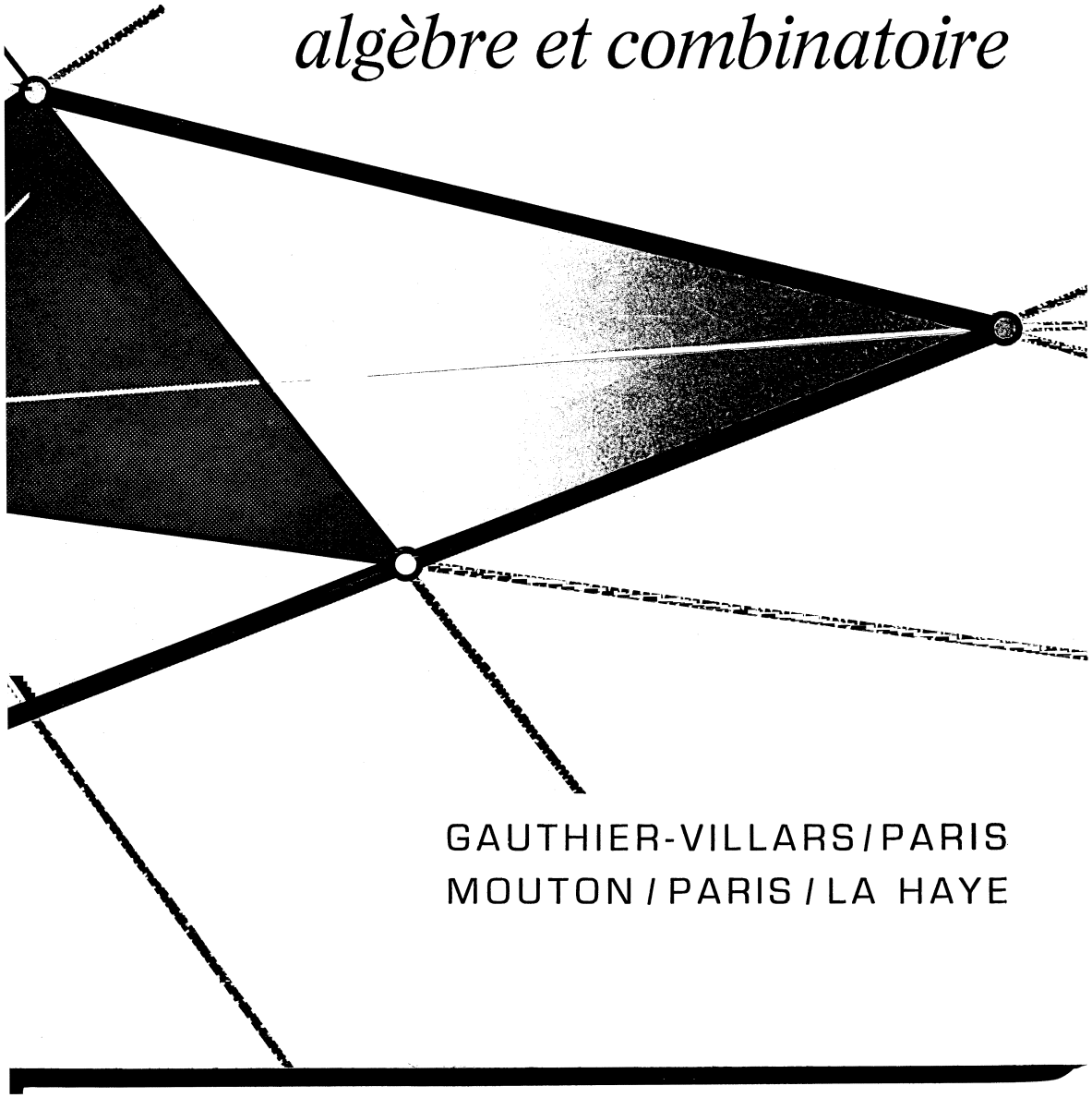
CAHIERS MATHEMATIQUES

3

algèbre et combinatoire



GAUTHIER-VILLARS / PARIS
MOUTON / PARIS / LA HAYE

M. P. SCHUTZENBERGER

Sur les contraintes définissant certains modèles formels de langage

L'un des objectifs partiels de la linguistique est l'établissement de grammaires, c'est-à-dire, pour une langue naturelle donnée, l'établissement d'un système explicite de règles qui permettent, en théorie du moins, de décider pour toute suite arbitraire de phonèmes si cette suite appartient ou non à l'ensemble des phrases grammaticalement correctes de la langue considérée. L'expérience semble indiquer qu'à un certain niveau une partie de ces règles est d'une nature assez simple pour pouvoir être discutée par des procédés purement formels. C'est là le cas des règles d'emploi des parenthèses ou, si l'on préfère, le formalisme de l'emboîtement des syntagmes et le but de cet exposé est de présenter quelques résultats obtenus dans cette direction par N. Chomsky, ses amis et leurs élèves ; je renvoie aux travaux de cet auteur et G. A. Miller pour une bibliographie complète de la question. Comme les problèmes proprement linguistiques ne seront pas abordés ici, il sera commode d'utiliser les notations suivantes qui ne risquent pas d'entraîner de confusion bien que les termes de « lettres » et de « mots » y soient utilisés d'une façon qui apparaîtra sans doute incongrue aux linguistes.

Soit X un ensemble *fini* dont les éléments sont appelés conventionnellement des *lettres*, F l'ensemble de toutes les séquences finies que l'on peut former avec ces lettres, ces séquences étant elles-mêmes appelées des *mots*. Techniquement, F est le *monoïde libre* engendré par X. Ceci veut dire que le mot vide (noté e) appartient à F et qu'à toute paire de mots f, f' de F est associé leur *produit* ff', c'est-à-dire par définition, le mot formé de f suivi de f'. Nous supposerons désormais que les lettres de X sont indexées par les nombres $\pm i$ ($1 \leqq i \leqq n$) c'est-à-dire que X est l'ensemble $\{ x_1, x_{-1}, x_2, x_{-2}, \ldots, x_n, x_{-n} \}$. Un mot f sera dit *réduit* s'il ne contient aucune paire de lettres consécutives ayant des indices opposés, c'est-à-dire

