



HAL
open science

**Formalisation de la description des niveaux
d'interprétation des langues naturelles : étude menée en
vue de l'analyse et de la génération au moyen de
transducteurs**

Sylviane Chappuy

► **To cite this version:**

Sylviane Chappuy. Formalisation de la description des niveaux d'interprétation des langues naturelles : étude menée en vue de l'analyse et de la génération au moyen de transducteurs. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1983. Français. NNT: . tel-00306957

HAL Id: tel-00306957

<https://theses.hal.science/tel-00306957>

Submitted on 28 Jul 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée à

l'Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR DE 3ème CYCLE
"Informatique"

par

Sylviane CHAPPUY

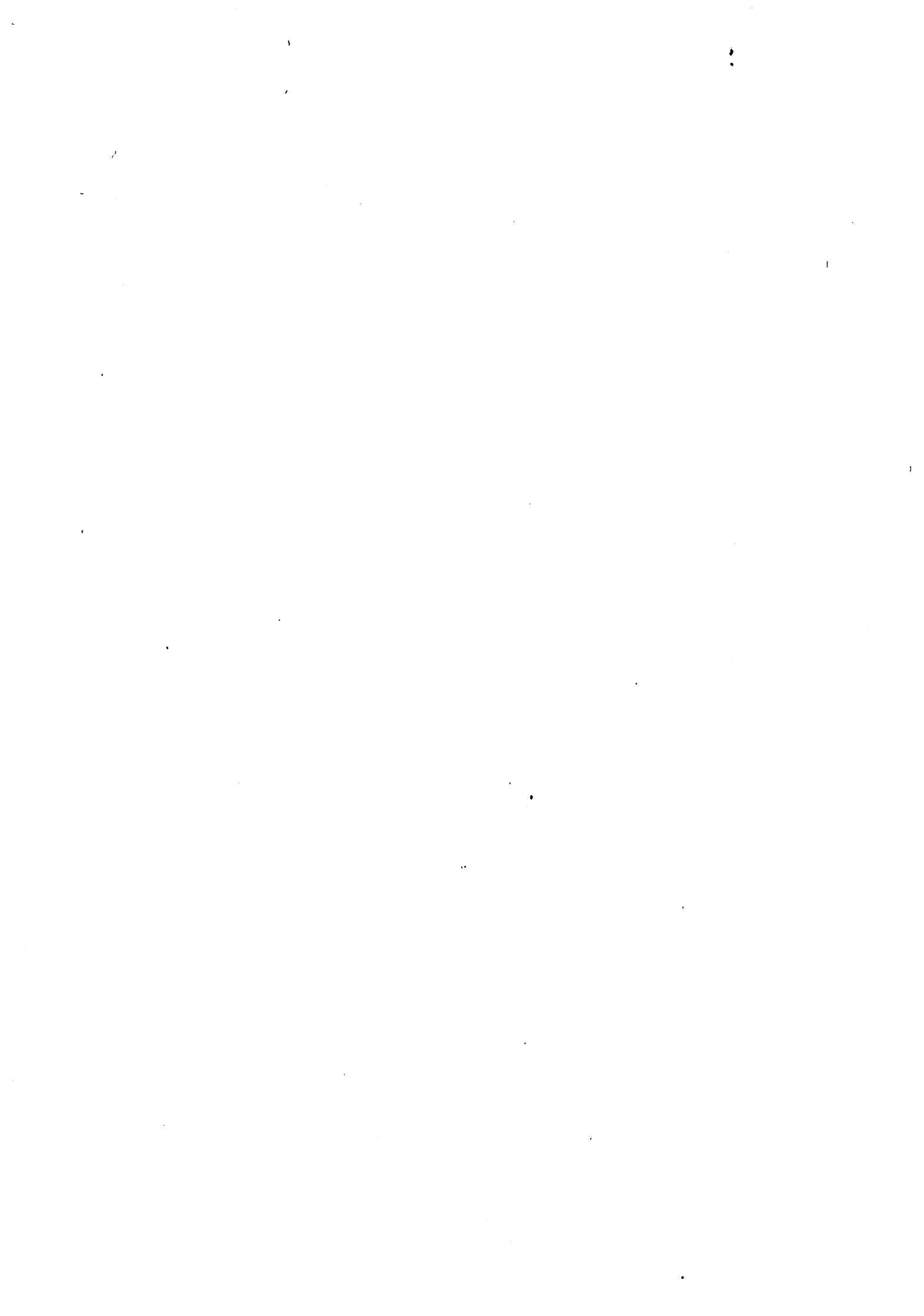


**FORMALISATION DE LA DESCRIPTION DES NIVEAUX
D'INTERPRETATION DES LANGUES NATURELLES.
ETUDE MENEES EN VUE DE L'ANALYSE
ET DE LA GENERATION AU MOYEN DE TRANSDUCTEURS.**



Thèse soutenue le 2 juillet 1983 devant la Commission d'Examen :

Monsieur	P.-C. SCHOLL	:	Président
Messieurs	C. BOITET J.-C. LATOMBE F. PECCOUD B. VAUQUOIS	}	Examineurs



A mes parents.

En témoignage de mon affection et de ma reconnaissance.



Les recherches qui font l'objet de cette thèse, ont été effectuées au sein du Groupe d'Etudes pour la Traduction Automatique (GETA). Je tiens à remercier Monsieur VAUQUOIS, Professeur à l'Université Scientifique et Médicale de Grenoble (USMG) et Directeur du GETA, d'avoir bien voulu diriger cette thèse et d'avoir toujours su me prodiguer les meilleurs conseils tout au long de ces années.

Monsieur BOITET, Professeur à l'USMG, a bien voulu s'intéresser à ce travail. Par sa lecture et sa critique détaillée de la rédaction provisoire de cette thèse, il m'a aidé à préciser de nombreux points importants. Je le prie de trouver ici l'expression de ma reconnaissance.

Monsieur SCHOLL, Professeur à l'USMG, m'a fait l'honneur de présider le jury, ce dont je le remercie.

Monsieur Latombe, Professeur à l'Institut National Polytechnique de Grenoble (INPG) et Monsieur PECCOUD, Professeur à l'Université des Sciences Sociales de Grenoble (USSG) ont accepté d'examiner ce travail et d'en juger les résultats. Je les remercie profondément.

Je voudrais, enfin, parler de l'excellent accueil qui m'a été réservé par les membres du GETA. Je tiens tout particulièrement à les remercier de l'aide qu'ils m'ont apportée, le plus souvent elle fut le résultat de discussions passionnantes et passionnées, auxquelles j'ai pris part avec plaisir.

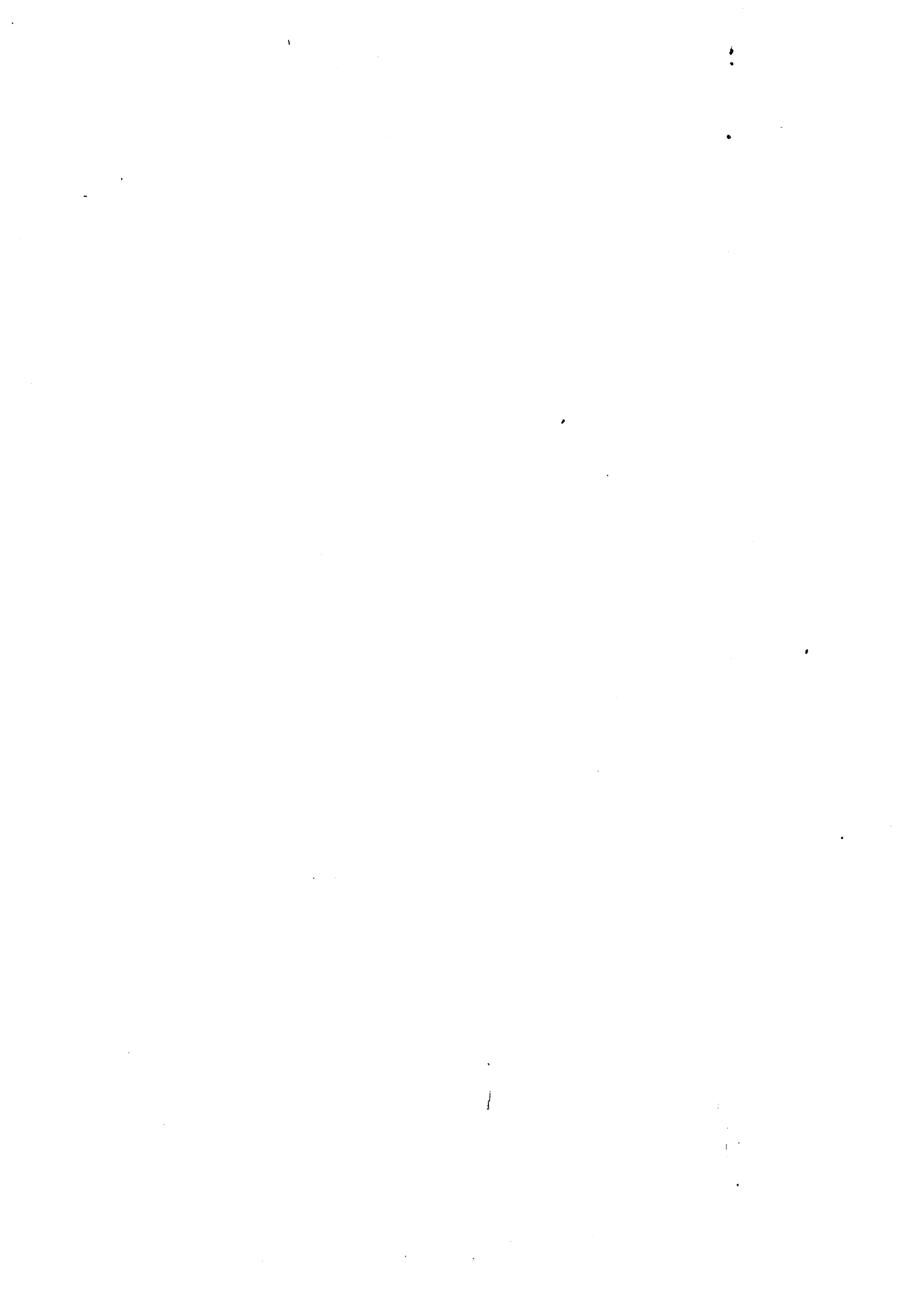


-TABLE DES MATIERES-

Notations.	1
Situation du travail présenté dans cette thèse.	2
<u>CHAPITRE I: INTRODUCTION AUX PROBLEMES DU TRAITEMENT AUTOMATIQUE DES LANGUES NATURELLES.</u>	5
1. Qu'est-ce qu'une langue? Qu'est-ce qu'un langage?	7
1. Point de vue du linguiste.	7
2. Point de vue de l'informaticien.	9
3. Point de vue du mathématicien.	10
2. Problèmes intéressants à propos des langages.	10
Génération et grammaires.	
Reconnaissance et automates accepteurs.	
Interprétation et transducteurs.	
Différentes approches de ces problèmes.	11
<u>CHAPITRE II: INTERPRETATION ET TRANSDUCTION.</u>	13
1. Description formelle des niveaux d'interprétation.	15
1. Choix des descripteurs.	17
2. Représentation en constituants immédiats.	18
3. Représentation en termes de dépendance.	19
4. Discussion à propos de ces deux types de représentation.	21
1. Passage constituants immédiats --> dépendance.	21
2. Passage dépendance --> constituants immédiats.	23
3. Expression des deux niveaux sur une même structure.	24
2. Accès à ces niveaux.	27
1. Par "Parsing" Hors-Contexte.	29
1. Analyseur descendant (top-down).	30
2. Analyseur ascendant (bottom-up).	32
	34
3. Problèmes liés à ces analyseurs et solutions.	34
1. Problème du non-déterminisme.	34
Solution 1: analyse avec retour-arrière (backtracking).	35
Solution 2: stratégie d'analyse parallèle.	35
2. Problèmes liés à l'aspect combinatoire.	35
Algorithme d'Earley.	39
Grammaires LL(k).	39
Grammaires LR(k).	40
2. Par filtrage dans un multigraphe.	

3. Par transduction.	43
1. Un modèle d'adjonction (graphe de chaînes d'arbres-graphe de chaînes d'arbres): les Systèmes-Q de Colmerauer.	43
2. Un modèle de création (chaîne-arbre étiqueté): les A.T.N. de W.A.Woods.	45
3. Un modèle de substitution (arbre décoré-arbre décoré): ROBRA.	48
4. Apports relatifs transducteurs-analyseurs.	52
<u>CHAPITRE III: CHOIX ET REPRESENTATION D'UN MODELE</u>	
<u>"TEXTE <--> INTERPRETATION".</u>	55
1. Choix des niveaux d'interprétation.	57
1. Buts à atteindre en traduction multilingue.	59
2. Notion de niveau d'interprétation.	61
3. Solutions apportées par le G.E.T.A.	65
1. Un modèle multiniveau.	65
2. Choix de structure pour ce modèle.	66
3. Choix des niveaux d'interprétation sur cette structure.	66
4. Représentation de cette structure.	67
1. Au niveau de la géométrie.	67
2. Au niveau de la décoration.	68
5. Exemple.	71
2. Représentation d'une grammaire statique.	75
Introduction aux grammaires statiques.	77
1. Autres travaux de description des liens texte-interprétation.	79
2. Etude d'un formalisme de représentation.	80
1. Formalisme d'écriture de la première zone.	82
1. Formalisme d'écriture des chaînes.	82
2. Formalisme de construction des structures.	85
3. Correspondances chaînes-structures.	92
4. Représentation du contexte.	94
2. Formalisme d'écriture de la deuxième zone.	98
3. Formalisme d'écriture de la troisième zone.	102
4. Articulation entre les planches.	104
5. Partie entête d'une planche statique.	106
6. Exemples de planches statiques.	107
1. Exemple 1 (groupe nominal simple du français).	107
2. Exemple 2 (groupe nominal complexe du français).	109
7. Généralisation.	111
1. Utilisation de ce formalisme pour écrire des grammaires statiques en morphologie.	111
2. Utilisation de ce formalisme pour écrire des grammaires statiques en transfert.	114

3. Application: extraits d'une grammaire statique expérimentale du français.	115
1. Attributs lexicaux.	117
2. Hiérarchie adoptée.	118
3. Quelques planches morphologiques.	119
4. Quelques planches structurales.	134
4. A propos des grammaires statiques et de leur utilisation dynamique.	165
1. Grammaires dynamiques d'analyse.	169
2. Grammaires dynamiques de transfert.	171
3. Grammaires dynamiques de génération.	172
<u>CHAPITRE IV: CONCLUSION.</u>	175
Annexe I: Attributs lexicaux du français. Extrait.	179
Annexe II: Stratégie globale du modèle d'analyse de l'anglais.	187
Bibliographie.	207



-NOTATIONS GENERALES UTILISEES-

$A = \{ x / P(x) \}$ ensemble des éléments ayant la propriété P.
|A| cardinal de A.

\in appartenance à un ensemble.

\notin non-appartenance à un ensemble.

\subset Inclusion stricte dans un ensemble.

\supseteq Inclusion large dans un ensemble.

\cap Intersection ensembliste.

\cup union ensembliste.

\cdot concaténation.

A^* monoïde libre engendré par A.

|x| longueur d'une chaîne x, $x \in A^*$.

ϵ , neutre (chaîne vide).

$A^+ = A^* - \{ \epsilon \}$.

$\alpha \rightarrow \beta$ règle de réécriture.

\Rightarrow dérivation directe.

\Rightarrow^* dérivation.

\Rightarrow^r

dérivation droite.

\Rightarrow^m

dérivation gauche.

\Rightarrow^l

\forall quel que soit.

\exists il existe.

\Rightarrow implication logique.

\Leftrightarrow équivalence logique.

\vee ou logique.

$\&$ et logique.

\neg non logique.

-ABREVIATIONS-

EF Etats Finis.

HC Hors-Contexte.

SC Sous-Contexte.

MT Machine de Turing.

ST Système Transformationnel.

GC Graphe de Contrôle.

GT Grammaire Transformationnelle.

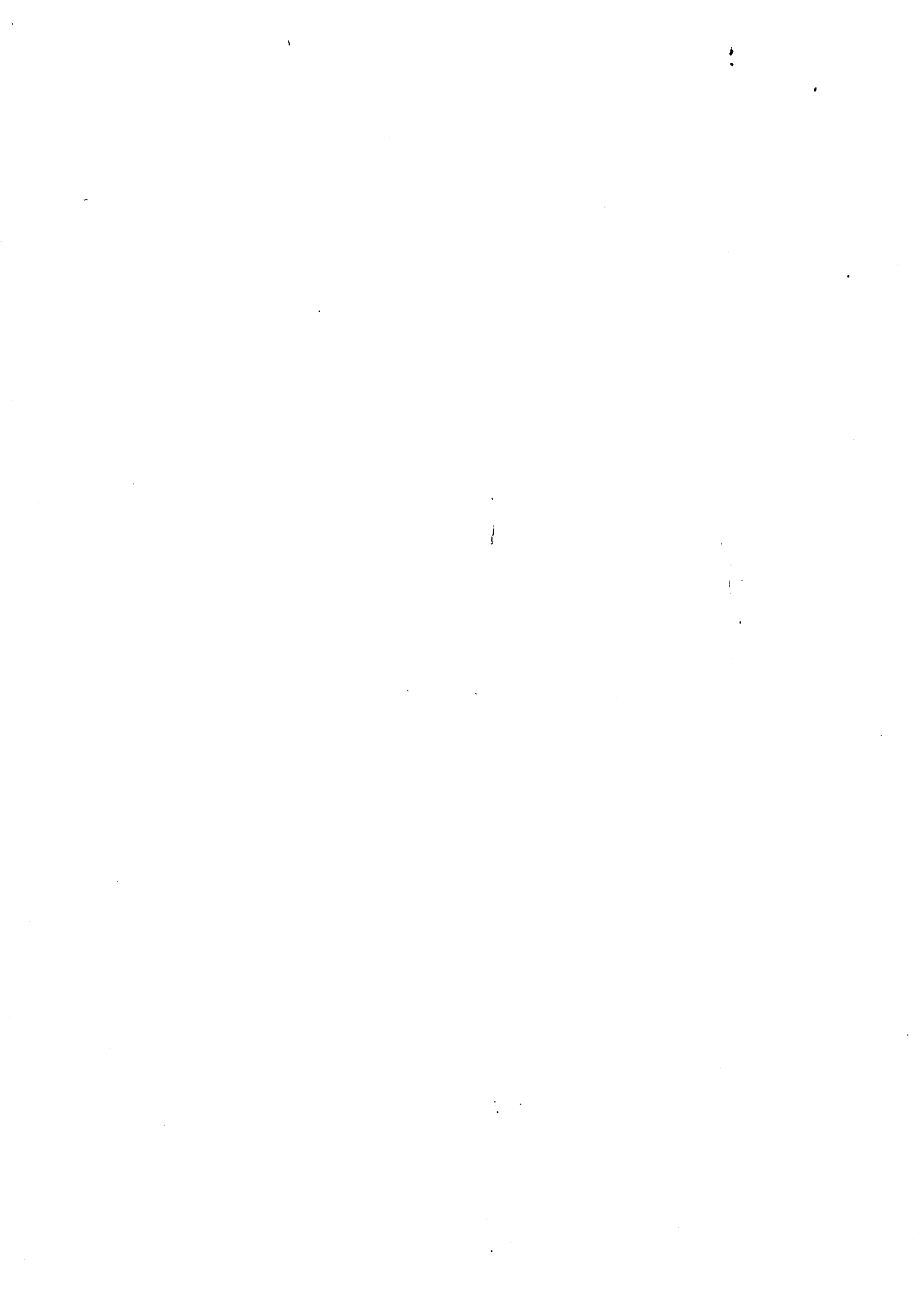
RP Règle de Production.

AO Arbre Objet.

RT Racine de Transformation.

l.e. "id est" (c'est-à-dire).

vs. "versus" (par opposition à).

