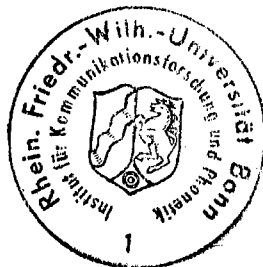


1965 International Conference on Computational Linguistics

SYNTAXE ET INTERPRETATION

B. Vauquois, G. Veillon, J. Veyrunes

C. E. T. A. B.P. 8 St Martin d'Hères - Isère -



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

Dans un système de traduction automatique fondé sur une succession de modèles (où chacun d'eux représente un niveau de la langue source ou de la langue cible) il faut établir leur enchainement. Dans la phase d'analyse (relative à la langue source) les liaisons d'enchainement sont dites "dirigées de bas en haut" car les modèles successifs correspondent à des niveaux de hiérarchie de plus en plus élevés dans la langue ; au contraire, dans la phase de synthèse (relative à la langue cible) les liaisons sont dirigées de haut en bas.

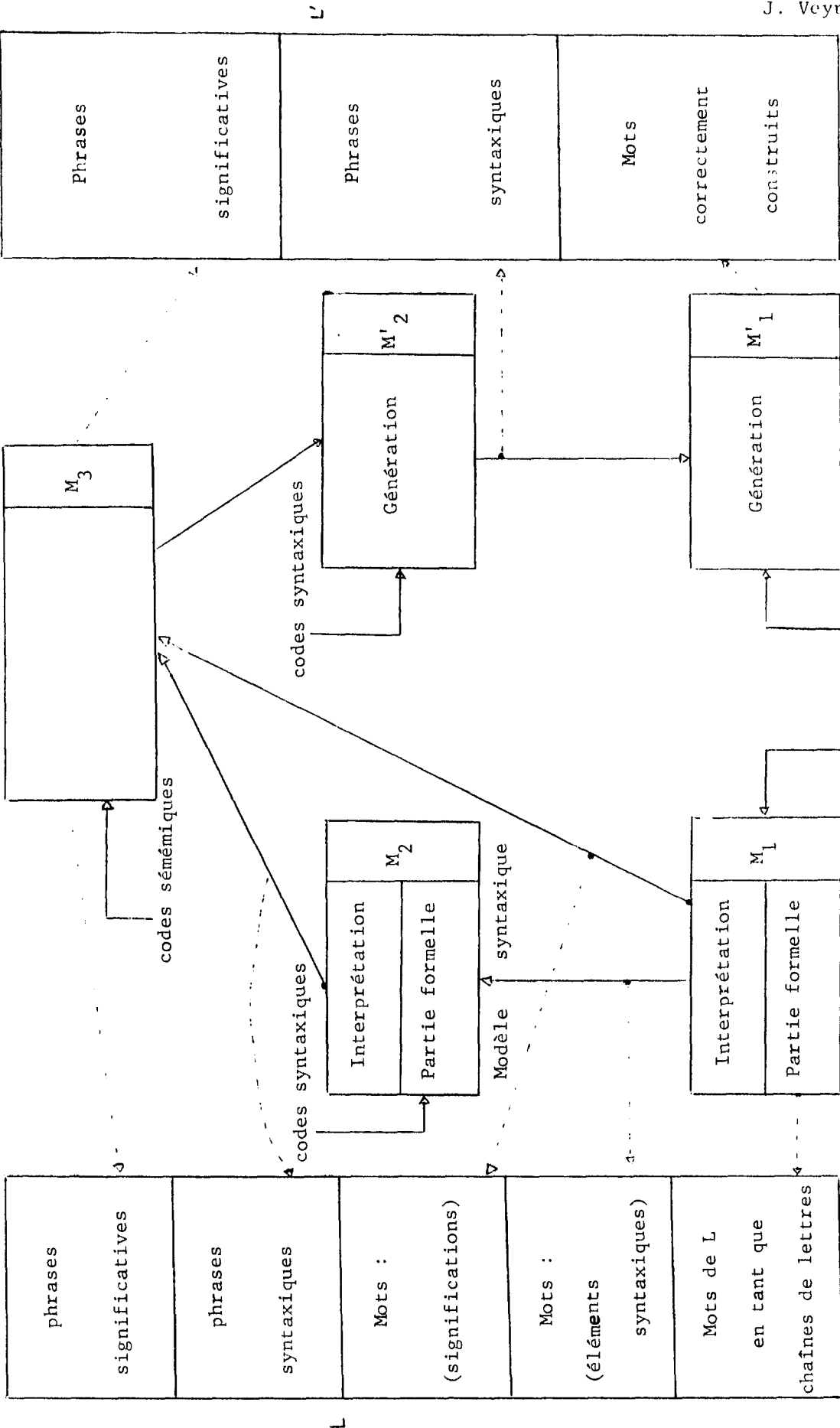
L'analyse d'un texte de la langue source L consiste à trouver dans le modèle de niveau le plus élevé, appelé modèle M_3 , une formule ou une suite de formules dont la représentation dans L est le texte donné. Si toute formule de M_3 a pour représentation dans L au moins une phrase de L on dira que l'analyse se fait phrase à phrase. Toutes les phrases de L qui sont les diverses représentations d'une même formule de M_3 sont dites "équivalentes" par rapport au modèle. Toute phrase de L qui est une représentation commune à deux formules ou plus de M_3 est dite "ambigüe" dans le modèle. On admettra que le modèle M_3 possède aussi une représentation dans la langue cible choisie L', sans se soucier de savoir si M_3 possède encore d'autres représentations dans des langues L'', L''', etc... Dans ces conditions, à toute phrase non ambigüe de L on peut faire correspondre une phrase de L'. Appelons degré d'ambigüité d'une phrase de L par rapport au

modèle, le nombre de formules distinctes qui ont cette phrase pour représentation dans L. A toute phrase de degré d'ambiguïté n dans L correspond alors au plus n phrases dans L'. La traduction ne pourra comporter qu'une seule phrase dans L' si les représentations des n formules de M_3 dans L' ont une intersection non vide.

Le schéma du système de traduction automatique envisagé est alors le suivant :

Synthèse

analyse



Modèle morphologique + codes morphologiques
 Dictionnaire de constituants + codes morphologiques
 Dictionnaire de constituants + codes morphologiques

Fig. 1

Ainsi les modèles M_1 et M_2 contiennent deux parties :

- a) La partie formelle qui résoud le problème de décision :
"la chaîne proposée à l'entrée appartient elle au langage artificiel de ce modèle ? Dans l'affirmative il s'agit de trouver toutes les structures associées à la chaîne.

- b) L'interprétation qui doit assurer la liaison avec les modèles de niveau supérieur.

Dans sa partie "synthèse", ce schéma correspond à quelques variantes pris au modèle d'automate linguistique proposé par S. LAMB.

L'exemple le plus simple est celui du modèle morphologique M_1 : le problème de décision attaché à la partie formelle est l'acceptation ou le refus d'une chaîne de morphèmes comme forme d'un mot de la langue. L'interprétation syntaxique d'une chaîne acceptée consiste à la transformer en éléments du vocabulaire terminal du modèle syntaxique (catégorie syntaxique, valeurs de variables grammaticales, règles interdites). L'interprétation sémantique de cette même chaîne consiste à donner sa signification soit sous forme d'équivalents dans la langue cible soit sous forme d'unités sémantiques dans un langage pivot.

L'étude du modèle syntaxique M_2 est beaucoup plus compliquée. La partie formelle de ce modèle consiste également à résoudre un problème de décision. Etant donnée une chaîne de syntagmes élémentaires fournis par l'interprétation du modèle morphologique, il s'agit d'accepter ou de

refuser cette chaîne en tant que phrase syntaxiquement correcte dans la langue source. En réalité, à une chaîne de mots unique dans la langue source l'interprétation morphologique fait correspondre en général une famille de chaînes de syntagmes élémentaires à cause des homographies syntaxiques. Ainsi, à moins d'explorer successivement toutes les chaînes de la famille, la résolution pratique doit les prendre en charge simultanément.

En outre, étant donné que même une phrase constituée de mots non syntaxiquement ambigus peut admettre dans la langue naturelle plusieurs constructions syntaxiques, le modèle doit rendre compte de cette multiplicité de structures, lorsqu'elle existe, sur chaque chaîne de syntagmes correspondant à cette phrase.

Ainsi, la partie formelle du modèle syntaxique consiste à refuser les chaînes de syntagmes qui ne correspondent à aucune phrase et à fournir pour les chaînes acceptées toutes les structures admissibles.