pour $f = x_{i_1}x_{i_2} \ldots x_{i_m}$ si $i_1 \neq -i_2$, $i_2 \neq -i_3$, $\ldots$, $i_{m-1} \neq -i_m$. Il est clair qu'à chaque mot f on peut associer un mot réduit, disons αf, obtenu en effaçant dans f toutes les paires de lettres consécutives dont les indices sont opposés et en répétant cette opération aussi longtemps qu'elle est possible. Par exemple

$$\alpha(x_1x_{-2}x_3x_{-3}x_2x_2x_{-1}x_1)$$

est égal à x_1x_2 à la suite des opérations suivantes: effacement de x_3x_{-3} et de $x_{-1}x_1$ ce qui donne $x_1x_{-2}x_2x_2$; effacement de $x_{-2}x_2$ ce qui donne finalement x_1x_2. On peut vérifier que l'ordre dans lequel sont effectuées les opérations d'effacement est sans effet sur le résultat final αf qui est donc, pour chaque f, un mot bien défini. Plus généralement, f et f' étant deux mots quelconques, on a l'identité: $\alpha(ff') = \alpha(\alpha f \alpha f')$. Par exemple, pour $f = x_1x_{-2}x_3x_{-3}$, $f' = x_2x_2x_{-1}x_1$, on a:

$$\alpha f = x_1x_{-2}, \qquad \alpha f' = x_2x_2$$

et comme plus haut $\alpha(x_1x_{-2}x_2x_2) = x_1x_2$. Cette construction définit un ensemble D de mots privilégiés: ceux dont la forme réduite est le mot vide e ou, si l'on veut, ceux qui peuvent être réduits à rien par les opérations successives d'effacement. Par exemple le mot

$$f = x_1x_{-2}x_3x_{-3}x_2x_2x_{-1}x_1x_{-2}x_{-1}$$

appartient à D puisqu'avec des notations évidentes l'on a:

$$f = x_1(x_{-2}(x_3x_{-3})x_2)(x_2(x_{-1}x_1)x_{-2})x_{-1}$$

par contre le mot

$$f' = x_1x_{-2}x_3x_{-3}x_2x_2x_{-1}x_1x_{-1}x_{-2}$$

obtenu en permutant les deux dernières lettres de f n'appartient pas à D parce que $\alpha f'$ est le mot non vide $x_1x_2x_{-1}x_{-2}$. D est défini algébriquement comme le noyau de l'homomorphisme β de F sur le groupe libre (engendré par l'ensemble X^+ des $x \in X$ d'indices positifs) qui satisfait pour chaque lettre x_i la condition que βx_{-i} soit l'inverse de βx_i (dans le groupe évidemment).