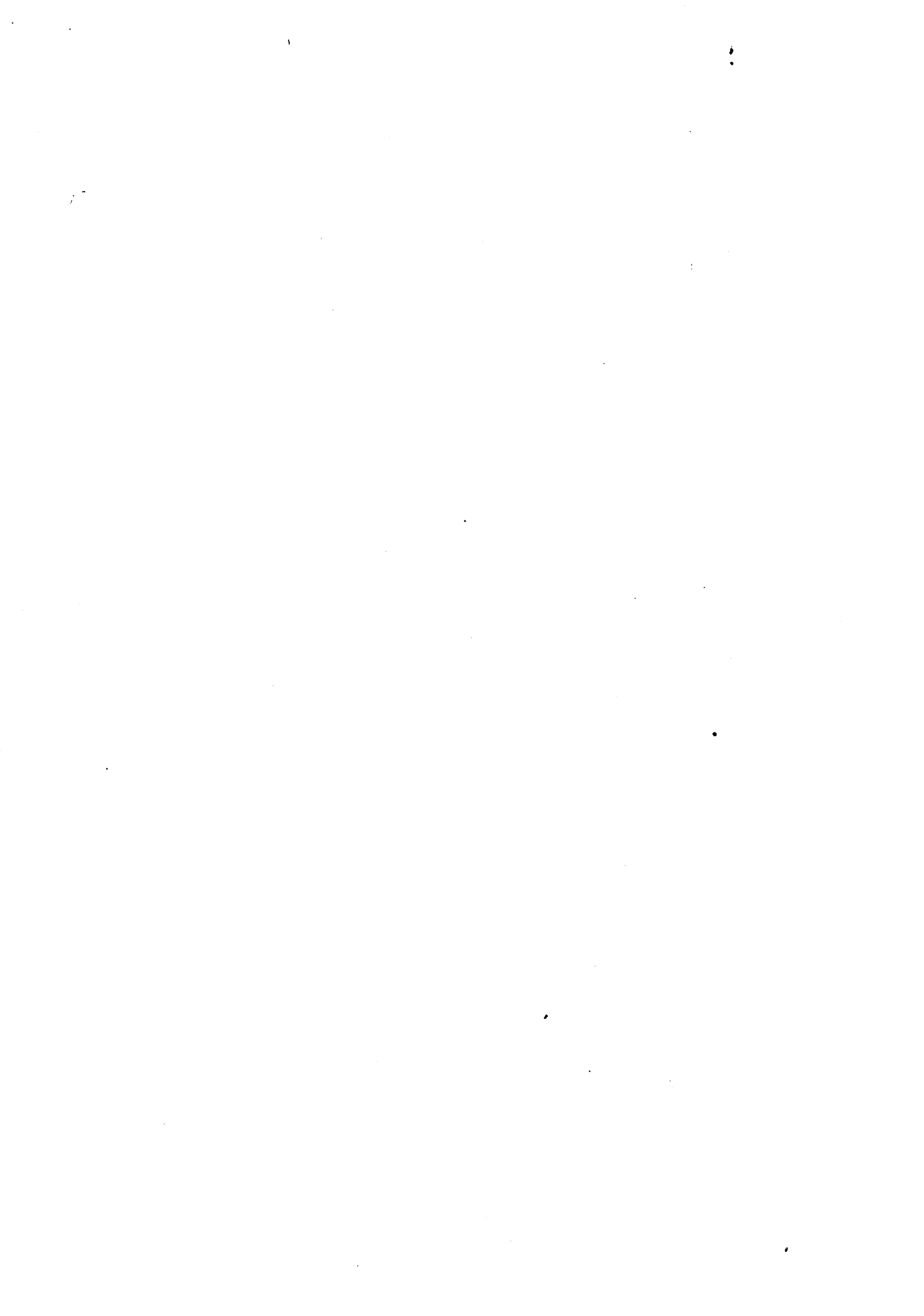


Situation du travail présenté dans cette thèse.

Les recherches en Traduction Automatique se font actuellement dans deux directions et à plusieurs niveaux. Essentiellement, il y a le point de vue Informatique qui consiste en la recherche de nouveaux algorithmes, de nouveaux types de transducteurs et de nouveaux métalangages pour augmenter les performances des systèmes de traduction, et le point de vue linguistique qui se concentre sur l'étude des modèles d'analyse, de transfert et de génération et sur la meilleure façon d'organiser les règles, grammaires et dictionnaires pour améliorer la qualité des traductions en utilisant les systèmes existants. L'étude de structures de données adéquates est un travail qui se fait plus ou moins dans les deux directions.

Le but des travaux présentés dans cette thèse est l'établissement d'un nouveau formalisme de représentation des langues naturelles. Dans le but d'aider à l'élaboration des grammaires dynamiques, nous introduirons la notion de grammaires statiques qui sont des grammaires de description des langues dans un modèle multiniveau que nous aurons présenté auparavant, et nous donnerons un formalisme d'écriture de ces grammaires statiques. Nous montrerons enfin comment utiliser ces grammaires statiques dans des grammaires dynamiques.

Le contenu de cette thèse est le suivant: le chapitre I résume les différentes manières d'approcher les Langues Naturelles et les problèmes soulevés par leur étude; le chapitre II fait quelques rappels historiques sur les modèles linguistiques, les analyseurs syntaxiques et les transducteurs; le chapitre III présente les notions de niveaux d'interprétation, de structure multiniveau, établit la notion de grammaire statique et donne un formalisme de représentation de ces grammaires statiques, il applique ensuite ce formalisme à une grammaire statique expérimentale du français, et introduit quelques notions de stratégies dynamiques sur ces modèles statiques.



CHAPITRE I

INTRODUCTION AUX PROBLEMES DU TRAITEMENT AUTOMATIQUE

DES LANGUES NATURELLES.



L'anglais ne connaît qu'un seul mot: "language", là où le français en utilise deux: "langue" et "langage".

1. QU'EST-CE QU'UNE LANGUE? QU'EST-CE QU'UN LANGAGE?

A cette question on peut fournir quantité de réponses, selon que l'on est linguiste, sociologue, informaticien, mathématicien. De manière générale, une langue est un ensemble de mots et de manières de les combiner en usage dans une communauté afin de permettre à des individus de communiquer entre eux et d'échanger des informations. Si les individus ne sont pas des êtres humains, mais des machines, des animaux, on parle de langage (langage Algol, langage des abeilles, etc.). Le mot langage peut même prendre un sens plus particulier (langage des fleurs, langage cinématographique).

Langues ou langages engendrent un processus d'interprétation qui permet le passage du niveau de l'expression qui consiste en une chaîne de signes, à celui du contenu, c'est-à-dire à la signification de cette chaîne de signes.

Nous allons considérer la notion de langage sous différentes optiques.

1.1. POINT DE VUE DU LINGUISTE.

Nous nous inspirerons ici essentiellement des travaux de Noam Chomsky [CHOM.57] [CHOM.65] et du livre de John Lyons [LYON.70].

Le linguiste s'intéresse au langage humain. Il appelle "langue" un ensemble fini ou infini de "phrases", chacune d'elles étant de longueur finie et composée d'un ensemble fini d'éléments. Toutes les langues naturelles sous leur forme écrite ou parlée répondent à cette définition puisqu'elles comportent toutes un ensemble de phonèmes (ou de lettres dans leur alphabet) et que chaque phrase peut être représentée comme une suite finie de ces phonèmes (ou lettres) bien que le nombre de phrases soit infini.

Le linguiste qui décrit une langue a pour tâche, entre autres, de déterminer pour cette langue, aussi simplement que possible, quelles phrases sont acceptables et de le faire dans le cadre d'une théorie générale linguistique. Une phrase acceptable étant une phrase qui a été produite (ou pourrait l'être) par un locuteur natif, dans un contexte approprié et que les autres locuteurs natifs acceptent (ou accepteraient) comme appartenant à la langue. Tout locuteur d'une langue peut comprendre et produire, non seulement les phrases qu'il a déjà entendues à un moment ou à un autre, mais également un nombre infiniment grand de phrases nouvelles qu'il n'a jamais entendu prononcer par d'autres locuteurs de cette langue. Les phrases nouvelles satisfont aux mêmes tests opérationnels d'acceptabilité, pour les autres locuteurs natifs, que celles qui pourraient simplement avoir été produites de mémoire, d'où l'idée qu'elles présentent des régularités et que des règles peuvent en rendre compte. Autrement dit, c'est la classe des énoncés potentiels que le linguiste doit identifier comme les phrases de la langue et, dans toutes les langues naturelles, le nombre des énoncés potentiels est illimité.

Le fondement traditionnel de la pensée linguistique en Amérique du Nord a été le point de vue **structuraliste**: la langue est un phénomène acoustique observable dont la nature et la structure sont susceptibles d'être étudiées. D'autre part, il existe un point de vue **communicatif**: la langue peut être considérée comme une correspondance entre le son et la signification, c'est-à-dire l'expression et le contenu (son et signification n'appartiennent alors pas au langage). Pour que le langage puisse servir à la communication, il faut que l'expression possède un contenu. Le contenu est le message que le locuteur transmet à celui ou à ceux auxquels il s'adresse.

Pour étudier le langage, il faut, non seulement étudier la structure de l'expression, mais aussi étudier la structure du contenu, ainsi que la relation entre l'expression et le contenu.

On distingue différentes orientations dans l'étude linguistique:

- La phonologie qui décrit les sons d'une langue donnée; par opposition à la morphologie qui s'intéresse à la structure interne des mots.
- La grammaire décrit la forme de ces mots et la façon dont ils se combinent pour former des syntagmes, des propositions, des phrases.
- La sémantique décrit le sens ou contenu de ces mots et des unités qu'ils composent.
- La pragmatique qui réfère à l'usage et témoigne d'une étroite adhésion au sens du réel, au milieu.

1.2. POINT DE VUE DE L'INFORMATICIEN.

Dans un premier temps, l'informaticien s'est intéressé exclusivement aux langages de programmation; un tel langage étant plus ou moins évolué, selon qu'il est plus proche du langage naturel ou du langage machine. Il a défini des langages spécialisés: Fortran pour le calcul scientifique, Cobol pour la gestion, Lisp pour l'intelligence artificielle, etc. Ces langages diffèrent des langues naturelles en de nombreux points; en particulier, ils servent au calcul et non au dialogue, ils doivent être totalement explicites.

L'informaticien s'intéresse de plus en plus aux langues naturelles, dans le cadre de travaux visant à une communication entre l'homme et la machine. L'ordinateur est devenu un outil indispensable pour calculer, organiser, prévoir. Avec le micro-processeur il entre dans la vie quotidienne. Pourquoi dans ces conditions ne pas penser qu'il parlera, entendra et comprendra notre propre langue? Un grand nombre de recherches sont menées dans ce but, elles suivent deux voies: une voie technologique qui se préoccupe de concevoir des dispositifs analogues à l'oeil ou à l'oreille, c'est-à-dire capables de reconnaître un signe dans une courbe graphique ou une suite de signaux acoustiques; une voie plus tournée vers les problèmes de représentation et de formalisation de la connaissance humaine, qui cherche à modéliser le phénomène de la compréhension. Ces deux voies sont complémentaires et constituent deux aspects parmi tous ceux d'un seul phénomène étudié sous le nom d'Intelligence Artificielle.

Les expériences menées en traitement automatique des langues naturelles sont nombreuses. Si les champs d'application auxquels elles s'intéressent, correspondent à des objectifs d'utilisation différents, il n'en reste pas moins qu'elles visent toutes à atteindre un même but: donner à une machine une aptitude à comprendre le langage humain.

Parmi les champs d'application notons:

- traduction automatique;
- documentation automatique;
- système de questions-réponses;
- enseignement assisté par ordinateur;
- interrogation de bases de données;
- traitement de textes;

Autant de domaines où des parties, plus ou moins comparables, du phénomène du langage sont étudiées, pour être cernées, réduites, modélisées.

1.3. POINT DE VUE DU MATHÉMATICIEN.

Les mathématiciens se sont essentiellement intéressés aux langages de chaînes: langages des parties du monoïde libre, c'est-à-dire ensemble des séquences finies (mots) d'éléments (lettres) d'un ensemble fini (alphabet); langages algébriques sur le magma libre de Maurice Nivat [NIVA.72]

Très peu nombreux sont ceux qui se sont intéressés à d'autres types de langages. Parmi eux, Claude Pair [PA&QU.68] qui n'a pas utilisé le mot langage, mais le mot bi-langage, pour des langages sur des structures arborescentes qu'il assimile à de véritables langages à deux dimensions. Il présente un formalisme qui permet de les traiter algébriquement et généralise la notion de langage régulier à celle de bi-langage régulier.

2. PROBLÈMES INTÉRESSANTS A PROPOS DES LANGAGES

[VAUQ.69] [HO&UL.69] [SALO.73].

Un langage formel étant considéré comme un ensemble de chaînes sur un vocabulaire donné, le premier problème qui se pose, est celui de la caractérisation d'un tel langage.

A ce sujet, on peut distinguer deux points de vue. Celui qui consiste à se donner les moyens de construire toutes les chaînes du langage (ou encore de les énumérer); c'est l'aspect **génération du langage**. Celui qui consiste à déterminer un processus, à l'issue duquel une chaîne sera acceptée comme appartenant au langage ou refusée; c'est l'aspect **reconnaissance du langage**.

La **génération** d'un langage est la réalisation d'un processus qui permet de produire toutes les phrases du langage. Si l'énumération peut être infinie (dans le cas le plus fréquent, celui des langages infinis), le processus de production de chaque phrase de ce langage doit être fini. La base d'un tel processus est une **grammaire**.

La **reconnaissance** consiste à déterminer un processus, à l'issue duquel une chaîne sera acceptée ou refusée. Un tel processus est réalisé au moyen d'un **automate de reconnaissance** ou **accepteur**.

Un ensemble de langages peut avoir un certain nombre de propriétés, caractéristiques de cet ensemble. Par exemple, ces langages peuvent être tous engendrés par des grammaires du même type logique ou reconnus par des automates correspondant à la même définition. On peut alors étudier des propriétés conséquentes de ces ensembles telles que fermeture par rapport aux opérations que l'on peut définir sur ces langages (conjonction, disjonction, concaténation, étoile, plus, etc.). Un problème particulièrement important consiste à comparer les familles de langages engendrés

par certains types de grammaires d'une part et reconnus par certains types d'automates accepteurs d'autre part; il y a, en effet, un intérêt à connaître les grammaires et automates qui caractérisent les mêmes langages.

Le troisième problème intéressant est celui de l'interprétation. Il échappe à la théorie des langages formels proprement dite et apparaît au niveau des applications. Il s'agit des rapports entre l'expression formelle et le contenu sémantique, ce dernier hors d'atteinte d'une définition générale.

On définit des automates, plus généraux que les accepteurs, qui produisent une chaîne x' sur un "vocabulaire de sortie" quand, à l'entrée de l'automate, est présentée une chaîne x sur un "vocabulaire d'entrée" (éventuellement le même que le vocabulaire de sortie). De tels automates sont dits transducteurs.

La théorie des langages formels étudie la classe des langages images, en fonction de la famille des langages sources et du type d'automate utilisé pour la transduction.

DIFFERENTES APPROCHES DE CES PROBLEMES.

L'objectif fondamental de l'analyse linguistique d'une langue est de séparer les suites grammaticales qui sont des phrases du langage, des suites agrammaticales qui n'en sont pas et d'étudier la structure des suites grammaticales. La grammaire du langage sera ainsi un mécanisme engendrant toutes les suites grammaticales et aucune des suites agrammaticales.

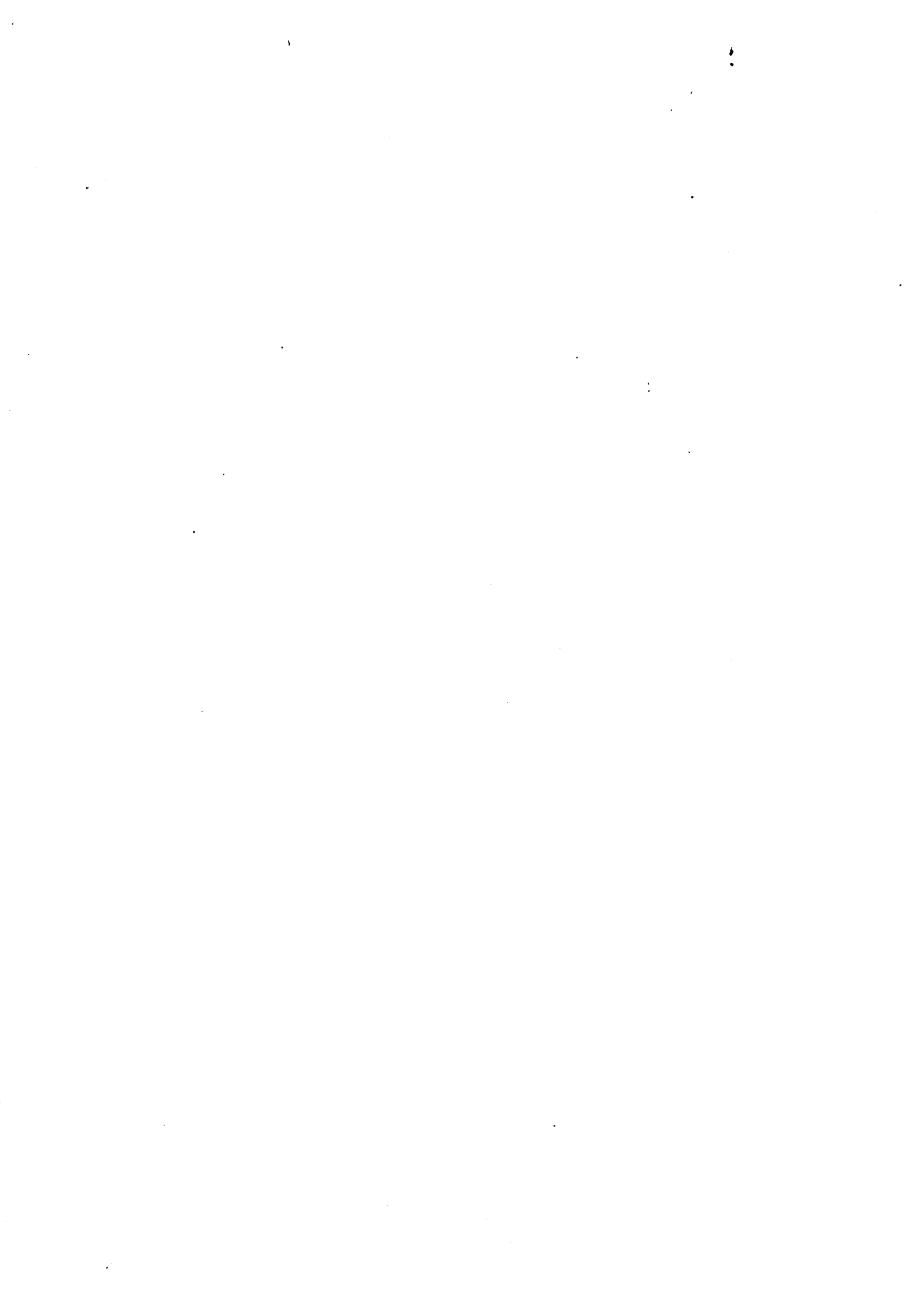
La notion de grammaticalité ne peut être associée à celle d'interprétabilité (tout au plus se borne-t-on à décrire la structure syntaxique, au sens de la syntaxe de surface, des phrases de la langue). L'approche linguistique de la langue est essentiellement générative. De même est générative l'approche mathématique.

Dans les applications à la compilation des langages de programmation ou à la traduction des langues naturelles, on procède par transduction. Le niveau d'interprétation souhaité peut être proche de la syntaxe de surface ou au contraire être beaucoup plus ambitieux et atteindre le niveau des relations logiques et sémantiques et refléter alors, non plus la "forme" mais le "sens" du texte. Le transducteur doit préserver l'interprétation que l'on a choisie, pour que l'on soit assuré du résultat. L'approche informatique du langage est essentiellement analytique et interprétative.



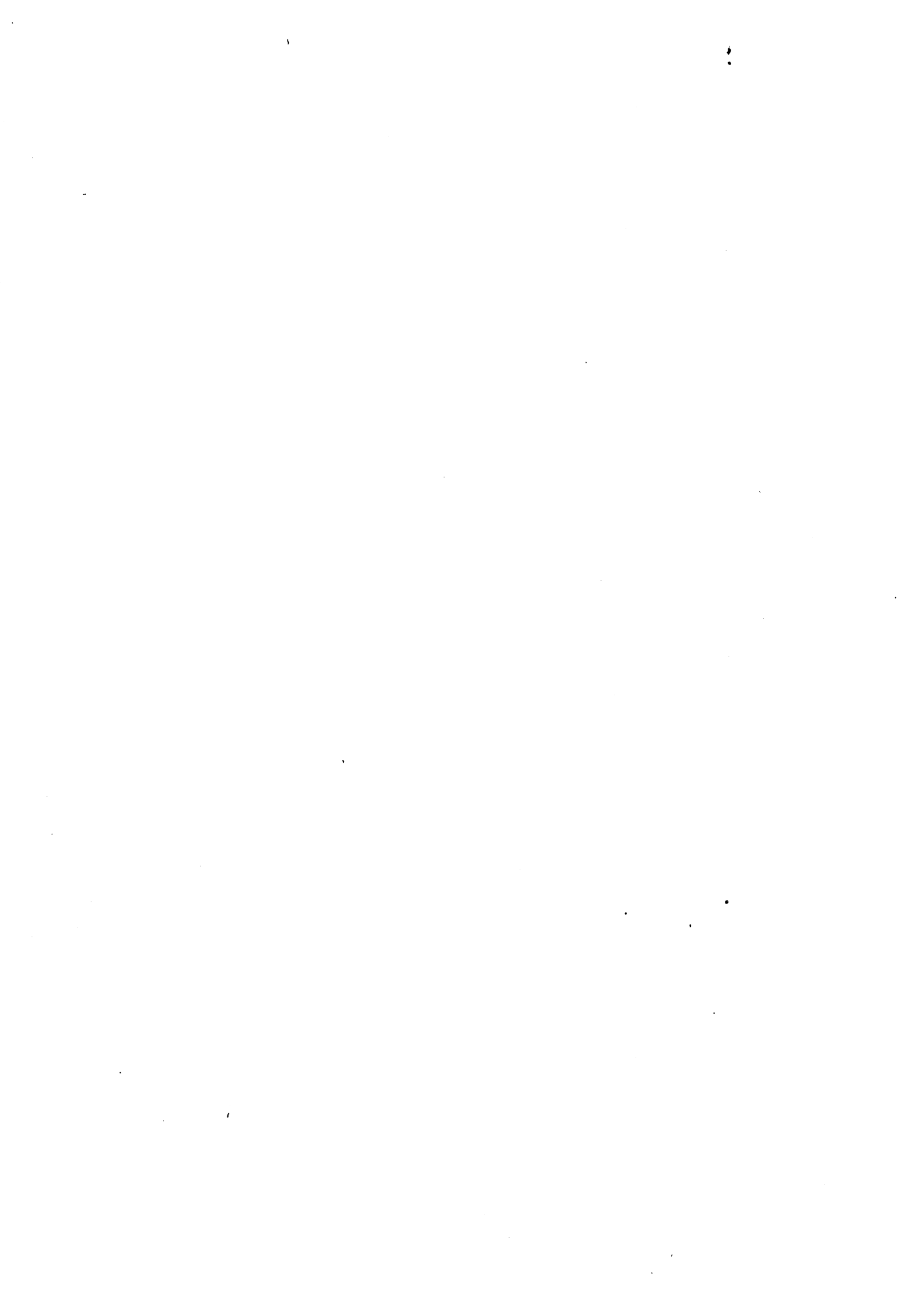
CHAPITRE II

INTERPRETATION ET TRANSDUCTION



CHAPITRE II
INTERPRETATION ET TRANSDUCTION

PREMIERE PARTIE
DESCRIPTION FORMELLE
DES NIVEAUX D'INTERPRETATION



1. DESCRIPTION FORMELLE DES NIVEAUX D'INTERPRETATION.

Toute analyse d'un texte représente un certain niveau d'interprétation du texte. Si cette interprétation est très éloignée des contingences grammaticales des langues particulières et si elle représente l'organisation logique et sémantique du texte, alors la structure associée est proche de l'universalité.

