4$$

Soit :

$$(S = \widehat{Y}_0, \widehat{Y}_1, \ldots, \widehat{Y}_p, \ldots, \widehat{Y}_n, \widehat{X})$$

la S-dérivation de $\widehat{X}$ résultant de cette modification. Nous avons :

$$\forall p \in) m) \quad : \qquad \widehat{Y}_p = \widehat{X}_p$$

Soit m' le plus petit indice (s'il existe) pour lequel :

$$\widehat{Y}_{m'} \xrightarrow[\mathfrak{G}']{s'_1} \widehat{Y}_{m'+1} \quad , \quad m' > m + h_2 - 2,$$

s'_1 étant une des règles spéciales à $\mathfrak{G}'$ introduisant un symbole nouveau spécial à $\mathfrak{V}'$; éventuellement $s'_1 = s_1$, mais il s'agit alors d'une application différente de celle étudiée ci-dessus ($m' > m + h_2 - 2$). Nous pouvons encore considérer les étapes suivantes :

$$\widehat{Y}_{m'} \xrightarrow[\mathfrak{G}']{s'_1} \widehat{Y}_{m'+1} \xrightarrow[\mathfrak{G}']{\widehat{a'_1}} \widehat{Y}_{m'+h'_1} \xrightarrow[\mathfrak{G}]{s'_2} \widehat{Y}_{m'+h'_1+1} \Longrightarrow$$
$$\ldots \xrightarrow[\mathfrak{G}']{\widehat{a'_3}} \widehat{Y}_{m'+h'_3} \xrightarrow[\mathfrak{G}']{s'_4} \widehat{Y}_{m'+h'_3+1}$$

dont l'ordre pourra être modifié pour donner la dérivation :

$$\widehat{Y}_{m'} \xrightarrow[\mathfrak{G}']{\widehat{a'_1}\widehat{a'_3}\varphi(r'_2)\widehat{a'_3}} \widehat{Y}_{m'+h'_3+1} \quad , \quad \varphi(r'_2) = s'_1 s'_2 s'_3 s'_4$$

Il est essentiel de remarquer que le regroupement des règles s_1, s_2, s_3, s_4, réalisé précédemment, est conservé. Ainsi au bout d'un nombre fini de telles modifications nous aboutissons à une S-dérivation de $\widehat{X}$, selon $\mathfrak{G}'$, satisfaisant à l'hypothèse ($\mathcal{H}_3$).

Ainsi s'achève la démonstration de la proposition fondamentale.

CONCLUSION. — *Langages C S.* — Dans la littérature, un langage C S (pour « context-sensitive ») est une partie $\mathfrak{L}$ du monoïde libre $\mathfrak{L}(\mathfrak{V}_T)$ telle que :

$$\mathfrak{L} \in \mathfrak{L}\{\alpha_2\}.$$

Compte tenu de la proposition fondamentale de ce paragraphe 4, à tout langage C S on pourra aussi associer une grammaire de type 1 l'engendrant. Par abus de langage, nous dirons que les grammaires de type 1 ou de type 2 sont des grammaires C S. D'après les lemmes du paragraphe 3, ces grammaires C S peuvent être supposées normales, ou même linéaires bornées. Cela va nous être utile pour établir certaines propriétés des langages C S.

5. GRAMMAIRES C S AVEC MARQUANT [6]

5.1. Définitions.

— Une grammaire C S avec marquant :

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

est définie par les données suivantes :

. L'ensemble $\mathfrak{V}_N$ des symboles non terminaux contient un deuxième élément distingué, représenté, exceptionnellement, non par une capitale latine, mais par $\#$, et appelé marquant de fin de phrase, ou simplement marquant.

. Les règles de production, $r = (\widehat{A}, \widehat{B})$, de la grammaire $\mathfrak{G}$ satisfont aux conditions suivantes :

a) $\widehat{A}$ contient au moins un symbole (non terminal) distinct du marquant $\#$,

b) les règles sont de l'un des quatre types suivants (où $\widehat{A}$ et $\widehat{B}$ désignent des mots de $\mathfrak{L}(\mathfrak{V})$ ne contenant pas le marquant $\#$) :

$$r = (\widehat{A}, \widehat{B}) \quad ; \quad \# r = (\# \widehat{A}, \# \widehat{B})$$

$$r \# = (\widehat{A} \#, \widehat{B} \#) \quad ; \quad \# r \# = (\# \widehat{A} \#, \# \widehat{B} \#)$$

— Le langage engendré par une grammaire C S avec marquant, $\mathfrak{G}$, est l'ensemble des mots terminaux $\widehat{X}$ tels que : $\# \widehat{X} \#$ dérivent de $\# S \#$ selon $\mathfrak{G}$, i. e. :

$$\mathfrak{L}[\mathfrak{G}] = \left\{ \widehat{X} \mid \widehat{X} \in \mathfrak{L}(\mathfrak{V}_T) \quad ; \quad \exists \widehat{r} \in \mathfrak{L}(\mathfrak{R}) \quad : \quad \# S \# \underset{\mathfrak{G}}{\overset{\widehat{r}}{\Longrightarrow}} \# \widehat{X} \# \right\}$$

5.2. Proposition.

Toute grammaire C S avec marquant est équivalente à une grammaire C S sans marquant.

DÉMONSTRATION. — Soit $\mathfrak{G}$ une grammaire C S avec marquant :

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \#, \mathfrak{R})$$

et considérons $\mathfrak{G}'$ la grammaire C S sans marquant définie de la façon suivante :

. $\mathfrak{V}'_N = (\mathfrak{V}_N \cup \mathfrak{V}^{\#}) - \{\#\}$, où :

$$\mathfrak{V}^{\#} = \left\{ {}^{\#}A_i, A_i{}^{\#}, {}^{\#}A_i{}^{\#} \quad \middle| \quad A_i \in (\mathfrak{V} - \{\#\}) \right\};$$

$\mathfrak{V}^{\#}$ et $\mathfrak{V}_N$ n'ayant aucun symbole en commun, i. e. :

$$\mathfrak{V}^{\#} \cap \mathfrak{V}_N = \emptyset$$

(i. e. on associe à chaque symbole de $\mathfrak{V}$, distinct du marquant, trois nouveaux symboles non terminaux, indicés respectivement à gauche, à droite ou des deux côtés par $\#$).

. $S' = {}^{\#}S^{\#}$ est le symbole initial de $\mathfrak{G}'$.

Si $\widehat{A} = A_{i_1} \dots A_{i_r}$ est un mot de $\mathfrak{L}(\mathfrak{V})$ on posera :

$${}^{\#}\widehat{A} = A_{i_1} \dots A_{i_{r-1}} A_{i_r}{}^{\#}$$

$${}^{\#}\widehat{A} = {}^{\#}A_{i_1} A_{i_2} \dots A_{i_r}$$

$${}^{\#}\widehat{A}{}^{\#} = {}^{\#}A_{i_1} A_{i_2} \dots A_{i_{r-1}} A_{i_r}{}^{\#}$$

${}^{\#}\widehat{A}$, $\widehat{A}{}^{\#}$ et ${}^{\#}\widehat{A}{}^{\#}$ étant des mots de $\mathfrak{L}(\mathfrak{V}')$.

De même si $r = (\widehat{A}, \widehat{B})$ est une règle de $\mathfrak{G}$ on posera :

$${}^{\#}r = ({}^{\#}\widehat{A}, {}^{\#}\widehat{B})$$

$$r^{\#} = (\widehat{A}{}^{\#}, B^{\#})$$

$${}^{\#}r^{\#} = ({}^{\#}\widehat{A}{}^{\#}, {}^{\#}\widehat{B}{}^{\#})$$

. L'ensemble $\mathcal{R}'$ est alors défini comme suit. Il comporte :

— des règles de production correspondant à celles de $\mathcal{R}$:

à r, règle de $\mathfrak{G}$ correspond dans $\mathfrak{G}'$: r, ${}^{\#}r$, $r^{\#}$, ${}^{\#}r^{\#}$;

de même si $\quad \# r \in \mathcal{R}$ alors ${}^{\#}r \in \mathcal{R}'$ et ${}^{\#}r^{\#} \in \mathcal{R}'$

si $\quad r \# \in \mathcal{R}$ alors $r^{\#} \in \mathcal{R}'$ et ${}^{\#}r^{\#} \in \mathcal{R}'$

enfin si $\# r \# \in \mathcal{R}$ alors ${}^{\#}r^{\#} \in \mathcal{R}'$;

— des règles de production terminales spéciales à $\mathfrak{G}'$:

$\forall A_i \in \mathcal{U}_T$: $\quad {}^{\#}t_i = ({}^{\#}A_i, A_i)$,

$t_i{}^{\#} = (A_i{}^{\#}, A_i)$,

et $\quad {}^{\#}t_i{}^{\#} = ({}^{\#}A_i{}^{\#}, A_i)$,

sont des règles de $\mathfrak{G}'$.

Il s'agit de prouver que

$$\mathfrak{L}[\mathfrak{G}'] = \mathfrak{L}[\mathfrak{G}];$$

c'est-à-dire que, quel que soit $\widehat{X} \in \mathfrak{L}(\mathcal{U}_T)$:

$$\# S \# \underset{\mathfrak{G}}{\Longrightarrow} \# \widehat{X} \# \text{ si et seulement si } {}^{\#}S^{\#} = S' \underset{\mathfrak{G}'}{\Longrightarrow} \widehat{X}.$$

Nous allons transformer une $\# S \#$-dérivation selon $\mathfrak{G}$ en une S'-dérivation selon $\mathfrak{G}'$ et réciproquement, essentiellement en identifiant $\# \widehat{A}, \widehat{A} \#$ et $\# \widehat{A} \#$ avec ${}^{\#}\widehat{A}, \widehat{A}^{\#}$ et ${}^{\#}\widehat{A}^{\#}$ respectivement.

1° $\mathfrak{L}[\mathfrak{G}] \subset \mathfrak{L}[\mathfrak{G}']$.

$\widehat{X}$ étant une phrase de $\mathfrak{L}[\mathfrak{G}]$, il existe, par définition, une $\# S \#$-dérivation de $\# \widehat{X} \#$ selon $\mathfrak{G}$:

$$(\# S \# = \# \widehat{X}_0 \#, \# \widehat{X}_1 \#, \ldots, \# \widehat{X}_i \#, \ldots, \# \widehat{X}_n \# = \# \widehat{X} \#)$$

où pour tout i, $i \in)n)$, $\widehat{X}_i$ ne contient pas le marquant $\#$, étant donné la nature des règles d'une grammaire C S avec marquant. Nous en déduisons la S'-dérivation de $\widehat{X}$, selon $\mathfrak{G}'$:

$$({}^{\#}S^{\#} = S', {}^{\#}\widehat{X}_1{}^{\#}, \ldots, {}^{\#}\widehat{X}_i{}^{\#}, \ldots, {}^{\#}\widehat{X}_n{}^{\#} = {}^{\#}\widehat{X}^{\#}, \widehat{X}^{\#}, \widehat{X}).$$

Si $\# \widehat{X}_i \# = \# \widehat{G}\widehat{A}\widehat{D} \#$ et $\# \widehat{X}_{i+1} \# = \# \widehat{G}\widehat{B}\widehat{D} \#$, le tableau ci-dessous nous montre que quelle que soit la règle de $\mathfrak{G}$ permettant de dériver $\# X_{i+1} \#$

de $\# \widehat{X}_i \#$ (la nature de cette règle dépend en partie de $\widehat{G}$ et $\widehat{D}$, égaux ou non à $\widehat{\emptyset}$), il existe toujours dans $\mathfrak{G}'$, une règle permettant de dériver :

$$\#\widehat{X}_{i+1}\# \quad \text{de} \quad \#\widehat{X}_i\#.$$

$$\# \widehat{X}_i \# = \# \widehat{G}\widehat{A}\widehat{D} \# ; \qquad \#\widehat{X}_i\# = \#\widehat{G}\widehat{A}\widehat{D}\#$$

$$\# \widehat{X}_{i+1} \# = \# \widehat{G}\widehat{B}\widehat{D} \# ; \qquad \#\widehat{X}_{i+1}\# = \#\widehat{G}\widehat{B}\widehat{D}\#$$

$\widehat{G}$	$\widehat{D}$	Règles de $\mathfrak{G}$ susceptibles de servir à $\#\widehat{X}_i\# \xrightarrow[\mathfrak{G}]{} \#\widehat{X}_{i+1}\#$	Règles de $\mathfrak{G}'$ susceptibles de servir à $\#\widehat{X}_i\# \xrightarrow[\mathfrak{G}']{} \#\widehat{X}_{i+1}\#$
$\neq \widehat{\emptyset}$	$\neq \widehat{\emptyset}$	$r = (\widehat{A}, \widehat{B})$	r
$= \widehat{\emptyset}$	$\neq \widehat{\emptyset}$	r ou $\# r$	$\#_r$
$\neq \widehat{\emptyset}$	$= \widehat{\emptyset}$	r ou $r \#$	$r^{\#}$
$= \widehat{\emptyset}$	$= \widehat{\emptyset}$	r, $\# r$, $r \#$ ou $\# r \#$	$\#_r\#$

On a donc :

$$\#S\# = S' \underset{\mathfrak{G}'}{\Longrightarrow} \#\widehat{X}\#$$

et comme $\widehat{X}$ est un mot terminal, par exemple :

$$X = A_{j_1} \ldots A_{j_k} \ldots A_{j_m} \quad , \quad \forall k, k \in) m) \ : \ A_{j_k} \in \mathfrak{V}_T$$

nous avons :

$$\#\widehat{X}\# \xrightarrow[\mathfrak{G}']{\#_{t_{j_1}}} \widehat{X}\# \xrightarrow[\mathfrak{G}']{t_{jm}\#} \widehat{X}.$$

Évidemment dans les cas particuliers où $|\widehat{X}| = 1$, par exemple $\widehat{X} = A_j \in \mathfrak{V}_T$, on aurait :

$$\#\widehat{X}\# \xrightarrow[\mathfrak{G}']{\#_{t_j}\#} \widehat{X}$$

Dans tous les cas nous avons prouvé que $\widehat{X}$ phrase de $\mathfrak{L}[\mathfrak{G}]$, est aussi une phrase de $\mathfrak{L}[\mathfrak{G}']$.

2° $\mathfrak{L}[\mathfrak{G}'] \subset \mathfrak{L}[\mathfrak{G}]$.

$\widehat{X}$ étant une phrase de $\mathfrak{L}[\mathfrak{G}']$ soit :

$$({}^{\#}S^{\#} = S', {}^{\#}\widehat{X}_1{}^{\#}, \ldots, {}^{\#}\widehat{X}_i{}^{\#}, \ldots, {}^{\#}\widehat{X}_n{}^{\#} = {}^{\#}\widehat{X}^{\#}, \widehat{X}^{\#}, \widehat{X})$$

une S'-dérivation de $\widehat{X}$ selon $\mathfrak{G}'$, où les règles terminales spéciales à $\mathfrak{G}'$ $({}^{\#}t_i, t_i{}^{\#}, {}^{\#}t_i{}^{\#})$ s'appliquent en fin de dérivation. Cela est toujours possible, car si nous avons, par exemple :

$${}^{\#}\widehat{X}_i{}^{\#} \xrightarrow[\mathfrak{G}']{} \widehat{X}_i{}^{\#} \xRightarrow[\mathfrak{G}']{} \widehat{X}^{\#} \xrightarrow[\mathfrak{G}']{} \widehat{X}$$

alors $\widehat{X}^{\#}$ dérive de $\widehat{X}_i{}^{\#}$ par une suite de règles de la forme r ou $r^{\#}$. Mais on peut donc aussi dériver ${}^{\#}\widehat{X}^{\#}$ de ${}^{\#}X_i{}^{\#}$, selon $\mathfrak{G}'$, car si r (resp. $r^{\#}$) est une règle de $\mathfrak{G}'$ il en est de même de ${}^{\#}r$ (resp. ${}^{\#}r^{\#}$).

Dans la S'-dérivation de $\widehat{X}$, indiquée ci-dessus, ${}^{\#}\widehat{X}_{i+1}{}^{\#}$ dérive directement de ${}^{\#}\widehat{X}_i{}^{\#}$ par application d'une règle de $\mathfrak{G}'$ correspondant (par définition de $\mathfrak{R}'$) à une règle de $\mathfrak{G}$ qui permet, d'une façon évidente, de dériver $\# \widehat{X}_{i+1} \#$ de $\# \widehat{X}_i \#$ et on a donc :

$$\# S \# \xRightarrow[\mathfrak{G}]{} \# \widehat{X} \#$$

$\widehat{X}$ est bien une phrase de $\mathfrak{L}[\mathfrak{G}]$.

Ainsi s'achève la démonstration de la proposition de Landweber; démonstration que nous avons faite sans tenir compte des lemmes du § 3 pour que le résultat puisse s'appliquer à des grammaires C S non réduites.

N. B. — D'après la démonstration faite cette proposition s'étend aussi aux grammaires C F ; en effet, si $r = (\widehat{A}, \widehat{B})$ est une règle de type C F $(\widehat{A} \in \mathfrak{V}_N)$, il en est de même toutes les règles associées : ${}^{\#}r^{\#}$, ${}^{\#}r$, $r^{\#}$.