Soit d'autre part X_1 un sous-ensemble de X, V un ensemble de mots de deux lettres et $R(X_1, V)$ l'ensemble de tous les mots de F qui commencent par une lettre de X_1 et dont aucune paire de lettres consécutives ne forme un mot de V. Par exemple pour $X_1 = X^+$ comme plus haut et

$$V = \{ x_ix_j : i < 0 < j \},$$

$R(X_1, V)$ est l'ensemble des mots de la forme ff' où toutes les lettres de f ont des indices positifs (ce que l'on notera $f \in F_+$), et où toutes les lettres

de f' ont des indices négatifs. Nous dirons que l'ensemble $L = D \cap R(X_1, V)$ des mots appartenant à la fois à D et à $R(X_1, V)$ *est* un *langage CF* (Context Free) *standard.* Pour l'exemple qui vient d'être donné, L est formé de tous les mots de la forme $f\bar{f}$ où f appartient à F_+ et où $\bar{f}$ est obtenu en « retournant » le mot f, ce qui donne $\tilde{f}$, et en remplaçant dans $\tilde{f}$ chaque lettre x_i par la lettre d'indice opposé x_{-i} (Par exemple pour $f = x_1x_1x_3x_2$, on pose

$$\tilde{f} = x_2x_3x_1x_1 \qquad \text{et} \qquad \bar{f} = x_{-2}x_{-3}x_{-1}x_{-1}).$$

Plus généralement, soit donné un homomorphisme φ de F dans lui-même, c'est-à-dire pour chaque lettre x_i de X un certain mot φx_i (éventuellement vide). Considérant un langage CF standard L, on forme l'ensemble de tous les mots φf où f est dans L et ceci constitue ce que nous appellerons un langage CF *général.* Je rappelle que si

$$f = x_{i_1}x_{i_2}x_{i_3} \ldots x_{i_n}$$

le mot φf est défini comme le produit

$$\varphi x_{i_1}\varphi x_{i_2}\varphi x_{i_3} \ldots \varphi x_{i_n}$$

et φ n'est donc pas autre chose qu'une règle systématique de réécriture. Reprenant l'exemple donné plus haut de $L = \{ f\bar{f} : f \in F_+ \}$ et posant

$$\varphi x_i = \varphi x_{-i} = x_i$$

pour chaque x_i de X^+, on obtient le langage $M = \{ f\tilde{f} : f \in F_+ \}$ formé des *mots en miroir* (ou « palindromes ») $f\tilde{f}$ dont la seconde moitié $\tilde{f}$ est précisément la première moitié « retournée ».

La définition qui vient d'être donnée d'un langage CF général n'est pas la définition originale de Chomsky, mais il est facile de voir que ces deux définitions sont à très peu près équivalentes. Cette définition et les raisons pour lesquelles ces langages sont dits « context free » sont données dans l'ouvrage de Chomsky et Miller indiqué en références.

Ceci dit nous pouvons examiner les contraintes qu'imposent aux lettres d'un mot l'hypothèse que celui-ci appartient à un langage CF donné L et, afin de simplifier, nous supposerons que L est standard. Ces contraintes sont d'une double nature : les premières résultent de l'appartenance à $R(X_1, V)$ et peuvent être considérées comme purement *locales* en ce sens qu'un être dont la mémoire se limiterait à la dernière lettre lue pourrait vérifier pour une séquence arbitraire de lettres si elles sont ou non satisfaites. Il n'est pas utile de rappeler les illustrations linguistiques ou pseudo-linguistiques bien connues de ce type extrêmement banal de contraintes (cf. plus bas).