Pendant les premières années de la traduction automatique, le niveau visé a consisté à décrire la structure syntaxique, au sens de la syntaxe de surface, des phrases écrites dans une langue naturelle.

Les modèles linguistiques qui se sont prêtés le plus aisément au traitement par ordinateur furent celui des constituants immédiats et celui des dépendances.

Le premier correspond à la classe des langages Hors-Contextes (HC) en théorie des langages formels. Bon nombre d'algorithmes ont été développés pour obtenir la structure syntaxique (sous forme d'arborescence étiquetée) recherchée.

Le second couvre la même classe de langages et a donné lieu à des algorithmes dérivés des précédents ou encore à des algorithmes à filtres. Les structures obtenues sont aussi des arborescences étiquetées.

1.1. CHOIX DES DESCRIPTEURS.

La structure de données est l'arborescence, issue de l'analyse syntaxique de chaque phrase du texte, arborescence correspondant au niveau linguistique appelé "syntaxe de surface". En ce qui concerne la représentation de ces arborescences, deux écoles (fondées sur les deux modèles linguistiques, de conceptions différentes, cités) s'affrontent et entrent en compétition. Il s'agit donc, d'une part de la représentation sous forme de constituants immédiats et d'autre part de la représentation sous forme de dépendance (bien que Halm Gaïfman [GAIF.65] ait montré en 1965 que tout langage descriptible dans l'un quelconque des deux systèmes, l'est aussi dans l'autre).

Les étiquettes, dans le premier type de représentation, établissent des propriétés attachées à chaque noeud de la structure, dans le deuxième type de représentation, des relations entre ces noeuds.

Nous pouvons mettre l'opposition: propriété attachée à chaque noeud de la structure vs. relation entre noeuds en parallèle avec deux types de grammaires: **grammaires syntagmatiques hors-contexte vs. grammaires de dépendances**, qui permettent respectivement la génération des structures syntaxiques correspondantes.

1.2. REPRESENTATION EN CONSTITUANTS IMMEDIATS.

[CHOM.57] [VAUQ.68].

Le concept fondamental est celui d'**unités syntaxiques**. La phrase est représentée par une arborescence dont les feuilles sont des symboles terminaux, **catégories syntaxiques** des mots de la phrase et les noeuds des symboles non-terminaux portant des étiquettes de **classes syntagmatiques**.

Considérons la phrase suivante:

"des arborescences représentent les graphes structuraux".

Les symboles terminaux nécessaires sont:

articles (les,des)
noms communs (arborescences,graphes)
adjectif (structuraux)
verbe-personnel (représentent)

Les non-terminaux sont:

Phrase (P)
Groupe-Nominal (GN)
Groupe-Verbal (GV)

Ainsi la représentation en terme de constituants immédiats de cette phrase sera la suivante:

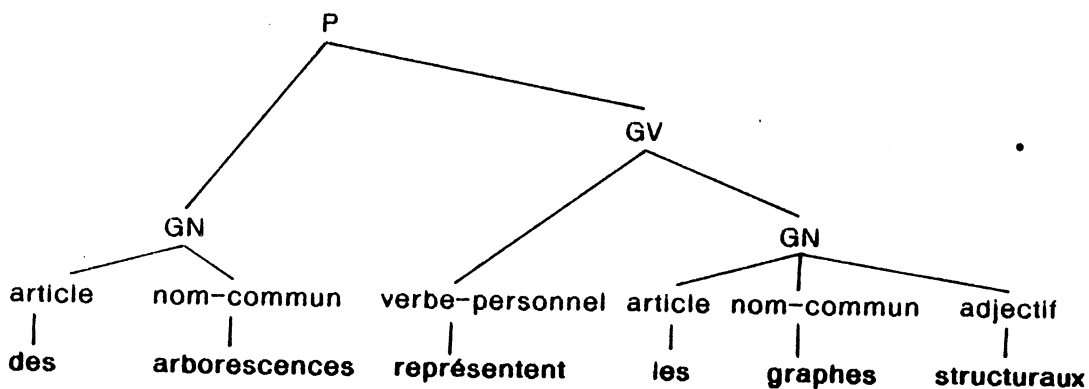


Figure II.1

Cette représentation correspond à un parenthésage de cette phrase en terme de classes:

$(P(GN_{\text{article nom-commun}}(GN) (GV_{\text{verbe-personnel}}(GN_{\text{article nom-commun}} \text{adjectif}(GN)GV)P))$ qui est un parenthésage fermé.

On associe à ce type de représentation les grammaires syntagmatiques hors-contextes. Elles permettent de décrire chaque unité syntaxique comme une chaîne d'unités syntaxiques ou de symboles terminaux.

Par exemple, la phrase est une unité lexicale représentée par P qui est la suivante:

P-->GN GV
GN-->article nom-commun/ article nom-commun adjectif
GV-->verbe-personnel GN

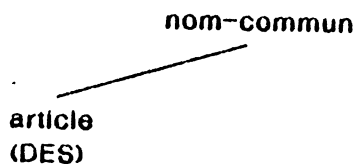
Cette grammaire est très partielle. Non seulement elle ne tient pas compte de toutes les structures de phrases, mais encore, sur le petit nombre de cas envisagés elle est incorrecte. En effet, si elle convient pour détecter la structure syntaxique de notre phrase, elle accepte une quantité impressionnante de phrases incorrectes (en particulier, elle ne tient aucun compte des accords grammaticaux). Elle se contente de vérifier que la chaîne des symboles terminaux est susceptible d'une représentation parenthésée d'unités syntaxiques.

1.3. REPRESENTATION EN TERMES DE DEPENDANCE.

[GAIF.65] [VAUQ.75]

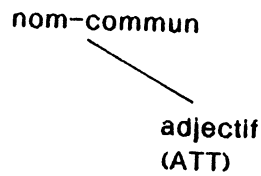
Le concept fondamental est celui de **relation de dépendance** entre deux classes de mots. Ainsi entre un article et un nom-commun, la relation de dépendance exprime que le nom-commun est le "gouverneur" et que l'article est son "dépendant".

La chaîne: "article nom-commun" donne naissance au graphe:



où "article" est le dépendant gauche de "nom-commun". cette relation exprime la désignation (DES) du nom-commun par l'article. On donne un nom à cette relation. Il sera porté par le sommet source.

La chaîne: "nom-commun adjectif" conduit à:



où "adjectif" est le dépendant droit de son gouverneur "nom-commun" dont il est attribut (ATT).

Cette relation de dépendance est donc représentée par un arc. Un élément n'est dépendant que d'un seul gouverneur. Une classe n'a pas de gouverneur. On obtient donc une arborescence qui pour notre exemple est la suivante:

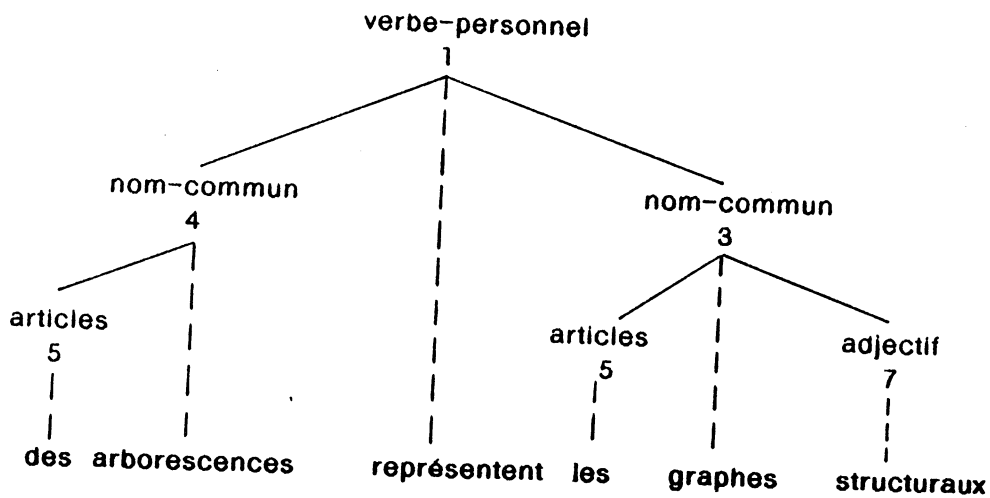


Figure II.2

Elle représente un parenthésage de notre phrase en terme de dépendance, parenthésage externe.

$(_{4}({}_{5}\text{article}_{5}) \text{nom-commun}_{4}) \text{verbe-personnel}$

$(_{3}({}_{5}\text{article}_{5}) \text{nom-commun} ({}_{7}\text{adjectif}_{7}))_{3}$

On associe à ce type de représentation une **grammaire de dépendance**, c'est-à-dire un ensemble de règles qui mettent en évidence des relations de gouverneurs à dépendants.

D'après Galfman, une règle de dépendance exprime les différentes valences d'un élément "gouverneur" par rapport à ses dépendants:

Soit X_i un élément gouverneur.

Solent $X_{j1}, X_{j2}, \dots, X_{jn}$ ses dépendants gauches.

Solent $X_{jn+1}, X_{jn+2}, \dots, X_{jk}$ ses dépendants droits.

La règle de dépendance s'écrit:

$$X_i(X_{j1}, X_{j2}, \dots, X_{jn}, *X_{jn+1}, \dots, X_{jk})$$

Une règle terminale sera de la forme: $X_i(*)$.

Une règle initiale: $*(X_i)$.

Cette définition suppose la projectivité partout.

La grammaire nécessaire à la génération de l'exemple est la suivante:

1. $*(\text{verbe-personnel})$
2. $\text{verbe-personnel}(\text{nom-commun} * \text{nom-commun})$
3. $\text{nom-commun}(\text{article} *)$
4. $\text{nom-commun}(\text{article} * \text{adjectif})$
5. $\text{article}(*)$
6. $\text{nom-commun}(*)$
7. $\text{adjectif}(*)$

1.4. DISCUSSION A PROPOS DE CES DEUX TYPES DE REPRESENTATION.

Notre but est de montrer que ces deux types de grammaires caractérisent les mêmes langages. Nous allons le faire à partir des travaux de H.Galfman [GAIF.65].

Tout d'abord nous remarquons que dans les graphes de dépendance tous les symboles sont des symboles terminaux. Il faudra projeter l'arborescence pour obtenir la chaîne d'entrée. Nous isolerons donc les grammaires et langages projectifs de ceux qui ne le sont pas.

1.4.1. PASSAGE CONSTITUANTS IMMEDIATS ---> DEPENDANCE.

On sait pouvoir mettre n'importe quelle grammaire hors-contexte sous forme de Chomsky:

$$A \rightarrow BC / a$$

où A,B,C sont des éléments de V_N .

où a est un élément de V_T .

A partir de la forme de Chomsky, il suffit d'ajouter aux règles binaires de la grammaire une indication notée "g" ou "d", selon que l'on considère le constituant de gauche ou de droite, comme principal pour former le syntagme résultant.

On définit alors une métrique: la distance entre le non-terminal en partie gauche et son constituant principal est nulle. On définit ainsi des classes d'équivalence, les éléments d'une même classe sont situés à des distances nulles. On aura autant de classes que de terminaux.

En utilisant cette propriété d'équivalence et en identifiant les sommets d'une même classe on obtient un graphe que l'on aurait pu générer directement par une grammaire de dépendance.

Reprenons notre exemple, écrivons notre grammaire sous forme de Chomsky et ajoutons lui une métrique.

d P ----> GN GV
g GV ----> V GN
d GN ----> DET GN1
g GN1 ----> GN1 GADJ
GN1 ----> nom-commun
V ----> verbe-personnel
DET ----> article
GADJ ----> adjectif

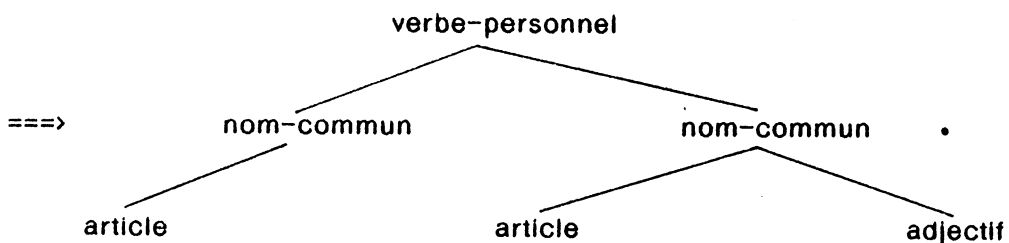
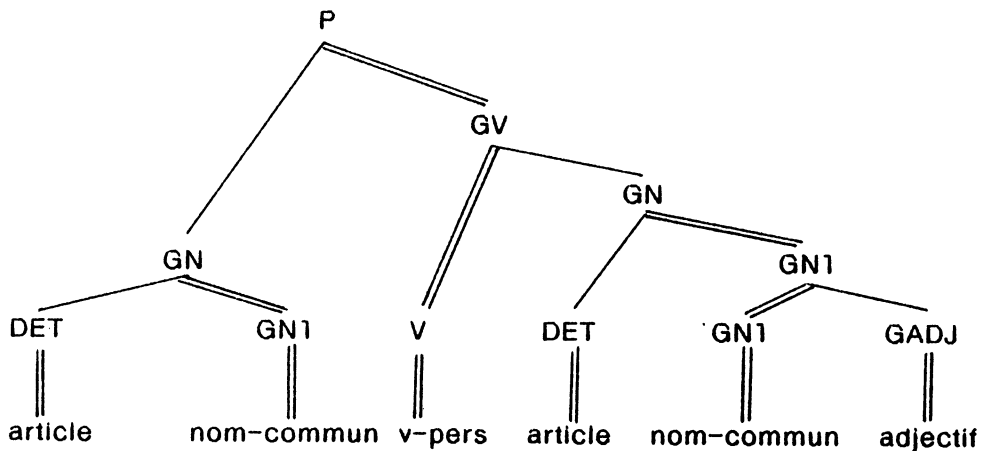


Figure II.3 Passage d'une arborescence en termes de constituants immédiats à une arborescence en termes de dépendance.

1.4.2. PASSAGE DEPENDANCE ----> CONSTITUANTS IMMEDIATS.

Pour tout système de dépendance il existe un système syntagmatique induit par lui. Il est construit comme suit:

Soit W_1, W_2, \dots, W_k toutes les catégories du système de dépendance pour lesquelles il n'y a pas de règles de la forme:

$$W_i (X_1, X_2, \dots, X_k, * X_{k+1}, \dots, X_n) \text{ où } n > 1.$$

Soit V_1, V_2, \dots, V_m les autres.

Pour chaque règle du système de dépendance, de la forme:

$$V_i (X_1, \dots, X_k, * X_{k+1}, \dots, X_n).$$

on met dans l'ensemble des règles P du système syntagmatique la règle:

$$V_i \text{ ----> } X_1 X_2 \dots X_k V_i^w X_{k+1} \dots X_n.$$

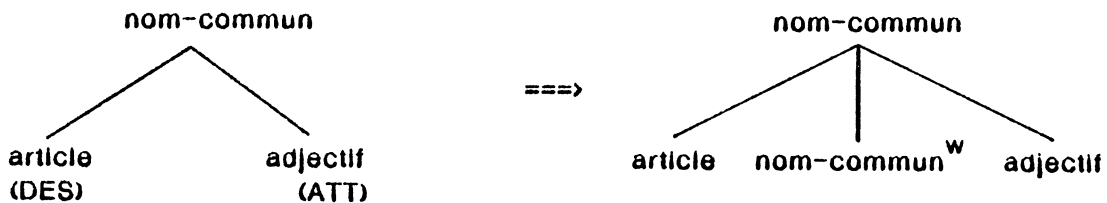
Les catégories du système syntagmatique seront:

$$W_1, \dots, W_k, V_1, \dots, V_m, V_1^w, \dots, V_m^w$$

Un mot sera dans le système syntagmatique de catégorie: W_i , s'il est de cette catégorie dans le système de dépendance.

Il sera dans le système syntagmatique de catégorie V_i^w , s'il est de catégorie V_i , dans le système de dépendance.

Le système induit exprime pleinement la relation de dépendance.



$$\text{nom-commun}(\text{article} * \text{adjectif}) \text{====>} \text{nom-commun} \text{---->} \text{article nom-commun}^w \text{ adjectif}$$

Figure II.4 Passage d'une arborescence en termes de dépendance à une arborescence en termes de constituants immédiats.

1.4.3. EXPRESSION DES DEUX NIVEAUX SUR UNE MEME STRUCTURE.

L'interprétation en termes de grammaire syntagmatique met en évidence le niveau des classes syntaxiques. Le passage à une grammaire de dépendance nécessite un niveau supplémentaire d'interprétation, celui des relations syntaxiques. On peut représenter sur une même structure ces deux niveaux d'interprétation.

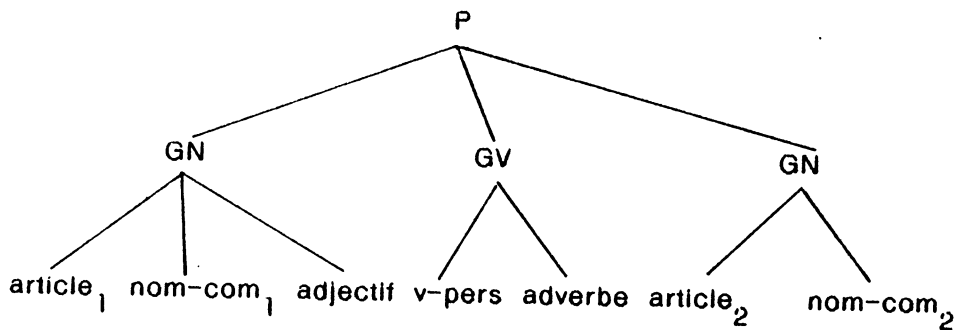


Figure II.5a arborescence en terme de classes.

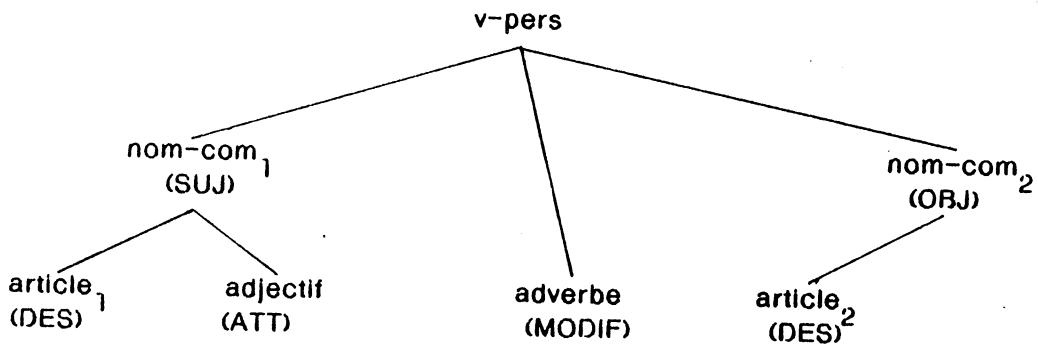


Figure II.5b arborescence en termes de relations.