5.3. Exemple.

S. Ginsburg et E. H. Spannier ont introduit [4] lors de leur étude du quotient de deux langages C F (cf. § 7) le langage suivant défini sur le vocabulaire terminal $\mathfrak{V}_T = \{ A, B \}$:

$$\mathfrak{L} = \{ AB, A^4, B^2A^3, B^4A^2, B^6A, B^8, A^3B^7, A^6B^6, \ldots, A^{24}, \ldots \}$$

Il s'agit là d'une suite de mots $\widehat{X}_i$, tels que :

(1) si $\widehat{X}_i = \widehat{Y}_i A$ alors $\widehat{X}_{i+1} = B^2\widehat{Y}_i$

(2) si $\widehat{X}_i = \widehat{Y}_i B$ alors $\widehat{X}_{i+1} = A^3\widehat{Y}_i$

On démontre que ce langage n'est pas C F, essentiellement en remarquant que si $\mathfrak{L}$ était C F il existerait une grammaire syntaxique C F l'engendrant (§ 3.1.3). En supprimant de cette grammaire la (ou les) règle terminale ayant pour second membre B, on obtiendrait une grammaire C F qui engendrerait :

$$\mathfrak{L}' = \{ A^4, A^{24}, \ldots, A^{4 \times 6^n}, \ldots \}$$

Or ce langage $\mathfrak{L}'$ n'est pas C F, car les exposants, 4×6^n, $n \in N$, forment une progression géométrique (Cf. *infra*, § 8.1).

Montrons que le langage $\mathfrak{L}$ est du moins C S, en esquissant une grammaire C S avec marquant, $\mathfrak{G}$, l'engendrant.

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \#, \mathfrak{R})$$

avec :

$$\mathfrak{V}_T = \{ A, B \} \quad , \quad \mathfrak{V}_N = \{ S, \#, A_1, A_2, B_1, B_2 \}.$$

Les règles peuvent être groupées de la façon suivante :

1° Règle initiale ($\# S \#$, $\# A_1 B_1 \#$).

2° Règles relatives au passage de $\widehat{Y}_i B$ à $A^3 \widehat{Y}_i$:

$$(B_1 \#, (A_2)^3 \#)$$
$$(B_1 A_2, A_2 B_1)$$
$$(A_1 A_2, A_1 A_1) \quad \text{ou} \quad (\# A_2, \# A_1).$$

3° Règles relatives au passage de $\widehat{Y}_i A$ à $B^2 \widehat{Y}_i$:

$$(A_1 \#, (B_2)^2 \#)$$
$$(A_1 B_2, B_2 A_1)$$
$$(B_1 B_2, B_1 B_1) \quad \text{ou} \quad (\# B_2, \# B_1).$$

4° Règles terminales :

$$(A_1, A) \quad \text{et} \quad (B_1, B).$$

On prouverait facilement que la grammaire engendre toutes les phrases de $\mathfrak{L}$ et elles seules, i. e. :

$$\mathfrak{L}[\mathfrak{G}] = \mathfrak{L}$$

Il en résulte, en tenant compte de la proposition de Landweber (§ 5.2), que le langage étudié, $\mathfrak{L}$, est bien C S.

On aurait pu décrire une grammaire C S, sans marquant :

$$\mathfrak{G}' = (\mathfrak{V}', \mathfrak{V}_T, S', \mathfrak{R}')$$

engendrant aussi le langage $\mathfrak{L}$. En s'inspirant des mêmes méthodes de génération que celles utilisées ci-dessus, le vocabulaire non terminal au lieu de comporter 6 symboles en comporterait 13 :

$$\mathfrak{V}'_N = \{ S', A_1, {}^{\#}A_1, A_1{}^{\#}, A_2, {}^{\#}A_2, A_2{}^{\#}, B_1, {}^{\#}B_1, B_1{}^{\#}, B_2, {}^{\#}B_2, B_2{}^{\#} \}$$

et le nombre des règles serait porté de 11 à 21. Ainsi les règles relatives au passage de $\widehat{Y}_i B$ à $A^3\widehat{Y}_i$ seraient les suivantes :

— $(B_1{}^{\#}, (A_2)^2 A_2{}^{\#})$;

— (B_1A_2, A_2B_1) ; $({}^{\#}B_1A_2, {}^{\#}A_1B_1)$; $(B_1A_2{}^{\#}, A_1B_1{}^{\#})$

— $({}^{\#}A_1A_2, {}^{\#}A_1A_1)$; (A_1A_2, A_1A_1) ; $(A_1A_2{}^{\#}, A_1A_1{}^{\#})$

Sur cet exemple nous voyons l'économie, de symboles et de règles, réalisée par l'utilisation de la proposition précédente.

6. PROPRIÉTÉS DE CLÔTURE DES LANGAGES C S

Nous rappelons qu'un ensemble $\mathfrak{E}$ est dit stable (ou fermé) relativement à une opération, notée $\circ$, pour exprimer qu'en appliquant cette opération à tout élément de $\mathfrak{E}$ si elle est unitaire, à tout couple d'éléments de $\mathfrak{E}$ si elle est binaire, etc., le résultat appartient toujours à l'ensemble. Formellement dans le cas d'une opération binaire :

$$\forall \alpha \in \mathfrak{E}, \forall \beta \in \mathfrak{E} : \quad (\alpha \circ \beta) \in \mathfrak{E}$$

PROPOSITION. — **L'ensemble $\mathfrak{L}\{\alpha_2\}$ (ou $\mathfrak{L}\{\alpha_1\}$, cf. Proposition fondamentale, § 4) des langages C S est stable pour les opérations suivantes :**

1° **Union :** $(\mathfrak{L}_1, \mathfrak{L}_2) \longrightarrow \mathfrak{L}_1 \cup \mathfrak{L}_2$

2° **Produit :** $(\mathfrak{L}_1, \mathfrak{L}_2) \longrightarrow \mathfrak{L}_1\mathfrak{L}_2$

$$\mathfrak{L}_1\mathfrak{L}_2 = \{ \widehat{X} \mid \widehat{X} = \widehat{X}_1\widehat{X}_2; \quad \widehat{X}_1 \in \mathfrak{L}_1 \text{ et } \widehat{X}_2 \in \mathfrak{L}_2 \}$$

3° **Miroir :** $\mathfrak{L} \longrightarrow \tilde{\mathfrak{L}}$

$\mathfrak{L}$ étant l'ensemble des mots de $\mathfrak{L}$ « écrits à l'envers » i. e. :

$$\forall \widehat{X} \in \mathfrak{L} \ , \ \widehat{X} = A_1 \dots A_n \ : \ \tilde{X} = A_nA_{n-1} \dots A_1 \in \tilde{\mathfrak{L}}$$

4° **Étoile :** $\mathfrak{L} \longrightarrow \mathfrak{L}^*$

$$\mathfrak{L}^* = \mathfrak{L} \cup \mathfrak{L}^2 \cup \dots \cup \mathfrak{L}^n \cup \dots \qquad n \in (N - \{0\})$$

REMARQUE. — Le lecteur vérifiera facilement que des démonstrations qui suivent, les points 1, 2, 3, restent valables si l'on se restreint à la classe des langages C F qui est donc stable relativement aux trois opérations : union, produit et miroir. La stabilité de cette classe par rapport à l'opération étoile a aussi été établie. Mais l'extension de la stabilité de cette opération à l'ensemble des langages C S présente quelques difficultés et n'avait pas encore été établie, nous semble-t-il.

DÉMONSTRATION. — Rappelons (cf. § 1.5.2) que tous les langages considérés ici sont définis sur le même vocabulaire terminal $\mathfrak{V}_T$. D'autre part lorsque nous considérerons deux langages $\mathfrak{L}_1$ et $\mathfrak{L}_2$ nous pourrons toujours supposer que les grammaires $\mathfrak{G}_1$ et $\mathfrak{G}_2$ les engendrant sont syntaxiques (lemme 3.1.2) et possèdent des vocabulaires auxiliaires non terminaux, $\mathfrak{V}_{N_1}$ et $\mathfrak{V}_{N_2}$ respectivement, disjoints. Il en résultera, en particulier que les grammaires $\mathfrak{L}_1$ et $\mathfrak{L}_2$ n'auront aucune règle de production commune.

6.1. Union.

Soient :

$$\mathfrak{G}_1 = (\mathfrak{V}_1, \mathfrak{V}_T, S_1, \mathfrak{R}_1) \quad , \quad \mathfrak{G}_2 = (\mathfrak{V}_2, \mathfrak{V}_T, S_2, \mathfrak{R}_2)$$

les grammaires C S syntaxiques engendrant $\mathfrak{L}_1$ et $\mathfrak{L}_2$ respectivement. Considérons la grammaire :

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

telle que :

— $\mathfrak{V} = \mathfrak{V}_1 \cup \mathfrak{V}_2 \cup \{ S \}$;

— l'élément nouveau S, $S \notin \mathfrak{V}_1 \cup \mathfrak{V}_2$, est le symbole initial de $\mathfrak{G}$;

— $\mathfrak{R} = \mathfrak{R}_1 \cup \mathfrak{R}_2 \cup \{ s_1, s_2 \}$,

$$s_1 = (S, S_1) \quad , \quad s_2 = (S, S_2).$$

$\mathfrak{G}$ est bien une grammaire C S. Prouvons qu'elle engendre l'union des deux langages $\mathfrak{L}_1$ et $\mathfrak{L}_2$, i. e. :

$$\mathfrak{L}[\mathfrak{G}] = \mathfrak{L}_1 \cup \mathfrak{L}_2$$

1° $\mathfrak{L}_1 \cup \mathfrak{L}_2 \subset \mathfrak{L}[\mathfrak{G}]$.

En effet $\widehat{X}$ étant une phrase de l'union des deux langages $\mathfrak{L}_1$ et $\mathfrak{L}_2$, $\widehat{X}$ est par exemple une phrase de $\mathfrak{L}_1$, i. e. :

$$\exists \widehat{r} \in \mathfrak{L}(\mathfrak{R}_1) \subset \mathfrak{L}(\mathfrak{R}) \quad : \quad S_1 \underset{\mathfrak{G}_1}{\overset{\widehat{r}}{\Longrightarrow}} \widehat{X}$$

et donc :

$$S \underset{\mathfrak{G}}{\overset{s_1\widehat{r}}{\Longrightarrow}} \widehat{X}$$

Ainsi $\widehat{X}$ est bien une phrase de $\mathfrak{L}[\mathfrak{G}]$. Il en serait de même si $\widehat{X}$ était une phrase de $\mathfrak{L}_2$.

2° $\mathfrak{L}[\mathfrak{G}] \subset \mathfrak{L}_1 \cup \mathfrak{L}_2$.

$\widehat{X}$ étant une phrase de $\mathfrak{L}[\mathfrak{G}]$, soit :

$$(S = \widehat{X}_0, \widehat{X}_1, \ldots, \widehat{X}_i, \ldots, \widehat{X}_n = \widehat{X}', \ldots, \widehat{X})$$

une S-dérivation de $\widehat{X}$ selon $\mathfrak{G}$, où $\widehat{X}'$ est le mot de $\mathfrak{L}(\mathfrak{V}_N)$ dont $\widehat{X}$ dérive par une suite, $\widehat{t}$, de règles terminales; cela est toujours possible, la grammaire $\mathfrak{G}$ étant syntaxique comme $\mathfrak{G}_1$ et $\mathfrak{G}_2$. D'après la nature des règles de $\mathfrak{G}$ nous avons : $\widehat{X}_1 = S_1$, ou bien $\widehat{X}_1 = S_2$. Si $\widehat{X}_1 = S_1$, la S_1-dérivation de $\widehat{X}'$ selon $\mathfrak{G}$, ne peut appliquer que des règles de $\mathfrak{G}_1$, car, pour tout i, $i \in)n)$, $\widehat{X}_i$ est un mot de $\mathfrak{L}(\mathfrak{V}_{N_1})$ étant donné la remarque préliminaire. On a donc :

$$S_1 \underset{\mathfrak{G}_1}{\Longrightarrow} \widehat{X}$$

et $\widehat{X}$ est une phrase de $\mathfrak{L}_1$. De même, si $\widehat{X}_1 = S_2$ on prouverait que $\widehat{X}$ est une phrase de $\mathfrak{L}_2$. Dans tous les cas $\widehat{X}$ est une phrase de $\mathfrak{L}_1 \cup \mathfrak{L}_2$.

6.2. **Produit.**

Les grammaires $\mathfrak{G}_1$ et $\mathfrak{G}_2$ étant définies comme ci-dessus, soit :

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

la grammaire C S où :

— $\mathfrak{V} = \mathfrak{V}_1 \cup \mathfrak{V}_2 \cup \{ S \}$;
— S, l'élément nouveau, $S \notin \mathfrak{V}_1 \cup \mathfrak{V}_2$, étant le symbole initial;
— $\mathfrak{R} = \mathfrak{R}_1 \cup \mathfrak{R}_2 \cup \{ s \}$, $s = (S, S_1S_2)$.

Il s'agit de prouver que :

$$\mathfrak{L}[\mathfrak{G}] = \mathfrak{L}_1\mathfrak{L}_2.$$

1° $\mathfrak{L}_1\mathfrak{L}_2 \subset \mathfrak{L}[\mathfrak{G}]$.

Résumons formellement cette démonstration.

$$\forall \widehat{X} \in \mathfrak{L}_1\mathfrak{L}_2 \quad : \quad \exists \widehat{X}_1 \in \mathfrak{L}_1 \quad \text{et} \quad \widehat{X}_2 \in \mathfrak{L}_2 \quad ; \quad \widehat{X} = \widehat{X}_1\widehat{X}_2;$$

$$\widehat{X}_1 \in \mathfrak{L}_1 \quad : \quad \exists \widehat{r}_1 \in \mathfrak{L}(R_1) \subset \mathfrak{L}(\mathcal{R}) \quad ; \quad S_1 \underset{\mathfrak{G}_1}{\overset{\widehat{r}_1}{\Longrightarrow}} \widehat{X}_1$$

$$\widehat{X}_2 \in \mathfrak{L}_2 \quad : \quad \exists \widehat{r}_2 \in \mathfrak{L}(\mathcal{R}_2) \subset \mathfrak{L}(\mathcal{R}) \quad ; \quad S_2 \underset{\mathfrak{G}_2}{\overset{\widehat{r}_2}{\Longrightarrow}} \widehat{X}_2$$

Il en résulte que : $S \underset{\mathfrak{G}}{\overset{s \widehat{r}_1 \widehat{r}_2}{\Longrightarrow}} \widehat{X} = \widehat{X}_1 \widehat{X}_2$

i. e. :

$$\widehat{X} \in \mathfrak{L}[\mathfrak{G}].$$

2° $\mathfrak{L}[\mathfrak{G}] \subset \mathfrak{L}_1 \mathfrak{L}_2$.

$\widehat{X}$ étant une phrase de $\mathfrak{L}[\mathfrak{G}]$ nous avons :

$$S \underset{\mathfrak{G}}{\overset{s \widehat{r}}{\Longrightarrow}} \widehat{X}, \qquad \widehat{r} \in \mathfrak{L}(\mathcal{R}_1 \cup \mathcal{R}_2)$$

mais $\mathfrak{G}_1$ et $\mathfrak{G}_2$ n'ayant aucune règle commune, i. e. :

$$\mathcal{R}_1 \cap \mathcal{R}_2 = \tilde{\emptyset}$$

nous pouvons écrire, en modifiant s'il le faut l'ordre des dérivations directes :

$$\widehat{r} = \widehat{r}_1 \widehat{r}_2, \qquad \widehat{r}_1 \in \mathfrak{L}(\mathcal{R}_1), \qquad \widehat{r}_2 \in \mathfrak{L}(R_2)$$

et la S-dérivation de $\widehat{X}$ précédente se décompose de la façon suivante :

$$S \xrightarrow[\mathfrak{G}]{s} S_1 S_2 \xrightarrow[\mathfrak{G}_1]{\widehat{r}_1} \widehat{X}_1 S_2 \xrightarrow[\mathfrak{G}_2]{\widehat{r}_2} \widehat{X}_1 \widehat{X}_2 = \widehat{X}$$

i. e. :

$$\widehat{X} \in \mathfrak{L}_1 \mathfrak{L}_2$$

6.3. **Miroir.**

Soit $\mathfrak{L}$ le langage engendré par la grammaire C S

$$\mathfrak{G} = (\mathcal{V}, \mathcal{V}_T, S, \mathcal{R})$$

i. e. :

$$\mathfrak{L} = \mathfrak{L}[\mathfrak{G}].$$

Montrons que son miroir

$$\tilde{\mathfrak{L}} = \{ \tilde{X} \mid \widehat{X} \in \mathfrak{L} \}$$

$\tilde{X}$ étant le mot $\widehat{X}$ écrit à l'envers) est engendré par la grammaire C S

$$\tilde{\mathfrak{G}} = (\mathcal{V}, \mathcal{V}_T, S, \tilde{\mathcal{R}})$$

où :

$$\forall r_i = (\widehat{A}_i, \widehat{B}_i) \in \mathcal{R} :$$

$$\tilde{r}_i = (\tilde{A}_i, \tilde{B}_i) \in \tilde{\mathcal{R}}$$

Si

$$\widehat{r} = r_1 \dots r_k \in \mathfrak{L}(\mathcal{R})$$

on posera :

$$\tilde{r} = \tilde{r}_1 \dots \tilde{r}_k \in \mathfrak{L}(\tilde{\mathcal{R}})$$

1° $\tilde{\mathfrak{L}} \subset \mathfrak{L}[\tilde{\mathcal{G}}]$.