Par contre, les contraintes imposées par l'appartenance à D sont un

peu moins simples : elles peuvent s'étendre aussi loin que l'on veut et elles nécessitent en général une mémoire non bornée. En effet, d'après l'identité

$$\alpha(\alpha f \alpha f') = \alpha f f',$$

deux mots f_1 et f_2 sont tels qu'il existe un mot f'' pour lequel $f_1 f''$ et $f_2 f''$ appartiennent simultanément à D, si et seulement si on a $\alpha f_1 = \alpha f_2$. Réciproquement, quand cette relation est vérifiée pour tout f'' tel que $f_1 f''$ soit dans D on a aussi $f_2 f''$ dans D. Par conséquent, ayant déjà lu un segment initial f_1 d'un mot il est nécessaire de conserver trace en mémoire au moins de sa forme réduite αf_1 afin de pouvoir déterminer si le mot complet $f_1 f''$ appartient ou non à D. Notons qu'il existe $((2n-1)^k - 1)n(n-1)^{-1}$ mots réduits distincts de longueur au plus k ; donc pour vérifier par lecture séquentielle de gauche à droite l'appartenance à D d'un mot quelconque de longueur $\leqq 2k$, il faut disposer d'une mémoire pouvant accumuler approximativement $k(1 + \mathrm{Log}_2 n)$ bits d'informations. En outre, l'exemple donné plus haut de

$$\mathrm{L} = x_{i_1} x_{i_2} \ldots x_{i_m} x_{-i_m} \ldots x_{-i_2} x_{-i_1}$$

montre qu'une fraction arbitrairement grande de ces informations doit être gardée en mémoire un temps arbitrairement long, à savoir depuis le début jusqu'à la fin du mot. Par contre, et c'est ce qui fait peut-être l'intérêt des langages CF, les contraintes existant entre les lettres d'un mot de D n'interviennent chacune essentiellement qu'une fois et ne se croisent pas les unes les autres. Soit par exemple un mot $f \in \mathrm{D}$ de la forme $f_1 x_i f_2$ où f_1 ne contient ni la lettre x_i ni la lettre x_{-i}. Le mot f_2 est soumis à la condition d'avoir la forme $f_3 x_{-i} f_4$ où f_3 ne contient ni x_i ni x_{-i} et où en outre $\alpha f_3 = e$: c'est en ce sens que nous dirons que la contrainte imposée par la présence de x_i ne joue essentiellement qu'une fois, bien qu'évidemment il puisse y avoir un nombre arbitrairement grand de lettres x_i dont chacune impose l'existence d'une lettre x_{-i} qui en permet l'effacement ultérieur. De plus, dans ce même exemple, si

$$f_3 = f_5 x_j f_6$$

où f_5 ne contient aucune des lettres $x_{\pm j}$, le fait que l'on doive avoir $\alpha f_3 = e$ entraîne que f_6 soit égal à $f_7 x_{-j} f_8$ où cette fois $\alpha f_7 = e$. On a donc nécessairement :

$$f = f_1 x_i f_5 x_j f_7 x_{-j} f_8 x_{-i} f_4$$

avec

$$\alpha f_7 = \alpha f_5 f_8 = \alpha f_1 f_4 = e$$

et, dans un sens assez évident, la contrainte associée à la lettre x_i « passe par dessus » la contrainte associée à la lettre x_j sans se croiser avec elle.

Par contre, l'ensemble des mots de la forme ff, avec f un mot quelconque de F^+, ne peut pas être un langage CF (standard ou non). En effet ici les contraintes consistant en ce que chaque lettre de la seconde moitié du mot correspond à une lettre de la première moitié s'entrecroisent et il peut y avoir un nombre arbitrairement grand de telles intersections (ceci n'est évidemment pas une preuve formelle que $\{ ff : f \in F^+ \}$ n'est pas CF). De même on montrerait que les ensembles de mots :

$$L_1 = \{ x_1^n x_2^n x_3^{n'} : n, n' > 0 \}$$

et

$$L_2 = \{ x_1^n x_2^{n'} x_3^{n'} : n, n' > 0 \}$$

sont tous deux des langages CF mais qu'il n'en est pas de même de leur intersection