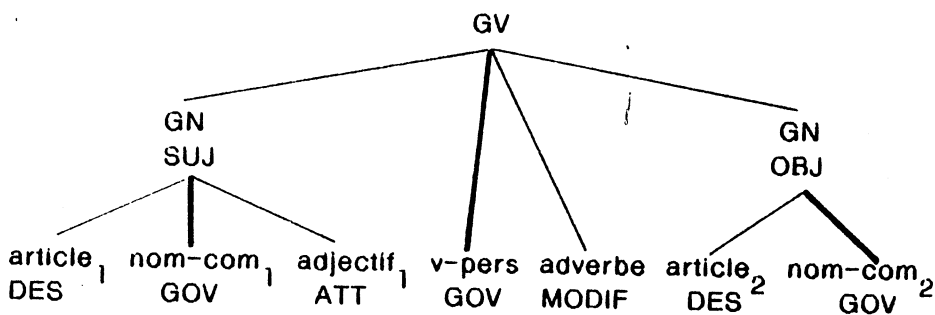


Figure II.5c structure à deux niveaux d'interprétation
relations syntaxiques + classes syntaxiques

Par la suite, des projets se sont proposés d'atteindre des niveaux d'interprétation plus ambitieux comme les structures profondes chomskiennes [CHOM.65], structures logico-syntaxiques ou encore structures logico-syntaxiques profondes. Dans tous les cas, le passage par la structure syntaxique superficielle est une étape obligatoire. Un système de règles transformationnelles permet le passage de la structure profonde à la structure de surface (la structure de surface constitue l'expression de la phrase, la structure profonde détermine son sens).

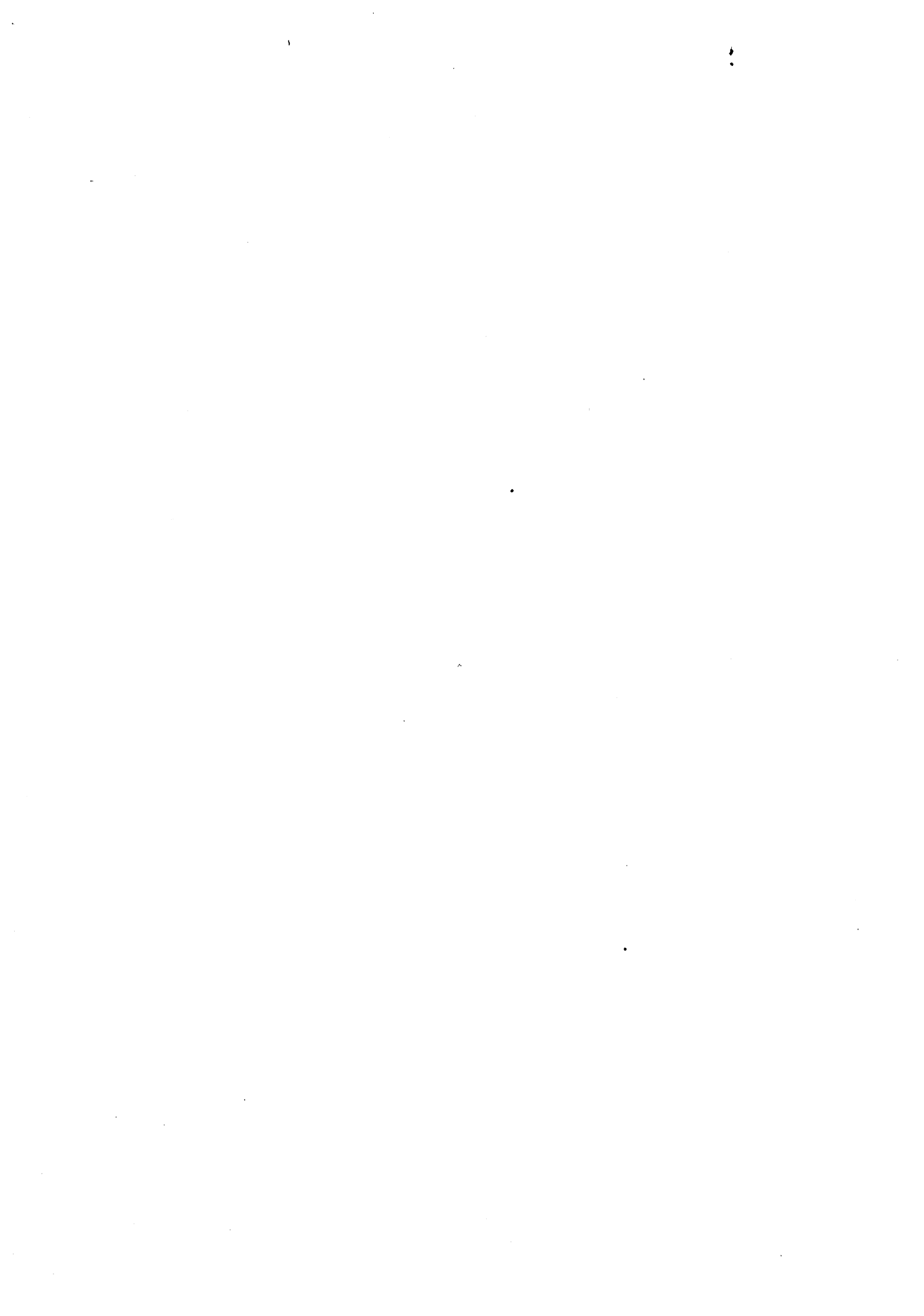
Quel que soit le niveau choisi et quelle que soit la théorie linguistique qui justifie la structure, les arborescences étiquetées n'ont dans la plupart des cas qu'une étiquette simple par sommet. L'emploi d'étiquettes complexes qui permet d'associer à chaque point du graphe plusieurs valeurs appartenant à diverses classes grammaticales et sémantiques, n'a été retenu, dans un premier temps, que par un petit nombre de chercheurs.



CHAPITRE II
INTERPRETATION ET TRANSDUCTION

DEUXIEME PARTIE

ACCES A CES NIVEAUX



2. ACCES A CES NIVEAUX.

2.1. PAR "PARSING" HORS-CONTEXTE [AH&UL.72] [HARR.78].

Un "parsing" est la "délinéarisation" d'une entrée linguistique. c'est-à-dire que la syntaxe et d'autres sources de connaissance sont utilisées pour déterminer les fonctions des mots dans la phrase en entrée, dans le but de créer une structure de données, telle qu'un arbre de dérivation, qui pourra être utilisée pour obtenir la "signification" de la phrase.

Nous disons qu'une phrase λ d'un langage $L(G)$, pour une quelconque grammaire H-C G , a été syntaxiquement analysée quand nous connaissons un de ses arbres de dérivation (ou peut-être tous).

Dans un processus de traduction, cet arbre peut être "physiquement" construit en mémoire, mais sa représentation peut être beaucoup plus subtile; nous verrons que le plus souvent elle est simplement déduite des pas de l'analyseur syntaxique.

On a cherché à découvrir des algorithmes de plus en plus performants pour les analyseurs syntaxiques. Dans tous les cas, l'algorithme provient de la théorie des automates, où le but même est la détection des phrases bien formées. Les analyseurs syntaxiques ou parsers ont donc été des "automates accepteurs" ayant à répondre par oui ou par non, au sujet de la validité de toute chaîne qui leur est soumise. La structure d'arbre déduite (en cas de réponse positive), qui est le résultat le plus intéressant, n'est en fait qu'un sous-produit de ces analyseurs.

Dans le traitement des langues naturelles, ces analyseurs font appel à un algorithme combinatoire qui pour chaque phrase recherche toutes les solutions compatibles avec la grammaire, ensemble statique de toutes les règles, et les dictionnaires. (Le dictionnaire affecte des classes aux éléments; la grammaire établit des relations entre ces classes; l'algorithme essaie toutes les configurations permises par les règles de grammaire). Pour une phrase donnée, toutes les configurations qui répondent aux critères d'acceptation sont fournies, d'où, parfois, un grand nombre de structures qui expriment l'ambiguïté de la grammaire formelle (ambiguïté souvent due à la langue elle-même, au niveau d'interprétation où on se place ou au formalisme choisi).

Nous avons besoin, dans un premier temps, d'établir quelques définitions et conventions sur les analyseurs syntaxiques.

Soit $G=(V_T, V_N, S, P)$,
où V_T est le vocabulaire terminal
où V_N est le vocabulaire auxiliaire
où S est l'axiome de la grammaire
et où P est l'ensemble des productions, une grammaire H-C dont les productions sont numérotées de 1 à p .

Une **analyse gauche (left parse)** d'une chaîne α est une séquence de productions utilisées dans une dérivation gauche de α à partir de S dans G .

Une **analyse droite (right parse)** de α est l'inverse d'une séquence de productions utilisées dans une dérivation droite de α à partir de S dans G .

Puisqu'il s'agit non seulement d'accepter ou de refuser une phrase donnée, mais encore de préciser la structure syntaxique, on peut imaginer la construction de cette arborescence soit à partir de la racine S en développant progressivement jusqu'à la coïncidence avec la phrase, soit à partir des symboles terminaux en construisant successivement les unités syntaxiques jusqu'au symbole S .

2.1.1. ANALYSEUR DESCENDANT (TOP-DOWN).

La construction de l'arbre se fait à partir de l'axiome de la grammaire; en effet, $\Pi=i_1 i_2 \dots i_n$ étant une analyse gauche d'une phrase λ de $L(G)$, on peut construire un arbre d'analyse de λ de la manière suivante. Nous commençons par la racine S alors i_1 donne la production qui doit être utilisée pour étendre S .

Si $i_1: S \rightarrow X_1 \dots X_k$

alors nous créons les k descendants de S et nous les notons $X_1 X_2 \dots X_k$.

Si $X_1 X_2 \dots X_{i-1}$ sont des terminaux

alors les $i-1$ premiers symboles de λ doivent être $X_1 \dots X_{i-1}$ et la production i_2 doit être de la forme $X_i \rightarrow Y_1 Y_2 \dots Y_l$ et nous continuerons à construire l'arbre d'analyse syntaxique de λ en étendant le noeud X_i .

Nous pouvons en procédant de cette façon construire la totalité de l'arbre d'analyse de λ , arbre correspondant à l'analyse gauche Π .

Analyse gauche d'une chaîne λ sur V_T^* .

Une façon de voir le problème, est de dire que nous connaissons sa racine et sa frontière et que notre but est de trouver les noeuds intermédiaires. Un analyseur gauche suggère que nous établissions l'analyse en partant de la racine, en travaillant de gauche à droite, jusqu'à obtention de la frontière.

Chaque symbole examiné conduit à effectuer deux actions:

- a) S'assurer que le symbole terminal réalise la prédiction de tête, d'une liste de prédiction et effacer cette prédiction satisfaite.
- b) Enregistrer dans l'ordre les prédictions imposées par ce symbole et les placer en tête de la liste des prédictions.

Au départ, la liste contient la seule prédiction S. A chaque pas, la liste de prédictions ajoutées peut éventuellement être vide. La phrase est acceptée si et seulement si, après examen du dernier symbole, la liste des prédictions est vide.

Du point de vue algorithmique, cette analyse consiste à réaliser un automate à pile (à un seul état) qui accepte les chaînes du langage par critère de pile vide, et à l'associer à un compteur de position dans la chaîne d'entrée.

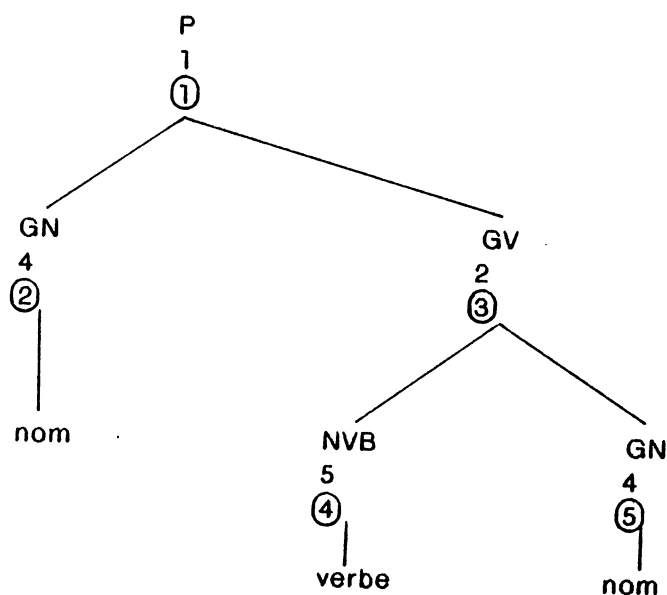
Exemple:

- P:
1. P ----> GN GV
 2. GV ----> NVB GN
 3. GN ----> NVB
 4. GN ----> nom
 5. NVB ----> verbe

	CHAINE D'ENTREE	PILE	ACTION	REGLFS
1	nom verbe nom \$	P \$	prédiction	1
2		GN GV \$	prédiction	4
3		nom GV \$	lecture	
4	verbe nom \$	GV \$	prédiction	2
5		NVB GN \$	prédiction	5
6		verbe GN \$	lecture	
7	nom \$	GN \$	prédiction	4
8		nom \$	lecture	
9	\$	\$		
10	ε	ε		Π

Figure II.6a Analyse descendante.

La chaîne a été reconnue, puisqu'en fin d'analyse la pile est vide. La lecture de la suite des règles utilisées $\Pi=1,4,2,5,4$ permet de déduire la structure syntaxique correspondant à cette analyse.



(étapes de construction numérotées de 1 à 5).

Figure II.6b Arbre déduit de l'analyse descendante.

Cet exemple nous permet de prendre conscience d'un certain nombre de faiblesses de l'algorithme:

(1) L'analyseur gauche en général n'est pas déterministe. Au cours de l'analyse, nous avons dû effectuer un choix au pas 4. Un autre choix que celui fait aurait conduit, après exploration d'un ou de plusieurs symboles, à une situation de blocage (un symbole lu par la suite n'aurait pas satisfait la prédiction avancée). Dans d'autres circonstances, plusieurs choix conduiraient à une situation acceptable, ce qui signifierait que la phrase présenterait une ambiguïté de structure dont on ne pourrait rendre compte.

(2) Pour certaines grammaires, une simulation complète est impossible; par exemple, certains mots possèdent une infinité d'analyses gauches si la grammaire n'est pas sans cycle. Bien plus, la simulation échoue pour une large classe de grammaires, celles qui sont récursives à gauche.

2.1.2. ANALYSEUR ASCENDANT (BOTTOM-UP).

C'est une méthode d'analyse opposée à la précédente, en ce sens que l'arborescence se construit progressivement depuis les feuilles en remontant vers la racine, à partir de l'analyse droite de la chaîne.

Une analyse droite pour une chaîne λ dans la grammaire G est une séquence de productions qui peut être utilisée pour réduire une chaîne λ au symbole de phrase S .

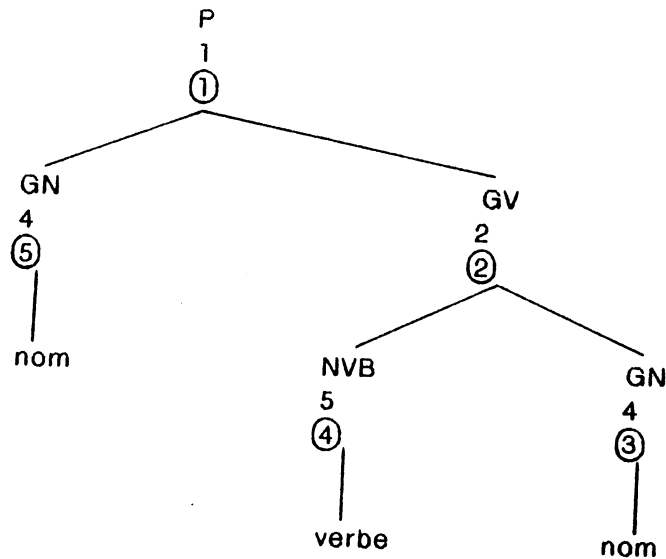
Ce type d'analyse nécessite un automate à pile qui reconnaîtra le langage par critère de pile vide. Un mouvement consiste à examiner la chaîne qui est au sommet de pile, pour voir s'il y a correspondance avec une partie droite d'une règle de production. Si oui, on substitue aux symboles de sommet de pile reconnus la partie gauche de cette production. Si plus d'une réduction est possible, on ordonne les réductions de façon arbitraire. Si aucune réduction n'est possible, on fait glisser un nouveau symbole d'entrée en sommet de pile et on recommence.

Exemple:

- | | | | | |
|---|----|-----|-------|--------|
| P | 1. | P | ----> | GN GV |
| | 2. | GV | ----> | NVB GN |
| | 3. | GV | ----> | NVB |
| | 4. | GN | ----> | nom |
| | 5. | NVB | ----> | verbe |

	PILE	CHAINE D'ENTREE	ACTION	REGLE
1	ε	nom verbe nom ‡	lecture	
2	nom	verbe nom ‡	réduction	4
3	GN		lecture	
4	GN verbe	nom ‡	réduction	5
5	GN NVB		réduction	3->Ech
<p>Le choix d'une telle réduction amènera à la présence de S seul au sommet de pile avec une chaîne d'entrée non totalement lue. Pour que l'analyse de la chaîne "nom verbe nom" puisse être effectuée, il faut associer à cet analyseur un système de retour-arrière.</p>				
7	GN NVB nom	‡	réduction	4
8	GN NVB GN		réduction	2
9	GN GV		réduction	1
10	P		réduction	1->Suc

Figure II.7a Analyse ascendante.



(Etapes de construction numérotées de 1 à 5)

Figure II.7b Arbre déduit de l'analyse ascendante.

2.1.3. PROBLEMES LIES A CES ANALYSEURS ET SOLUTIONS.

2.1.3.1. PROBLEME DU NON-DETERMINISME.

Solution 1: Analyse avec "backtracking" (retour-arrière).

Pour une chaîne en entrée λ et pour un automate A non-déterministe, si l'on n'est intéressé que par une seule sortie pour λ , la première séquence valide permet l'arrêt de A. Naturellement, si aucune séquence valide n'est trouvée, il faut que toutes les séquences aient été traitées.

Nous pouvons définir dans cette optique l'analyse avec backtracking (retour-arrière) dans les termes suivants:

Les séquences de mouvements ont été arrangées dans un ordre tel qu'il est possible de simuler le prochain mouvement, en revenant en arrière jusqu'à une configuration où un choix a été fait (plutôt qu'un ou plusieurs autres) et en changeant ce choix.

Le backtracking permet de simuler le déterminisme d'analyseurs gauches et droits, pour une large classe de langages H-C. Toutefois, dans certains cas, une telle simulation peut devenir extrêmement coûteuse en temps: fonction exponentielle de la longueur n de la chaîne $O(r^n)$ (espace: $O(r.n)$).

Solution 2: stratégie d'analyse parallèle.

Le processus est proche de l'analyse descendante. L'analyse commence à la racine qui est le symbole de départ, puis tous les chemins qui peuvent résulter de la racine sont générés; cela implique qu'aucun choix sur les règles de production ne soit fait. Les (ou le) chemins possibles sont choisis en comparant le noeud courant de chaque arbre partiel résultant avec le symbole d'entrée courant. S'il y a correspondance, alors ces (ou ce) chemins sont pris et on génère tous les chemins possibles résultants. Les chemins où il n'y a pas correspondance sont éliminés immédiatement, le processus s'effectue récursivement jusqu'à ce qu'on ait parcouru toute la chaîne d'entrée.

L'arbre d'analyse est "complet" s'il y a un "chemin" dominé par la racine et se terminant avec la totalité de la chaîne d'entrée en tant que "feuilles".

Cette méthode construit tous les arbres partiels, mais elle permet de trouver toutes les solutions et élimine le retour-arrière.

2.1.3.2. REDUCTION DE L'ASPECT COMBINATOIRE.

Deux méthodes d'analyse se distinguent par leurs performances:

- l'algorithme de Cocke-Younger-Kasami [HAYS.67] [YOUN.67] [KASA.65] [KA&TO.69] [HARR.69]
- l'algorithme de Earley [EARL.70].

Le premier requiert:

- un temps moyen en n^3
- et une place en n^2 .

Le second requiert:

- un temps maximum en $O(n^3)$
- une place en $O(n^2)$

mais aussi:

- pour de grammaires non-ambigües, un temps en $O(n^2)$
- pour des grammaires déterministes, un temps en $O(n)$.

C'est ce dernier que nous allons exposer ici.

Algorithme d'EARLEY [EARL.70] [AHO.73].

Soit une grammaire $G=(V_T, V_N, S, P)$.

L'algorithme d'Earley réduit une chaîne: $X=X_1 X_2 \dots X_n$. Pour cela il regarde en avant un nombre fixé k , de symboles. A la réduction de chaque symbole X_i , un ensemble d'items, S_i , est construit qui représente les conditions du processus de reconnaissance à ce niveau de la réduction.

Chaque item dans l'ensemble est représenté comme une production avec un point à l'intérieur, suivie par une chaîne et un entier.

La production est telle que nous pouvons dériver une partie de la chaîne d'entrée à partir de sa partie droite;

Le point, dans cette production, exprime "la quantité de partie droite" qui a déjà été reconnue;

L'entier est un pointeur vers la position dans la chaîne d'entrée (représentée par l'ensemble d'items dans lequel cet état fut créé), source de la recherche de cette production.

La chaîne de k symboles est un successeur syntaxiquement reconnu à ce cas de la production.

Ce quadruplet est représenté ici comme une production, avec un point à l'intérieur, suivie par un entier et une chaîne.

L'algorithme fonctionne suivant trois opérations de base: prédiction, réduction, complétion. Il fournit un ensemble d'items S_i pour chaque symbole d'entrée X_i .

Il construit S_0 comme suit:

Pour tout $S \rightarrow \alpha$ dans P

Ajouter $[S \rightarrow \cdot \alpha, 0]$ à S_0

Si $[A \rightarrow \alpha.B\beta, 0]$ dans S_0 et $B \rightarrow \gamma$ dans P
alors Ajouter $[B \rightarrow \cdot \gamma, 0]$ dans S_0 .

com: prédiction.