Ce résultat se démontre facilement en remarquant que si :

$$S \underset{\mathcal{G}}{\overset{\widehat{r}}{\Longrightarrow}} \widehat{X}$$

alors :

$$S \underset{\tilde{\mathcal{G}}}{\overset{\tilde{r}}{\Longrightarrow}} \widehat{X}$$

En effet dans le cas d'une dérivation directe :

$$\widehat{E} = \widehat{G}\widehat{A}_i\widehat{D} \xrightarrow[\mathcal{G}]{\widehat{G}, r_i, \widehat{D}} \widehat{F} = \widehat{G}\widehat{B}_i\widehat{D} \qquad r_i = (\widehat{A}_i, \widehat{B}_i) \in \mathcal{R}$$

implique :

$$\tilde{E} = \tilde{D}\tilde{A}_i\tilde{G} \xrightarrow[\tilde{\mathcal{G}}]{\tilde{D}, \tilde{r}_i, \tilde{G}} \tilde{F} = \tilde{D}\tilde{B}_i\tilde{G}$$

2° $\mathfrak{L}[\tilde{\mathcal{G}}] \subset \tilde{\mathfrak{L}}$.

Ceci résulte immédiatement du 1° en remarquant que l'opération miroir est :

— d'une part compatible avec l'inclusion, i. e. :

$$\mathfrak{L}_1 \subset \mathfrak{L}_2 \quad \text{implique} \quad \tilde{\mathfrak{L}}_1 \subset \tilde{\mathfrak{L}}_2$$

— d'autre part involutive, i. e. :

$$\tilde{\tilde{\mathfrak{L}}} = \mathfrak{L}$$

on a de même :

$$\tilde{\tilde{\mathcal{G}}} = \mathcal{G}$$

Ainsi le 1° nous permet de poser en remplaçant $\mathcal{G}$ par $\tilde{\mathcal{G}}$:

$$\widetilde{\mathfrak{L}[\tilde{\mathcal{G}}]} \subset \mathfrak{L}[\tilde{\tilde{\mathcal{G}}}] = \mathfrak{L}$$

et donc :

$$\mathfrak{L}[\tilde{\mathcal{G}}] \subset \tilde{\mathfrak{L}}$$

6.4. Étoile.

Soit :

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

la grammaire C S engendrant $\mathfrak{L}$. On va supposer que cette grammaire est non seulement syntaxique mais aussi linéaire bornée (§ 3.3). D'après la définition d'une telle grammaire la règle :

$$s = (S, SS)$$

n'appartient pas à la grammaire $\mathfrak{G}$. Considérons donc la grammaire :

$$\mathfrak{G}' = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R}')$$

où

$$\mathfrak{R}' = \mathfrak{R} \cup \{ s \}$$

$\mathfrak{G}'$ est aussi une grammaire C S syntaxique.

1° Il est immédiat que toutes les phrases de $\mathfrak{L}^*$:

$$\mathfrak{L}^* = \mathfrak{L} \cup \mathfrak{L}^2, \ldots \cup \mathfrak{L}^n \cup \ldots, n \in \mathbf{N}, \qquad n \geqslant 1,$$

sont engendrées par la grammaire $\mathfrak{G}'$, i. e. :

$$\mathfrak{L}^* \subset \mathfrak{L}[\mathfrak{G}']. \tag{1}$$

En effet si $\widehat{X}$ est une phrase de $\mathfrak{L}^*$ il existe un entier n tel que :

$$\widehat{X} \in \mathfrak{L}^n$$

i. e. :

$$\widehat{X} = \widehat{X}_1 \ldots \widehat{X}_n \quad , \quad \forall i \in)n) \quad : \quad \widehat{X}_i \in \mathfrak{L}$$

et donc $\exists \widehat{r_i} \in \mathfrak{L}(\mathfrak{R})$:

$$S \underset{\mathfrak{G}}{\overset{\widehat{r_i}}{\Longrightarrow}} \widehat{X}_i$$

Il en résulte que :

$$S \underset{\mathfrak{G}'}{\overset{s^{n-1}\widehat{r}}{\Longrightarrow}} \widehat{X} \quad , \quad \widehat{r} = \widehat{r_1} \ldots \widehat{r_i} \ldots \widehat{r_n}$$

i. e. :

$$\widehat{X} \in \mathfrak{L}[\mathfrak{G}'].$$

2° Cependant l'inclusion (1) du 1° peut être stricte. En effet, supposons la situation suivante :

$p = (A_1A_2, A_3A_4)$ est une règle de $\mathfrak{G}$

et on a :

$$S \xrightarrow[\mathfrak{G}']{s} SS \underset{\mathfrak{G}'}{\Longrightarrow} \widehat{X}_1A_1S \underset{\mathfrak{G}'}{\Longrightarrow} (\widehat{X}_1A_1)(A_2\widehat{X}_2) \xrightarrow[\mathfrak{G}']{p} \widehat{X}_1A_3A_4\widehat{X}_2 \underset{\mathfrak{G}'}{\Longrightarrow} \widehat{X}$$

$\widehat{X} \in \mathfrak{L}(\mathfrak{V}_T)$. Il en résulte que $\widehat{X}$ est une phrase de $\mathfrak{L}[\mathfrak{G}']$, mais rien ne nous permet d'affirmer, d'une façon générale, que $\widehat{X}$ est aussi une phrase de $\mathfrak{L}^*$. Ainsi il est possible, par exemple, que dans la grammaire $\mathfrak{G}$ la règle p ait été « inefficace », en ce sens qu'elle ne pouvait jamais intervenir dans une S-dérivation, selon $\mathfrak{G}$, aboutissant à une phrase de $\mathfrak{L}[\mathfrak{G}]$... Pour éviter qu'une règle du type $p = (A_1A_2, A_3A_4)$ ne s'applique dans une dérivation selon $\mathfrak{G}'$, lorsque A_1 « termine » (à droite) un mot dérivant de S et A_2 « débute » (à gauche) un autre mot dérivant d'un autre symbole initial S, nous allons considérer une nouvelle grammaire $\mathfrak{G}_1$ équivalente à $\mathfrak{G}$:

$$\mathfrak{G}_1 = (\mathfrak{V}_1, \mathfrak{V}_T, S, \mathcal{R}_1)$$

définie comme suit :

— $\mathfrak{V}_1 = \mathfrak{V} \cup \mathfrak{V}'$

. $\mathfrak{V}'$ ne comporte que des symboles nouveaux, i. e. :

$$\mathfrak{V} \cap \mathfrak{V}' = \emptyset$$

. il existe une bijection ψ de $\mathfrak{V}_N$ sur $\mathfrak{V}'$. On posera :

$$\forall A_i \in \mathfrak{V}_N \quad : \quad A'_i = \psi(A_i)$$

— Le symbole initial S n'a pas changé.

— $\mathcal{R}_1$ comporte toutes les règles de $\mathfrak{G}$, sauf l'unique règle d'une grammaire linéaire bornée (§ 3.3) faisant disparaître en cours de dérivation le symbole initial S et notée ici

$$(S, A_0)$$

On la remplace dans $\mathfrak{G}_1$ par la règle :

$$(S, A'_0)$$

et à toute règle non terminale de $\mathfrak{G}$:

$$r = (A_iA_j, A_kA_l) \quad ; \quad \text{resp. : } r = (A_i, A_k)$$

on ajoutera dans $\mathfrak{G}_1$ la règle :

$$r' = (A'_iA_j, A'_kA_l) \quad ; \quad \text{resp. : } r' = (A'_i, A'_k)$$

De même à toute règle terminale de $\mathfrak{G}$:

$$t = (A_i, A_k)$$

s'ajoutera dans $\mathfrak{G}_1$, la règle

$$t' = (A'_i, A_k).$$

De cette façon l'on constate facilement que la grammaire $\mathfrak{G}_1$ est aussi puissante que la grammaire $\mathfrak{G}$, i. e. :

$$\mathfrak{L}[\mathfrak{G}] \subset \mathfrak{L}[\mathfrak{G}_1]$$

D'autre part :

$$\mathfrak{L}[\mathfrak{G}_1] \subset \mathfrak{L}[\mathfrak{G}]$$

car l'homomorphisme φ de $\mathfrak{L}(\mathfrak{V}_1)$ dans $\mathfrak{L}(\mathfrak{V})$ tel que :

$$\forall A_i \in \mathfrak{V} \quad : \quad \varphi(A_i) = A_i$$

et

$$\forall A'_i \in \mathfrak{V}' \quad : \quad \varphi(A'_i) = A_i$$

est un homomorphisme de grammaires, ce que l'on vérifierait facilement. On a donc :

$$\mathfrak{L}[\mathfrak{G}_1] = \mathfrak{L}$$

Ayant substitué la grammaire $\mathfrak{G}_1$ à la grammaire $\mathfrak{G}$ nous allons de même substituer la grammaire $\mathfrak{G}'_1$ à la grammaire $\mathfrak{G}'$

$$\mathfrak{G}'_1 = (\mathfrak{V}_1, \mathfrak{V}_T, S, \mathfrak{R}'_1)$$

avec :

$$\mathfrak{R}'_1 = \mathfrak{R}_1 \cup \{ s \} \qquad\qquad s = (S, SS)$$

Le raisonnement du 1° reste valable, nous avons donc toujours :

$$\mathfrak{L}^* \subset \mathfrak{L}[\mathfrak{G}'_1]$$

Mais nous avons maintenant aussi :

$$\mathfrak{L}[\mathfrak{G}'_1] \subset \mathfrak{L}^*$$

car pour toute phrase $\widehat{X}$ de $\mathfrak{L}[\mathfrak{G}'_1]$

$$S \underset{\mathfrak{G}'_1}{\overset{s^{n-1}}{\Longrightarrow}} S^n \underset{\mathfrak{G}_1}{\overset{\widehat{r}}{\Longrightarrow}} X \qquad \widehat{r} \in \mathfrak{L}(\mathfrak{R}_1)$$

Dans $\mathfrak{G}_1$ la situation gênante signalée au début du 2° ne peut se produire. $\mathfrak{G}_1$ étant une grammaire linéaire bornée, chaque symbole S de S^n ne peut disparaître que par application de la règle (S, A'_0) et on aboutit ainsi pour chacun de ces symboles S à un mot de $\mathfrak{L}$, i. e. :

$$\widehat{X} = \widehat{X}_1 \ldots \widehat{X}_n \in \mathfrak{L}^n \subset \mathfrak{L}^*$$

REMARQUE. — On constatera facilement que si :

$$\mathfrak{L} = \mathfrak{V}_T$$

$$\mathfrak{V}_T^* = \mathfrak{L}(\mathfrak{V}_T)$$

Cela explique pourquoi le monoïde libre engendré par un vocabulaire fini $\mathfrak{V}$ est parfois noté $\mathfrak{V}^*$ au lieu de $\mathfrak{L}(\mathfrak{V})$.

Les résultats établis dans ce paragraphe vont être complétés par ceux du paragraphe suivant notamment en ce qui concerne l'intersection et le quotient de deux langages C S.

7. PROLONGEMENT D'UN LANGAGE DANS UN AUTRE

7.1. Définitions. Propriétés élémentaires.

7.1.1. Quotient de deux langages.

Étant donné deux langages $\mathfrak{L}_1$ et $\mathfrak{L}_2$ sur un même vocabulaire terminal $\mathfrak{V}_T$ on définit le quotient à droite (resp. à gauche) de $\mathfrak{L}_1$ par $\mathfrak{L}_2$, noté $\mathfrak{L}_1/\mathfrak{L}_2$ (resp. $\mathfrak{L}_1\backslash\mathfrak{L}_2$), comme suit :

$$\mathfrak{L}_1/\mathfrak{L}_2 = \{ \hat{X} \mid \exists \hat{B} \in \mathfrak{L}_2 \quad : \quad \hat{X}\hat{B} \in \mathfrak{L}_1 \}$$

$$(\text{resp. } \mathfrak{L}_1\backslash\mathfrak{L}_2) = \{ \hat{X} \mid \exists \hat{B} \in \mathfrak{L}_2 \quad : \quad \hat{B}\hat{X} \in \mathfrak{L}_1 \}$$

S. Ginsburg et E. H. Spannier ont étudié cette opération dans le cas des langages C F et des langages réguliers ou langages linéaires d'un côté [4]. Nous rappelons qu'un langage linéaire à droite (resp. à gauche) est engendré par une grammaire, dite aussi linéaire à droite (resp. à gauche), dont toutes les règles sont de l'un des deux types suivants :

$$(A_i,\ D_k\, A_h) \qquad (\text{resp. } (A_i,\ A_h\, D_k))$$

$$(A_i,\ D_k)$$

où A_i et A_h sont des symboles non terminaux et D_k un symbole terminal. Tout langage linéaire à droite étant aussi linéaire à gauche, on parle simplement de langages linéaires d'un côté ou de langages réguliers ou même de langages d'états finis car ils correspondent aux automates à un nombre

fini d'états [3]. S. Ginsburg et E. H. Spannier ont notamment établi les deux résultats suivants (9 juillet 1962) :

1° Le problème, de savoir si le quotient de deux langages C F est C F, est indécidable.

2° Le quotient d'un langage C F par un langage régulier est un langage C F. Des résultats semblables ont été établis antérieurement pour l'intersection.

Dans le cas des langages C S, Landweber a démontré la stabilité de l'intersection [6]. Il ne nous semble pas qu'un tel résultat puisse être étendu au quotient. Mais la recherche de ce résultat nous a amené à considérer une nouvelle opération sur les langages.

7.1.2. Prolongement d'un langage dans un autre.

Étant donné deux langages $\mathfrak{L}_1$ et $\mathfrak{L}_2$, sur un même vocabulaire terminal $\mathfrak{V}_T$, on appellera prolongement à gauche (resp. à droite) du langage $\mathfrak{L}_2$ dans le langage $\mathfrak{L}_1$, l'ensemble des phrases de $\mathfrak{L}_1$ commençant à droite (resp. à gauche) par une phrase de $\mathfrak{L}_2$; on notera ce sous-ensemble de $\mathfrak{L}_1$, $\mathfrak{L}_1[\mathfrak{L}_2]$ (resp. $[\mathfrak{L}_2]\mathfrak{L}_1$).

Formellement on a donc :

$$\mathfrak{L}_1[\mathfrak{L}_2] = \{ \widehat{A} \mid \widehat{A} \in \mathfrak{L}_1 \quad , \quad \exists \widehat{B} \in \mathfrak{L}_2 \quad : \quad \widehat{A} = \widehat{X}\widehat{B} \} \qquad (1)$$

$$(\text{resp. } [\mathfrak{L}_2]\mathfrak{L}_1) = \{ \widehat{A} \mid \widehat{A} \in \mathfrak{L}_1 \quad , \quad \exists \widehat{B} \in \mathfrak{L}_2 \quad : \quad \widehat{A} = \widehat{B}\widehat{X} \}$$

On remarquera les rapports existant entre cette nouvelle opération prolongement et l'opération quotient définie ci-dessus. Ainsi le quotient $\mathfrak{L}_1/\mathfrak{L}_2$ s'obtient à partir du prolongement $\mathfrak{L}_1[\mathfrak{L}_2]$ en substituant, dans toutes les phrases $\widehat{A} = \widehat{X}\widehat{B}$ de $\mathfrak{L}_1[\mathfrak{L}_2]$, à l'extrémité droite $\widehat{B}$ appartenant à $\mathfrak{L}_2$, le mot vide, $\widehat{\varnothing}$. D'autre part le quotient à droite de $\mathfrak{L}_1$ par $\mathfrak{L}_2$ n'est pas modifié si l'on remplace $\mathfrak{L}_1$ par son sous-ensemble $\mathfrak{L}_1[\mathfrak{L}_2]$, i. e. :

$$\mathfrak{L}_1/\mathfrak{L}_2 = \mathfrak{L}_1[\mathfrak{L}_2]/\mathfrak{L}_2$$

On notera aussi que les phrases de $\mathfrak{L}_1 \cap \mathfrak{L}_2$ sont celles de $\mathfrak{L}_1[\mathfrak{L}_2]$, $\widehat{A} = \widehat{X}\widehat{B}$ pour lesquelles on a $\widehat{X} = \widehat{\varnothing}$. Ainsi :

$$\mathfrak{L}_1 \cap \mathfrak{L}_2 \subset \mathfrak{L}_1[\mathfrak{L}_2]$$

$\mathfrak{L}_1[[\mathfrak{L}_2]]$ désignant le prolongement strict à gauche de $\mathfrak{L}_2$ dans $\mathfrak{L}_1$, i. e. :

$$\mathfrak{L}_1[[\mathfrak{L}_2]] = \{ \widehat{A} \mid \widehat{A} \in \mathfrak{L}_1, \quad \exists \widehat{B} \in \mathfrak{L}_2 \quad : \quad \widehat{A} = \widehat{X}\widehat{B}, \quad \widehat{X} \neq \widehat{\varnothing} \}$$

on a :

$$\mathfrak{L}_1[\mathfrak{L}_2] = (\mathfrak{L}_1 \cap \mathfrak{L}_2) \cup (\mathfrak{L}_1[[\mathfrak{L}_2]])$$

$\mathfrak{L}_1 \cap \mathfrak{L}_2$ et $\mathfrak{L}_1[[\mathfrak{L}_2]]$ ne constituent cependant pas toujours une partition de $\mathfrak{L}_1[\mathfrak{L}_2]$, n'étant pas nécessairement disjoints.