$$L_1 \cap L_2 = \{ x_1^n x_2^n x_3^n : n > 0 \}.$$

On pourrait illustrer, à l'aide de règles tirées de la grammaire de langues naturelles, ces deux contre-exemples et ceci montrerait, s'il le fallait, que les contraintes de type CF ne concernent qu'un horizon limité de la réalité linguistique. Une fois encore, je me bornerai à renvoyer aux travaux de Chomsky pour une discussion approfondie de ces problèmes, de la valeur explicative en linguistique des règles CF et enfin de leur rôle dans la construction de grammaires moins formelles. Cependant il convient de rappeler que dès le niveau des langages CF, les questions les plus évidentes sont, en général, *indécidables* au sens technique du terme. Ainsi, Bar-Hillel, Perles et Shamir ont montré à l'aide du contre-exemple classique de Post qu'il n'existe aucun algorithme permettant de décider pour deux homomorphismes φ_1 et φ_2 quelconques si les langages CF

$$L_1 = \{ \varphi_1 f\bar{f} : f \in F_+ \}$$

et

$$L_2 = \{ \varphi_2 f\bar{f} : f \in F_+ \}$$

ont ou non au moins un mot en commun.

D'autre part, il résulte des définitions que la mémoire d'un dispositif destiné à tester séquentiellement l'appartenance des mots à un langage CF standard donné n'est utilisée que d'une manière extrêmement restreinte qui rappelle certains aspects élémentaires de la technique de programmation connue sous le nom de « *push down storage* ». De façon schématique, (en laissant de côté la mémoire bornée qui vérifie l'appartenance à $R(X_1, V)$), si αf est le mot enregistré en mémoire après la lecture du segment initial f, et si la lettre suivante est x_i, il suffit d'ajouter celle-ci à la fin de αf quand αf se termine par x_j où $j \neq -i$ et d'effacer x_j dans le cas contraire où $j = -i$. Cette procédure ne nécessite donc en fait que la considération de l'extrémité finale du mot mis en mémoire, d'où le terme « push down ».

Il existe bien entendu d'autres types de contraintes définissant des familles de langage et les plus étudiées d'entre elles constituent ce que l'on appelle après Kleene les *événements réguliers*, qui contiennent en particulier *tous* les langages n'ayant qu'un nombre fini de mots et les langages de la forme $R(X_1, V)$ utilisés plus haut. Avec nos notations, tout événement régulier P est obtenu en prenant un sous-ensemble X_1 de X^+, un ensemble V contenant tous les mots de deux lettres $x_i x_j$ avec i négatif et j positif, ce qui donne un certain langage CF standard dont, finalement, P sera l'image par un homomorphisme φ tel que φx_i soit le mot vide pour chacun des i négatifs. Plus simplement, on peut définir P comme l'image homomorphique de $R(X'_1, V', X'_2)$ où l'homomorphisme φ n'est soumis à aucune restriction et où l'événement régulier $R(X'_1, V', X'_2)$ est l'ensemble des mots commençant par une lettre du sous-ensemble X'_1 de X, n'ayant aucun facteur de longueur deux dans V′ et se terminant par une lettre du sous-ensemble X'_2 de X. Donc, pour les événements réguliers, il est possible de remplacer toutes les contraintes résultant de l'appartenance à D par la seule contrainte supplémentaire que le mot se termine par une lettre appartenant à un ensemble distingué. De façon plus directe, Kleene avait défini un événement régulier comme un ensemble P de mots tel qu'il soit possible de reconnaître si un mot arbitraire appartient à P par lecture séquentielle de ce mot, en ne gardant en mémoire à chaque lettre qu'une quantité d'information bornée par une valeur finie (ne dépendant que de P).

Par exemple, l'ensemble de tous les mots de longueur paire qui ne contiennent pas plus de trois fois la lettre x_2 forme un événement régulier puisque l'appartenance à P peut être reconnue en ne gardant en mémoire pour chaque segment initial que la parité de la longueur de ce segment (soit 1 bit d'information) et le nombre (au plus égal à 3) de fois où x_2 est déjà apparu (soit 2 bits d'information).