Le choix du type logique de modèle, le formalisme d'écriture de la grammaire ainsi que l'algorithme d'exploitation de cette grammaire font l'objet de la première partie de cette communication.

La deuxième partie s'attache à montrer la transformation qu'il faut faire subir aux structures fournies par la partie formelle du modèle, pour qu'elles soient acceptables en tant qu'"entrées" du modèle M_3 . Pour justifier le bien-fondé de ces contraintes, on indiquera enfin quelques éléments de M_3 en troisième partie.

L'adaptation d'un tel formalisme à l'analyse syntaxique d'une langue naturelle conduit à un très grand nombre d'éléments terminaux et non terminaux. On a donc été amené à utiliser un formalisme équivalent conduisant à une grammaire de dimension acceptable [5].

On écrit :

- (5) N° Règle - a // VVa >--- A // VVA - SAT
 (6) N° Règle - B // VVB | C // VVC - VIV >--- A // VVA - SAT [2] [7]

I LE VOCABULAIRE TERMINAL du modèle syntaxique est constitué par 3 sortes d'éléments :

1 - Des catégories syntaxiques notées a, b, c, ... dans les règles de type (5).

Exemples : Substantif commun

Adjectif qualificatif

Conjonction de coordination.

2 - Des valeurs de variables grammaticales notées VVa, VVb, ... dans les règles de type (6).

On associe en effet à chaque catégorie syntaxique K p variables grammaticales V_{K_i} $1 \leq i \leq p$, chaque variable V_{K_i} pouvant prendre $n(V_{K_i})$ valeurs - on utilise le produit de valeurs de variables grammaticales, chacune appartenant à une variable différente -

Exemples : "Nominatif singulier inanimé" ; "Indicatif présent première personne" etc...

- 3 - Des numéros de règles interdites : Les règles de type (5) et (6) sont repérées par un numéro de règles : N° règle :

On désire interdire au résultat de l'application d'une règle de grammaire de figurer dans une ou plusieurs autres règles. On dit que ces règles sont saturées. Cette liste de règles, éventuellement vide figure dans les règles de type (5) et (6) sous la rubrique SAT.

II LE VOCABULAIRE NON-TERMINAL est également constitué par 3 éléments :

- I - Des catégories non-terminales notées A, B, C ...

L'une d'entre elles est distinguée des autres et caractérise l'obtention d'une structure de phrase.

Exemple : Groupe nominal général, verbe général, ...

- 2 - Des variables grammaticales associées, cette fois, aux catégories non-terminales.

3 - Des numéros de règles interdites, comme précédemment.

Les règles de construction peuvent, en effet, aussi bien que les règles lexicales, saturer des listes de règles.

Les principaux éléments constitutifs des règles de type (5) et (6) ont été définis comme constituants du vocabulaire terminal et du vocabulaire non terminal.

VIV signifie Valeurs Identiques de Variables. C'est une condition permettant de valider une règle de type (6) seulement si B et C ont en commun certaines valeurs d'une ou plusieurs variables données.

Exemple : I2 - B // | C // - CAS >---

La règle I2 ne s'applique que si B et C ont en commun des valeurs de la variable CAS.

- // | >--- sont des séparateurs.

l'ensemble des éléments précédant >- s'appelle la partie gauche de la règle

l'ensemble des éléments suivant >- s'appelle la partie droite de la règle

Dans la partie gauche,

l'ensemble des éléments précédant | s'appelle le 1er constituant

l'ensemble des éléments suivant | s'appelle le 2ème constituant

Ils sont notés I et 2 si nécessaire.

Exemple : La règle précédente complète :

I2 - B // | C // - CAS >--- A // CAS (I.2) -

Le point symbolisant l'intersection, les valeurs du cas A seront celles qui constituent l'intersection de l'ensemble des valeurs du cas du 1er constituant de partie gauche avec l'une des valeurs du cas du 2ème.

III VARIABLES GRAMMATICALES [2]

Leur intérêt consiste à réaliser des partitions en classes d'équivalence sur le vocabulaire terminal V_T associé aux règles (2). Les ensembles quotients sont les catégories syntaxiques en nombre restreint.

Les conditions d'applications, restituant l'information négligée lors des différentes partitions, sont de 2 types :

- 1) Valeurs des variables imposées (VVA, VVB).
- 2) Intersections de valeurs non vides sur des variables communes (VIV).

IV REGLES INTERDITES - SATURATIONS [5] [7]

Définition :

La règle numérotée I sature à gauche (respectivement à droite), la règle numérotée J relativement à la catégorie non terminale A, si A étant obtenue par l'application de I, on interdit à A d'être le 1er constituant (respectivement le 2ème) de la partie gauche de la règle J.

Les éléments de SAT sont : J_g , J_d selon qu'il s'agit de saturation gauche ou droite. On note J s'il n'y a pas d'ambiguïté.

Transmission de saturations :

Dans le cas de règles récursives on peut décider de la transmission des saturations de la partie gauche vers la partie droite.

Exemple : I - B // | A // - >---- A // - 2
 2 - A // | C // - >---- B // -
 3 a // >---- A
 4 et // >---- C
 5 2 // >---- C

L'utilisation des saturations présente 2 intérêts :

- 1) Regrouper dans la même catégorie non terminale des catégories syntaxiques différentes : les saturations portées par les 2 règles lexicales correspondantes différencient leur comportement syntaxique futur.
- 2) Diminuer le nombre de structures jugées équivalents obtenus dans l'application de la grammaire.

Ainsi l'exemple précédent permet d'obtenir une seule structure dans l'analyse des énumérations de type

a, a,, a et a

V EXTENSIONS DU FORMALISME PROPOSE :

Le formalisme ci-dessus reste context-free [†]. On peut songer à l'étendre pour traiter les problèmes de constituants discontinus qui n'appartiennent pas aux modèles "context-free".

I - Variables véhiculaires [6]

Il s'agit d'une généralisation de la notion de variables grammaticales permettant de réaliser des accords à distance entre 2 occurrences (par ex : accord du pronom relatif avec son antécédent) et donc de traiter d'une manière générale le problème des constituants discontinus [5],[7].

Les variables véhiculaires créées lors de l'application d'une règle, se transmettent vers l'élément de partie droite jusqu'à ce qu'une règle y fasse appel.

2 - Empilement de variables véhiculaires - traitement de structures context-sensitive.

L'utilisation des variables véhiculaires en nombre limité permet de traiter des structures Context-Free ainsi que des structures de constituants discontinus qui se ramènent aisément à des structures Context-Free.

L'utilisation d'un empilement de variables véhiculaires - comme dans un automate à pile ordinaire - permet le traitement de structures essentiellement context-sensitive. C'est le cas par exemple des structures utilisant le mot "respectivement".

Exemple : La chaîne A B C R A'B'C' implique les livraisons
 entre A et A'
 B et B'
 C et C'

On écrira

RA'	>---	R/	\mathcal{V}_A
RB'	>---	R/	\mathcal{V}_B
RC'	>---	R/	\mathcal{V}_C
CR	>---	R/	$\bar{\mathcal{V}}_C$
BR	>---	R/	$\bar{\mathcal{V}}_B$
AR	>---	R/	$\bar{\mathcal{V}}_A$

\mathcal{V}_A indique l'empilement de la variable véhiculaire associé au couple A,A'

$\bar{\mathcal{V}}_A$ indique la suppression dans la pile de la variable véhiculaire associé au couple A,A'.

On peut imaginer en outre plusieurs sortes de variables véhiculaires, pouvant constituer plusieurs piles distinctes.

Les langages ainsi reconnus sont inclus dans les langages context-sensitive. Il reste à prouver, éventuellement, qu'ils se confondent avec eux.

B - ALGORITHME D'EXPLOITATION DE LA GRAMMAIRE.

I BALAYAGE.

L'analyse d'une phrase selon la grammaire de type Context-Free normale doit fournir une ou plusieurs structures arborescentes binaires

On peut concevoir une recherche systématique de ces structures en considérant tout d'abord la construction des n groupements de niveau 1 (c'est-à-dire l'application des règles du lexique), puis les groupements de niveau 2, c'est-à-dire correspondant à la combinaison de 2 syntagmes de niveau 1. Plus généralement, on cherchera les syntagmes de niveau p en formant pour chacun d'entre eux les $(p-1)$ possibilités : $(1, (p-1)), (2, p-2), (i, p-i) \dots (p-1, 1)$.