EXEMPLES :

1) $\mathfrak{L}_1 = \{ A^n B^n C^n \mid n \in \mathbf{N}^* \}$, $\mathfrak{L}_2 = \{ A^{2^i} / i \in \mathbf{N} \}$

$\mathfrak{L}_1[\mathfrak{L}_2] = \mathfrak{L}_1 \cap \mathfrak{L}_2 = \emptyset$

$[\mathfrak{L}_2]\mathfrak{L}_1 = [[\mathfrak{L}_2]]\mathfrak{L}_1 = \{ A^{2^i} B^{2^i} C^{2^i} \mid i \in \mathbf{N} \}$

2) $\mathfrak{L}_1 = \{ A^n B^n C^p \mid n \in \mathbf{N}^*, p \in \mathbf{N}^* \}$, $\mathfrak{L}_2 = \{ A^m B^q C^q \mid m \in \mathbf{N}^*, q \in \mathbf{N}^* \}$

$\mathfrak{L}_1[\mathfrak{L}_2] = \{ A^n B^n C^n \mid n \in \mathbf{N}^* \} = \mathfrak{L}_1 \cap \mathfrak{L}_2$

Il est connu [3], que $\mathfrak{L}_1$ et $\mathfrak{L}_2$ sont des langages C F tandis que $\mathfrak{L}_1[\mathfrak{L}_2]$ n'en est pas. Sur cet exemple, on constate que le prolongement d'un langage C F dans un langage C F n'est pas nécessairement C F.

Nous allons étudier les propriétés de stabilité de l'opération « prolongement » relativement aux principales classes de langages. Mais auparavant nous constatons facilement, en partant de la définition, que :

$$\mathfrak{L}_1[\mathfrak{L}_2] = \mathfrak{L}_1 \cap (\mathfrak{V}_T^* . \mathfrak{L}_2) \tag{2}$$

(de même $[\mathfrak{L}_2]\mathfrak{L}_1 = \mathfrak{L}_1 \cap (\mathfrak{L}_2 . \mathfrak{V}_T^*)$)

$\mathfrak{V}_T^*$ désignant d'après une remarque du § 6.4, le monoïde libre $\mathfrak{L}(\mathfrak{V}_T)$. Sachant d'une part que $\mathfrak{V}_T^*$ est toujours un langage régulier et que d'autre part les opérations produit et intersection sont stables pour la classe des langages réguliers, la formule (2) permet d'affirmer directement la stabilité de la nouvelle opération relativement à cette même classe. Ce raisonnement pourrait aussi s'appliquer pour la classe des langages C S en tenant compte du théorème de Landweber sur l'intersection de deux langages C S. Nous préférons établir la stabilité de l'opération prolongement relativement aux langages C S et en déduire le théorème de Landweber. Mais auparavant nous signalons quelques autres propriétés de cette nouvelle opération.

7.1.3. Propriétés élémentaires.

Elles s'établissent facilement en partant de la définition ou de la formule (2).

$$\widetilde{[\mathfrak{L}_2]\mathfrak{L}_1} = \tilde{\mathfrak{L}}_1[\tilde{\mathfrak{L}}_2] \tag{3}$$

On rappelle qu'étant donné un langage $\mathfrak{L}$, $\tilde{\mathfrak{L}}$ désigne son miroir, c'est-à-dire l'ensemble des phrases de $\mathfrak{L}$ écrites à l'envers. Cette opération miroir étant involutive et stable relativement à la classe des langages C S (ou C F) (cf. § 6.3) la formule (3) permet de remplacer les prolongements à droite par des prolongements à gauche à l'étude desquels nous nous limiterons par la suite.

$$\mathfrak{L}_1 \subset \mathfrak{L}_2 \quad \text{implique} \quad \mathfrak{L}_1[\mathfrak{L}_2] = \mathfrak{L}_1 \tag{4}$$

$$\mathfrak{L}_2 \subset \mathfrak{L}_3 \quad \text{implique} \quad \mathfrak{L}_1[\mathfrak{L}_2] \subset \mathfrak{L}_1[\mathfrak{L}_3] \tag{5}$$

$$(\mathfrak{L}_1 \cup \mathfrak{L}_2)[\mathfrak{L}_3] = \mathfrak{L}_1[\mathfrak{L}_3] \cup \mathfrak{L}_2[\mathfrak{L}_3] \tag{6}$$

$$\mathfrak{L}_1[\mathfrak{L}_2 \cup \mathfrak{L}_3] = \mathfrak{L}_1[\mathfrak{L}_2] \cup \mathfrak{L}_1[\mathfrak{L}_3] \tag{7}$$

$$(\mathfrak{L}_1 \cap \mathfrak{L}_2)[\mathfrak{L}_3] = \mathfrak{L}_1[\mathfrak{L}_3] \cap \mathfrak{L}_2[\mathfrak{L}_3] \tag{8}$$

$$\mathfrak{L}_1[\mathfrak{L}_2 \cap \mathfrak{L}_3] \subset \mathfrak{L}_1[\mathfrak{L}_2] \cap \mathfrak{L}_1[\mathfrak{L}_3] \tag{9}$$

On peut préciser cette dernière inclusion en remarquant que

$$\mathfrak{L}_2 \cap \mathfrak{L}_3 \subset \mathfrak{L}_2[\mathfrak{L}_3]$$

d'où, en utilisant la propriété (5) :

$$\mathfrak{L}_1[\mathfrak{L}_2 \cap \mathfrak{L}_3] \subset \mathfrak{L}_1[\mathfrak{L}_2[\mathfrak{L}_3]] \subset \mathfrak{L}_1[\mathfrak{L}_2] \cap \mathfrak{L}_1[\mathfrak{L}_3] \tag{9'}$$

7.2. Langages C F.

Nous intéressant spécialement aux langages C S, nous ne donnerons pas, dans ce paragraphe, des démonstrations rigoureuses mais l'énoncé de quelques résultats. Au paragraphe précédent, on a remarqué sur un exemple, que le prolongement d'un langage C F dans un langage C F n'est pas nécessairement C F. Mais on peut établir :

7.2.1. Le prolongement d'un langage régulier dans un langage C F est C F.

On remarquera, tout d'abord, qu'étant donné une grammaire linéaire à droite, par exemple,

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathfrak{R})$$

il existe une grammaire équivalente $\mathfrak{G}'$, linéaire du même côté et dont aucune règle ne contient le symbole initial dans son second membre. Il suffit de prendre

$$\mathfrak{G}' = (\mathfrak{V}', \mathfrak{V}_T, S', \mathfrak{R}')$$

avec :

— $\mathfrak{V}' = \mathfrak{V} \cup \{ S' \}$, S' étant un nouveau symbole,

— $\mathcal{R}'$ comporte toutes les règles de $\mathfrak{G}$; et, de plus, si

$$(S, D_k A_i) \in \mathcal{R} \quad \text{alors} \quad (S', D_k A_i) \in \mathcal{R}'.$$

Soient donc $\mathfrak{L}_1$, un langage C S, et $\mathfrak{L}_2$ un langage régulier engendrés respectivement par $\mathfrak{G}_1$ et $\mathfrak{G}_2$:

$$\mathfrak{G}_1 = (\mathfrak{V}_1, \mathfrak{V}_T, S_1, \mathcal{R}_1) \qquad \mathfrak{G}_2 = (\mathfrak{V}_2, \mathfrak{V}_T, S_2, \mathcal{R}_2)$$

où :

$$\mathfrak{V}_T = \{ D_1, \ldots, D_p \}$$

$$\mathfrak{V}_{N_1} = \{ A_1, \ldots, A_p, A_{p+1}, \ldots, A_n \} \quad ; \quad \mathfrak{V}_{N_2} = \{ B_1, \ldots, B_m \}$$

$$S_1 = A_n \qquad S_2 = B_m$$

Considérons la grammaire C F

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathcal{R})$$

où :

— $\mathfrak{V}_N = \left\{ C_i^k \,\middle|\, i, k \in (m), \quad j \in)n) \right\}$

(i. e. $\mathfrak{V}_N$ est un ensemble de symboles à trois indices, dont deux varient de 0 à m, et l'autre de 1 à n).

— $S = C_m^0{}_n$

— $\mathcal{R}$ comporte :

. des règles non terminales déduites de celles de la grammaire $\mathfrak{G}_1$ que l'on suppose normale :

si $(A_i, A_k A_l)$ est une règle non terminale de $\mathfrak{G}_1$,

$$\left(C_h^{h'}{}_i, C_h^g{}_k C_g^{h'}{}_l \right)$$

est une règle de $\mathfrak{G}$, ceci pour tout h, h' et g compris entre 0 et m i. e., h, h', $g \in (m)$, avec la restriction suivante : si $h = h' = m$ on pose aussi $g = m$;

. des règles terminales déduites des règles de la grammaire $\mathfrak{G}_2$ (d'après la remarque préliminaire on suppose que $\mathfrak{G}_2$ ne comporte pas de règle de la forme $(B_i, A_j B_m)$ quels que soient i et j) :

— si $(B_i, D_j B_k)$ est une règle de $\mathfrak{G}_2$ $(k \neq m)$ alors $\left(C_i^k{}_j, D_j \right)$ est une règle terminale de $\mathfrak{G}$;

— de même si $(B_i, D_j) \in \mathcal{R}_{T_2}$ alors $\left(C_i^0{}_j, D_j \right) \in \mathcal{R}_T$.

— enfin $\forall i \in)p)$:

$$\left(C_m^m i, A_i\right) \in \mathcal{R}_T.$$

On prouvera que :

$$\mathfrak{L}[\mathfrak{G}] = \mathfrak{L}_1[\mathfrak{L}_2]$$

Pour cela on remarquera que la grammaire $\mathfrak{G}$ est normale et donc si $\widehat{A}$ est une phrase de $\mathfrak{L}[\mathfrak{G}]$, il existe deux suites de règles, $\widehat{r} \in \mathfrak{L}(\mathcal{R}_N)$ et $\widehat{t} \in \mathfrak{L}(\mathcal{R}_T)$:

$$S = C_m^0 n \underset{\mathfrak{G}}{\overset{\widehat{r}}{\Longrightarrow}} \widehat{A}' \underset{\mathfrak{G}}{\overset{\widehat{t}}{\Longrightarrow}} \widehat{A}$$

De plus si :

$$\widehat{A} = D_{i_1} \ldots D_{i_q} \qquad \forall j \in)q) \quad : \quad i_j \in)p).$$

alors $\widehat{A}'$ se présente nécessairement sous la forme suivante :

$$\widehat{A}' = C_m^m i_1 \ldots C_m^m i_k \, C_m^{j_k+1} i_{k+1} \ldots C_{j_{q-1}}^0 i_q$$

avec j_r différent de m pour tout r, $r \geqslant k+1$, étant donné la restriction imposée aux règles non terminales et la nature des règles terminales. On pourra alors établir facilement que :

— d'une part $\widehat{A}$ est une phrase de $\mathfrak{L}_1$,

— d'autre part que $\widehat{B} = D_{i_{k+1}} \ldots D_{i_q}$ est une phrase de $\mathfrak{L}_2$ et donc $\widehat{A} = \widehat{X}\widehat{B}$ est une phrase de $\mathfrak{L}_1[\mathfrak{L}_2]$.

La réciproque : toute phrase de $\mathfrak{L}_1[\mathfrak{L}_2]$ peut être engendrée par $\mathfrak{G}$, s'établira sans difficulté.

Ce résultat a des conséquences immédiates relatives à l'intersection et au quotient.

1° Si de la grammaire $\mathfrak{G}$ précédente on élimine les règles terminales : $\forall i \in)p)$: $\left(C_m^m i, A_i\right)$; il est facile de vérifier que la grammaire C F qui en résulte engendre uniquement $\mathfrak{L}_1 \cap \mathfrak{L}_2$, ce qui nous permet d'énoncer :

L'intersection d'un langage C F et d'un langage régulier est C F.

2° De même, partant toujours de la grammaire, $\mathfrak{G}$, si l'on substitue à toute règle terminale $\left(C_i^k j, D_j\right)$ la règle $\left(C_i^k j, \widehat{\varnothing}\right)$ et à toute règle terminale $\left(C_i^0 j, D_j\right)$ la règle $\left(C_i^0 j, \widehat{\varnothing}\right)$, la grammaire qui en résulte engendre le quotient $\mathfrak{L}_1/\mathfrak{L}_2$, ce que l'on vérifie facilement. Or étant donné un langage C F, $\mathfrak{L}$, engendré

par une grammaire $\mathfrak{G}_1$ contenant en plus de règles C F ordinaires, des règles de la forme $(A_j, \widehat{\emptyset})$, A_j étant un symbole non terminal, on démontre qu'il existe toujours une grammaire C F, $\mathfrak{G}_2$, ne contenant pas de telles règles et engendrant $\mathfrak{L} - \{\widehat{\emptyset}\}$. Ceci nous permet d'énoncer avec S. Ginsburg et E. H. Spannier [4] :

— Le quotient d'un langage C F par un langage régulier est C F.

Le prolongement d'un langage C F dans un langage C F n'est pas, nous l'avons vu, nécessairement C F. Mais nous avons plus.

7.2.2. **Il est récursivement indécidable de déterminer, pour deux langages C F arbitraires, $\mathfrak{L}_1$ et $\mathfrak{L}_2$, si, oui ou non le prolongement $\mathfrak{L}_1[\mathfrak{L}_2]$ est C F.**

Ce résultat va être établi par une méthode, inspirée de S. Ginsburg et E. H. Spannier [4], qui permettra de montrer aussi l'indécidabilité des problèmes de savoir si l'intersection ou le quotient de deux langages C F est C F.

Pour chaque n-uplet

$$\Sigma = (\widehat{Z}_1, \widehat{Z}_2, \ldots, \widehat{Z}_n)$$

de mots, non vides, définis sur le vocabulaire { E, F }, on peut considérer le langage $\mathfrak{L}(\Sigma)$ défini sur le vocabulaire { D, E, F }

$$\mathfrak{L}(\Sigma) = \left\{ \mathrm{D}\,\widehat{Z}_{i_1} \ldots \widehat{Z}_{i_k}\,\mathrm{D\,E\,F}^{i_k}\,\mathrm{E\,F}^{i_{k-1}} \ldots \mathrm{E\,F}^{i_1} \,\middle|\, k \geqslant 2;\ i_1, i_2, \ldots, i_k \in)n) \right\}$$

Un tel langage est toujours C F car il peut être engendré par la grammaire ayant pour règles :

$$.(\mathrm{S}, \mathrm{D\,S}_1)$$

$$\forall i \in)n) \qquad .(\mathrm{S}_1, \widehat{Z}_i \mathrm{S}_1\,\mathrm{E\,F}^i)$$

$$\forall i \in)n) \qquad .(\mathrm{S}_1, \widehat{Z}_i\,\mathrm{D\,E\,F}^i)$$

S étant le symbole initial, S_1 un autre symbole auxiliaire.

$$\Gamma = (\widehat{X}_1, \widehat{X}_2, \ldots, \widehat{X}_n) \quad \text{et} \quad \Lambda = (\widehat{Y}_1, \widehat{Y}_2, \ldots, \widehat{Y}_n)$$

étant deux n-uplets de mots, non vides, de $\mathfrak{L}(\{ \mathrm{E, F} \})$, il est facile de voir que le problème, de savoir si :

$$\mathfrak{L}(\Gamma)\,[\mathfrak{L}(\Lambda)] = \mathfrak{L}(\Gamma) \cap \mathfrak{L}(\Lambda)$$

est vide ou non, se ramène au problème de correspondance de Post (§ 1.6) pour les n-uplets de paires de mots :

$$\{ (\widehat{X}_1, \widehat{Y}_1), (\widehat{X}_2, \widehat{Y}_2), \ldots, (\widehat{X}_n, \widehat{Y}_n) \}.$$

Or un tel problème est indécidable.

Soient $\mathfrak{L}_1$ et $\mathfrak{L}_2$, deux langages C F particuliers définis sur le vocabulaire $\{ A, B, C \}$ (cf. § 7.1.2)

$$\mathfrak{L}_1 = \{ A^n B^n C^p \mid n, p \in \mathbf{N}^* \}$$
$$\mathfrak{L}_2 = \{ A^m B^q C^q \mid m, q \in \mathbf{N}^* \}$$

Les langages $\mathfrak{L}_1\mathfrak{L}(\Gamma)$ et $\mathfrak{L}_2\mathfrak{L}(\Lambda)$, définis sur le vocabulaire terminal

$$\mathfrak{V}_T = \{ A, B, C, D, E, F \},$$

produits de langages C F sont eux-mêmes C F (§ 6.2). Or :

$$(1) \quad (\mathfrak{L}_1 \cdot \mathfrak{L}(\Gamma))\,[\mathfrak{L}_2 \cdot \mathfrak{L}(\Lambda)] = (\mathfrak{L}_1 \cdot \mathfrak{L}(\Gamma)) \cap (\mathfrak{L}_2 \cdot \mathfrak{L}(\Lambda)) = (\mathfrak{L}_1 \cap \mathfrak{L}_2) \cdot (\mathfrak{L}(\Gamma) \cap \mathfrak{L}(\Lambda))$$

est, suivant que le problème de correspondance de Post, signalé ci-dessus, admet une solution ou non, ou bien non C F, ou bien vide. En effet, dans le 1er cas, s'il existait une grammaire C F, supposée syntaxique, engendrant le langage précédent (1), la grammaire C F qui s'en déduirait en remplaçant les règles terminales (A_i, D), (A_j, E), (A_k, F) par les règles $(A_i, \widehat{\emptyset})$, $(A, \widehat{\emptyset})$, $(A_k, \widehat{\emptyset})$, engendrerait :

$$\mathfrak{L}_1 \cap \mathfrak{L}_2 = \{ A^n B^n C^n \mid n \in \mathbf{N}^* \}$$

Or ce langage n'est pas C F [3]. On a donc le résultat annoncé ainsi que le suivant :

— Le problème, de savoir si l'intersection de deux langages C F est C F, est indécidable.