Si $[A \rightarrow \alpha \cdot, 0]$ dans S_0

alors pour tout item de S_0 de la forme $[B \rightarrow \beta.A\gamma, 0]$

Ajouter $[B \rightarrow \beta.A \cdot \gamma, 0]$ à S_0 .

com: complétion.

Il construit S_j comme suit:

Pour tout $[A \rightarrow \alpha.X_j\beta, i]$ dans S_{j-1}

Ajouter $[A \rightarrow \alpha.X_j \cdot \beta, i]$ à S_j .

com: réduction.

Si $[A \rightarrow \alpha.B\beta, i]$ dans S_j et $B \rightarrow \gamma$ dans P

alors Ajouter $[B \rightarrow \cdot \gamma, j]$ à S_j .

com: prédiction.

Si $[A \rightarrow \alpha \cdot, i]$ dans S_j

alors pour tout item de la forme $[B \rightarrow \beta.A\gamma, k]$ dans S_i

Ajouter $[B \rightarrow \beta.A \cdot \gamma, k]$ à S_j .

com: complétion.

La chaîne $X=X_1 X_2 \dots X_n$ est dans $L(G)$ ssi $[S \rightarrow \alpha \cdot, 0]$ est dans S_n .

Cet algorithme développe toutes les analyses simultanément et combine les sous-analyses; cela décroît énormément l'effort de combinatoire et élimine aussi le problème de la récursivité à gauche. L'utilisation de $k=1$ est généralement suffisante, car la plupart des langages utilisent seulement un

caractère de contexte pour lever les ambiguïtés de structure, et si deux caractères sont nécessaires, l'algorithme sera plus long mais ne se bloquera pas sur le problème.

La structure arborescente est construite à partir de la liste d'analyses développée par l'algorithme (liste des ensembles d'items successifs).

Exemple:

```

P      0:      R ----> P $
      1:      P ----> GN PRED $
      2:      PRED----> GV $
      3:      GV ----> NVB GN $
      4:      GN ----> NVB $
      5:      GN ----> nom $
      6:      NVB ----> verbe $
    
```

Chaîne d'entrée: "nom verbe nom".

La génération des états par: prédiction, réduction et complétion est la suivante:

S0	R	---->.P	\$ 0
	P	---->.GN PRED	\$ 0
	GN	---->.nom	\$ 0

Initialement.

Par prédictions successives sur P puis GN on obtient S0

P ne peut être suivi que d'une fin de texte \$

Par réduction avec X1="nom"

(nom étant à la droite du "." dans GN---->.nom on détermine S1

```
S1 P ---->nom. $ 0
```

Par complétion avec X2="verbe", puisqu'il y a un point à la fin d'une production, on revient à l'ensemble d'items indiqué par le pointeur (le pointeur à 0 pour l'ensemble d'états S0) et on transforme les productions qui avaient à la droite du point le terminal qui était à l'origine de la production.

```
P ---->GN. PRED $ 0
```

Par prédiction (sur .PRED) on crée des productions au niveau 1 comme précédemment.

S1	GN	---->nom.	\$ 0
	P	---->GN. PRED	\$ 0
	PRED	---->.GV	\$ 1
	GV	---->.NVB GN	\$ 1
	GV	---->.NVB	\$ 1
	NVB	---->.verbe	\$ 1

réduction avec X1.

complétion.

Prédictions successives sur PRED, GV.NVB (création au rang donc pointeur à "1").

De la même façon on obtient S2 puis S3 avec successivement X2="verbe", X3="nom".

S2	NVB	---->verbe.	\$ 1	Réduction avec X2="verbe"
	GV	---->NVB. GN	\$ 1	Par complétion dans S1 on transforme les règles qui ont .NVB
	GV	---->NVB.	\$ 1	
	GN	---->.nom	\$ 2	Par prédiction (.GN) au niveau 2.

S3	GN	---->nom.	\$ 2	Par réduction avec X3="nom"
	GV	---->NVB GN.	\$ 1	Par complétion au niveau 2 sur .GN
	PRED	---->GV.	\$ 1	Par complétion au niveau 1 sur .GV
	P	---->GN PRED.	\$ 0	Par complétion au niveau 1 sur .PRED
	R	---->P.	\$ 0	Par complétion au niveau 0 sur .P

Figure II.8a Analyse utilisant l'algorithme de Earley.

Le processus se termine en S3 ensemble d'items qui correspond à X3.

R---->P. \$ 0 est dans S3, la chaîne "nom verbe nom" est dans L(G).

A partir de cette analyse on obtient un seul arbre:

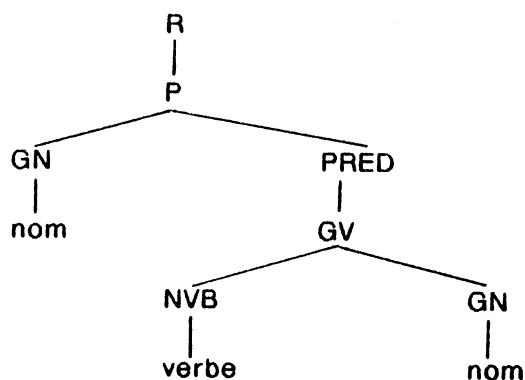


Figure II.8b Arbre déduit de l'analyse précédente.

Notons l'utilisation, faite par Chang May See [CHAN.78], de cet algorithme pour une analyse du malais fort intéressante.

Les classes de grammaires auxquelles nous allons nous intéresser sont les grammaires LL(K) pour lesquelles l'analyseur gauche peut être rendu déterministe s'il regarde k symboles à droite du symbole courant en entrée, et

et les grammaires LR(k) pour lesquelles l'analyseur droit peut travailler de façon déterministe s'il lui est possible de regarder k symboles en entrée au delà de sa position courante.

Sur ces grammaire l'analyseur effectue $C1 \cdot n$ opérations et occupe un espace de l'ordre de $C2 \cdot n$ (où C1 et C2 sont des constantes petites).

Nous allons donc définir ces deux classes de langages H-C pour lesquelles le problème de l'analyse échappe à la multiplicité des essais infructueux dus à l'aspect combinatoire des remplacements de sous-chaînes par des symboles auxiliaires en utilisant les règles de toutes les manières possibles.

Grammaires LL(k).

Soit $G=(V_T, V_N, S, P)$ une grammaire H-C. Nous disons que G est LL(k) pour un entier k fixé, si pour tout couple de dérivations gauches:

$$(1) S \xRightarrow{*} wA\alpha \xRightarrow{*} w\beta\alpha \xRightarrow{*} wx$$

$\text{Im } x \quad \text{Im} \quad \text{Im}_k$

$$(2) S \xRightarrow{*} wA\alpha \xRightarrow{*} w\gamma\alpha \xRightarrow{*} wy$$

$\text{Im} \quad \text{Im} \quad \text{Im}$

tels que $\text{Premier}_k(x) = \text{Premier}_k(y)$ on a $\gamma = \beta$

$(\text{Premier}_k(x) = \{w \in V_T / (x \xRightarrow{*} w\beta \text{ et } |w|=k) \vee (x \xRightarrow{*} w \text{ et } |w| < k)\})$

Dans l'automate général, cela signifie qu'en examinant k symboles à droite du symbole courant dans la chaîne d'entrée, l'expansion du non-terminal en sommet de pile est unique (analyse descendante k-prédictive). Une grammaire est LL si elle est LL(k) pour un k, LL(0) si elle est LL(k) pour tout k.

Grammaires LR(k).

Une grammaire $G=(V_T, V_N, S, P)$ est LR(k), $k \geq 0$ si pour tout couple (1), (2) de dérivations dans la grammaire augmentée G'

$$(G' = (V_T, V_N \cup \{S\}, S', P \cup \{S' \rightarrow S\}))$$

$$(1) S' \xRightarrow{*} \alpha A w \xRightarrow{*} \alpha \beta w$$

$\text{rm } x \quad \text{rm}$

$$(2) S' \xRightarrow{*} \gamma B x \xRightarrow{*} \alpha \beta y$$

$\text{rm} \quad \text{rm}$

tels que $\text{Premier}_k(w) = \text{Premier}_k(y)$

alors $\alpha A w = \gamma B x$ (c'est-à-dire $\alpha = \gamma, A = B, x (=w) = y$)

Ainsi, dans la dérivation la plus à droite, de deux chaînes qui coïncident jusqu'au même symbole de l'occurrence du dernier remplacement, les deux autres chaînes obtenues à l'avant dernier pas de dérivation coïncident jusqu'au même symbole de l'occurrence du dernier remplacement. Le processus d'analyse balaye la chaîne d'entrée de gauche à droite avec une avance de k symboles. Une grammaire est LR si elle est LR(k) pour un k, LR(0) si elle est LR(k) pour tout k.

2.2. PAR FILTRAGE DANS UN MULTIGRAPHE.

La construction du graphe de dépendance se prête à une stratégie d'analyse qui n'a pas d'équivalent dans l'autre représentation. A ce sujet, l'algorithme de O.Kulagina [KU&VA.59] [KULA.59] [KU&ME...] est particulièrement significatif.

Le principe est de construire tous les arcs de dépendance compatibles avec une "table de dépendance"; Cette construction fournit toutes les relations possibles entre les éléments pris deux à deux, car elle doit permettre l'obtention de toutes les relations pertinentes; de ce fait elle comporte un certain nombre de relations qui, elles, ne sont pas pertinentes pour une phrase donnée. Par la suite, le but de l'algorithme sera de les déterminer; pour cela, il utilisera des règles d'élimination (ou filtres) d'abord générales, puis poursuivra par des filtres de plus en plus particuliers.

Construction de la table de dépendance.

La table établit des relations entre deux syntagmes élémentaires: le gouverneur et son dépendant.

Chaque relation est caractérisée par: le type de liaison auquel elle appartient, les conditions grammaticales sur le gouverneur, les conditions grammaticales sur le dépendant, l'ordre relatif du gouverneur et de son dépendant dans la phrase (cet ordre est éventuellement indifférent).

TYPE DE LIAISON	COND SUR DEP	COND SUR GOV	ORDRE
1.détermination	art.mas.sing art.fém.sing dém.mas.sing	nmc.mas.sing nmc.fém.sing nmc.mas.sing	d-1-g d-1-g d-1-g
2.verbe-sujet	nmc.sing nmc.plur pr.1.sing formule ...	3.sing 3.plur 1.sing 3.sing ...	d-2-g d-2-g d-2-g d-2-g
3.modification	adv-md-vb	vb	d-3-g

ETC.....

Figure II.9

Les mots n'ayant pas de gouverneur comme les verbes, les adverbes modificateurs de toute une phrase, etc. sont signalés comme étant leur propre gouverneur.

A partir de la table de dépendance, on construit, pour une phrase donnée, toutes les relations possibles entre les différents éléments. On obtient, pour cette phrase, un multigraphe qui contient toutes les relations pertinentes pour cette phrase ainsi qu'un certain nombre de relations qui ne le sont pas.

Elimination des relations non-pertinentes.

Elle repose sur l'utilisation par des algorithmes de trois jeux de règles:

(1) le premier jeu de règles établit les liaisons incompatibles.

Exemple: un verbe qui gouverne comme objet un nom ou un infinitif ne peut gouverner les deux.

Dans "SAVOIR FAIRE QUELQUE CHOSE" la règle exige que l'une des deux relations

SAVOIR---->FAIRE
SAVOIR---->CHOSE disparaisse.

C'est l'algorithme qui choisira. (Dans le cas présent, il se fondera sur le critère suivant: un mot ne peut avoir qu'un seul gouverneur, pour le moment CHOSE en a deux, on éliminera donc SAVOIR---->CHOSE qui est incompatible avec ce critère. Un autre critère serait d'éliminer les liaisons qui détruisent la projectivité du graphe de dépendance).

(2) le deuxième jeu de règles consiste à décrire les liaisons interdites, par la présence dans la chaîne d'une unité syntaxique (ou lexicale) particulière, située entre le gouverneur et son dépendant.

Par exemple, une règle de ce type imposera l'élimination des arcs de dépendance qui relient des mots de part et d'autre d'un signe de ponctuation forte (":", ";", "."), où, dans beaucoup de cas, de part et d'autre d'un verbe.

(3) le troisième jeu de règles établit des préférences: elles permettent de choisir entre plusieurs gouverneurs en attribuant des poids (a priori) à ces gouverneurs; poids pouvant être modifiés par calcul en fonction des liaisons déjà effectuées.

L'algorithme d'utilisation de ces trois jeux de règles est le suivant:

```
tantque nbreliasons >= nbremots
faire
début
  tantque "¬stabilité" et nbreliasons >= nbremots
  faire exécuter(1)
  si nbreliasons >= nbremots
  alors
  début
    exécuter(2)
    si nbreliasons >= nbremots et "(2)n'a pas éliminé d'arc"
    alors
    début
      faire exécuter(3)
      si "(3) n'a pas éliminé d'arc"
      alors nbreliasons := nbremots - 1
    fsi
  fsi
fin.
```

Le processus s'arrête (car exécuter(x) s'arrête) dès que le nombre de liaisons restantes est égal au nombre de mots de la phrase moins un. En effet, il n'y a alors plus rien à éliminer.

2.3. PAR TRANSDUCTION.

A ce niveau apparaît clairement la nécessité de définir un autre type d'analyseurs capables de résoudre les problèmes soulevés par l'utilisation d'accepteurs.

Il faut se dégager de la contrainte, très forte pour les langues naturelles, de se fonder sur des modèles H-C. Il faut que la structure de sortie ne soit plus un sous-produit de l'analyseur. Il s'avère nécessaire d'introduire deux notions, celle de stratégie qui permet interruption et backtracking immédiat à la découverte d'une erreur, et celle d'heuristique qui consiste à donner priorité au chemin qui a la plus forte chance de succès. Pour des raisons pratiques, enfin, il est hautement intéressant de travailler avec des grammaires modulaires qui permettent un "décorticage" plus aisé en cas d'augmentation du volume de la grammaire ou de correction sur cette grammaire.

Les automates transducteurs opèrent des transformations à partir de la structure d'entrée jusqu'à l'obtention de la structure de sortie, dont le niveau d'interprétation est défini à l'avance.

Nous allons étudier les modèles de transduction sous les différents aspects: modèle de substitution, modèle de création, modèle d'adjonction dans divers cas de structures de données: chaînes, arborescences, graphes de chaînes d'arborescences. Pour cela nous nous référerons à trois d'entre eux: les Systèmes-Q de Colmerauer [COLM.71], les A.T.N. de Woods [WOOD.70] [WOOD.71] et le système ROBRA [BO&GU&QU.78] mis au point au G.E.T.A.

2.3.1. UN MODELE D'ADJONCTION (GRAPHE DE CHAINES D'ARBRES - GRAPHE DE CHAINES D'ARBRES): LES SYSTEMES-Q DE COLMERAUER [COLM.71].

Il s'agit d'un transducteur dont les données lexicales et grammaticales sont incorporées au système. La structure de données est un graphe de chaînes où chaque élément de chaîne est une arborescence étiquetée. Le graphe ne contient pas de circuit, et il a deux sommets respectivement appelés entrée et sortie qui sont tels que tout arc du graphe figure dans un chemin allant de l'entrée à la sortie.

Exemple:

HE+HAS+BROKEN+GLASSES

ou

HE HAS BROKEN GLASSES



(chaîne d'arborescences, ici un élément est une arborescence d'un seul sommet)

Une règle en système-Q comporte en partie gauche la description d'une chaîne et en partie droite celle qui permettra la "réécriture" de la partie gauche si elle est détectée, ainsi qu'une condition.

Une grammaire en système-Q est un ensemble de telles règles de transduction. Cet ensemble s'applique sur un graphe de chaînes donné et le transforme en un nouveau graphe de chaînes. La transformation se fait en deux phases distinctes: tout d'abord addition d'un certain nombre de flèches au graphe, ensuite suppression de certaines flèches jugées inutiles.

L'algorithme réalise la combinatoire totale. Toutes les règles sont essayées sur le graphe initial et sur tous les résultats intermédiaires, de toutes les manières possibles. Le processus s'arrête lorsqu'aucune règle n'est applicable (ou par saturation de la mémoire).

Les systèmes-Q constituent un remarquable outil où la grammaire est considérée comme une donnée. Les dictionnaires eux-mêmes sont incorporés au système. En effet, les articles de dictionnaires s'écrivent avec le même formalisme que les règles de transformation.

Exemple: HE == PRONOM (HE.3.MASC.SING).

Le système peut donc prendre en charge la compilation des grammaires et des données lexicales. Si des informations lexicales sont associées à chaque mot, on peut imaginer la structure:

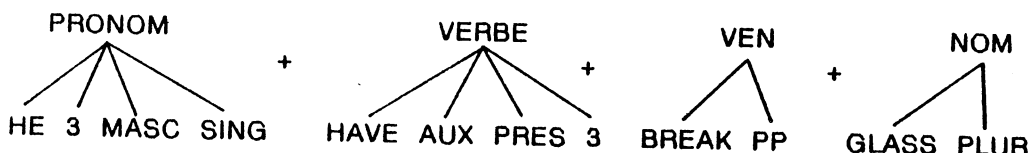


Figure II.10

et, au cours d'une construction, on peut imaginer la structure suivante qui fait apparaître les ambiguïtés:

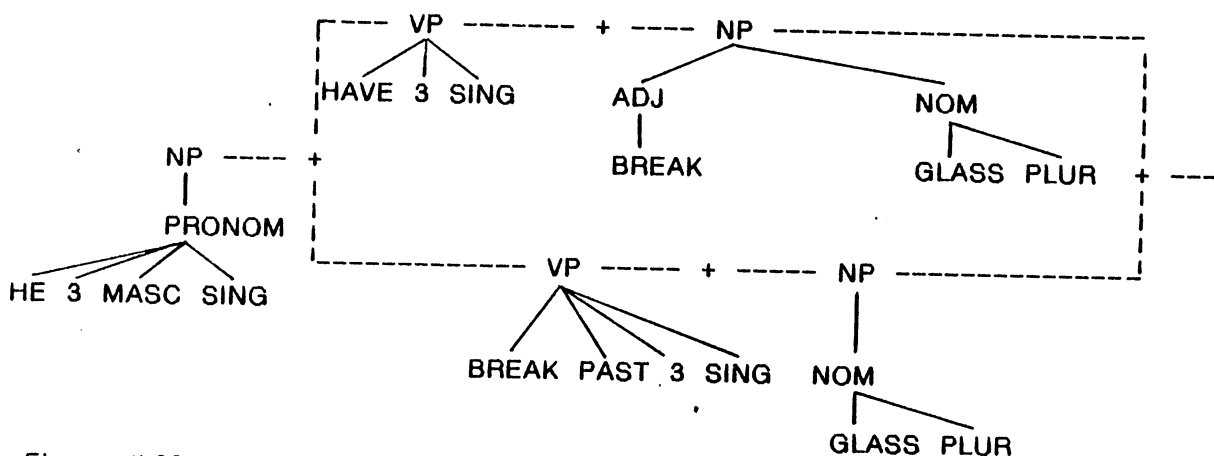


Figure II.11

Comme entrées et sorties des Systèmes-Q sont homogènes, d'une part ils peuvent être utilisés dans les deux sens (par application des règles à l'envers). Il faut, pour cela, qu'il y ait le même nombre de variables à gauche et à droite dans les règles.

D'autre part la modularité d'une grammaire de taille importante peut être obtenue par une séquence de grammaires de tailles plus petites.

Un traitement-Q transforme donc un graphe de chaînes par l'application de plusieurs phases, chacune de ces phases consistant en un système-Q (éventuellement inversé); le premier s'applique sur le graphe de départ, le second sur le graphe résultant de la phase précédente, etc. (La version de 1976 du système anglais-français mise au point à Montréal pour la météo nécessitait 15 systèmes-Q successifs). Un système-Q dispose de la puissance de calcul maximale (équivalence avec les machines de Turing; c'est peut-être même surpuissant, en tout cas, les constructeurs de modèles linguistiques ne sont plus embarrassés par les limites étroites des modèles H-C ou assimilés).

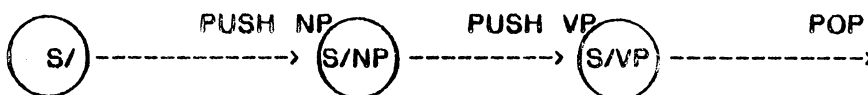
2.3.2. UN MODELE DE CREATION (CHAINE-ARBRE ETIQUETE): LES A.T.N. ('AUGMENTED TRANSITION NETWORKS') DE W.A.WOODS [WOOD.70] [WOOD.71].

Ils ont marqué le couronnement de ces analyseurs. Il s'agit d'un modèle de transduction de chaîne à arborescence. Il produit un arbre étiqueté (ou plusieurs en cas d'ambiguïté) associé à chaque phrase en entrée. L'idée maîtresse des A.T.N. est l'utilisation d'un graphe pour la représentation d'un processus qui reconnaît des langages plus complexes que des langages d'états finis (c'est une généralisation des diagrammes de Floyd). La grammaire est organisée en réseau dont les arcs sont des règles et les noeuds, les états de départ et d'arrivée des règles.