Cet algorithme dû à COKE, suppose connue la longueur n de la phrase. En utilisant un tel procédé, on voit que l'on construit simultanément tous les syntagmes de même niveau et couvrant les mêmes terminaux. Nous appellerons niveau p d'un syntagme le nombre de terminaux qu'il regroupe et q le rang du premier de ceux-ci dans la chaîne ; si l'on note σ_p^q un sommet d'une structure arborescente binaire ayant le niveau p et couvrant les sommets terminaux $\sigma_q, \sigma_{q+1} \dots, \sigma_{q+p-1}$, on peut associer à toute structure couvrant n terminaux $(2n - 1)$ sommets (en comptant les terminaux).

Exemple :

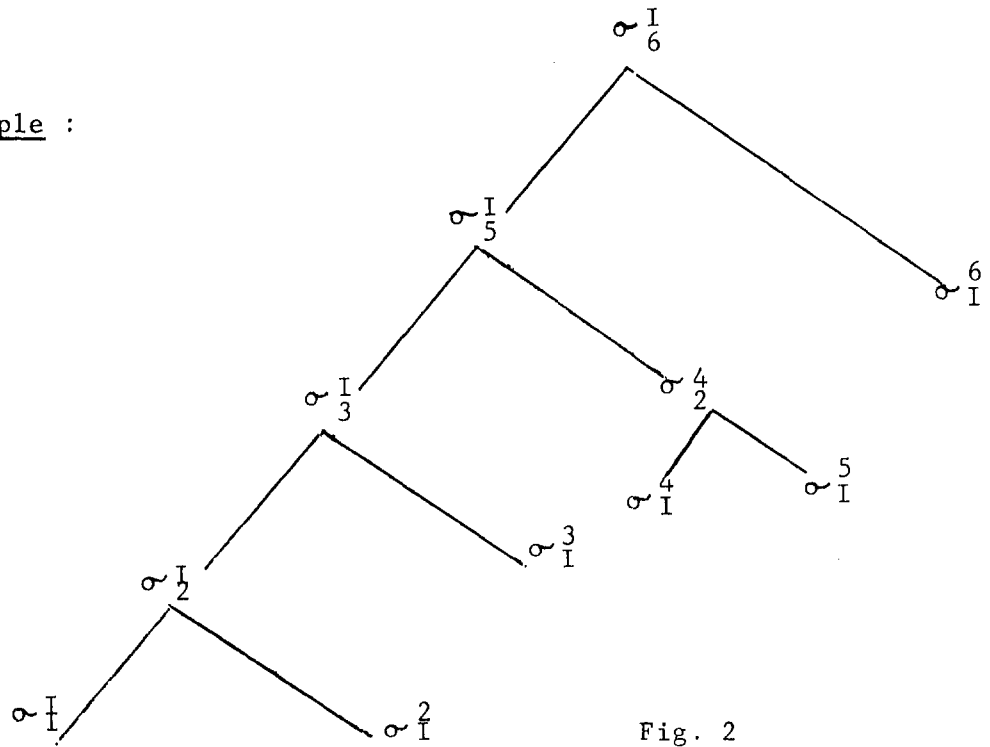


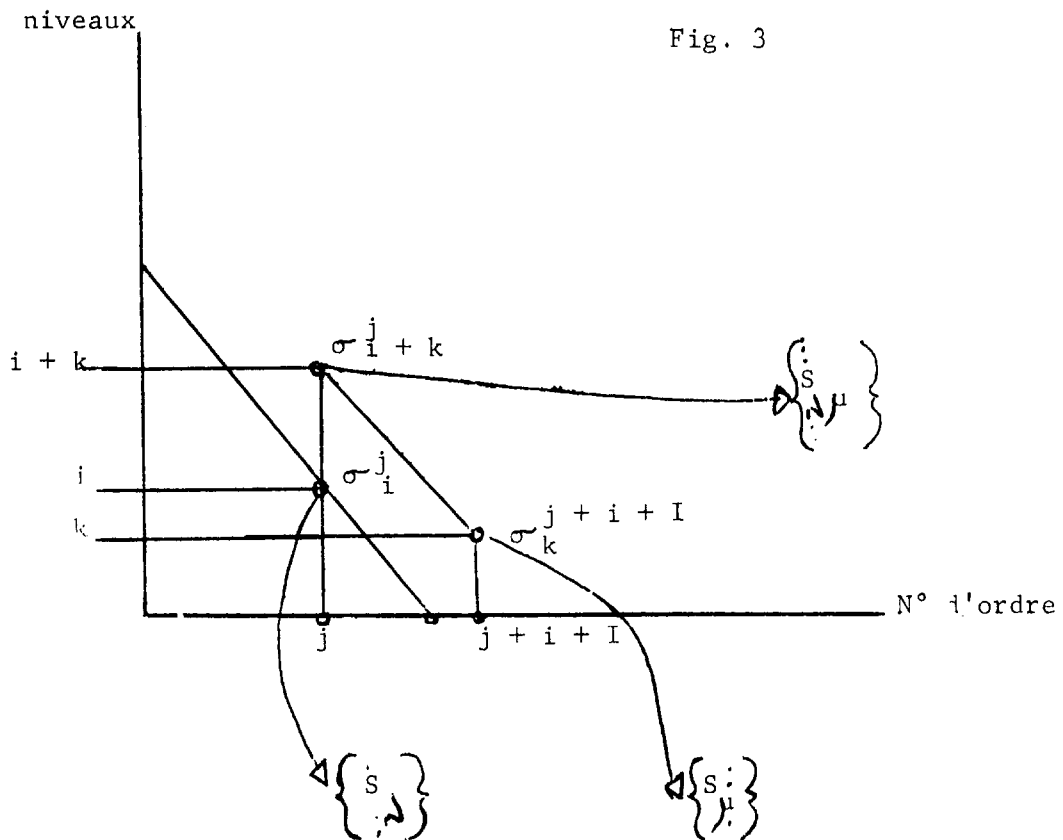
Fig. 2

D'autre part, il est clair qu'il n'y a au total que $\frac{n(n+1)}{2}$ sommets σ_p^q distincts : n de niveau 1, $(n-1)$ de niveau 2 ... et un seul de niveau n .

L'algorithme consiste à examiner tous les sommets σ_p^q possibles. LUKASIEWICZ a montré [1] que l'on peut associer $\frac{1}{2p-1} C_p^{2p-1}$ structures différentes à un sommet de niveau p .

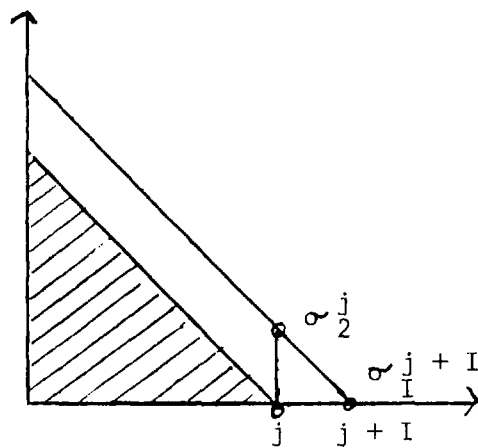
A un sommet donné est attachée la liste des syntagmes homogrophes. Au niveau 1, cette liste fournit les diverses homographies correspondant à une forme.

Le diagramme suivant permet de représenter les sommets d'une phrase de p mots. Les niveaux sont portés en ordonnée, et les numéros d'ordre en abscisse.



Aux syntagmes S_λ correspondant au sommet σ_i^j , on peut associer les syntagmes S_μ du sommet σ_k^{j+i+I} pour former les combinaisons $S_\lambda \mu$ associées au sommet σ_{i+k}^j .

Le programme utilise un tel cadre, chaque sommet σ étant une adresse de liste de syntagmes. Comme la longueur de la phrase n'est pas connue, les sommets sont parcourus par diagonales successives. Si l'on suppose construits tous les syntagmes associés aux j ($j + 1$) sommets correspondant aux j premiers terminaux, le $(j + 1)$ ième terminal permet de construire les syntagmes correspondants aux $j + 1$ sommets sur la diagonale. On commencera par examiner tous les σ de la

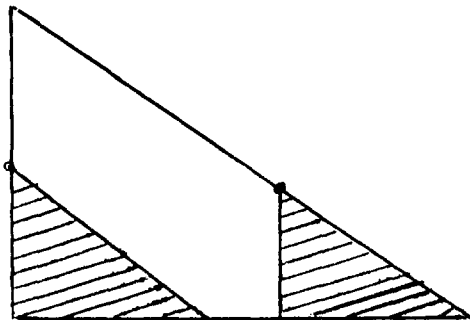


j ième diagonale avec σ_{j+1}^{j+1} (construction de syntagmes associés aux σ de la $j + 1$ ième diagonale). Puis les σ de la $(j - 1)$ ième diagonale avec $\sigma_{j+1}^j \dots$ etc...