Ceci équivaut à dire que P est un événement régulier si et seulement si il est l'image inverse d'un sous-ensemble de l'image de F par un homomorphisme de ce monoïde dans un monoïde *fini*. Donc, ces questions qui étaient en règle générale indécidables pour les langages CF généraux deviennent ici susceptibles d'une solution algorithmique par exhaustion de tous les cas possibles dans un certain ensemble fini. Cette propriété essentielle qui semble avoir fasciné les esprits épris de finitude, permet de rattacher très simplement aux événements réguliers les « grammaires probabilistes » n'utilisant qu'un nombre fini d'états (ce sont les « *finite state sources* » de Shannon) c'est-à-dire — en gros — les schémas markoviens finis et propose le problème de discuter la façon dont une telle grammaire peut approcher un langage CF propre.

Intuitivement, il semble bien évident qu'un langage tel que l'ensemble $M = \{f\tilde{f} : f \in F_+\}$ des *mots en miroir* qui joue un rôle central dans l'étude des langages CF ne puisse être approché que de façon tout à fait triviale par un événement régulier R. En effet, en dénotant par m_k, r_k et q_k le nombre des mots de longueur k de M, R et $Q = R \cap M$, on voit facilement que :

1) soit Q ne constitue qu'une fraction asymptotiquement nulle de M en ce sens que $\lim_{k\to\infty} q_k m_k^{-1} = 0$ (par exemple, si R est contenu dans M) ;

2) soit, au contraire, Q est une fraction asymptotiquement nulle de R, c'est-à-dire $\lim_{k\to\infty} q_k r_k^{-1} = 0$ (par exemple, si M est contenu dans R).

Plus généralement soit $A = \varphi L$ un langage CF (non nécessairement standard) satisfaisant la condition restrictive que chacun de ses mots soit l'image par φ d'un seul mot du langage standard L et cherchons à l'approximer par une séquence infinie strictement croissante d'événements réguliers $\{R_i\}$ $(i = 1, 2, \ldots)$ tous contenus dans A. Pour chaque i la valeur de cette approximation est fournie par la suite des différences $a_k - r_{i,k}$ (où a_k, $r_{i,k}$ sont, comme plus haut, le nombre des mots de longueur k de A et de R_i), soit encore, de façon condensée par le nombre $\bar{a} - \bar{r}_i$ où

$$\bar{a} = \sum_{k>0} a_k x^k, \qquad \bar{r}_i = \sum_{k>0} r_{i,k} x^k$$

avec

$$x = (2n + 1)^{-1}$$

ce qui revient à faire une moyenne (pondérée par les x^k) de toutes ces différences. Or, pour chaque i, $\bar{r}_i$ est une fraction rationnelle dont le dénominateur $\bar{p}_i$ peut être considéré comme une évaluation grossière de la complexité de R_i c'est-à-dire de la quantité d'information qu'il faut pouvoir garder en mémoire afin de tester l'appartenance d'un mot à R_i. Moyennant ces interprétations et le fait aisément vérifié que $\bar{a}$ est un nombre algébrique, des théorèmes classiques d'arithmétique montrent que pour chaque valeur ε de $\bar{a} - \bar{r}_i$, il n'existe qu'un nombre fini de R_i dont la « complexité » $\bar{p}_i$ soit inférieure à une certaine fonction (donnée par le théorème) de ε. C'est là le résultat cherché qui suggère comme une sorte de complémentarité entre les contraintes CF et les contraintes correspondant aux mémoires bornées.

Enfin, sur le plan des faits linguistiques observables, la nécessité d'une approximation des langues naturelles plutôt par des langues CF que par ces modèles à mémoire finie que sont les événements réguliers semble avoir été admise implicitement par les techniciens de la traduction automatique eux-mêmes, puisque tous les programmes présentés jusqu'ici sont basés

sur des méthodes qui sont essentiellement du type « push down » évoqué plus haut. Il me semble y avoir là une certaine ironie des choses sur laquelle je conclurai ces remarques.

Références

N. CHOMSKY et G. A. MILLER. *L'Analyse formelle des langues naturelles*. (Collection « Mathématiques et Sciences de l'Homme »), Paris, Gauthier-Villars, La Haye, Mouton, 1968.

A. LENTIN et M. GROSS. *Notions sur les grammaires formelles*. Paris, Gauthier-Villars, 1967.