La fonction de l'automate en question, est d'analyser la suite de symboles grammaticaux, représentative de toute phrase à étudier. Partant du plus à gauche, il examine les symboles un à un, et à chaque nouvel examen il change d'état. Un état exprime donc un certain avancement dans la reconnaissance des éléments grammaticaux. Toutes les constructions possibles, c'est-à-dire tous les chemins possibles ont été préalablement définis et communiqués à l'automate sous forme d'un réseau de changement d'états ou "réseau de transition".

Exemple: Fragment d'une grammaire de transition simple

S ----> NP VP
NP ----> DET N / NPR



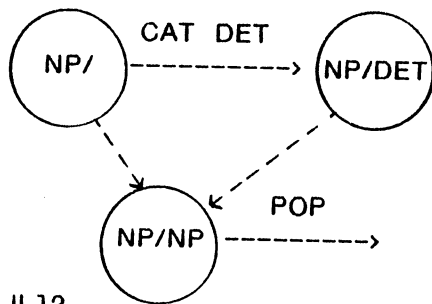


Figure II.12

Les cinq types d'arcs de base sont: CAT,TST,JUMP,POP,PUSH.
CAT représente une transition qui peut être empruntée si le mot courant, dans la chaîne d'entrée, est de la catégorie dite. Le mot courant est alors consommé.
TST agit comme CAT, mais la condition de transition est exprimée sur l'arc.
JUMP agit comme TST, mais ne consomme rien dans la chaîne d'entrée.
POP est un "pseudo" arc, il signifie que l'état dont il est issu est final.
PUSH est l'arc à la base de la récursion pour les grammaires de réseau de transition. La transition peut être faite s'il y a eu consommation d'une sous-chaîne acceptée par l'état indiqué en tant que label de l'arc PUSH.

Jusque là, ce système est formellement équivalent à un automate à pile et donc à une grammaire H-C.

Dans le cas des A.T.N. -réseaux de transition étendus- la seule rencontre d'un symbole grammatical ne suffit pas à l'automate pour lui permettre d'emprunter un arc; il vérifie, en outre, qu'un certain nombre de conditions sont remplies et, détail supplémentaire, si ces conditions sont remplies, l'automate emprunte l'arc, et au cours de ce passage des actions peuvent être exécutées. Ces conditions et ces actions qui donnent aux réseaux de transition de Woods le qualificatif d'étendus (augmented), consistent, sur le plan de la réalisation matérielle, en l'utilisation (tests et affectations) de registres spécialisés et de constructions de structures. Ces conditions et actions, en augmentant les possibilités de mémoire et de "calcul" de l'automate, lui permettent, lors de l'analyse de n'importe quel symbole de la suite étudiée, de fournir des conclusions dépendant du reste (encore inconnu) de la suite. A tout moment, l'automate est guidé par les analyses partielles auxquelles il s'est livré sur les symboles précédents, tout comme il peut poser des hypothèses d'analyse qui seront confirmées ou infirmées par des symboles ultérieurement rencontrés.

Les A.T.N. ont la puissance d'une Machine de Turing si l'utilisation des conditions sur les arcs est sans contrainte, mais cette puissance peut être réduite en plaçant des contraintes appropriées sur les types de conditions et d'actions autorisées.

Une action HOLD permet de retrouver par la suite un constituant qui a été déplacé en le préservant dans la "Hold-liste".

Une fonction BUILDQ permet de construire la structure issue de l'analyse.

Exemple:

Si les registres Sujet.Verbe.Objet contiennent respectivement
(NP(NPR Pierre)).(V aime).(NP(NPR Marie)),
la fonction
(BUILDQ(S+(VP++))Sujet Verbe Objet),
créera la construction
(S(NP(NPR Pierre))(VP(V aime)(NP(NPR Marie))))).

Fragment d'un A.T.N.

```
(S (PUSH NP/ T (SETR SUBJ *)  
      (SETR TYPE (QUOTE DCL) (TO Q1))
```

(Où: S état initial; PUSH NP/ test sur l'arc; T condition mise à vrai; SETR actions; Q1 état final. Explications: on recherche une chaîne acceptée par NP. Si on la trouve, on la range dans le registre Subj et la phrase est de type déclaratif; l'extrémité de l'arc est l'état S/NP).

```
(CAT AUX T (SETR AUX *) (TYPE TYPE (QUOTE Q)) (TO Q2)))
```

```
(Q1 (CAT V T (SETR AUX NIL)  
      (SETR V *) (TO Q4))
```

```
(CAT AUX T (SETR AUX *) (TO Q3)))
```

```
(Q2 (PUSH NP/ T (SETR SUBJ *) (TO Q3)))
```

```
(Q3 (CAT V T (SETR V *) (TO Q4)))
```

```
(Q4 (POP(BUILDQ(S+++ (VP+))TYPE SUBJ AUX VT)  
      (PUSH PP/ (SETR VP(BUILDQ(VP(V+)*V)) (TO Q5))
```

```
(Q5 (POP(BUILDQ(S++++)TYPE SUBJ AUX VP+)  
      (PUSH NP/ T (SETR VP(APPEND(GETR VP) (LIST *))) (TO Q5)))
```

On peut citer le langage de programmation REZO [STEW.74] [STEW.75], qui adapte les A.T.N. à la traduction automatique. Ce langage a été utilisé par TAUM-AVIATION pour l'analyse syntaxique de l'anglais.

2.3.3. UN MODELE DE SUBSTITUTION (ARBRE DECORE - ARBRE DECORE): ROBRA.

Robra [CHAUC.74] · [CHAU.75] et [BO&GU&QU.80] est un langage d'écriture de systèmes transformationnels agissant sur des arborescences étiquetées. En Ariane-78, Robra est utilisé pour écrire les trois phases d'analyse structurale, de transfert structural et de génération syntaxique.

Un système transformationnel (ST) opère sur une arborescence objet (AO), étiquetée sur un ensemble dénombrable Σ . En Robra, Σ est l'ensemble des masques de variables possibles. Un ST est défini par un **graphe de contrôle** (GC), un ensemble de **grammaires transformationnelles** (GT) et un ensemble de **règles de production** (RP). Une GT est un ensemble ordonné de RP et un GC est un graphe à noeuds étiquetés sur $GT \cup \{\&Nul\}$ dont les arcs peuvent porter des conditions sur T_{Σ} , ensemble des AO possibles.

Une règle (RP) se compose d'un nom de règle, d'une partie gauche (le schéma) et d'une partie droite (l'image). On peut la "comprendre" comme une "règle de production", le schéma étant la condition et l'image, l'action. Le schéma décrit la famille de sous-arbres que la règle peut transformer. Il se compose d'un modèle géométrique et de conditions sur les variables portées par les sommets. L'image est constituée par le "sous-arbre image", la "fonction de transfert" qui indique où rattacher les dépendants des sommets qui n'apparaissent pas dans le schéma et la partie "affectation" qui calcule les valeurs des attributs portées par les sommets images.

Exemple de règle transformationnelle (pris dans l'analyse structurale de l'anglais).

** MODAL+BE+VEN: simple modal and passive.

** SHOULD BE TRANSFORMED, CAN BE DONE, MIGHT BE SEEN.

```

MDBEVEN: (*0.&NIV=1) (0(1($A.2,*),*.3($B.4),*.5($V)) /
    1: $MODAL; 4: UL-E-"BE"; 5: SUBV-INC-VEN; $A: ADV; $B: $ADV;
== 0(5($A.2,$B,$V)) / * <-- 1,3,4 /
    2: 2, SF:=AUX;
    5: 5, K:=VCL, VOICE:=PAS, UL:="*VCL", NEG:=NEG(1),
        SUBV:=VB, NUM:=NUM(1),
        -SI- UL(2) -E- "COULD" -OU- UL(2) -E- "MIGHT"
        -ALORS- TENSE(5):=COND -U- PRES ;
        -SINON- TENSE(5) := TENSE(2)
        -FSI-.

```

"(*.&NIV)" signifie que le niveau 0 (racine) de la sous-arborescence traitée n'entre pas dans la transformation (*0); il est donc contextuel; d'autre par il est pris au niveau 1 (&NIV=1) dans l'arborescence objet.

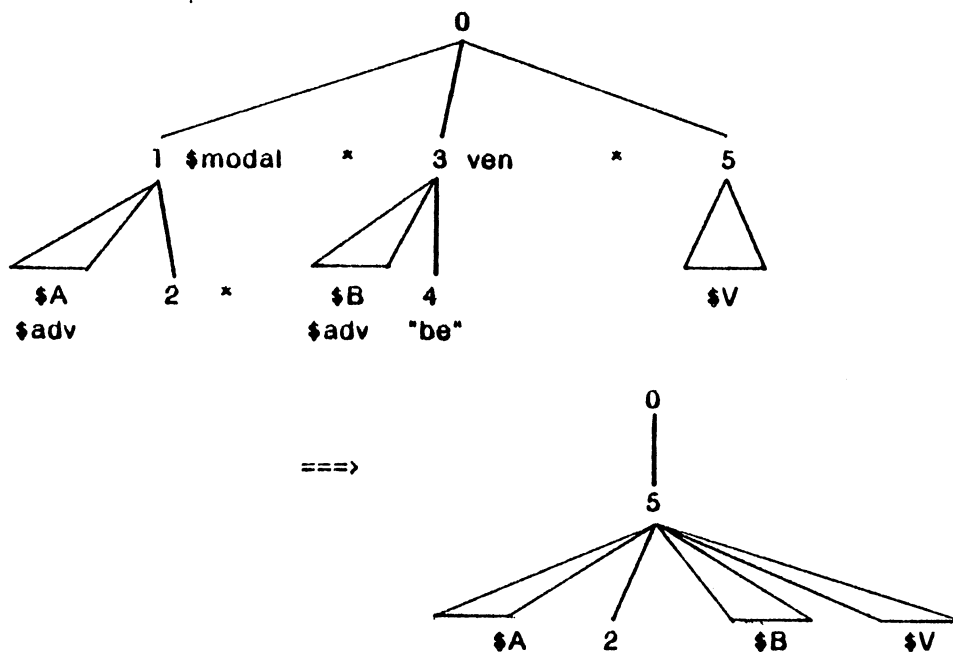
Dans le schéma d'arborescence, "*" signifie que l'on n'accepte rien entre les sommets cités (ici, rien entre 1 et 3 et rien entre 3 et 5); "\$" indique

une référence à une procédure de condition (\$MODAL fait ici référence à un processus qui vérifie que TYPE -E- MODAL, de même \$ADV vérifie que CAT -E- A -ET- SUBA -E- ADV).

"* <-- 1,3,4" Indique que les sommets 1,3,4, disparaissent dans la nouvelle construction, ainsi que leurs dépendants.

Enfin, dans la partie affectation, la notation 5: 5..... permet la recopie éventuelle des attributs portés par le sommet (ici sommet 5), suivie de nouvelles affectations.

Graphiquement, on peut représenter cette règle par:



où 5 et 2 portent leurs attributs initiaux, modifiés ou complétés par les affectations.

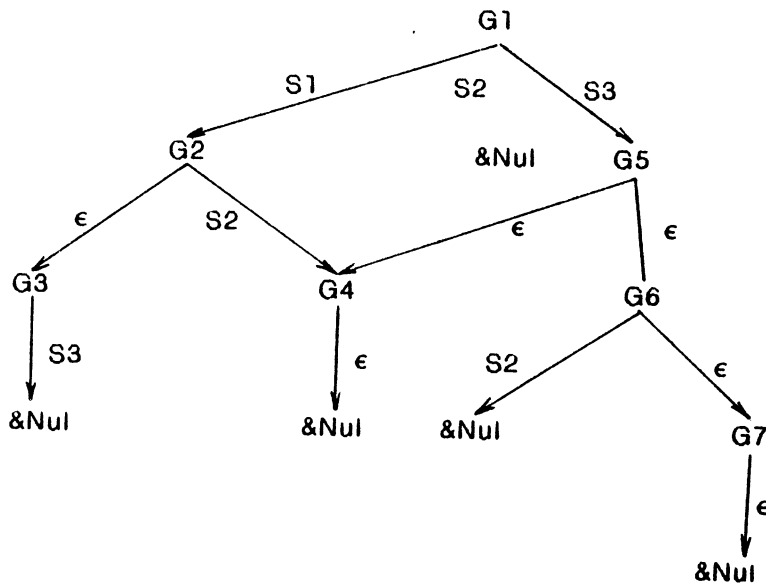
Figure II.13

Une grammaire transformationnelle (GT) est un ensemble ordonné de RP, cet ordre est propre à chaque grammaire et sert à résoudre des conflits entre les règles.

Le graphe de contrôle (GC) est un multigraphe orienté et ordonné, où chaque noeud porte le nom d'une GT avec mode d'application, ou l'ordre de sortie &Nul. Les arcs du graphe peuvent porter des conditions qui doivent être réalisées pour que l'arc puisse être traversé.

Exécution d'un système transformationnel.

L'exécution d'un système transformationnel (ST) se fait de la façon suivante: un sommet initial est choisi dans le graphe; l'automate, utilisant un algorithme de retour-arrière incorporé, parcourt le graphe de contrôle selon le premier chemin possible conduisant à un symbole de sortie, tout en appliquant chaque GT traversée à l'AO. Après l'application d'une GT, les arcs sortants, dont les conditions ne sont pas vérifiées par l'AO obtenu, ne peuvent être traversés.



Si le schéma S1 apparaît dans $G1(A0)$, on calculera $G3(G2(G1(A0)))=A3$. Si S3 apparaît dans A3, le système s'arrête et produit A3. Sinon il revient en arrière, à A2. Si, alors, S2 apparaît dans A2, on produit $G4(A2)$. Sinon on revient à $A1=G1(A0)$, etc.

Figure II.14 Graphe de contrôle.

Si une règle appliquée comporte un appel récursif d'un sous-système transformationnel, l'arborescence soumise à la récursion est traitée par le même GC en prenant le noeud correspondant à la grammaire comme noeud initial. (Remarquons que s'il n'existe aucun chemin possible, l'arbre de sortie sera identique à l'arbre d'entrée).

Certains modes d'exécution d'une GT indiquent son comportement lors du parcours du graphe de contrôle, certains contrôlent l'itération, d'autres déterminent le mode de choix entre plusieurs occurrences possibles.

- Si une grammaire est en mode "I" (impératif), et qu'aucune transformation ne s'est effectuée, le chemin est bloqué et l'automate doit revenir en arrière.
- Si une grammaire est en mode "Z" (zéro), l'automate ne peut revenir (en arrière) sur ce qui précède: le travail effectué précédemment sur l'AO ne peut donc être détruit.

Application d'une grammaire transformationnelle.

L'application d'une grammaire transformationnelle (GT) consiste en une (mode "unitaire") ou plusieurs (mode "exhaustif") application(s) élémentaire(s).

Application élémentaire simple.

Une application élémentaire simple consiste à exécuter en parallèle, une fois, un certain nombre d'occurrences des règles de la GT. Ceci est tout-à-fait analogue à un pas de dérivation parallèle dans une grammaire syntagmatique; le problème est cependant le passage à deux dimensions.

En effet il faut préciser la forme générale de la famille d'occurrences souhaitée (mode "coupe", mode "total") et les règles de résolution de conflit. En effet, il est clair que, pour appliquer en même temps deux occurrences de règles, il faut que les occurrences de leurs schémas ne partagent pas de points actifs. On dira alors que ces occurrences sont indépendantes.

Un application élémentaire simple se fait en trois étapes.

(1) Recherche des occurrences des règles de la GT dans l'arborescence objet (AO).

(2) Sélection de la sous-famille applicable, formée d'occurrences indépendantes, répondant aux critères global (total ou coupe). En mode total (et non en mode coupe), une occurrence peut en dominer strictement une autre. Un parcours canonique associé à la GT est utilisé pour cette sélection (mode "haut" ou préordre ou mode "bas" ou postordre).

(3) Application parallèle et marquage des racines de transformation (RT) images par une ou plusieurs règles. En effet, chaque point de l'AO est muni d'un ensemble, initialement vide, de "marques" de règles. Après l'exécution de l'occurrence Ok de la règle Ri, on ajoute aux marques de l'occurrence de la racine de transformation Ok(RTi) la marque Ri si la GT est en mode "ponctuel" ou les marques Rj telles que $j \leq i$ dans la GT si la GT est en mode "dessous". Les marques seront effacées quand on passe d'un noeud à l'autre du graphe de contrôle. Lors d'un appel récursif, on efface sur la racine de l'arbre soumis à la récursion les marques des règles intervenant dans cette récursion; au retour, on fera sur cette racine l'union des anciennes et des nouvelles marques.

Application élémentaire avec itération.

Si la GT est en mode exhaustif, on répète les applications élémentaires jusqu'à ce qu'il ne reste plus aucune règle applicable.

En mode "gardé" (par opposition au mode libre), si Rk est marqué sur un noeud, Rk ne peut être utilisé sur ce noeud, ni sous ce noeud, de sorte que le nombre d'applications potentielles diminue en général strictement et que le processus s'arrête.

Application élémentaire avec récursion.

Dans une grammaire transformationnelle G, une règle R_k peut appeler récursivement la sous-grammaire de G constituée par les règles R_{k1}, \dots, R_{ki} de G munies des mêmes modes d'exécution, ou la totalité de la grammaire G, ou le sous-système-transformationnel dont le noeud initial est G.

Dans G, chaque règle peut avoir un appel récursif et un seul. Pour exécuter un appel récursif, le système "détache" le sous-arbre-objet et le traite comme un nouvel arbre objet. Les sommets images non-validés ne peuvent participer activement à aucune transformation durant l'appel, ce qui réduit le nombre des occurrences potentielles. Le résultat de l'appel remplace le sous arbre d'entrée dans l'arbre origine.

Les appels récursifs, spécialement du second type, permettent d'utiliser différentes stratégies sur diverses parties d'une unité de traduction.

Robra peut être considéré comme un langage algorithmique de très haut niveau, du fait de ses types de données et de ses structures de contrôle. Les types de données sont les variables (ou attributs), les masques de variables (ou décoration) et les arborescences étiquetées avec des opérateurs associés (comparaison et affectation pour les deux premiers, transformation à l'aide d'une ou plusieurs règles pour le troisième). Les structures de contrôle comprennent des affectations conditionnelles (de variables), le parallélisme (dans l'application des règles d'une grammaire), l'itération (de grammaires), le non-déterminisme (dans le GC), enfin les règles peuvent provoquer des appels récursifs de sous-grammaires et de sous-systèmes transformationnels.

2.4. APPORTS RELATIFS TRANSDUCTEURS-ANALYSEURS.

Le choix d'un transducteur, plutôt que d'un analyseur, s'appuie sur un certain nombre d'arguments. Le premier est que la structure produite par un analyseur est un sous-produit puisque le rôle de cette analyseur est d'accepter ou de refuser son entrée; en cas d'échec rien n'est produit, or malheureusement les grammaires ne sont jamais parfaites et les textes d'entrée sont souvent incorrects par rapport aux grammaires ou même à l'usage courant. Un transducteur peut traiter des entrées imparfaites et donner un résultat dans tous les cas. Dans le cas d'un analyseur, la grammaire impose la structure obtenue alors que par transduction, on définit les structures d'entre et de sortie. Suivent, deux autres arguments en faveur des transducteurs: possibilité de modularité et aspect heuristique. Nous allons rappeler l'intérêt de nos trois types de transducteurs à ce propos.

Les Systèmes-Q, du fait de l'homogénéité des structures d'entrée et de sortie, permettent la modularité par enchaînement d'un certain nombre de systèmes-Q, chacun prenant comme structure d'entrée la structure de sortie du précédent. L'utilisation de cette modularité, pour décomposer la grammaire en phases indépendantes, est un aspect organisationnel intéressant.

Les A.T.N. ont accès à la modularité par la décomposition du réseau en sous-réseaux. L'aspect heuristique réside dans l'utilisation de la Hold liste qui permet de conserver des informations, des hypothèses auxquelles on peut à tout moment référer.