Fig. 4

L'intérêt de ce procédé est de supprimer la contrainte sur la longueur de la phrase. L'analyse progresse mot par mot et s'arrête au mot p si il existe un syntagme de phrase associé au sommet σ_p^1 .

D'autre part, il est facile d'éviter un grand nombre de balayages lorsque l'on peut savoir que tous les syntagmes associés à un sommet donné ne sont éléments de gauche d'aucune règle.



La figure matérialise cette famille de points.

Fig. 5

$k-1$ k

II REPRESENTATION DES STRUCTURES SYNTAXIQUES :

A chaque sommet σ_p^q est associée une liste des syntagmes correspondants. Ces syntagmes comprennent, outre les informations syntaxiques, (c'est-à-dire la catégorie, les saturations et les variables grammaticales), le numéro de la règle qui a servi à les construire, et les adresses des deux syntagmes, droite et gauche, qui forment la partie gauche de cette règle.

Nous avons ainsi le schéma suivant :

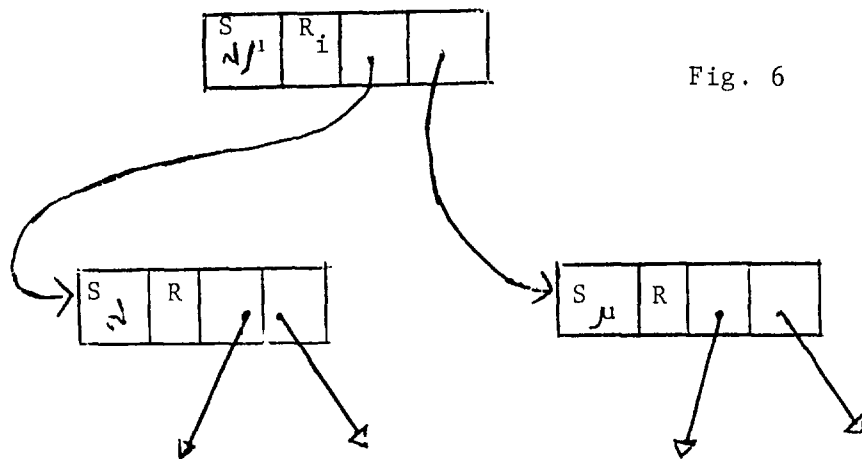


Fig. 6

III REDUCTION DU NOMBRE D'HOMOGRAPHIES :

La liste des syntagmes homographes associée à un sommet donné peut être considérablement réduite en ne conservant que les syntagmes ayant des valeurs syntaxiques différentes. Les syntagmes associés à un sommet σ_p^q sont alors décrits comme une liste de syntagmes auxquels

est associée une liste de règles :

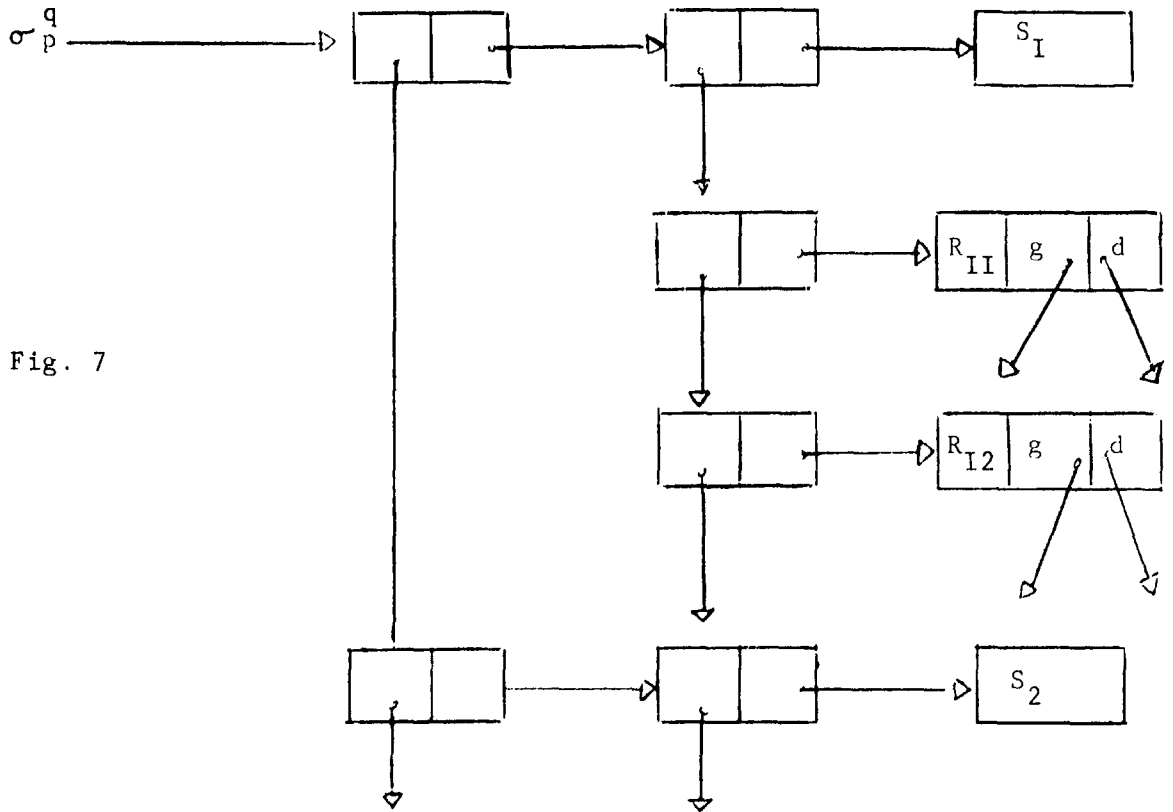


Fig. 7

Ceci évite la prolifération des structures homographes ; dans la chaîne :

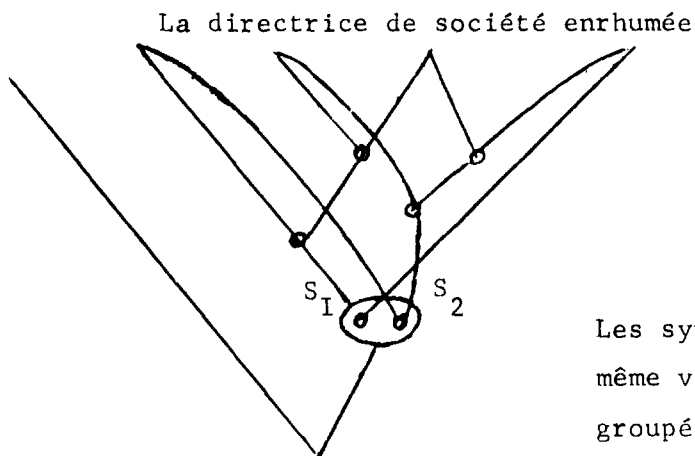


Fig. 8

Les syntagmes S_I et S_2 ayant même valeur syntaxique seront groupés.

Cette homographie ne produira pas de multiplicité de structures au niveau supérieur.

IV EXPLOITATION DE LA GRAMMAIRE PROPREMENT DITE.

L'exploitation de la grammaire est liée à l'algorithme de balayage. Pour un nouveau syntagme donné, on recherche tout d'abord l'existence de ce syntagme comme composant de droite d'une règle. Ceci utilise à la fois la catégorie et les saturations portées par le syntagme.

Lorsque de telles liaisons sont permises, on applique les règles de grammairesaux divers couples :

Détermination de la règle, satisfaction des variables grammaticales et des saturations, calcul du syntagme résultant. La codification interne de la grammaire est réalisée par un compilateur, qui prend en charge les règles données selon le formalisme décrit ci-dessus (I, A).

V FORME DU RESULTAT :

Lorsqu'un syntagme de type $S \begin{matrix} I \\ j \end{matrix}$ correspond à une phrase, l'analyse de la chaîne est arrêtée. Le résultat correspond à la famille des structures associées au syntagme de phrase trouvé.

Elle se présente comme une structure de demi treillis représentant l'ensemble des arborescentes binaires, contenant en un seul graphe connexe toutes les sous-structures communes ou homographes.