S. Ginsburg et E. H. Spannier ont établi [4], un résultat semblable pour le quotient en procédant de la façon suivante : quels que soient les langages $\mathfrak{L}'$ et $\mathfrak{L}''$, le quotient

$$\mathfrak{L}' \cdot \mathfrak{L}(\Gamma) / \mathfrak{L}'' \cdot \mathfrak{L}(\Lambda)$$

est égal à $\mathfrak{L}'/\mathfrak{L}''$ ou vide suivant que le problème de correspondance de Post considéré admet une solution ou non. Ce problème étant indécidable, il ne reste donc plus, comme ci-dessus, qu'à trouver deux langages C F particuliers $\mathfrak{L}'$ et $\mathfrak{L}''$ tels que le quotient $\mathfrak{L}'/\mathfrak{L}''$ ne soit pas C F.

Étant donné trois mots $\widehat{M}_1$, $\widehat{M}_2$, $\widehat{M}_3$ sur le vocabulaire { A, B, C }, le langage

$$\mathfrak{L}(\widehat{M}_1, \widehat{M}_2, \widehat{M}_3) = \{ \widehat{M}_{i_1} \dots \widehat{M}_{i_k} D \widehat{N}_{i_1} \dots \widehat{N}_{i_k} \mid k \geqslant 1;\ i_1, \dots, i_k \in)3) \}$$

où $\widehat{N}_1 = A$, $\widehat{N}_2 = B$, $\widehat{N}_3 = C$ est toujours C F, car il peut être engendré par la grammaire ayant pour règles :

$$\forall i \in)3) \qquad .(S, \widehat{M}_i S \widehat{N}_i)$$
$$.(S, \widehat{M}_i D \widehat{N}_i)$$

Il en résulte en particulier que $\mathfrak{L}' = \mathfrak{L}(B^2, A^3, ABC)$ et $\mathfrak{L}'' = \mathfrak{L}(A, B, C)$ sont des langages C F. Or on prouverait que :

$$\mathfrak{L}'/\mathfrak{L}'' = \{ AB, A^4, B^2, A^3, B^4A^2, B^6A, B^8, A^3B^7, \dots, A^{24}, \dots \} ;$$

et un tel langage n'est pas C F (cf. § 5.2). On peut donc énoncer :

— Il est récursivement indécidable de déterminer pour deux langages C F arbitraires, $\mathfrak{L}_1$ et $\mathfrak{L}_2$, si le quotient $\mathfrak{L}_1/\mathfrak{L}_2$ est C F ou non.

7.3. Langages C S.

7.3.1. Prolongement.

THÉORÈME. — **Le prolongement d'un langage C S dans un langage C S est C S.**

DÉMONSTRATION. — *Rappels.* — 1° Au paragraphe 3.3 on a établi que tout langage C S pouvait être engendré par une grammaire linéaire bornée, c'est-à-dire une grammaire dont les seules règles comportant le symbole initial S sont :

$$(S, SW) \qquad \text{et} \qquad (S, S')$$

W et S' étant deux symboles particuliers; les autres règles non terminales étant toutes de la forme (A_iA_j, A_kA_l), les règles terminales étant d'ordre un comme pour les grammaires syntaxiques. Il sera bon d'avoir présent à l'esprit la remarque du paragraphe 3.3.2 indiquant les caractéristiques d'une S-dérivation selon une grammaire linéaire-bornée.

2° Au paragraphe 5 on a prouvé la proposition de Landweber énonçant que toute grammaire C S avec marquant est équivalente à une grammaire C S sans marquant.

Soient $\mathfrak{L}_1$ et $\mathfrak{L}_2$ deux langages C S engendrés respectivement par $\mathfrak{G}_1$ et $\mathfrak{G}_2$, deux grammaires C S linéaires bornées :

$$\mathfrak{G}_1 = (\mathfrak{V}_1, \mathfrak{V}_T, S_1, \mathfrak{R}_1) \quad ; \quad \mathfrak{G}_2 = (\mathfrak{V}_2, \mathfrak{V}_T, S_2, \mathfrak{R}_2),$$

avec :

$$\mathfrak{V}_T = \{ D_1, D_2, \ldots, D_p \}$$

$\mathfrak{V}_{N_1} = \{ A_1, \ldots, A_p, A_{p+1}, \ldots, A_n \}$, $S_1 = A_n$ étant le symbole initial de $\mathfrak{G}_1$, $S'_1 = A_{n-1}$ et $W_1 = A_{p+1}$ correspondant aux symboles particuliers d'une grammaire linéaire bornée (cf. rappels 1°);
$\mathfrak{V}_{N_2} = \{ B_1, \ldots, B_p, B_{p+1}, \ldots, B_m \}$, $S_2 = B_m$ étant le symbole initial de $\mathfrak{G}_2$. Ces deux vocabulaires non terminaux sont évidemment supposés disjoints. Les règles terminales de ces deux grammaires syntaxiques sont :

$$\forall i \in)p) \ : \ t_{1,i} = (A_i, D_i) \in \mathfrak{R}_{T_1}$$

$$t_{2,i} = (B_i, D_i) \in \mathfrak{R}_{T_2}$$

(cf. démonstration du lemme 3.1.2).

Nous allons maintenant définir une grammaire C S, avec marquant, $\mathfrak{G}$, engendrant le prolongement $\mathfrak{L}_1[\mathfrak{L}_2]$:

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \#, \mathfrak{R}).$$

— $\mathfrak{G}$ a pour vocabulaire non terminal :

$$\mathfrak{V}_N = \mathfrak{V}_{N_1} \cup \mathfrak{V}_{N_2} \cup \mathfrak{V}' \cup \{ S, S_3, \# \}$$

où :

$$\mathfrak{V}' = \{ C_i^j \mid \forall i \in)n), \qquad \forall j \in)p) \}$$

C_i^j pourra faire penser simultanément aux symboles A_i et D_j ;

— S est son symbole initial.

— Les règles de $\mathfrak{G}$ peuvent être cataloguées en 5 groupes. Nous indiquerons après la description de chaque groupe les dérivations essentielles que permettent ses règles.

1er groupe: les règles initiales

$$(\# S \#, \# S_3 S_2 \#) \qquad \text{et} \qquad (\# S \#, \# S_2 \#)$$

débutant toute dérivation aboutissant à une phrase de $\mathfrak{L}_1[[\mathfrak{L}_2]]$ et de $\mathfrak{L}_1 \cap \mathfrak{L}_2$ respectivement.

2e groupe: $\mathfrak{R}_{N_3}$, toute règle non terminale de $\mathfrak{G}_2$ est une règle de $\mathfrak{G}$. Ces premières règles nous permettent les dérivations suivantes :

$$\# S \# \underset{\mathfrak{G}}{\Longrightarrow} \# S_3 B_{j_r} \ldots B_{j_2} B_{j_1} \#$$

$$\# S \# \underset{\mathfrak{G}}{\Longrightarrow} \# B_{j_r} \ldots B_{j_2} B_{j_1} \#$$

pour toute suite $(j_1, \ldots, j_r)$ d'indices compris entre 1 et p, telle que :

$$\widehat{B} = D_{j_r} \ldots D_{j_1} \text{ soit une phrase de } \mathfrak{L}_2.$$

3e groupe: règles spéciales permettant de passer aux symboles C_i^j

$\forall j \in]p), \forall j' \in]p)$: $(B_j \#, C_{p+1}^j \#)$ (on rappelle que $W_1 = A_{p+1}$)

$(B_{j'} C_{p+1}^j, C_{p+1}^{j'} C_{p+1}^j)$

$(S_3 C_{p+1}^j, A_n C_{p+1}^j)$, $(A_n = S_1)$

$(\# C_{p+1}^j, \# C_{n-1}^j)$, $(A_{n-1} = S'_1)$

On peut, grâce à ces règles, dériver de $\# S \#$ les mots de l'un des deux types suivants :

(1) $$\# A_n C_{p+1}^{j_r} \ldots C_{p+1}^{j_2} C_{p+1}^{j_1} \#$$

(2) $$\# C_{n-1}^{j_r} C_{p+1}^{j_r} \ldots C_{p+1}^{j_2} C_{p+1}^{j_1} \#$$

avec : $$\widehat{B} = D_{j_r} \ldots D_{j_1} \in \mathfrak{L}_2.$$

On sait d'autre part (cf. remarque 3.3.2) que toute S_1-dérivation selon $\mathfrak{G}_1$ aboutissant à une phrase $\widehat{A}$ de $\mathfrak{L}_2$ passe par les étapes suivantes :

$$S_1 = A_n \underset{\mathfrak{G}_1}{\Longrightarrow} A_n A_{p+1} \ldots A_{p+1} \underset{\mathfrak{G}_1}{\longrightarrow} A_{n-1} A_{p+1} \ldots A_{p+1} \underset{\mathfrak{G}_1}{\Longrightarrow} \widehat{A}$$

Les règles suivantes, déterminées par celles de $\mathfrak{G}_1$, vont permettre de prolonger les dérivations, comme dans $\mathfrak{G}_1$, en agissant uniquement sur les indices inférieurs; les indices supérieurs n'étant pas modifiés, conserveront en cours de dérivations l'information de la production de la phrase $\widehat{B}$ par la grammaire $\mathfrak{G}_2$.

4e groupe: $\mathfrak{R}_{N_1}$ et d'autres règles définies comme suit :

— $(A_n, A_n A_{p+1})$ et (A_n, A_{n-1}) sont des règles de $\mathfrak{G}$, la première permet notamment de prolonger par une dérivation selon $\mathfrak{G}$, les mots de type (1) signalé ci-dessus;

— à toute autre règle non terminale $(A_i A_j, A_k A_l)$ s'ajoutent les $(p^2 + p)$ règles suivantes :

$\forall h \in]p)$: $(A_i C_j^h, A_k C_l^h)$

$\forall h \in]p), \forall h' \in]p)$: $(C_i^h C_j^{h'}, C_k^h C_l^{h'})$

Grâce à ces règles on prolonge les dérivations comme si l'on était dans $\mathfrak{G}_1$, et l'on dérive de $\# S \#$ des mots de l'un des deux types suivants :

(1') $$\# A_{i_{r+s}} \dots A_{i_{r+1}} C_{i_r}^{j_r} \dots C_{i_1}^{j_1} \#$$

(2') $$\# C_{i_r}^{j_r} \dots C_{i_1}^{j_1} \#$$

$D_{i_{r+s}} \dots D_{i_1}$ et $D_{i_r} \dots D_{i_1}$ étant des phases de $\mathfrak{L}_1$.

5e groupe: $\mathcal{R}_T$, toute règle terminale de $\mathfrak{G}_1$

$$t_{1i,} = (A_i, D_i) \quad , \quad i \in)p),$$

est une règle terminale de $\mathfrak{G}$ ainsi que la règle associée $t_i = (C_i^i, D_i)$.

On pourra donc appliquer ces règles terminales pour aboutir à une phrase de $\mathfrak{L}(\mathfrak{G})$ si et seulement si par les règles non terminales des quatre premiers groupes on a formé un mot de l'un des deux types suivants :

(1'') $$\# A_{i_{r+s}} \dots A_{i_{r+1}} C_{i_r}^{i_r} \dots C_{i_1}^{i_1} \#$$

(2'') $$\# C_{i_r}^{i_r} \dots C_{i_1}^{i_1} \#$$

Mais étant donné la nature des règles non terminales, le lecteur verra facilement que cela aussi est possible si et seulement si :

— d'une part $\widehat{B} = D_{i_r} \dots D_{i_1}$ est une phrase de $\mathfrak{L}_2$;

— d'autre part :

. dans le cas (1'') $\widehat{A} = D_{i_{r+s}} \dots D_{i_{r+1}} \widehat{B}$ est une phrase de $\mathfrak{L}_1$, mais alors $\widehat{A} \in \mathfrak{L}_1[[\mathfrak{L}_2]]$,

. dans le cas (2'') $\widehat{A} = \widehat{B}$ est aussi une phrase de $\mathfrak{L}_1$ et donc $\widehat{A} \in \mathfrak{L}_1 \cap \mathfrak{L}_2$.

Nous avons bien :

$$\mathfrak{L}[\mathfrak{G}] = \mathfrak{L}_1[\mathfrak{L}_2] = \mathfrak{L}_1[[\mathfrak{L}_2]] \cup (\mathfrak{L}_1 \cap \mathfrak{L}_2)$$

Un premier avantage de cette démonstration est de permettre d'en déduire directement le théorème de Landweber sur l'intersection des langages C S [6].

7.3.2. Intersection.

COROLLAIRE. — **L'intersection de deux langages C S est C S.**

En effet, si, de la grammaire $\mathfrak{G}$ ci-dessus, nous supprimons la règle initiale ($\# S \#$, $\# S_3 S_2 \#$), les $\# S \#$-dérivations, selon la grammaire C S, $\mathfrak{G}'$,

qui en résulte, n'engendrent plus que les mots (2), (2'), (2''), et donc d'après la démonstration du paragraphe précédent :

$$\mathfrak{L}[\mathfrak{G}'] = \mathfrak{L}_1 \cap \mathfrak{L}_2$$

On démontrerait de même que, si l'on appelle $\mathfrak{G}''$ la grammaire C S déduite de la grammaire $\mathfrak{G}$ par suppression de la règle ($\neq$ S $\neq$, $\neq$ S_2 $\neq$), on a :

$$\mathfrak{L}[\mathfrak{G}''] = \mathfrak{L}_1[[\mathfrak{L}_2]]$$

Enfin la démonstration du théorème précédent va nous permettre d'étudier le quotient de deux langages C S.

7.3.3. **Quotient.**

Une grammaire engendrant le quotient $\mathfrak{L}_1/\mathfrak{L}_2$ est celle déduite de la grammaire $\mathfrak{G}$ (§ 7.3.1) par substitution à toute règle terminale

$$t_i = (C_i^i,\ D_i), \qquad i \in)p),$$

de la règle

$$t_i' = (C_i^i,\ \widehat{\emptyset})$$

Mais une telle grammaire n'est plus C S ; car $|\widehat{\emptyset}| = 0$ et $|C_i^i| = 1$, si bien que t'_i ($i \in)p$)) est une règle pour laquelle le second membre est de longueur strictement inférieure à celle du premier membre... Dans le cas des grammaires C F on a déjà signalé le résultat suivant (§ 7.2.1) : étant donné une grammaire C F engendrant un langage $\mathfrak{L}$, contenant le mot vide $\widehat{\emptyset}$, on peut lui associer une grammaire C F très « voisine » engendrant $\mathfrak{L} - \{\widehat{\emptyset}\}$. En particulier la nouvelle grammaire ne contient plus de règle de la forme $(A_i, \widehat{\emptyset})$. Un tel résultat ne semble pas applicable aux grammaires C S...

On peut toutefois généraliser la notion de grammaire C S de façon à pouvoir engendrer le mot vide ; c'est-à-dire, étant donné une grammaire C S, $\mathfrak{G}$, engendrant $\mathfrak{L}$, on peut lui associer une grammaire, $\mathfrak{G}'$, « très voisine », engendrant $\mathfrak{L} \cup \{\widehat{\emptyset}\}$. En effet, on peut supposer que $\mathfrak{G}$ est une grammaire C S linéaire bornée; pour obtenir la grammaire $\mathfrak{G}'$ nous ajoutons aux règles de $\mathfrak{G}$ l'unique règle $(S, \widehat{\emptyset})$, S étant le symbole initial. On établira facilement que la nouvelle grammaire $\mathfrak{G}'$ engendre uniquement $\mathfrak{L} \cup \{\widehat{\emptyset}\}$, en remarquant qu'il n'y a que deux autres règles où intervient le symbole initial S (S, SW) et (S, S') (cf. rappels 1°, § 7.3).