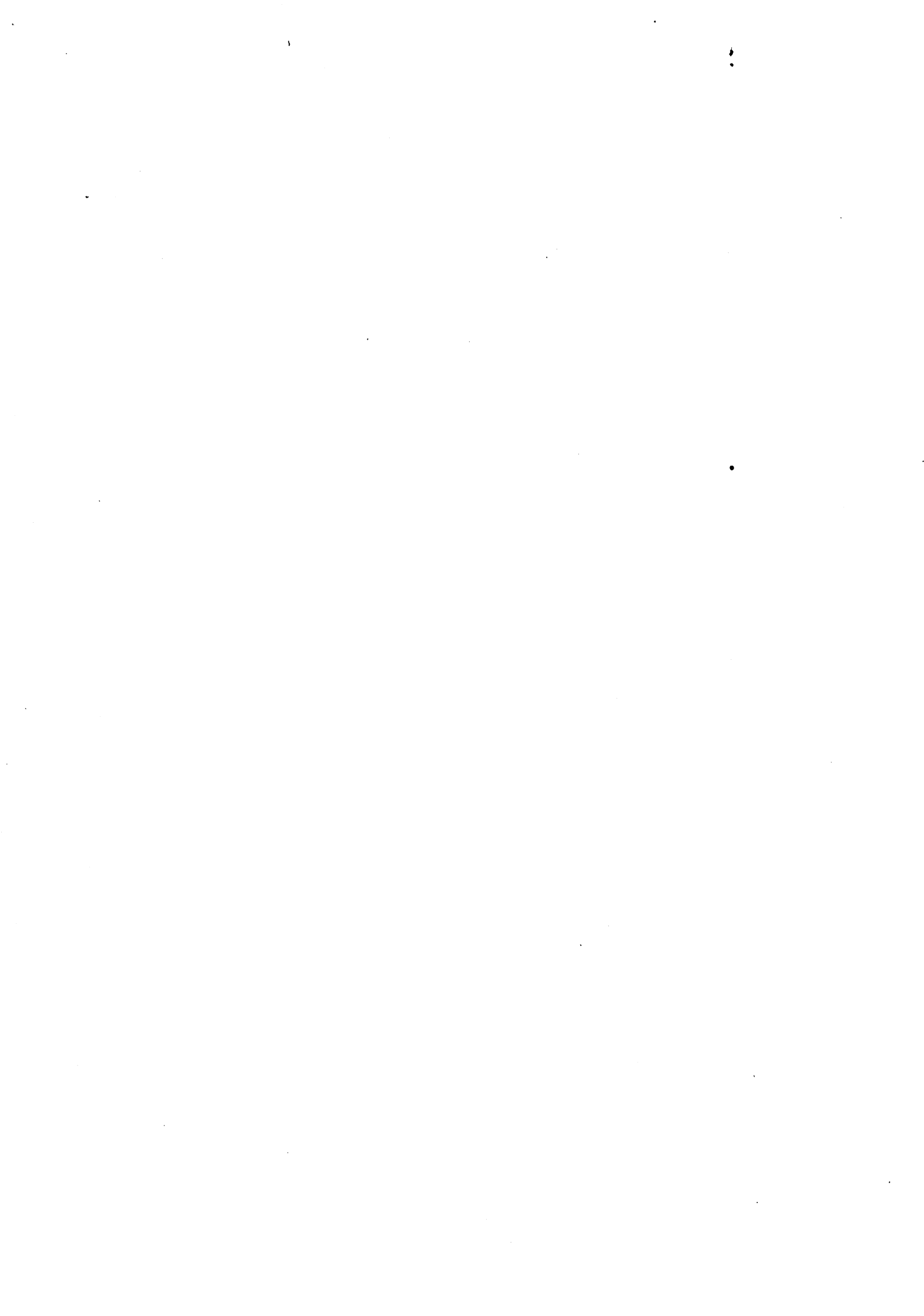
Enfin, dans Robra, la structure de sortie, homogène à la structure d'entrée, permet d'utiliser la composition des grammaires transformationnelles pour donner de la modularité. L'aspect heuristique se révèle au niveau du travail du linguiste qui dispose du choix des grammaires élémentaires, de leur organisation en hiérarchie en une grammaire de contrôle, du choix du statut de chaque règle dans chaque grammaire et du mode d'application des grammaires elles-mêmes.



CHAPITRE III

CHOIX ET REPRESENTATION D'UN MODELE

"TEXTE \longleftrightarrow INTERPRETATION".



CHAPITRE III

CHOIX ET REPRESENTATION D'UN MODELE

"TEXTE \longleftrightarrow INTERPRETATION"

PREMIERE PARTIE

CHOIX DES NIVEAUX D'INTERPRETATION

Dans cette partie, nous nous appuierons essentiellement sur des publications du G.E.T.A., traitant du choix et de la représentation de la structure intermédiaire.

[VAUQ.78] [GETA.80] [BO&DF&GU&NF&VA.82].



1. CHOIX DES NIVEAUX D'INTERPRETATION.

1.1. BUTS A ATTEINDRE EN TRADUCTION MULTILINGUE.

Rappelons que la traduction s'effectue en trois phases:

-Analyse du texte en langue source, pour extraire une structure logico-sémantique libérée le plus possible des contraintes morphologiques et syntaxiques de la langue source. Cette phase est monolingue.

-Transfert des unités lexicales du texte d'entrée vers les unités lexicales de la langue cible, réalisation des choix convenables en résolvant le plus possible d'ambiguïtés, modification, le cas échéant, de la structure issue de l'analyse pour la rendre cohérente avec les données linguistiques de la langue cible. Cette phase, essentiellement bilingue, doit être réduite au minimum dans une économie de traduction multilingue.

-Génération du texte de sortie à partir de la structure logico-sémantique munie des unités lexicales de la langue cible. Cette phase est strictement monolingue.

L'organisation des modèles linguistiques doit donc, en premier lieu, se fonder sur cette articulation en trois phases représentée ci-dessous.

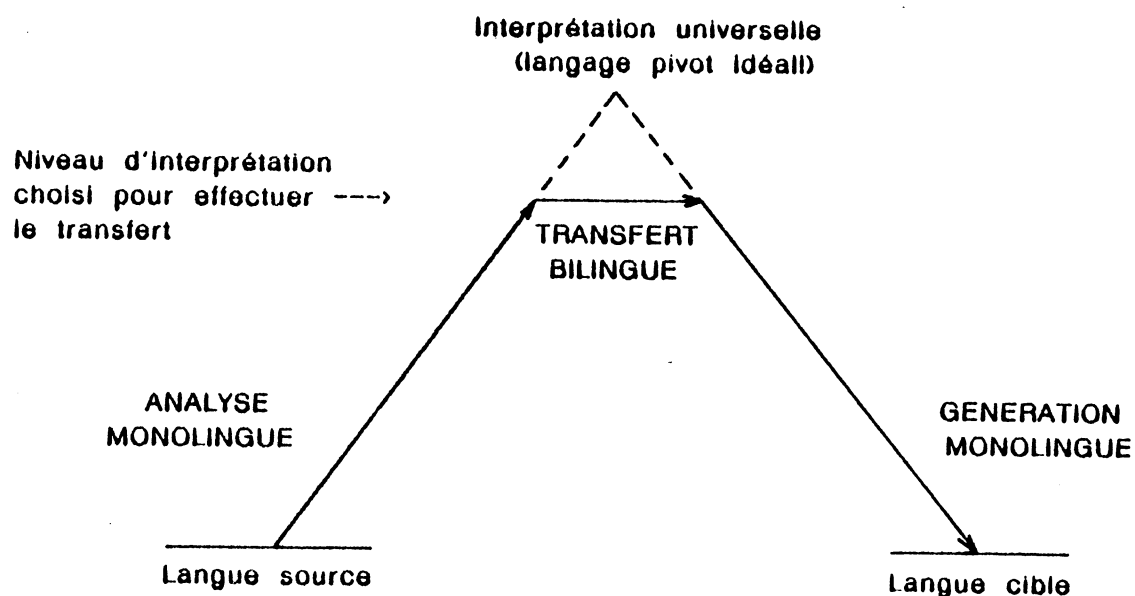


Figure III.1

Il y a donc pour un système de traduction automatique de la langue A vers la langue B:

- Un modèle d'analyse de A.
- Un modèle de génération de B.
- Un modèle de transfert de A vers B.

En traduction multilingue, le même modèle d'analyse convient pour les modèles de transfert vers n'importe quelle autre langue. De même, un modèle de génération ne dépend pas de la langue source. Seul le modèle de transfert est propre à un couple de langues donné.

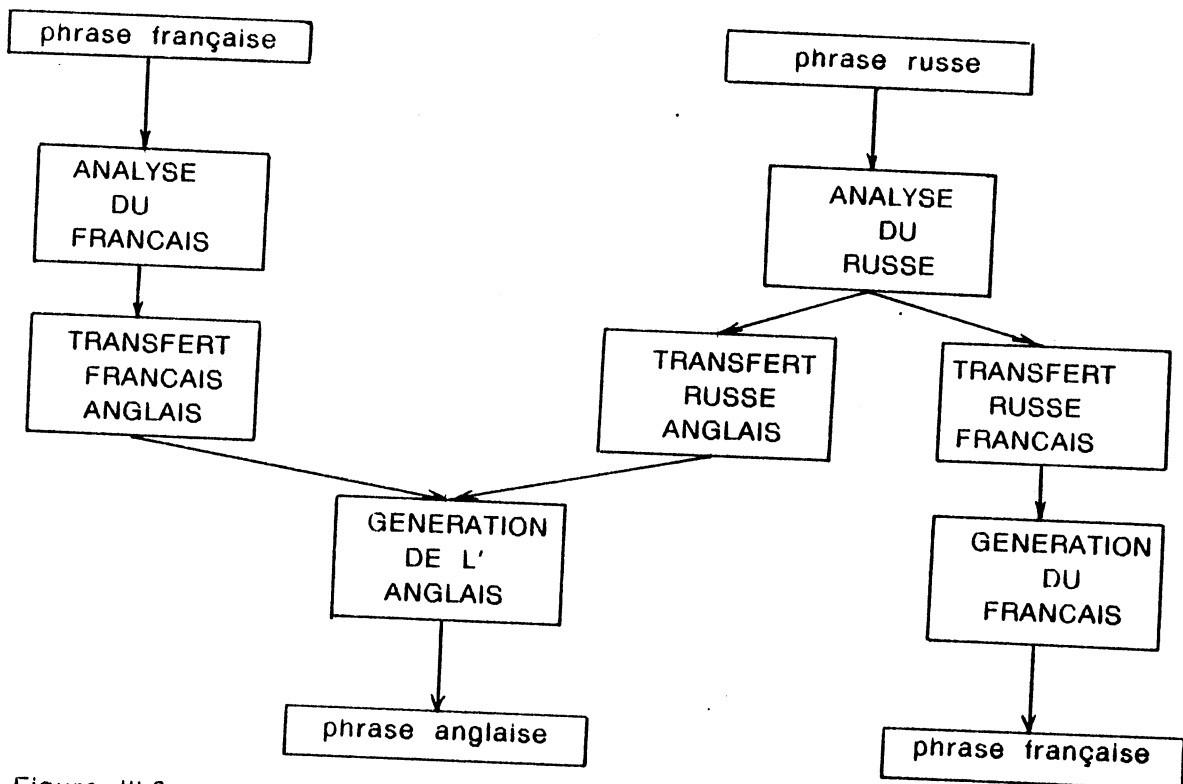


Figure III.2

La structure, résultat de l'analyse du texte en langue source, doit donc permettre un transfert vers les différentes langues cibles avec un minimum d'opérations à mettre en oeuvre, car le nombre de dictionnaires de transfert augmente comme le carré du nombre de langues envisagées.

Toute structure intermédiaire (structure résultat de l'analyse) représente un certain niveau d'interprétation du texte donné en langue source. Si cette interprétation est très éloignée des contingences grammaticales, alors la structure associée à un tel niveau aura un caractère proche de l'universalité.

1.2. NOTION DE NIVEAU D'INTERPRETATION.

Le problème est le choix du niveau d'interprétation auquel il faudra arrêter l'analyse pour aborder le transfert.

Nous allons illustrer ce problème par quelques exemples. Soient les phrases françaises suivantes:

- (1) beaucoup d'équations sont résolues par itération.
- (2) on résoud beaucoup d'équations par itération.
- (3) beaucoup d'équations se résolvent par itération.
- (4) il se résoud beaucoup d'équations par itération.

On estime que ces quatre phrases sont synonymes. Pourtant, chacune d'elles possède une structure syntaxique qui lui est propre.

Cet exemple montre que la traduction automatique ne peut se satisfaire des seuls niveaux -déjà abordés- des classes syntagmatiques et des fonctions syntaxiques. En effet, si l'on admet qu'en anglais la seule expression possible est la phrase:

(1') many equations are solved by iteration.

Seule la traduction (1)→(1') est pratiquement mot à mot, les structures syntaxiques des autres ne correspondent pas à la structure anglaise.

Les structures suivantes (figures III.3) représentent les phrases (1), (2), (3) et (4) au niveau des classes syntagmatiques et catégories syntaxiques, d'une part, et au niveau des fonctions syntaxiques d'autre part. Ce dernier est exprimé entre parenthèses.

(1) beaucoup d'équations sont résolues par itération

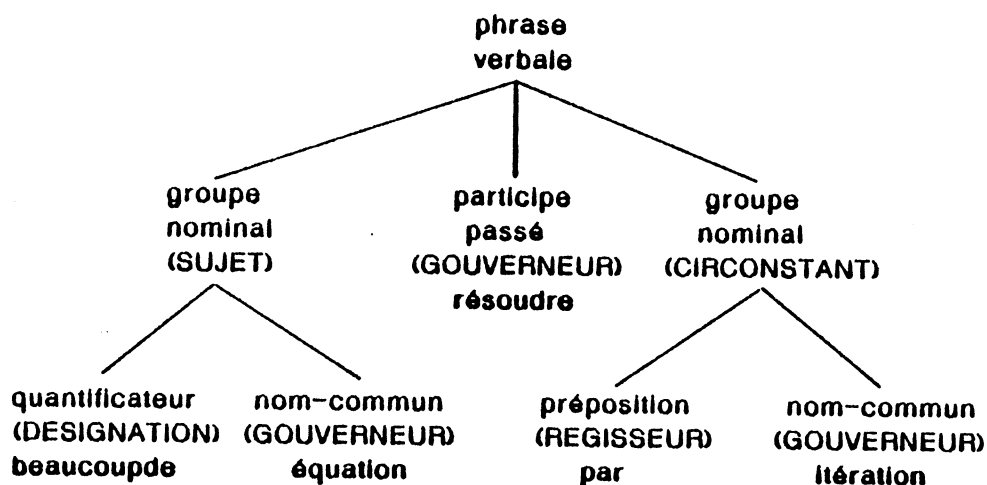


Figure III.3a.

(2) on résoud beaucoup d'équations par itération

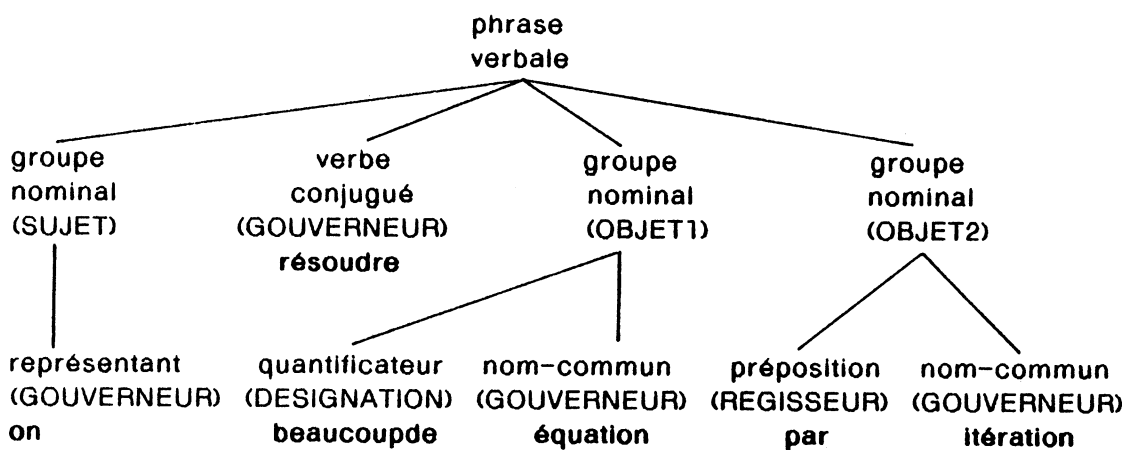


Figure III.3b.

(3) beaucoup d'équations se résolvent par itération

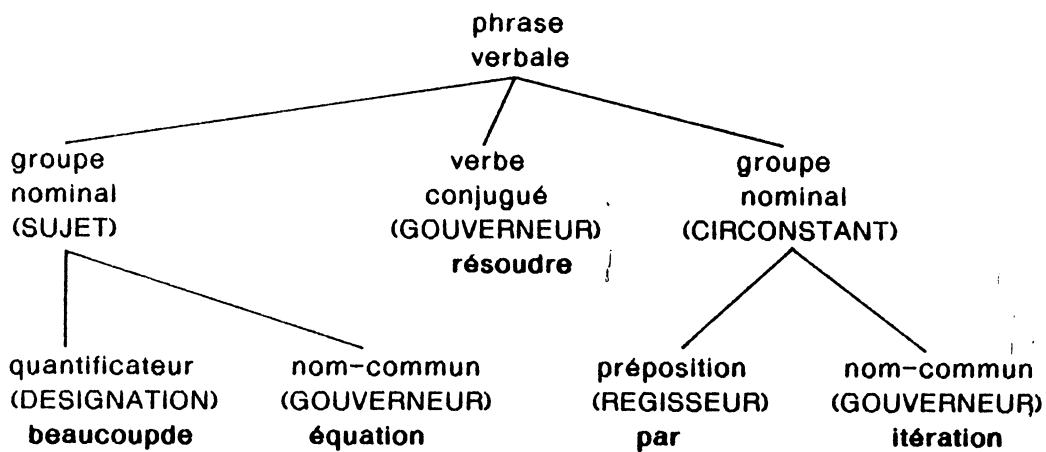


Figure III.3c.

(4) Il se résoud beaucoup d'équations par itération

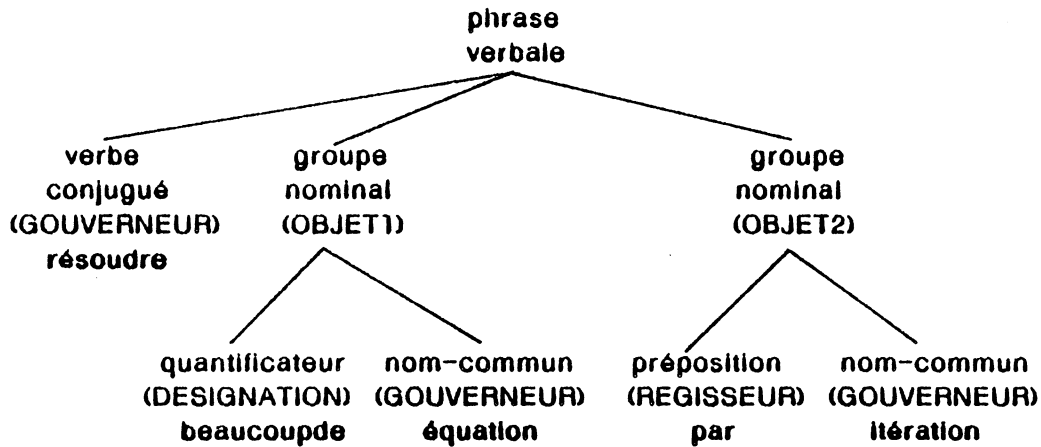


Figure III.3d.

Si l'on veut effectuer la traduction dans tous les cas, il est nécessaire d'analyser les phrases (1) à (4) jusqu'à un niveau d'interprétation qui fournit pour toutes, une structure commune, ou éventuellement des structures différentes qui seront jugées équivalentes.

Pour cela, on définit le niveau des relations logiques et sémantiques. Les relations logiques expriment des relations entre un prédicat et ses arguments qui peuvent être au nombre de trois. Le prédicat est un verbe ou dérivé de verbe et les places d'argument ARG0, ARG1, ARG2 correspondent respectivement aux fonctions syntaxiques de sujet, 1^e objet, 2^e objet dans la phrase verbale active correspondante.

Exemple: Le prédicat "extraire" a trois places d'argument.
"X extrait Y de Z"

Des hommes extraient le charbon de la mine.
arg0 / prédicat / arg1 / arg2

Le charbon est extrait de la mine par des hommes.
arg1 / prédicat / arg2 / arg0

L'extraction du charbon de la mine par des hommes.
gouverneur / arg1 / arg2 / arg0

(extraction est un nom dérivé du prédicat "extraire".
Il conserve ses arguments.)

Les relations sémantiques expriment des relations sémantiques simples d'origine, de destination, de résultat, etc., ainsi que divers types de liens (qualifications particularisatrices).

Regardons maintenant nos phrases (1) à (4) aux différents niveaux, nous noterons les fonctions syntaxiques FS, et les relations logiques et sémantiques respectivement RL et RS

	(1) <u>beaucoup d'équations</u> <u>sont résolues</u> <u>par itération.</u>			
FS		prédicat		
RL	sujet			circonstant
RS	arg1			Instrument
	(2) <u>on résoud</u> <u>beaucoup d'équations</u> <u>par itération.</u>			
FS		prédicat		
RL	sujet	objet1		circonstant
RS	arg0	arg1		Instrument
	(3) <u>beaucoup d'équations</u> <u>se résolvent</u> <u>par itération.</u>			
FS		prédicat		
RL	sujet			circonstant
RS	arg1			Instrument
	(4) <u>il se résoud</u> <u>beaucoup d'équations</u> <u>par itération.</u>			
FS		prédicat		
RL		objet1		circonstant
RS		arg1		Instrument

Les relations logiques et sémantiques sont conservées dans chacune de ces paraphrases, on les retrouvera dans leur traduction commune en anglais (1').

La structure profonde commune (ou jugée équivalente) à ces quatre phrases est la suivante:

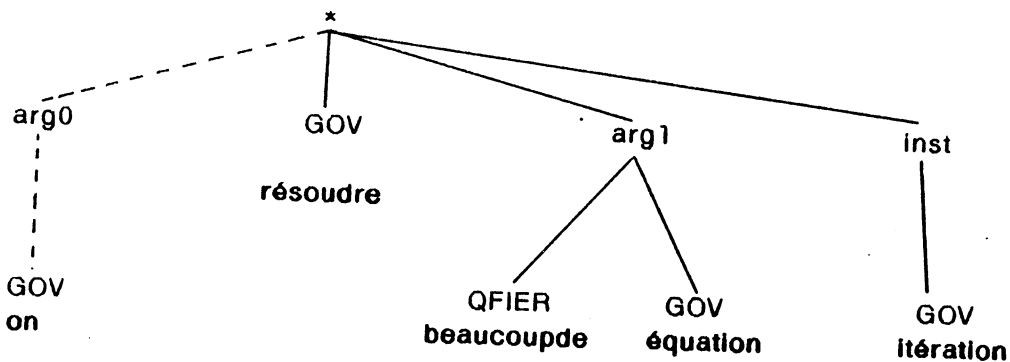


Figure III.4.