D E U X I E M E P A R T I E

INTERPRETATION DU MODELE SYNTAXIQUE

A - FORME DES STRUCTURES A INTERPRETER :

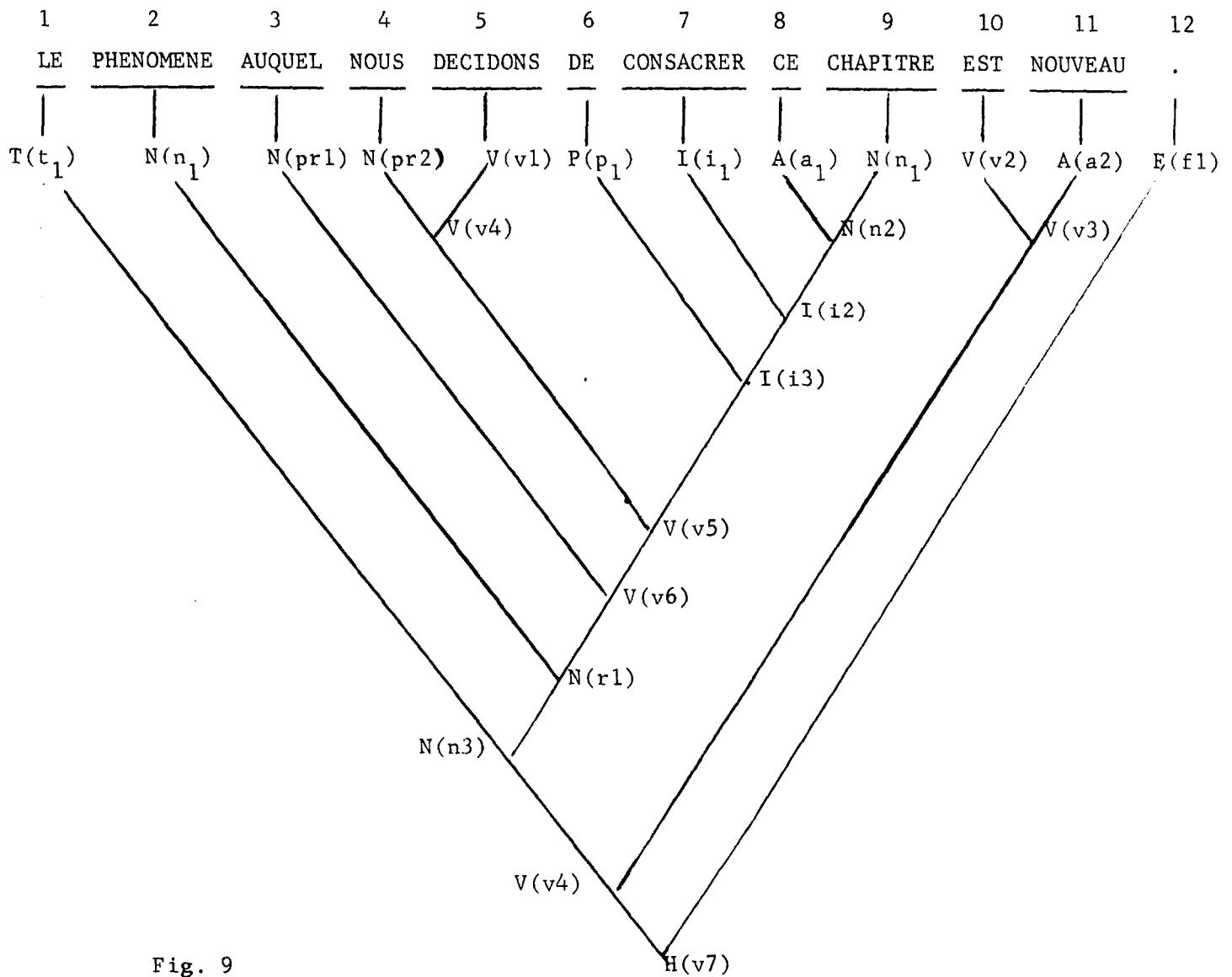


Fig. 9

Dans un but de simplification, et pour séparer la partie théorique de la mise en oeuvre pratique, nous ne considérons ici que le cas d'une structure simple sans homographes.

Il s'agit donc d'une structure arborescente binaire dans laquelle chaque sommet non terminal est un élément du vocabulaire non terminal de la grammaire (Syntagme). Les éléments terminaux de la structure appartiennent au vocabulaire terminal et sont reliés à des éléments non terminaux (règles lexicales) dont ils sont les seuls descendants.

En outre, en chaque sommet non terminal, s'ajoute au nom de ce sommet, le nom de la règle de grammaire (rj) qui a permis de le construire.

B - FORMES DE L'ENTREE DU MODELE M₃ (RESULTAT DE L'INTERPRETATION):

Alors que nous avons une structure sur des syntagmes, nous ne nous intéressons plus maintenant qu'à des fonctions, correspondant à une interprétation des règles de la grammaire. Les syntagmes ont permis de déterminer la cohérence de la phrase et d'en déduire une structure. L'interprétation doit fournir une nouvelle structure sur des règles. En particulier, la fonction d'ordre associée aux terminaux (ordre séquentiel des mots dans le texte d'entrée) peut être modifiée.

La structure obtenue est limitée à une arborescence. Les terminaux de cette nouvelle arborescence expriment dans le modèle M₂ les fonctions syntaxiques liées aux unités lexicales.

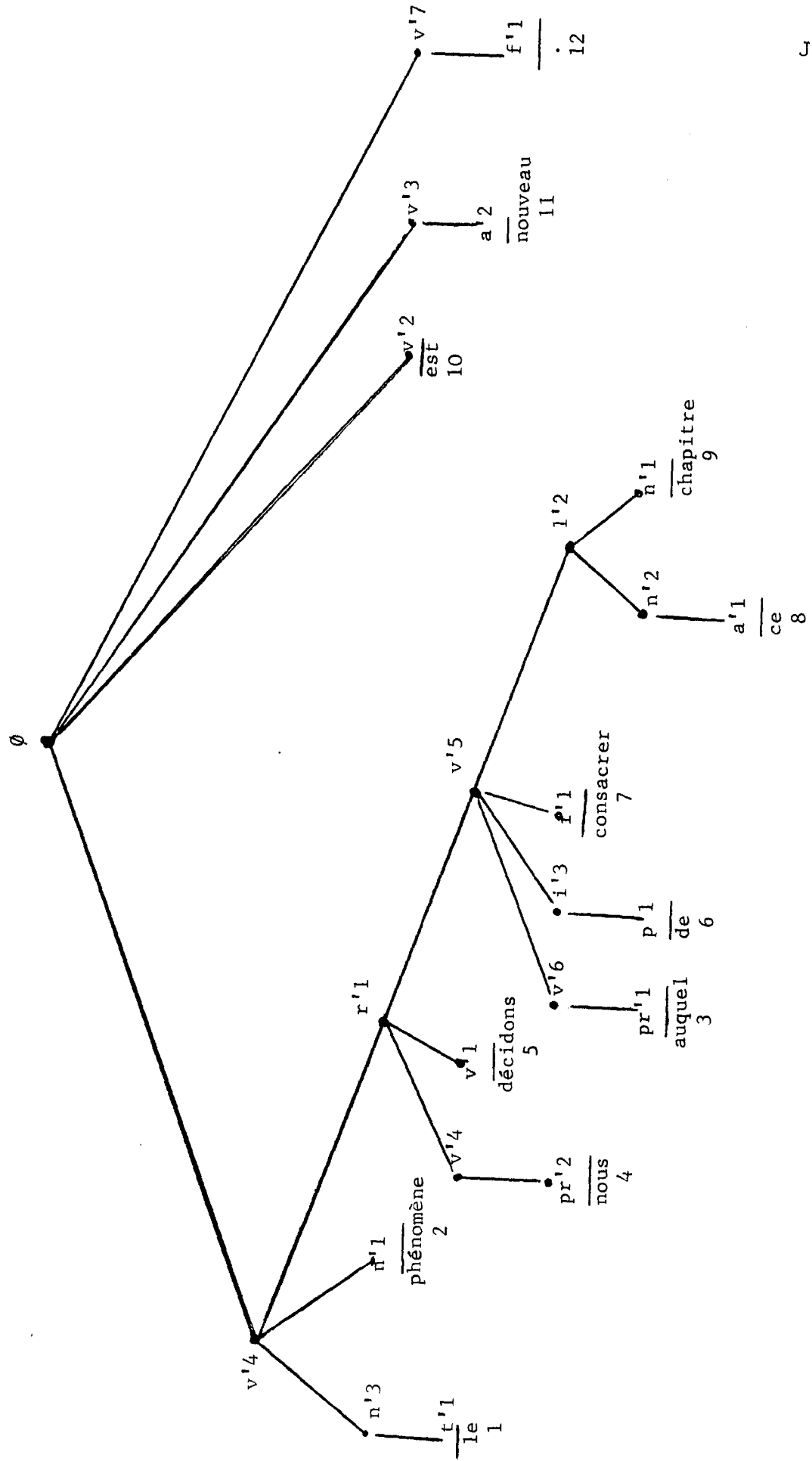


Fig. 10

Un noeud de la structure interprétée est une fonction syntaxique pour son antécédent. Il est muni lui même d'un certain nombre de fonctions qui le caractérisent. La structure est telle que toutes les informations nécessaires à caractériser un noeud sont données aux noeuds du niveau suivant.