Si des règles arbitraires du type $(C_i^i, \widehat{\emptyset})$ ne peuvent intervenir dans une grammaire C S, nous pouvons, tout de même, remplacer dans la grammaire $\mathfrak{G}$ ci-dessus (§ 7.3.1) les règles terminales $t = (C_i^i, D)$ $(i \in I)$ par la règle (C_i^i, D_0), D_0 étant un nouveau symbole terminal que nous supposerons figurer le « blanc ». La grammaire $\mathfrak{G}'''$, ainsi définie, n'engendre pas le quotient $\mathfrak{L}_1/\mathfrak{L}_2$ mais un langage, que l'on notera $\mathfrak{L}_1//\mathfrak{L}_2$, dont les phrases s'obtiennent à partir de celles de $\mathfrak{L}_1[\mathfrak{L}_2]$, $\widehat{A} = \widehat{X}\widehat{B}$, en remplaçant l'extrémité droite $\widehat{B}$, appartenant au langage $\mathfrak{L}_2$, non par le mot vide, $\widehat{\emptyset}$ (comme dans le quotient), mais par un mot de même longueur ne comportant que des symboles D_0, i. e. :

$$\left.\begin{aligned}\mathfrak{L}[\mathfrak{G}'''] = \mathfrak{L}_1 // \mathfrak{L}_2 = \{ \widehat{X} \mid \widehat{X} = D_{i_1} \dots D_{i_q}(D_0)^r, \quad \exists \widehat{B} \in \mathfrak{L}_2, \quad \exists \widehat{A} \in \mathfrak{L}_1 \\ \widehat{B} = D_{i_{q+1}} \dots D_{i_{q+r}}, \quad \widehat{A} = D_{i_1} \dots D_{i_q}\widehat{B}, \quad \forall j \in)q+r) \ : \ i_j \in (p) \}\end{aligned}\right.$$

8. EXEMPLES DE LANGAGES C S. APPLICATIONS

8.1. Langages C S artificiels.

Nous allons présenter dans ce paragraphe quelques langages C S définis sur le vocabulaire terminal, $\mathfrak{V}_T = \{ A \}$, réduit à une seule lettre. Une phrase d'un tel langage, A^k, est donc entièrement caractérisée par son exposant k. On sait qu'un tel langage infini, n'est C F que s'il contient un sous-ensemble dont les phrases ordonnées par longueur croissante est une suite pour laquelle les exposants forment une progression arithmétique. Le lecteur pourra ainsi vérifier facilement qu'aucun des langages indiqués ici n'est C F. Ces langages ont été choisis car ils seront étudiés à un autre point de vue par R. Guedj dans une thèse de 3e cycle à paraître.

8.1.1. 1er exemple :

$$\mathfrak{L}_{(i)} = \{ A^{i^n} \mid n \in \mathbf{N} \}, \qquad i \in \mathbf{N} \qquad i \geqslant 2$$

On va décrire une grammaire C S avec marquant engendrant $\mathfrak{L}_{(3)}$ (la généralisation se ferait sans difficulté) :

$$\mathfrak{G}_{(3)} = (\mathfrak{V}_{(3)}, \mathfrak{V}_T, S, \#, \mathfrak{R}_{(3)})$$

où :

— $\mathfrak{V}_{(3)} = \{ S, \#, A_1, A_2, A \}$,

— $\mathcal{R}_{(3)}$ comporte :

la règle initiale ($\# S \#$, $\# A_1 \#$)
les règles de production ($\# A_1$, $\# A_2$) (début)

$$(A_2A_1,\ A_1^3A_2)$$
$$(A_2 \#,\ A_1^3 \#) \qquad \text{(fin)}$$

la règle terminale (A_1, A).

Il est facile de vérifier que :

$$\mathfrak{L}[\mathfrak{G}_{(3)}] = \mathfrak{L}_{(3)} = \{ A^{3^n} \mid n \in \mathbf{N} \}.$$

En vertu de la stabilité du produit pour la classe des langages C S (§ 6.2) le langage

$$\mathfrak{L}_{(2)} \cdot \mathfrak{L}_{(3)} = \{ A^{2^n+3^m} \mid n \in \mathbf{N},\ m \in \mathbf{N} \}$$

est C S, mais on peut imposer une « contrainte » plus forte aux exposants.

8.1.2. 2e exemple :

$$\mathfrak{L}_{(2,3)} = \{ A^{2^n+3^n} \mid n \in \mathbf{N} \}$$

Soit $\mathfrak{G}_{(2,3)}$ la grammaire C S avec marquant

$$\mathfrak{G}_{(2,3)} = (\mathfrak{V}_{(2,3)},\ \mathfrak{V}_T,\ S,\ \#,\ \mathcal{R}_{(2,3)})$$

où :

— $\mathfrak{V}_{(2,3)} = (S,\ \#,\ A_1,\ A_2,\ B_1,\ B_2,\ A)$,
— $\mathcal{R}_{(3,3)}$ comporte :

la règle initiale ($\# S \#$, $\# A_1B_1 \#$),
les règles de production ($\# A_1$, $\# A_2$) (début)

$$(A_2A_1,\ A_1^2A_2)\ ;\quad (A_2B_1,\ A_1^2B_2)\ ;\quad (B_2B_1,\ B_1^3B_2)\ ;\quad (B_2 \#,\ B_1^3 \#) \qquad \text{(fin)}$$

et les règles terminales (A_1, A); (B_1, A).

Les dérivations selon cette grammaire ressemblent beaucoup aux dérivations selon la grammaire $\mathfrak{G}_{(3)}$ de l'exemple précédent; on vérifiera que :

$$\mathfrak{L}[\mathfrak{G}_{(2,3)}] = \mathfrak{L}_{(2,3)} = \{ A^{2^n+3^n} \mid n \in \mathbf{N} \}.$$

Ce résultat serait évidemment aussi susceptible de généralisation.

8.1.3. 3e exemple :

$$\mathfrak{L}(!) = \{ A^{n!} \mid n \in \mathbf{N}^* \}$$

Considérons la grammaire C S avec marquant :

$$\mathfrak{G}(!) = (\mathfrak{V}(!), \mathfrak{V}_T, S, \#, \mathfrak{R}(!)).$$

— $\mathfrak{V}(!)$ comprend les deux symboles distingués S et #, les éléments de $\mathfrak{V}_1$:

$$\mathfrak{V}_1 = \{ A_1, A_2, A_3, B, C, C_1, C_2, D \}$$

et ceux de

$$\mathfrak{V}'_1 = \{ X' \mid X \in \mathfrak{V}_1 \} \quad , \quad \mathfrak{V}''_1 = \{ X'' \mid X \in \mathfrak{V}_1 \} \quad , \quad \mathfrak{V}'''_1 = \{ X''' \mid X \in \mathfrak{V}_1 \}$$

et le symbole terminal A.

Nous allons décrire les règles de la grammaire $\mathfrak{G}(!)$ en indiquant les dérivations qu'elles permettent. L'idée directrice est la suivante : on passe de $A^{n!}$ à $A^{(n+1)!}$ en « copiant » $(n + 1)$ fois la première phrase $A^{n!}$:

$$A^{(n+1)!} = \underbrace{A^{n!}A^{n!} \ldots A^{n!}}_{(n+1) \text{ fois}}$$

Ainsi au niveau non terminal le passage de $A^{3!}$ à $A^{4!}$ correspondra à la dérivation :

$$\# A_1A_1A_1B^3 \# \underset{\mathfrak{G}(!)}{\Longrightarrow} \# A_1A_1A_1A_1B^{20} \#,$$

le nombre de symboles A_1 permet de savoir à quelle étape on est rendu. On montrera comment les règles décrites permettent de réaliser la dérivation précédente.

I. — Règle initiale (# S #, # A_1A_1 #).

II. — Règles permettant les dérivations du type :

$$\# A_1A_1A_1B^3 \# \underset{\mathfrak{G}(!)}{\Longrightarrow} \# DA_1A_1B^3(C_1)^6 \#$$

. (# A_1, # DA_2) (début)
. (A_2A_1, A_1A_2C); (A_2B, BA_2C) : règles de « recopiage »
. (CA_1, A_1C); (CB, BC)
(C #, C_1 #); (CC_1, C_1C_1) } mise en place des nouveaux symboles.
. (A_2C_1, C'_1C_1)
(YX', $Y'X$) $\forall X \in \mathfrak{V}_1$
$\forall Y \in \mathfrak{V}_1 - \{ A_3, D \}$ } passage à l'étape suivante.

Par la suite les symboles X et Y désigneront un symbole quelconque de $\mathfrak{V}_1$, les restrictions faites seront seules indiquées.

III. — Règles permettant les dérivations du type

$$\# DA_1A_1B^3(C_1)^6 \# \underset{\mathfrak{G}(!)}{\Longrightarrow} \# DA_1A_3B^3(C_1C_2C_2)^6 \#$$

. (DA'_1, DA''_3) ou $(A_3A'_1, A_1A''_3)$: début de la nouvelle étape.
. $(X''Y, XY'')$ $X \neq C_1$.
. (C''_1Y, C_1C_2Y''), $(C''_1 \#, C'_1C_2 \#)$: règles de « recopiage ».
. $(C''_2 \#, C'_2 \#)$.

D'une façon générale s'il s'agit du passage de $A^{n!}$ à $A^{(n+1)!}$ cette étape recommence $(n-1)$ fois.

IV. — Règles permettant les dérivations du type

$$\# DA_1A_3B^3(C_1C_2C_2)^6 \# \underset{\mathfrak{G}(!)}{\Longrightarrow} \# DA_1A_1A_1B^{20} \#$$

. $(A_3C'_1, A'''_1A'''_1)$ ou $(A_3B', A'''_1A'''_1)$: début, un nouveau symbole A^1 apparaît en vue d'une nouvelle étape éventuelle.
. $(X'''Y, XY''')$ $X \neq C_1$ et $X \neq C_2$.
. (C'''_1Y, BY''').
. (C'''_2Y, BY''') ou $(C'''_2 \#, B \#)$.

V. — Enfin les règles

. $(XA'''_1, X'''A_1)$,
. $(\# D''', \# A_1)$,

achèvent un développement complet (dans l'exemple choisi on aboutira à : $\# A_1A_1A_1A_1B^{20} \#$), et l'on pourra recommencer un nouveau développement (on passerait à : $\# A_1A_1A_1A_1A_1B^{115} \#$ en considérant le même exemple) ou appliquer les :

VI. — Règles terminales :

. $(\# S \#, \# A \#)$;
. (A_1, A);
. (B, A).

On vérifiera que :

$$\mathfrak{L}[\mathfrak{G}(!)] = \mathfrak{L}(!) = \{ A^{n!} \mid n \in \mathbf{N}^* \}$$

8.1.4. 4e **exemple :** Nombres premiers et langages C S.

Nous avons aussi démontré le résultat suivant : le langage

$$\mathfrak{L}(p) = \{ A^p \mid p \in \mathbf{N}, p \text{ premier} \}$$

est C S.

8.2. Nouvel énoncé du grand théorème de Fermat.

Le théorème de Fermat, considéré ici, énonce que les équations :

$$x^n + y^n = z^n,$$

pour n entier supérieur ou égal à 3, n'admettent pas de solutions à valeurs entières, i. e. :

$$\nexists\,(x, y, z) \in \mathbf{N}^* \times \mathbf{N}^* \times \mathbf{N}^* : \qquad x^n + y^n = z^n \quad , \quad n \in \mathbf{N} \quad n \geqslant 3.$$

Jusqu'à ce jour, on ne sait si ce théorème est vrai.

Avant d'en donner un énoncé faisant intervenir certains langages, nous allons prouver que ces langages sont C S.

8.2.1 $$\mathfrak{L}^{(i)} = \{ \mathrm{A}^{n^i} \mid n \in \mathbf{N}^* \}, \qquad \{ i \in \mathbf{N},\ i \geqslant 3 \}$$

On va étudier tout d'abord le cas particulier où $i = 3$

$$\mathfrak{L}^{(3)} = \{ \mathrm{A}^{n^3} \mid n \in \mathbf{N}^* \}$$

en définissant une grammaire C S avec marquant

$$\mathfrak{G}^{(3)} = (\mathfrak{V}^{(3)}, \mathfrak{V}_T, \mathrm{S}, \#, \mathfrak{R}^{(3)})$$

l'engendrant. On s'inspirera de la grammaire précédente $\mathfrak{G}(!)$, et on considérera comme exemple de génération, la génération de la phrase $\mathrm{A}^{4^3} = (\mathrm{A}^{4^2})^4 = \mathrm{A}^{64}$, en distinguant dans la $\#\,\mathrm{S}\,\#$-dérivation de cette phrase les étapes suivantes :

$$\#\,\mathrm{S}\,\# \underset{\mathfrak{G}^{(3)}}{\Longrightarrow} \#\,\mathrm{DA_1A_1A_1}\,\# \underset{\mathfrak{G}^{(3)}}{\Longrightarrow} \#\,\mathrm{DA_1A_1A_1B^{12}}\,\#$$
$$\underset{\mathfrak{G}^{(3)}}{\Longrightarrow} \#\,\mathrm{A_1A_1A_1A_1B^{60}}\,\# \underset{\mathfrak{G}^{(3)}}{\Longrightarrow} \#\,\mathrm{A^{64}}\,\#$$

. Le vocabulaire de la grammaire $\mathfrak{G}^{(3)}$ est le même que celui de $\mathfrak{G}(!)$:

$$\mathfrak{V}^{(3)} = \mathfrak{V}(!).$$

. On a distingué dans la $\#\,\mathrm{S}\,\#$-dérivation de $\#\,\mathrm{A^{64}}\,\#$ quatre étapes; décrivons les quatre groupes de règles permettant de réaliser chacune de ces étapes.

I′. — Règles permettant les dérivations du type

$$\#\,\mathrm{S}\,\# \underset{\mathfrak{G}^{(3)}}{\Longrightarrow} \#\,\mathrm{DA_1A_1A_1}\,\#$$

. (S, $\mathrm{SA_1}$).
. (S, D).

II′. — Règles permettant les dérivations du type :

$$\#\,\mathrm{DA_1A_1A_1}\,\# \underset{\mathfrak{G}^{(3)}}{\Longrightarrow} \#\,\mathrm{DA_1A_1A_1B^{12}}\,\#$$

Il s'agit des règles du groupe II, III, IV de la grammaire $\mathfrak{G}(!)$ seules, la règle ($\#\,\mathrm{A_1}$, $\#\,\mathrm{DA_2}$) du début de cette étape est remplacée par :
. ($\mathrm{DA_1}$, $\mathrm{DA_3A_2C}$).

et les règles $(A_3C'_1, A'''_1A'''_1)$, $(A_3B', A'''_1A'''_1)$ de IV par :

. $(A_3C'_1, A'''_1C'''_1)$ et
. (A_3B', A'''_1B''').

Les règles (A_2B, BA_2C) et (CB, BC) du groupe II n'interviendront qu'à l'étape suivante et la règle $(C'''_1 \#, B \#)$ doit être ajoutée au groupe IV. On peut distinguer dans la dérivation précédente les phases indiquées ci-après :

$$\# DA_1A_1A_1 \# \underset{\mathfrak{G}^{(3)}}{\overset{II}{\Longrightarrow}} \# DA_3A_1A_1(C_1)^4 \# \underset{\mathfrak{G}^{(3)}}{\overset{III}{\Longrightarrow}} \# DA_1A_1A_3(C_1C_2C_2)^4 \# \underset{\mathfrak{G}^{(3)}}{\overset{IV}{\Longrightarrow}} \# DA_1A_1A_1B^{12} \#$$

III'. — Règles permettant les dérivations du type

$$\# DA_1A_1A_1B^{12} \# \underset{\mathfrak{G}^{(3)}}{\Longrightarrow} \# A_1A_1A_1A_1B^{60} \#$$

Cette étape commence par application de la règle

. $(DA'''_1, A_1A_3A_2C)$ après application des règles,
. $(XA'''_1, X'''A_1)$, $X \neq D$

puis on applique les règles de l'étape précédente II' auxquelles on ajoute la règle :

. $(\# A'''_1, \# A_1)$;

on peut décomposer la dérivation indiquée de la façon suivante :

$$\# DA_1A_1A_1B^{12} \# \underset{\mathfrak{G}^{(3)}}{\Longrightarrow} \# A_1A_3A_1A_1B^{12}(C_1)^{16} \# \underset{\mathfrak{G}^{(3)}}{\Longrightarrow} \# A_1A_1A_1A_3B^{12}(C_1C_2C_2)^{16} \# \underset{\mathfrak{G}^{(3)}}{\Longrightarrow} \# A_1A_1A_1A_1B^{60} \#$$

IV'. — Règles terminales : celles de la grammaire $\mathfrak{G}(!)$

. $(\# S \#, \# A \#)$,
. (A_1, A).
. (B, A).

Dans le cas général, $i \geqslant 3$, il suffit d'introduire des symboles non terminaux supplémentaires :

$$D_{i-3}, D_{i-4}, \ldots, D_1, \quad D \text{ (pour } D_0)$$

de remplacer la règle (S, D) du groupe I', et la règle (DA_1, DA_3A_2C) du groupe II', de $\mathfrak{G}^{(3)}$, respectivement par les règles (S, D_{i-3}) et $(D_{i-3}A_1, D_{i-3}A_3A_2C)$ et d'ajouter à la règle $(DA'''_1, A_1A_3A_2C)$ du groupe III', les règles $(D_jA'''_1, D_{j-1}A_3A_2C)$ pour tout j, $j \in (i-3)$.