En général, il existe un élément distingué qui caractérise le noeud précédent. Cet élément (ou cette règle) pourrait définir le gouverneur dans un graphe de dépendance. **C'** est le cas par exemple de $v'2$ (EST) par rapport à \emptyset ou de $v'4$ (PHENOMENE) par rapport à $v'4$.

Cependant il existe des cas :

a) où l'on rencontre plusieurs règles distinguées :

Exemple : l'énumération ci-dessous dans laquelle n'_1 apparaît trois fois.

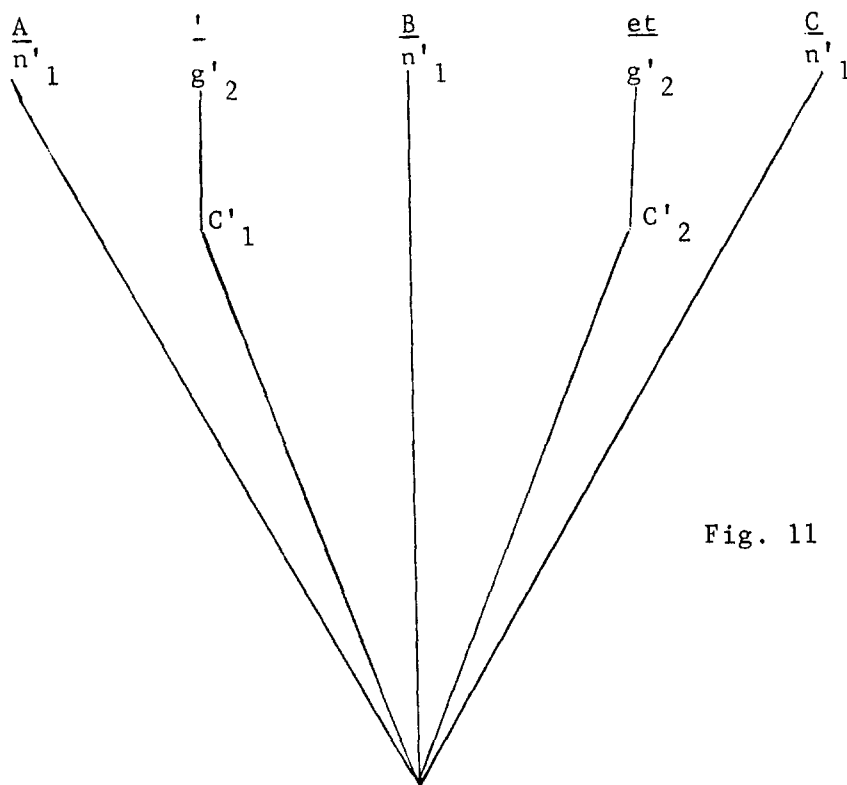


Fig. 11

b) où la règle distinguée ne conduit directement à aucun élément terminal.

Exemple :

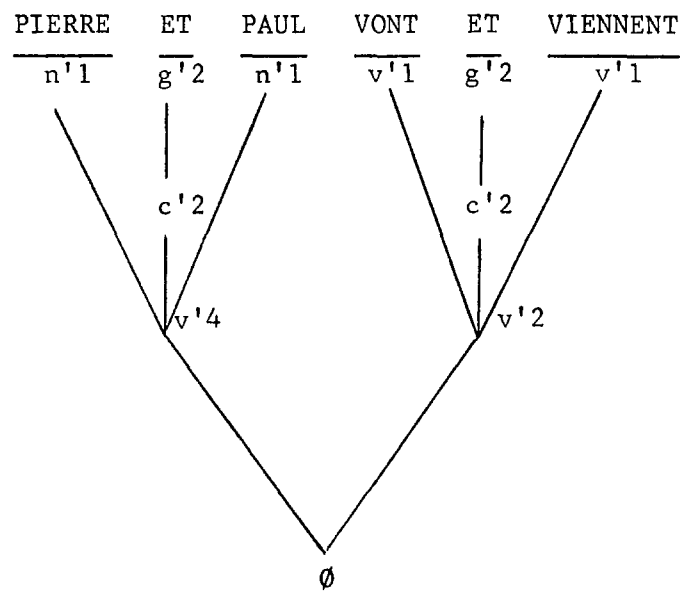


Fig. 12

Tel est le cas pour le noeud \emptyset où $v'2$, règle distinguée ne conduit pas directement à VONT (ou VIENNENT) et exige l'intermédiaire $v'1$.

C - PROCEDE DE CONSTRUCTION DE LA STRUCTURE INTERPRETEE :

L'exemple qui a servi d'illustration en (A) et en (B), correspondant aux figures 9 et 10, montre respectivement la structure formelle et la structure interprétée. Pour effectuer la transformation dans laquelle :

- a) les noms de syntagme disparaissent
- b) les règles r_j deviennent r'_j
- c) l'ordre des syntagmes élémentaires (terminaux) est éventuellement modifié
- d) l'arborescence ne possède plus une structure binaire, on fait appel d'une part à des données d'interprétation sur les règles r_j , d'autre part à des algorithmes d'exploitation de ces données.

I - DONNEES D'INTERPRETATION.

Les données d'interprétation sont les suivantes :

- Toute règle de construction binaire r_j de la forme $AB \rightsquigarrow C$ indique par le symbole "g" ou "d" que le constituant distingué est celui de gauche A ou celui de droit B.
- Toute règle contenant des variables véhiculaires VHL indique par son propre formalisme d'écriture s'il s'agit d'introduction, de transfert ou de destruction pour chacune de ces variables véhiculaires.

II - ALGORITHMES D'EXPLOITATION.

1) Algorithme de transformation :

Il s'agit de réaliser un certain nombre de changements de hiérarchie dans la structure présentée à la figure 9 afin de rétablir les liaisons correctes dans le cas des gouvernements discontinus. On définit la création d'une variable véhiculaire par le symbole \downarrow asso-

cié à la règle de création.

la transmission par le symbole * associé à la règle de transmission.

la suppression par le symbole \uparrow associé à la règle de suppression.

Ces symboles sont notés ainsi que "g" ou "d" dans la phase de reconnaissance formelle.

On appelle chemin étoilé un chemin du graphe tel que le noeud initial contienne le symbole \uparrow et les noeuds intermédiaires contiennent le symbole *.

Le noeud final est celui qui suit le noeud contenant le symbole \downarrow et que l'on atteint à partir de ce dernier en suivant l'information \bar{g} (respectivement \bar{d}).

Soit $C_{n_i}^*$ un chemin étoilé de longueur p commençant au noeud n_i

$$C_{n_i}^* = (N_i, N_{i+1}, \dots, N_{i+p+1})$$

A chaque noeud N_{i+j} de $C_{n_i}^*$ on associe le sous-graphe T_{i+j} de sommet N_{i+j} et ne contenant ni N_{i+j+1} ni ses descendants.

L'algorithme consiste à traiter successivement tous les $C_{n_i}^*$ du graphe, à partir de la racine de la structure.

Pour chacun d'eux, pris dans cet ordre le traitement consiste à

a) transformer $C_{n_i}^* = (N_i, N_{i+1}, \dots, N_{i+p}, N_{i+p+1})$ en le chemin (non

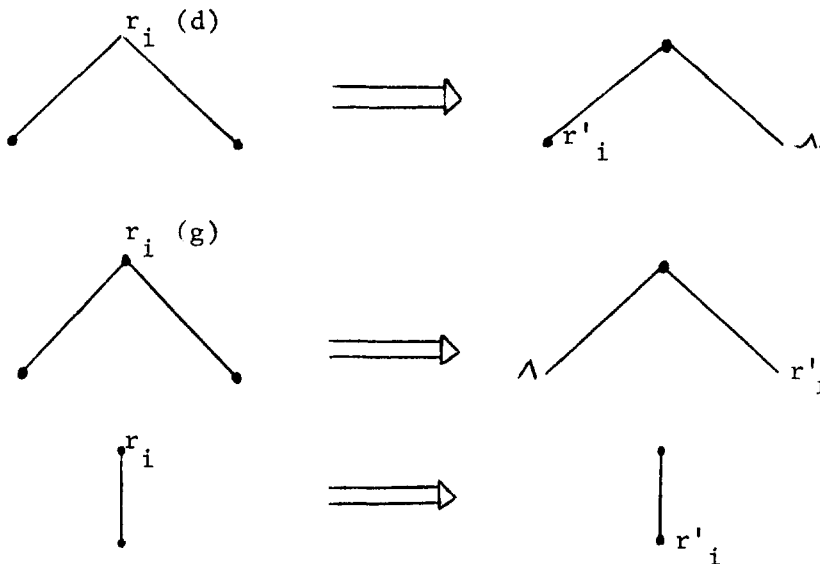
étoilé) de longueur p : $C_{ni+p} = (N_{i+1}, \dots, N_{i+p}, N_i, N_{i+p+1})$,
 restant attachés aux noeuds N_{i+j} auxquels ils étaient primitivement
 associés.

b) on note sur N_i autant de $*$ différents que l'on a interrompu de
 chemins étoilés entre N_{i+p} et N_{i+p+1} .

2) Algorithme de construction des noyaux.

Sur le plan théorique cet algorithme se divise en deux phases. En premier lieu on effectue la séquence d'opérations qui suit.

On procède, niveau par niveau, en commençant par le niveau terminal, à l'affectation pour chaque noeud d'un symbole noté, soit r'_j déduit du nom de règle r_j porté par le noeud immédiatement antérieur, soit \wedge . La règle d'affectation est donnée dans tous les cas possibles par les graphes suivants :



Exemple :

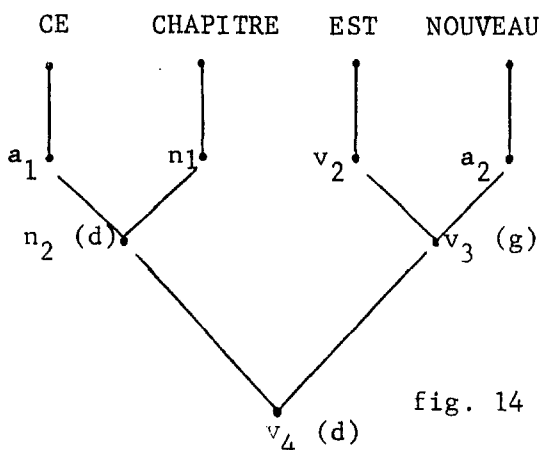


fig. 14

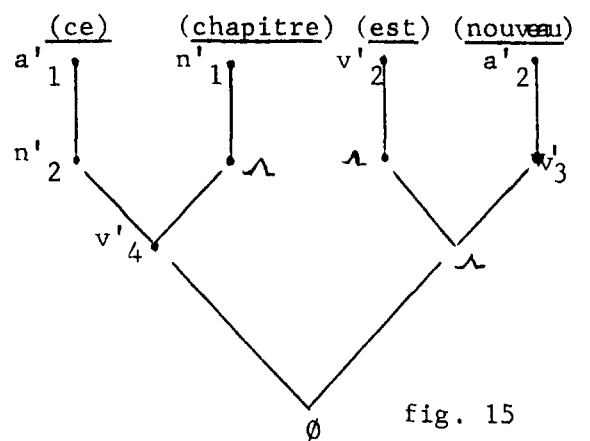
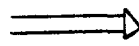
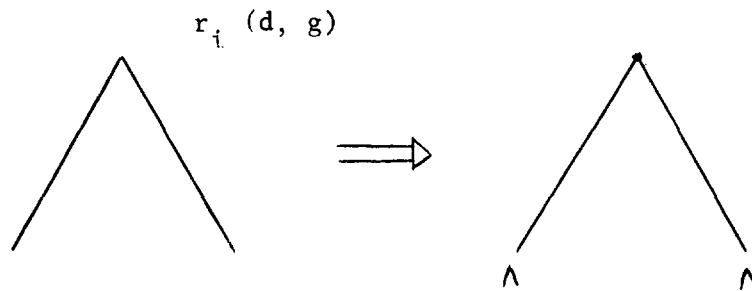


fig. 15

En outre, dans certains cas, nous aurons des règles du type $r_i(d, g)$ la règle d'application est la suivante :



il n'y a pas de r'_i .

La figure 16 montre le résultat de l'application de l'algorithme (1) et de cette phase de l'algorithme (2) sur la structure formelle de la figure 9.