8.2.2. Théorème de Fermat.

Les langages C S étant stables pour le produit (§ 6.2) et pour l'intersection (§ 7.3.2) le théorème de Fermat peut s'énoncer :

— Pour tout entier i supérieur ou égal à 3 les langages C S

$$(\mathfrak{L}^{(i)} \cdot \mathfrak{L}^{(i)}) \cap \mathfrak{L}^{(i)}$$

sont vides, i. e. :

$$\forall i, i \in \mathbf{N} \qquad i \geqslant 3 : \qquad (\mathfrak{L}^{(i)} \cdot \mathfrak{L}^{(i)}) \cap \mathfrak{L}^{(i)} = \emptyset.$$

En effet si l'un de ces langages n'était pas vide, il existerait un triplet d'entiers (n, m, k) tel que :

$$A^{i^n} \cdot A^{i^m} = A^{i^k}, \text{ i. e. : } \qquad i^n + i^m = i^k.$$

Pour chacun des langages précédents on peut déterminer une grammaire C S qui l'engendre. Mais le problème général, de savoir si le langage, engendré par une grammaire C S arbitraire, est vide, est indécidable [3] (le problème est décidable dans le cas des grammaires C F). Lever cette indécidabilité pour les grammaires particulières considérées ici, c'est résoudre le problème posé par le théorème de Fermat.

8.2.3. Cas particulier :

$$\mathfrak{L}^{(2)} = \{ A^{n^2} \mid n \in \mathbf{N} \}$$

Il est facile de voir que le langage $\mathfrak{L}^{(2)}$ peut être engendré par une grammaire déduite de $\mathfrak{G}^{(3)}$ en y remplaçant les règles du groupe III′ par celles du groupe V de $\mathfrak{G}(!)$. Mais on peut définir une grammaire plus simple en se basant sur le fait que l'on passe d'un carré, n^2, au carré immédiatement supérieur $(n+1)^2$, en ajoutant le nombre impair $(2n+1)$. Soit donc la grammaire C S avec marquant :

$$\mathfrak{G}^{(2)} = (\mathfrak{V}^{(2)}, \mathfrak{V}_T, S, \#, \mathfrak{R}^{(2)})$$

ayant pour vocabulaire non terminal :

$$\mathfrak{V}_N^{(2)} = (S, \#, A_1, A_2, B, B_1)$$

pour vocabulaire terminal :

$$\mathfrak{V}_T = \{ A \}$$

et pour règles :

. (S, B_1); $(\# A_1, \# A_2)$.
. (A_2A_1, A_1A_2); (A_2B, A_1B_1).
. (B_1A_1, A_1BA_2); (B_1B, A_1BB_1).
. $B_1 \#$, $A_1BBB \#)$.
. (A_1, A); (B, A).

Dans la $\# S \#$-dérivation de $\# A^{16} \#$, selon $\mathfrak{G}^{(2)}$, on peut distinguer les étapes suivantes :

$$\# S \# \xrightarrow[\mathfrak{G}^{(2)}]{} \# B_1 \# \xrightarrow[\mathfrak{G}^{(2)}]{} \# A_1BBB \# \xRightarrow[\mathfrak{G}^{(2)}]{} \# A_1A_1BA_1BA_1BBB \# \xRightarrow[\mathfrak{G}^{(2)}]{} \# A_1A_1A_1BA_1A_1BA_1A_1BA_1BA_1BBB \# \xRightarrow[\mathfrak{G}^{(2)}]{} \# A^{16} \#$$

On prouvera aisément que :

$$\mathfrak{L}[\mathfrak{G}^{(2)}] = \mathfrak{L}^{(2)}.$$

En vertu de la stabilité du produit et de l'intersection pour l'ensemble des langages C S on peut dire que le langage, non vide :

$$(\mathfrak{L}^{(2)}.\mathfrak{L}^{(2)}) \cap \mathfrak{L}^{(2)} = \{ A^n \mid n \in \mathbf{N}, \exists (m, h, k) \in \mathbf{N} \times \mathbf{N} \times \mathbf{N} \quad : \quad n = m^2 = h^2 + k^2 \}$$

est C S.

8.3. Langages C S et règles de mariage dans les sociétés primitives [7].

Dans certaines sociétés, les mariages sont soumis à des règles très strictes. Nous ne nous intéressons pas aux clans de ces sociétés, mais aux différents types de mariages possibles. Les règles de mariage considérées ici sont caractérisées par les axiomes suivants :

. Axiome 1. — On assigne à chaque individu un type de mariage.

. Axiome 2. — Deux individus ne sont autorisés à se marier que s'ils ont le même type.

. Axiome 3. — Le type d'un individu est déterminé par son sexe et par le type de ses parents.

. Axiome 4. — Deux garçons (ou deux filles) dont les parents sont de types différents sont eux-mêmes de types différents.

Exemple : Dans le système Kariera (Australie), il existe quatre types de mariages, notés M_1, M_2, M_3, M_4, attribués selon le tableau suivant :

Type des parents	Type du fils	Type de la fille
M_1	M_3	M_4
M_2	M_4	M_3
M_3	M_1	M_2
M_4	M_2	M_1

Dans le cas général, s'il existe n types de mariage, désignés par les symboles : $M_1, M_2, \ldots, M_n$, les axiomes 3 et 4 impliquent l'existence de deux permutations g et f de l'ensemble $)n) = \{1, 2, \ldots, n\}$ telles que : $M_{g(i)}$ (resp. $M_{f(i)}$) désigne le type des parents d'un homme H_i (resp. d'une femme F_i) de type M_i, $i \in)n)$. Les parents d'un homme de type M_i sont donc parfaitement définis quant à leur type et se noteront $H_{g(i)}F_{g(i)}$; de même les grands-parents pourront être désignés par le symbolisme suivant :

$$H_{g[g(i)]}F_{g[g(i)]}H_{f[g(i)]}F_{f[g(i)]},$$

on a tenu compte de l'axiome 2, et on a toujours écrit le père à gauche, la mère à droite. En fait le type des parents de H_i, à la deuxième génération est parfaitement défini par le symbolisme : $M_{g[g(i)]}M_{f[g(i)]}$. D'une façon précise, étant donné un individu de la société on désignera sa « signature clanique » d'ordre k par la suite des 2^{k-1} types de ses 2^k ancêtres à la $k^{\text{ième}}$ génération; c'est un mot, de longueur 2^{k-1}, défini sur le vocabulaire :

$$\mathfrak{V}_T = \{M_1, M_2, \ldots, M_n\}.$$

Le langage considéré sera donc l'ensemble de toutes les « signatures claniques » possibles pour une telle société. Montrons que ce langage est C S en esquissant une grammaire C S avec marquant l'engendrant :

$$\mathfrak{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \#, \mathfrak{R})$$

où :

— $\mathfrak{V}_T = \{M_1, M_2, \ldots, M_n\}$
— $\mathfrak{V}_N = \{S, \#, H_i, F_i, H'_i, F'_i \mid \forall i : i \in)n)\}$
— $\mathfrak{R}$ comporte trois groupes de règles définies $\forall i \in)n)$, $\forall j \in)n)$:

. Règles initiales :

. (S, H_i); (S, F_i).

. Règles de développement (permettant de considérer les ancêtres d'un individu) :

· $(\# H_i, \# H_{g(i)}F'_{g(i)})$ ou $(\# F_i, \# H_{f(i)}F'_{f(i)})$
· $(F'_jH_i, F_jH_{g(i)}F'_{g(i)})$ ou $(F'_jF_i, F_jH_{f(i)}F'_{f(i)})$
· $(F'_j \#, F_j \#)$

. Règles terminales :

· $(H_i, M_{g(i)})$
· $(F_i, M_{f(i)})$.

On remarquera que l'homme H_i de type M_i, et la femme F_j de type M_j, ont mêmes signatures claniques de tous ordres, si et seulement si :

$$g(i) = f(j).$$

Il en résulte que le nombre de signatures claniques distinctes d'ordre k, est toujours égal à n, le nombre de types de mariage, quel que soit k. Ainsi, dans le système Kariera, les quatre signatures claniques d'ordre $k = 4$ sont :

. $M_1\ M_2\ M_2\ M_1\ M_2\ M_1\ M_1\ M_2$
. $M_2\ M_1\ M_1\ M_2\ M_1\ M_2\ M_2\ M_1$
. $M_3\ M_4\ M_4\ M_3\ M_4\ M_3\ M_3\ M_4$
. $M_4\ M_3\ M_3\ M_4\ M_3\ M_4\ M_4\ M_3$.

9. UNE GÉNÉRALISATION DES GRAMMAIRES CONTEXT-FREE DE CHOMSKY

9.1. Composition et expression [*1*].

Nous avons vu (§ 3.1.3) qu'à toute grammaire de constituants

$$\mathcal{G} = (\mathfrak{V}, \mathfrak{V}_T, S, \mathcal{R})$$

on peut associer une *grammaire syntaxique* équivalente où l'on distingue les règles non terminales des règles terminales, $\mathcal{R} = \mathcal{R}_N \cup \mathcal{R}_T$. En linguistique, au lieu de parler de règles terminales, on dirait que l'on a groupé les mots de la langue (qui constituent le vocabulaire terminal

$$\mathfrak{V}_T = \{ D_i \mid i \in)p) \})$$

suivant les différentes parties syntaxiques qui constituent elles le vocabulaire non terminal : $\mathfrak{V}_N = \{ A_i \mid i \in (n) \}$, $A_0 = S$; ainsi à chaque symbole de $\mathfrak{V}_N$, A_i, se trouve associé un sous-ensemble de $\mathfrak{V}_T$, que l'on notera $\{ A_i \}$:

$$\{ A_i \} = \{ D_{ij} \mid D_{ij} \in \mathfrak{V}_T, \qquad (A_i, D_{ij}) \in \mathcal{R}_T \}$$

Ainsi $\{ S \}$ serait l'ensemble des phrases d'un seul mot : oh !, bonjour !, silence !, viens !, etc. Les mots du vocabulaire terminal ayant ainsi été classés suivant les diverses parties syntaxiques, la grammaire se trouve caractérisée par l'ensemble des règles non terminales qui sont essentiellement des règles de syntaxe. On peut, comme Chomsky, partir de l'unité la plus

grande, la phrase (S), pour la diviser en unités de plus en plus petites : c'est le point de vue de la *dérivation*; ou au contraire, partir des éléments pour construire avec eux des unités de plus en plus grandes : c'est le point de vue de la *composition.* Ce point de vue intéresse le codeur, le locuteur : avec des membres de phrase faire des phrases. Les deux points de vue sont, d'ailleurs, proches l'un de l'autre, si

$\widehat{A} \longrightarrow \widehat{B}$ est une règle de dérivation (notée encore $(\widehat{A}, \widehat{B})$),

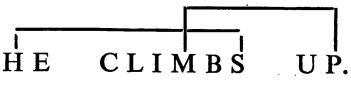
$\widehat{B} \longrightarrow \widehat{A}$ est la règle de composition correspondante.

Il suffit d'inverser les flèches.

Mais dans les grammaires de constituants de Chomsky on ne considère qu'une opération d'assemblage c'est le produit de juxtaposition, et on ne peut donc par ce formalisme qu'étudier les textes présentant un caractère strictement monoïdal. Or l'étude de textes (linguistiques, mathématiques, ...) fait apparaître fréquemment des mots fortement liés entre eux bien que ne se suivant pas immédiatement dans la chaîne; d'où l'idée, considérée notamment par J. P. Benzécri, d'introduire dans le formalisme linguistique des *mots non connexes* ou mots à plusieurs composantes, comme : « ne ... pas » en français, « bring ... up » en anglais, et des signes à plusieurs insertions comme les parenthèses $(\overline{\ldots})$, le $\overset{\sqcap}{\underset{\sqcup}{\square}}$ de Bourbaki, ceci en plus des mots et symboles ordinaires considérés alors comme des mots à une composante, des signes à une insertion... Au lieu de pouvoir simplement juxtaposer symboles et mots, on imbriquera signes et mots à plusieurs insertions de diverses façons.

Cela nous permet de distinguer dans la langue 2 niveaux : celui de la composition décrit ci-dessus et que l'on peut concevoir suffisamment général pour convenir à plusieurs langues et celui de l'*expression* variant d'une langue à l'autre. En effet si la règle de syntaxe : une phrase se compose d'un sujet et d'un verbe, se trouve dans plusieurs langues, cette règle s'exprimera différemment suivant chacune. Ainsi en anglais le verbe devra sans doute être considéré comme un mot à deux composantes : le verbe lui-même et sa postposition (éventuellement absente, auquel cas la deuxième composante serait réduite au mot vide) par exemple : C L I M B-U P; le sujet sera aussi un mot à deux composantes, par exemple : H E-S.

Ainsi « il grimpe » s'exprimera en anglais :



Le cadre mathématique adapté à de telles structures est celui des catégories. Ainsi J. P. Benzécri a étudié [2] une catégorie, appelée S E G, qui permettrait d'approcher la structure de l'expression. Nous allons nous intéresser tout d'abord au point de vue de la composition.

9.2. Catégorie S Y N.

9.2.1. Définitions.

On ne trouvera ici sur les catégories que les définitions qui nous sont nécessaires (pour une étude plus développée du sujet on se rapportera aux divers ouvrages spécialisés).

1° **Catégorie.** — Une catégorie $\mathcal{C}$ est la donnée :

1) d'une classe non vide d'objets;

2) pour tout couple (A, B) d'objets de $\mathcal{C}$, d'un ensemble noté Hom (A, B) et appelé ensemble des morphismes de A dans B. On parle aussi de morphismes de source A, de but B et on les désignera par des lettres grecques en précisant généralement la source et le but en indice supérieur et inférieur respectivement, ainsi : $\alpha_B^A \in \text{Hom}\ (A, B)$;

3) pour tout triple d'objets A, B, C de $\mathcal{C}$ d'une application de Hom (A, B) $\times$ Hom (B, C) dans Hom (A, C), l'application de composition notée

$$(\alpha_B^A,\ \beta_C^B) \rightsquigarrow \beta_C^B \circ \alpha_B^A$$

satisfaisant aux axiomes suivants :

a. L'application de composition est associative :

$$(\gamma_D^C \circ \beta_C^B) \circ \alpha_B^A = \gamma_D^C \circ (\beta_C^B \circ \alpha_B^A)$$

lorsque ceci est défini;

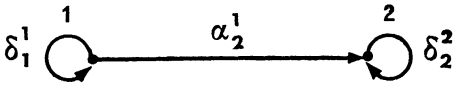
b. Pour tout objet A de $\mathcal{C}$, il existe un élément de Hom (A, A), que nous noterons δ_A^A et tel que :

$\forall \alpha_B^A \in \text{Hom}\ (A, B)$ et $\beta_A^B \in \text{Hom}\ (B, A)$

$$\alpha_B^A \circ \delta_A^A = \alpha_B^A \quad \text{et} \quad \delta_A^A \circ \beta_A^B = \beta_A^B.$$

EXEMPLES. — Ainsi nous avons la catégorie E N S dont les objets sont les ensembles et les morphismes les applications ensemblistes munies de la loi de composition usuelle. Mais les objets ne sont pas nécessairement des

ensembles. Dans ce cas la catégorie peut être considérée comme un diagramme : les points sont les objets et les flèches reliant ces points sont les morphismes. Soit par exemple la catégorie D I A G



réduite à deux objets, notés 1 et 2, et à trois morphismes avec les compositions évidentes; ici Hom (2, 1) est vide.

2° **Catégorie à produit direct.** — Étant donné deux objets A et B d'une catégorie C, on dira qu'un troisième objet C est produit direct de A et de B par les morphismes (projections) π_A^C et π_B^C si quel que soit l'objet X et les morphismes α_A^X et β_B^X, il existe un morphisme unique γ_C^X tel que :

$$\pi_A^C \circ \gamma_C^X = \alpha_A^X$$

$$\pi_B^C \circ \gamma_C^X = \beta_B^X$$

On notera $C = A \times B$ et $\gamma = \alpha \times \beta$.

La catégorie C sera dite à produit direct si pour tout couple d'objets (A, B) on peut trouver un troisième objet C qui en est le produit direct par des projections convenables que nous noterons désormais $\delta_A^{A\times B}$ et $\delta_B^{A\times B}$. E N S est une catégorie où le produit direct de deux objets correspond au produit cartésien de ces deux ensembles.

C étant une catégorie à produit direct on pourra définir sur les morphismes une nouvelle opération, appelée produit.

Soient deux morphismes quelconques,

$$\alpha_C^A \quad \text{et} \quad \beta_D^B$$

le morphisme produit, noté $\alpha \otimes \beta$, élément de Hom $(A \times B, C \times D)$ est défini comme suit :

$$\alpha_C^A \otimes \beta_D^B = (\alpha_C^A \circ \delta_A^{A\times B}) \times (\beta_D^B \circ \delta_B^{A\times B}).$$

Le lecteur remarquera que l'on a noté comme un produit cartésien, $\times$, le produit de deux morphismes de même source et comme un produit

tensoriel, $\otimes$, le produit de deux morphismes de sources non supposées identiques. Ces notations varient d'un auteur à l'autre. Celles adoptées ici correspondent à celles suggérées et choisies par J. P. Benzécri.