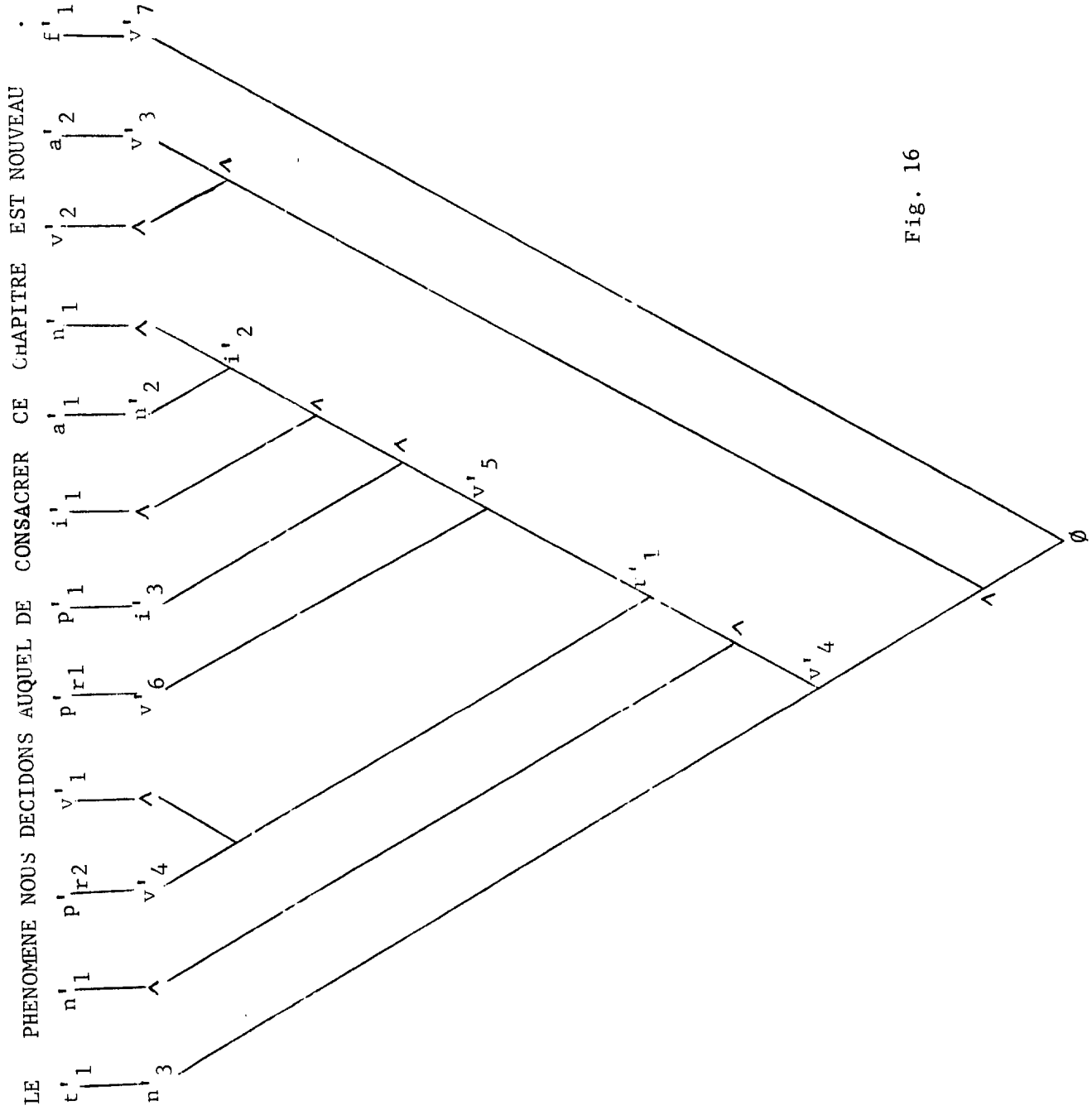


Fig. 16

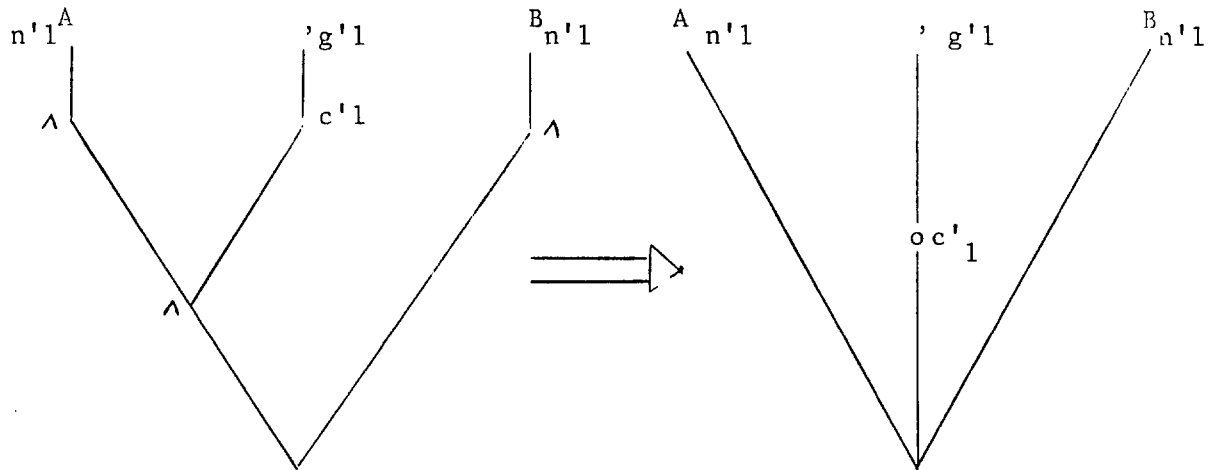
Ensuite, on transforme l'arborescence binaire pour constituer les noyaux de la phrase. Pour cela on considère dans le graphe obtenu tous les chemins du type :

$(R'_i, \wedge, \dots, \wedge)$ notés P'_i associés à chaque R'_i

On définit alors le graphe (P'_i, G) , avec $G(R'_i, \wedge, \dots, \wedge) = \Gamma(R'_i) \cup \Gamma(\wedge) \dots$

En pratique, on réalise ce graphe en supprimant les noeuds et en rétablissant les liaisons de \wedge à ses successeurs sur le R'_i . Ainsi, les noeuds R'_i sont conservés.

Dans le cas où il y a deux A sous un R'_i , pour définir le graphe transformé, il y a lieu de définir P'_i par l'union des chemins P'_i^d et P'_i^g . Ce cas se rencontrera, en particulier, pour la coordination :



T R O I S I E M E P A R T I E

LE MODELE M3.

C'est un langage artificiel dans lequel chaque formule a pour représentation une famille de phrases significatives équivalentes dans la langue source L (et aussi dans la langue cible pour que la traduction soit possible).

Le "degré de signification" que l'on peut atteindre dans L dépend évidemment du modèle. On se borne ici à mettre en évidence la signification syntaxique.

A partir de la structure fournie par l'interprétation syntaxique (figure 10) à propos de l'exemple choisi, la formule déduite dans M3 est celle donnée par le graphe de la figure .

Le modèle M3 accepte une structure interprétée de M2 si les règles de sa grammaire faisant intervenir à la fois les éléments r'j et les codes sémantiques associés aux unités lexicales permettent d'attacher aux noeuds des éléments du vocabulaire de M3 (par exemple : SUJET, ACTION, ATTRIBUTION, etc ...).

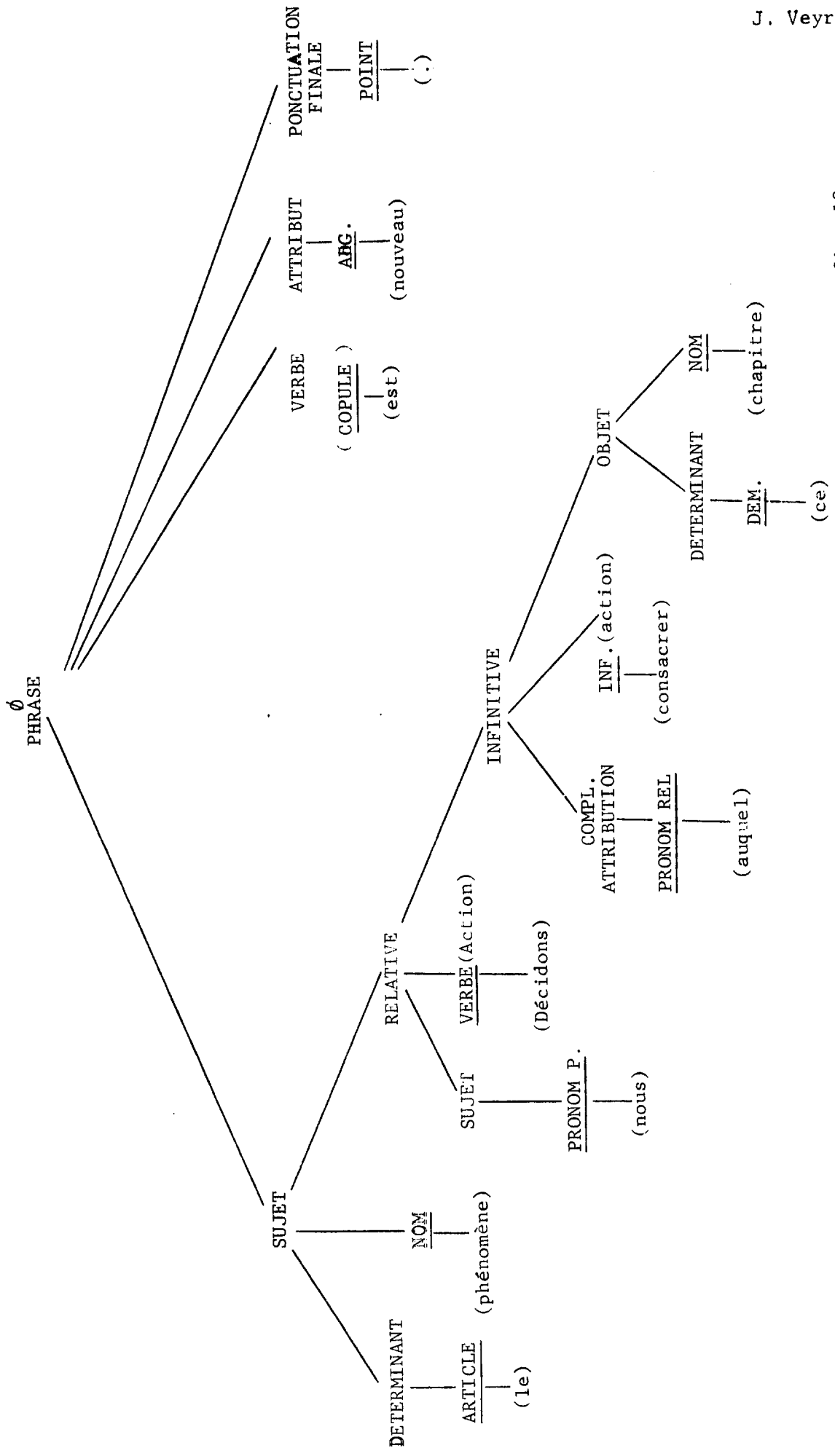


fig. 18

- B I B L I O G R A P H I E -

- (1) C. BERGE Théorie des Graphes et ses applications -
DUNOD - Paris.
- (2) J. COLOMBAUD Langages artificiels en analyse syntaxique -
Thèse de 3ème cycle - Université de Grenoble
(1964).
- (3) G. H. MATTHEWS Discontinuity and Asymmetry in Phrase
Structure Grammars - Inf. and Contr. 6 - 137 -
146 (1963).
- (4) G. H. MATTHEWS A Note on Asymmetry in Phrase Structure
Grammars - Inf. and Contr. 7 - 360 - 365 (1964)
- (5) N. NEDOBEJKINE Modèle de la syntaxe russe - I. Structures
L. TORRE abstraites dans une grammaire 'context-free'.
Document C.E.T.A. G - 201-1 (1964).
- (6) B. VAUQUOIS Application des grammaires formelles aux
G. VEILLON modèles linguistiques en traduction automatique.
J. VEYRUNES Communication au Congrès de Prague (Sept. 1964).
- (7) G. VEILLON Etude de la réalisation pratique d'une
J. VEYRUNES grammaire 'context-free' et de l'algorithme
associé. Document C.E.T.A. G - 2001- 1. (1964).
-