3° **Foncteur.** — Soient $\mathcal{C}$ et $\mathcal{C}'$ deux catégories. Un foncteur $\mathcal{F}$ de $\mathcal{C}$ dans $\mathcal{C}'$ est la donnée :

1) pour tout objet A de $\mathcal{C}$ d'un objet $\mathcal{F}(A)$ de $\mathcal{C}'$,

2) pour tout morphisme μ de A dans B d'un morphisme $\mathcal{F}(\mu)$ de $\mathcal{F}(A)$ dans $\mathcal{F}(B)$ tel que :

$$\mathcal{F}(\mu \circ \varphi) = \mathcal{F}(\mu) \circ \mathcal{F}(\varphi)$$

$$\mathcal{F}(\delta_A^A) = \delta_{\mathcal{F}(A)}^{\mathcal{F}(A)}$$

D'autre part, les foncteurs considérés par la suite seront toujours supposés compatibles avec le produit direct, c'est-à-dire, que si C est produit direct de A et B par les projections π et π', alors $\mathcal{F}(C)$ est produit direct de $\mathcal{F}(A)$ et $\mathcal{F}(B)$ par $\mathcal{F}(\pi)$ et $\mathcal{F}(\pi')$.

Nous sommes maintenant en mesure de définir une structure type correspondant à la composition dans une langue.

9.2.2. Catégorie S Y N.

Étant donné une grammaire de constituants de type context-free, supposée syntaxique (§ 3.1.3)

$$\mathcal{G} = (\mathcal{V}, \mathcal{V}_T, S, \mathcal{R})$$

la catégorie S Y N qui lui est associée est définie de la façon suivante :

— elle a pour objets les mots du monoïde libre $\mathcal{L}(\mathcal{V}_N)$.

Tout élément A de $\mathcal{V}_N$ s'appellera *objet-base*. Le produit direct défini sur les objets de S Y N correspond au produit de juxtaposition défini sur les mots de $\mathcal{L}(\mathcal{V}_N)$ et se notera de la même façon ; les morphismes projections associées s'écriront sous la forme $\delta_{\widehat{A}}^{\widehat{A}\widehat{B}}$; $\delta_{\widehat{A}}^{\widehat{A}}$ désignant le morphisme identique de Hom $(\widehat{A}, \widehat{A})$ pour tout $\widehat{A} \in \mathcal{L}(\mathcal{V}_N)$;

— en plus des morphismes-projections ainsi définis il y a les *morphismes-générateurs* en bijection avec les règles non terminales de la grammaire $\mathcal{G}$, $\mu_{A_i}^{\widehat{B}}$ sera le morphisme correspondant à la règle $A \to \widehat{B}$ de la grammaire.

Enfin il y a à considérer tous les morphismes se déduisant des précédents par composition et produit.

Cette catégorie S Y N peut être imaginée comme un diagramme dont les sommets correspondent aux mots de $\mathfrak{L}(\mathfrak{V}_N)$ et les flèches aux morphismes. Nous nous intéressons spécialement aux morphismes de but S car à chacun de ces morphismes correspond une structure syntaxique de phrase, correcte relativement à la grammaire $\mathfrak{G}$ associée. Ainsi nous allons définir la catégorie S Y N associée à la grammaire $\mathfrak{G}$, quelque peu modifiée, ayant servi d'exemple au § 1.5.2.

EXEMPLE. — Les objets-base de la catégorie correspondent aux éléments du vocabulaire non terminal de la grammaire $\mathfrak{G}$

$$\mathfrak{V}_N = \{ S, N, V, A_r, A_j, A_d, \overline{V} \}$$

chacun de ces symboles désignant la partie syntaxique indiquée au § 1.5.2.

— Les morphismes-générateurs déduits des règles de $\mathfrak{G}$ sont :

$$\alpha_S^{NV}; \quad \beta_N^{A_r NA_j}; \quad \gamma_{A_j}^{A_d A_j}; \quad \varepsilon_V^{VN}; \quad \eta_V^{VA_d}; \quad \theta_V^{\overline{V}}.$$

La phrase, dont la génération par $\mathfrak{G}$ avait été décrite à l'aide d'un arbre : UN ÉTUDIANT TRÈS DOUÉ NE RÉSOUT PAS TOUJOURS UN PROBLÈME DIFFICILE, aura dans S Y N une structure syntaxique définie par le morphisme suivant de but S :

$$\alpha_S^{NV} \left\{ \left[\beta_N^{A_r NA_j} \circ \left(\delta_{A_r N}^{A_r N} \otimes \gamma_{A_j}^{A_d A_j} \right) \right] \otimes \left[\varepsilon_V^{VN} \circ \left(\left(\eta_V^{VA_d} \circ \left(\theta_V^{\overline{V}} \otimes \delta_{A_d}^{A_d} \right) \right) \otimes \beta_N^{A_r NA_j} \right) \right] \right\}$$

Dans la grammaire en question les règles terminales étant peu nombreuses, les ensembles { N }, { V }, etc. (notation définie au § 9.1) se réduisent à un ou 2 éléments. On peut évidemment lever cette restriction et nous intéresser à l'ensemble des mots de la langue française catalogués suivant les différentes parties syntaxiques. Il en résulte que le morphisme précédent représente la structure syntaxique de toutes les phrases bâties sur le modèle de celle indiquée ci-dessus. Ainsi en passant dans la catégorie E N S par un foncteur convenable les morphismes α, β, ... apparaissent comme des fonctions à 2, 3, ... arguments respectivement et le morphisme composé précédent peut s'écrire sous la forme fonctionnelle suivante :

$$\alpha \{ [\beta(.,.,\gamma(.,.))], [\varepsilon(\eta(\theta(.),.)), \beta(.,.,.)] \}$$

Plus précisément l'on passera de S Y N dans E N S par l'intermédiaire d'une troisième catégorie convenable, E X P ou catégorie d'expression.

9.3. Catégorie E X P.

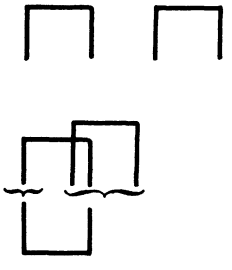
9.3.1. Définitions.

Nous allons nous intéresser uniquement aux langages pour lesquels au niveau de l'expression interviennent des constituants non connexes (cf. § 9.1 : CLIMB-UP). La catégorie adéquate est la catégorie S E G que nous ne décrirons pas ici d'une manière formelle (cf. [2]).

1° **Catégorie S E G.** — Les objets-bases sont dénombrés par les entiers 1, 2, ... il y en a autant que de type de mots, c'est-à-dire, que de nombres possibles d'insertions. On parle de « peignes » à 1, 2, ... dents, que l'on note : ⌊, ⌈⌉, ...

. Les morphismes correspondent à tous les procédés pour fabriquer un mot à plusieurs insertions (ou une liste de tels mots) à partir d'une liste de mots de types donnés. Nous allons considérer deux morphismes qui interviendront par la suite.

. Morphisme ρ :



Par un foncteur de S E G dans E N S, à ρ correspondra un procédé permettant de fabriquer d'une liste de deux mots ayant chacun deux composantes un troisième mot à deux composantes. En désignant aussi par ρ cette application ensembliste, on écrira :

$$\rho(\widehat{A} - \widehat{B}/\widehat{C} - \widehat{D}) = \widehat{A} - \widehat{C}\widehat{B}\widehat{D} ;$$

le tiret — sépare les composantes d'un mot non connexe, et la barre / les mots d'une liste.

Dans le schéma ci-dessus, la 1re (resp. la 3e) ligne désigne la source (resp. le but) du morphisme ρ. La 2e ligne indique comment se fait l'imbrication et la juxtaposition des dents de la source : on l'appellera le graphe du morphisme.

. Morphisme λ_2 :

C'est un morphisme simple que l'on appellera liaison d'ordre 2. Dans E N S on a :

$$\lambda_2(\widehat{A} - \widehat{B}) = \widehat{A}\widehat{B}$$

Le morphisme λ_2 ayant pour source le but du morphisme ρ, ces deux morphismes peuvent se composer, ainsi dans E N S :

$$\lambda_2 \circ \rho(\widehat{A} - \widehat{B}/\widehat{C} - \widehat{D}) = \widehat{A}\widehat{C}\widehat{B}\widehat{D}.$$

2° **Foncteur d'expression.** — Étant donné une catégorie S Y N très générale, une langue sera essentiellement caractérisée par un foncteur de S Y N dans E X P (ici S E G). A chaque objet-base de S Y N (symbole non terminal) correspond un objet-base de S E G (ainsi, pour la langue anglaise à V (verbe) correspondrait le peigne à deux dents ⌐¬|). De même à chaque morphisme de S Y N (règle de grammaire) correspond un morphisme de S E G qui régit l'assemblage considéré en tenant compte de la langue étudiée.

3° **Grammaires context-free à peignes.** — On a vu comment à une grammaire de constituants de type C F, définie par N. Chomsky on associe une catégorie S Y N (cf. 9.2.2). La catégorie d'expression associée est la catégorie M O N que l'on trouvera décrite dans [*1*]. Cela revient à dire que les mots considérés ne possèdent qu'une composante et que la seule opération définie sur les mots est le produit de juxtaposition.

. Si la catégorie d'expression est la catégorie S E G signalée ci-dessus, on parle de grammaires context-free à peignes (C F P, par abréviation). Dans une telle grammaire les symboles non terminaux sont des symboles à plusieurs insertions; les règles ayant un premier membre défini par un symbole à *n* insertions ont un mot à *n* composantes comme second membre, ce mot

est défini par une liste de symboles de base et un morphisme de S E G. Nous allons préciser ces notions sur un exemple tiré de [*1*] et revenir, par les règles, au point de vue de la dérivation.

9.3.2. Exemple : Construction des formes modales des verbes anglais.

La grammaire ci-dessous (inspirée de N. Chomsky) vise à construire les formes modales des verbes anglais. Elle comprend les règles non terminales :

$$\begin{aligned} S &\to \lambda_2 \circ \rho(N/V) \\ V &\to V_i \qquad\qquad i \in (3) \\ V_j &\to \rho(M_j/V_i) \qquad (i < j) \end{aligned}$$

et les règles terminales :

S → # THEY- ; S → # HE-S ; S → # THE # LINGUISTS-

Ici # représente le « blanc » d'un texte.

$$V_0 \to \# \text{ INSERT- } \# \quad ; \quad V_0 \to \# \text{ CLIMB-UP } \#$$

M₁ → # BE-EN ; M₂ → # BE-ING ; M₃ → # HAVE-EN

Dans ces règles V_i (verbe), N (sujet), M_j sont des symboles à deux insertions.

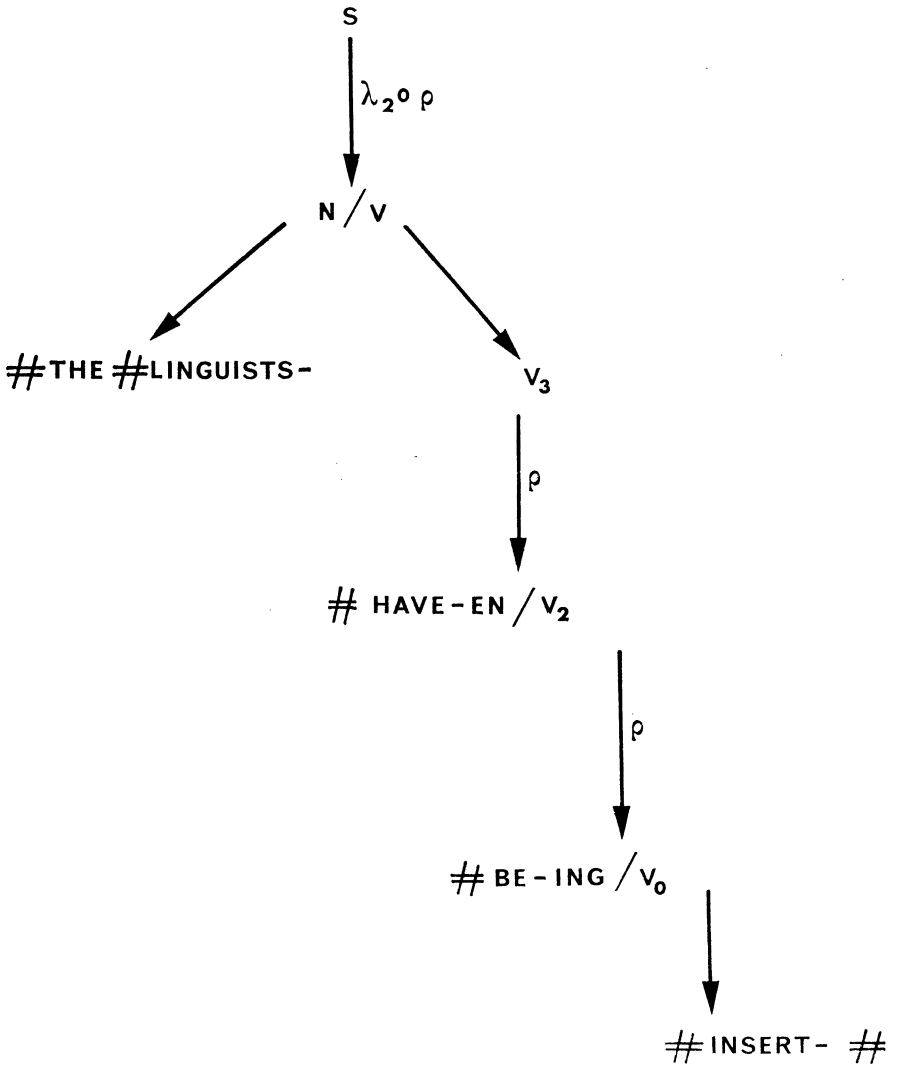
ρ et λ_2 correspondent (dans E N S) aux morphismes de S E G décrits ci-dessus (§ 9.3.1).

Pour engendrer une phrase selon cette grammaire, partons du symbole initial S (symbole à une insertion) et appliquons successivement les règles de la liste suivante :

$$\begin{aligned} S &\to \lambda_2 \circ \rho(N/V) \\ N &\to \# \text{ THE } \# \text{ LINGUISTS-} \\ V &\to V_3 \\ V_3 &\to \rho(M_3/V_2) \\ V_2 &\to \rho(M_2/V_0) \\ V_0 &\to \# \text{ INSERT- } \# \end{aligned}$$

On aura l'arbre de dérivation ci-contre.

Dans cet arbre, les flèches représentent des substitutions : au symbole placé en haut de la flèche, on substitue l'expression qui est en bas, ou, si



cette expression est une liste, le résultat de l'opération sur cette liste du morphisme indiqué à droite de la flèche. On obtient ainsi la phrase :

$\lambda_2 \circ \rho$(#THE#LINGUISTS/ρ(# HAVE-EN/ρ(#BE-ING/#INSERT-#)))

c'est-à-dire :

THE # LINGUISTS # HAVE # BEEN # INSERTING

9.4. Conclusion.

Dans ce dernier paragraphe nous avons associé à une grammaire de constituants de type context-free une catégorie S Y N. Cela nous a permis d'obtenir une généralisation des langages de type C F en considérant, au niveau de l'expression, la catégorie S E G (au lieu de M O N). Dans le cas d'une grammaire de constituants de type 1 ou de type 2 on pourrait aussi envisager de lui associer une catégorie S Y N adéquate...

Les langages C F P (langages engendrés par des grammaires C F P) constituent déjà une extension importante des langages C F au sens de Chomsky. On se contentera de faire remarquer que les langages C S suivants, déjà mentionnés dans cette étude, sont C F P (cf. R. Guedj, Thèse de 3e cycle, *à paraître*) :

. $\mathfrak{L}_1 = \{ A^n B^n C^n \mid n \in \mathbf{N}^* \}$ (§ 7.1.2)

. $\mathfrak{L}_2 = \{ AB, A^4, B^2A^3, B^4A^2, B^6A, B^8, A^3A^7, \ldots, A^{24}, \ldots \}$ (§ 5.2)

. $\mathfrak{L}_{(i)} = \{ A^{i^n} \mid n \in \mathbf{N} \}$ (§ 8.1.1).

Cependant nous avons démontré le résultat suivant :

THÉORÈME. — **Tout langage C F P est C S.**

Ce résultat a été complété par R. Guedj qui a montré qu'il existe des langages C S qui ne sont pas C F P. R. Guedj donne pour exemples des langages C S étudiés au § 8.1.

Mon désir est de consacrer un travail ultérieur à exposer le théorème précédent et étudier les grammaires à peignes.

BIBLIOGRAPHIE

[1] J. P. BENZÉCRI (Rennes, février 1965), Structures algébriques et constituants non connexes dans les grammaires.

[2] J. P. BENZÉCRI (Rennes, 1964), Cours de Linguistique Mathématique, 3e leçon.

[3] N. CHOMSKY, Formal Properties of Grammars, dans Handbook of Mathematical Psychology, vol. II (Ed. by D. Luce, E. Busch, E. Galanter), 1963, p. 323-418.

[4] S. GINSBURG et E. H. SPANNIER (juillet 1962), Quotient of Context-free languages.

[5] S. Y. KURODA (octobre 1963), Classes of Languages and Linear-bounded automata.

[6] P. S. LANDWEBER, Three theorems on phrase structure grammars of type 1, *Information and Control*, t. 6, 1963, p. 137-146.

[7] A. WEIL, Sur l'étude de certains types de lois de mariage (système Murngin) dans Appendice à la première partie, *Les structures élémentaires de la parenté*, par C. Lévi-Strauss, P. U. F., 1949.

[8] A. MARTINET, Éléments de linguistique générale ; Collection Armand Colin, n° 349.

TABLE DES MATIÈRES