Les expériences réalisées dans le passé montrent que les meilleures traductions (surtout dans les cas difficiles et entre langues de familles très différentes) sont obtenues grâce à des analyses très profondes et qui conduisent donc à des niveaux d'interprétation très ambitieux. Les expériences montrent aussi que dans les cas les plus simples, entre langues voisines, des niveaux d'interprétation peu ambitieux suffisent à donner des résultats convenables. Il est également montré que les analyses très profondes ne peuvent être obtenues que dans des domaines extrêmement limités et que leur coût devient prohibitif en cas d'extension à un domaine moins microscopique.

En conséquence, un point de vue "réaliste" sur le choix d'un niveau d'interprétation doit tenir compte de tous ces facteurs. Etant données les contraintes contradictoires qu'ils apportent, il en résulte qu'il n'existe pas de niveau d'interprétation idéal, pour une traduction multilingue, convenable pour toutes les situations.

1.3. SOLUTION APPORTEE PAR LE GETA.

1.3.1. UN MODELE MULTINIVEAU.

Dès 1974, les travaux du GETA se sont orientés vers une méthodologie d'analyse qui rassemble sur une même structure le niveau optimal des relations logiques et sémantiques et deux autres niveaux moins ambitieux, celui des relations syntaxiques (propre au texte en langue source) et celui, plus superficiel encore des groupes syntagmatiques.

Une telle structure est alors génératrice d'une structure pour chacun des niveaux choisis. Si l'on souhaite expliciter l'une d'elle, on l'obtient à partir de la structure à multiples niveaux d'interprétation par l'utilisation de règles fixes établies, une fois pour toutes, pour effectuer ce type de transformations. On a alors le schéma suivant:

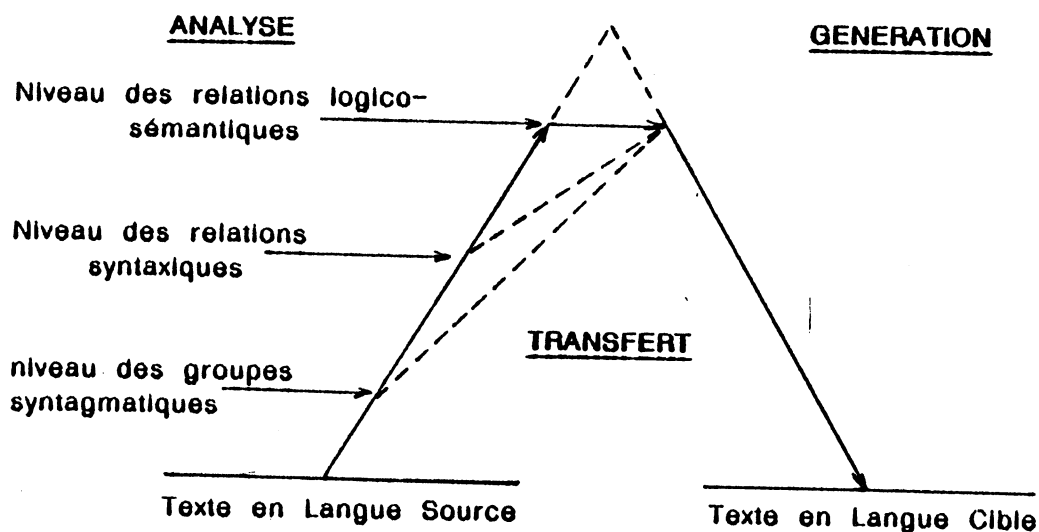


Figure III.5

Le fait de choisir une structure linguistique intermédiaire à plusieurs niveaux d'interprétation comporte à la fois des contraintes et des avantages.

Du côté des contraintes, on sera conduit à adopter la géométrie la plus simple possible et comportant un contenu d'information acceptable pour tous les niveaux.

Du côté des avantages, il faut compter:

-la possibilité de procéder à l'analyse en progressant dans tous les niveaux à la fois et d'utiliser les résultats acquis dans l'un quelconque d'entre eux pour résoudre les difficultés rencontrées dans les autres. On est ainsi libéré de l'analyse par pas successifs où l'on commence par la syntaxe de surface, pour continuer par la syntaxe profonde, pour finir par une analyse sémantique;

-la possibilité, en cas d'échec pour une phrase au niveau le plus ambitieux, d'obtenir une traduction à partir des niveaux d'analyse moindres.

1.3.2. CHOIX DE STRUCTURE POUR CE MODELE.

Pour parvenir au but énoncé, une structure à multiples niveaux d'interprétation en forme d'arborescence a été choisie. Il faut entendre par là une seule arborescence porteuse des informations nécessaires à l'interprétation de chacun des niveaux choisis. Les noeuds de cette structure seront décorés. Eventuellement, ces décorations représenteront des relations binaires entre sommets.

1.3.3. CHOIX DES NIVEAUX D'INTERPRETATION SUR CETTE STRUCTURE.

Actuellement, trois niveaux d'interprétation ont été définis. En indiquant ces niveaux par ordre de "profondeur" croissante, nous avons:

le niveau des classes morpo-syntaxiques.

C'est le plus proche de la langue. La structure associée est l'arborescence de ces classes; elle est déductible de la structure multiniveau. Un sommet terminal (feuille de l'arborescence) est porteur d'une classe terminale (catégorie syntaxique), un sommet non-terminal est porteur d'une classe non-terminale (groupe syntagmatique);

le niveau des fonctions syntaxiques.

Une fonction syntaxique est une relation entre deux sommets. Ce niveau peut être considéré comme représentatif de la classique arborescence de "dépendances syntaxiques" (mais les sommets non-terminaux sont conservés, ainsi on ne perd pas le parenthésage en termes de classes syntagmatiques);

le niveau des relations logiques et sémantiques.

Ce niveau décrit lui aussi des relations binaires entre sommets. Les relations sémantiques expriment des relations entre un prédicat et un terme qui n'occupe pas une place d'argument (circonstantiel en syntaxe, sujet en syntaxe, complément) ou entre deux termes à titre de particularisation (qualificatif en syntaxe, désignation...). Les relations logiques s'expriment entre un prédicat et l'un de ses arguments.

La décoration donne accès aux informations relatives aux niveaux d'interprétation. Une information de ce type est portée par un sommet. Si cette information est une relation binaire, elle sera portée par un sommet choisi de façon arbitraire par le formalisme et exprimera de façon implicite la relation (c'est-à-dire que, par son seul nom, elle nous dira où trouver les sommets source et cible de la relation). Cela permet de conserver la structure arborescente souhaitée.

1.3.4. REPRESENTATION DE CETTE STRUCTURE.

1.3.4.1. AU NIVEAU DE LA GEOMETRIE.

La géométrie de la structure doit avoir la signification la plus faible possible; nous avons vu que nous nous bornerons à représenter par la géométrie de l'arborescence le parenthésage de la phrase au moyen du plus petit nombre de groupes syntagmatiques.

Une "arborescence élémentaire" est une arborescence composée d'une racine et de sommets (au moins un) qui en sont des dépendants directs. Considérons le quadruplet de variables $[K \text{ ou } CAT, FS, RL, RS]$ où K est une valeur de classe non-terminale (classe morpho-syntaxique), CAT une classe terminale (catégorie syntaxiques), FS une valeur de fonction syntaxique et RL et RS des valeurs de relations logiques et sémantiques; la racine de l'arborescence élémentaire a nécessairement une valeur de classe morpho-syntaxique (classe non-terminale K); un seul sommet dépendant peut avoir la fonction syntaxique de gouverneur, ses relations logiques et sémantiques sont alors vides. Tous les autres sommets sont optionnels.

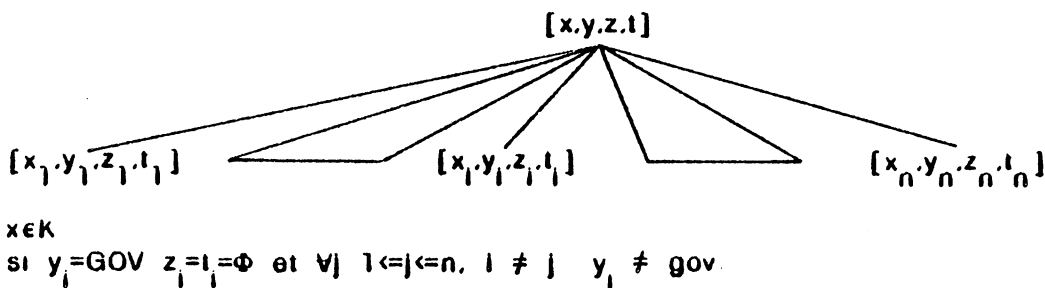


Figure III.6

L'"arborescence élémentaire initiale" est caractérisée par le fait que sa racine porte $[x.\Phi,\Phi,\Phi]$, où x est une valeur de K .

La construction multiniveau correspondant à une phrase donnée est une composition d'arborescences où chaque sommet $[x.y.z.t]$, où x est une valeur de K , est la racine d'une arborescence élémentaire initiale.

1.3.4.2. AU NIVEAU DE LA DECORATION.

Nous avons vu que chaque sommet de la structure était porteur d'une décoration et que parmi les étiquettes affectées à chaque sommet, certaines sont liées aux différents niveaux d'interprétation; elles sont alors soit dépendantes directes de la structure, c'est le cas des classes syntagma-tiques et des fonctions syntaxiques, soit imposées a priori et sont communes à toutes les grammaires successives, c'est le cas des relations logiques et sémantiques. Les relations logiques et sémantiques, de même que les fonctions syntaxiques, expriment des relations. Chacune de ces relations est signifiée par une étiquette particulière (choisie de façon arbitraire, mais imposée), portée par un sommet, qui désigne implicitement la cible, la source et le type de la relation.

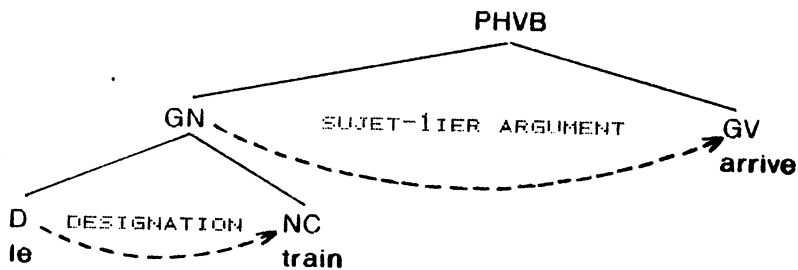
Exemples de types de relations.

A. Relation entre frères.

A1. Elle exprime une relation entre un élément et son gouverneur.

C'est le cas des fonctions syntaxiques de Sujet, Désignation, de la relation logique de premier argument entre un sommet (il sera alors porteur de la relation) et son gouverneur.

Exemple:



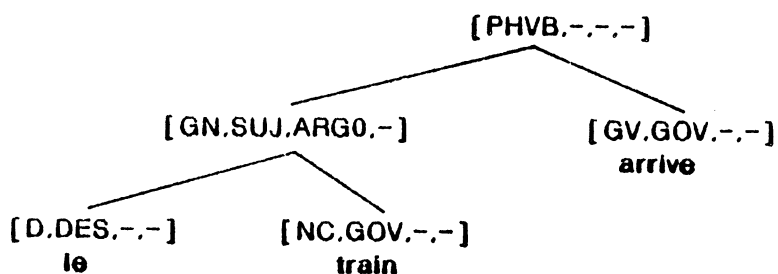


Figure III.7

A2. Elle exprime une relation entre deux éléments (aucun n'est gouverneur).

De la même façon, portée par le sommet source de la relation, elle exprime de façon implicite le sommet qui en est la cible.

Exemple:

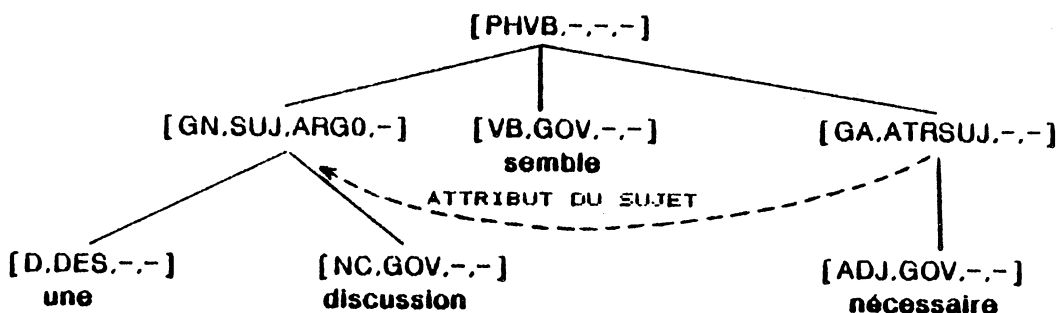


Figure III.8

B. relation entre un élément et son père.

B1. Le fils est source de la relation.

C'est le cas de la coordination d'un groupe syntagmatique à un autre de même type qui reçoit par identification les relations logiques et sémantiques du groupe qui l'absorbe.

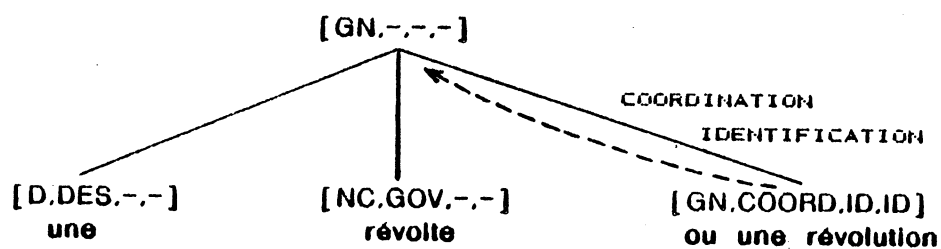


Figure III.9

B2. Le père est source de la relation.

Dans ce cas on définit une relation inverse. C'est le cas de Objet quantifié (qui est la relation inverse de quantification).

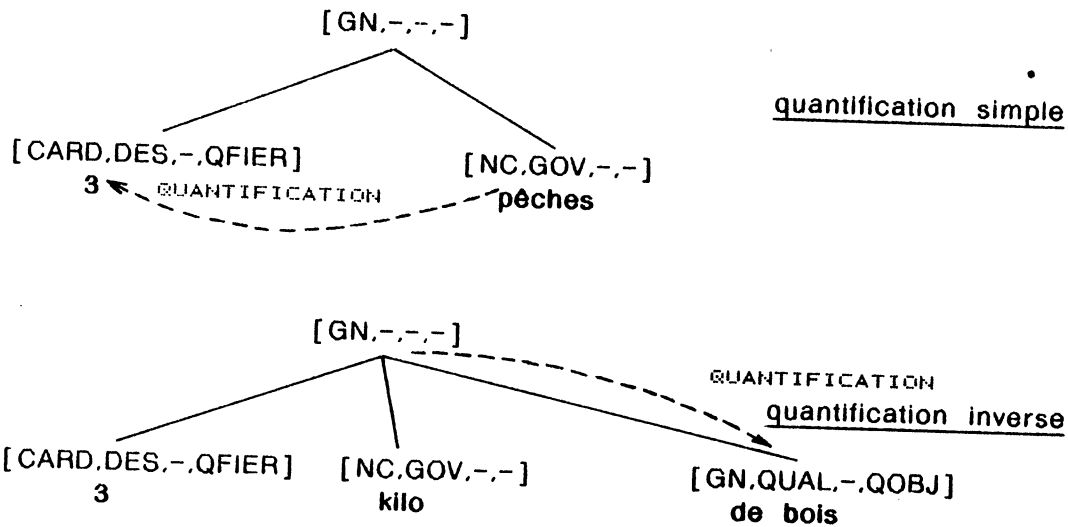


Figure III.10

C. Relations entre sommets de niveaux différents.

On peut imaginer des relations oncle-neveu par exemple. Deux solutions sont alors envisageables:

- (1) Répéter à l'un des niveaux le sommet source de la relation et donc créer un nouveau sommet pour conserver la structure arborescente, on perd alors le lien entre les deux occurrences du même mot dont l'une est "tactique".
- (2) Créer un nouveau type d'étiquette pour chaque nouveau cas rencontré. Chaque nouvelle étiquette exprimera de façon implicite les sommets cibles et source de la relation qu'elle induit.

Ainsi considérons l'exemple suivant:

"A software based system"

Le mot "system" est à la fois gouverneur et deuxième argument de "based" (to base something on something). Pour exprimer cette relation, on peut créer une nouvelle étiquette ARG-1 qui exprime que le sommet source de la relation: "est deuxième argument" (exprimée généralement par la relation ARG1) est le gouverneur du sommet porteur (et non pas le sommet porteur lui-même comme c'est le cas pour ARG1), la cible est le gouverneur de l'arborescence élémentaire dont le sommet porteur est racine.

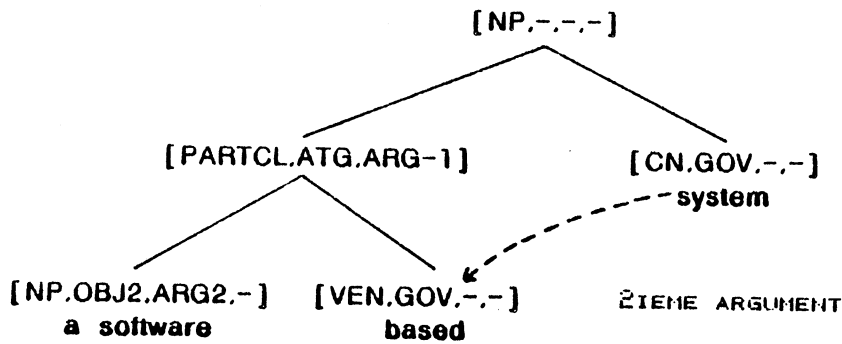


Figure III.11

On peut créer de cette façon toutes sortes de relations sans bouleverser la structure arborescente souhaitée. La première tâche à réaliser dans l'élaboration de grammaires consistera à établir le répertoire des étiquettes.

1.3.5. EXEMPLE.

Regardons les structures multiniveau associées aux phrases françaises (1) à (4) et les structures qu'elles génèrent au niveau des relations logiques et sémantiques.

(1) beaucoup d'équations sont résolues par itération

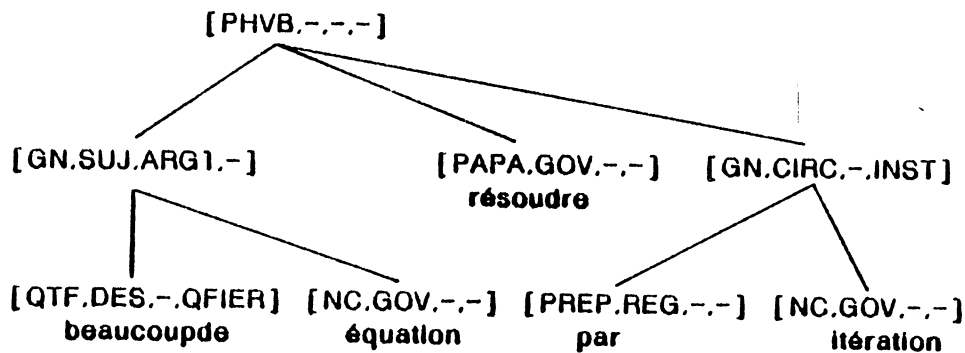


Figure III.12 Structure multiniveau associée à (1).

(2) on résoud beaucoup d'équations par itération

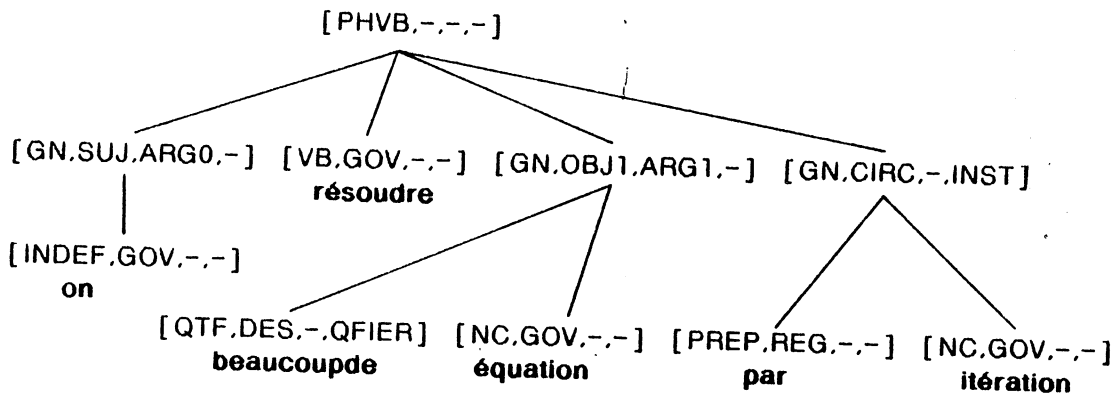


Figure III.13 Structure multiniveau associée à (2).

(3) beaucoup d'équations se résolvent par itération

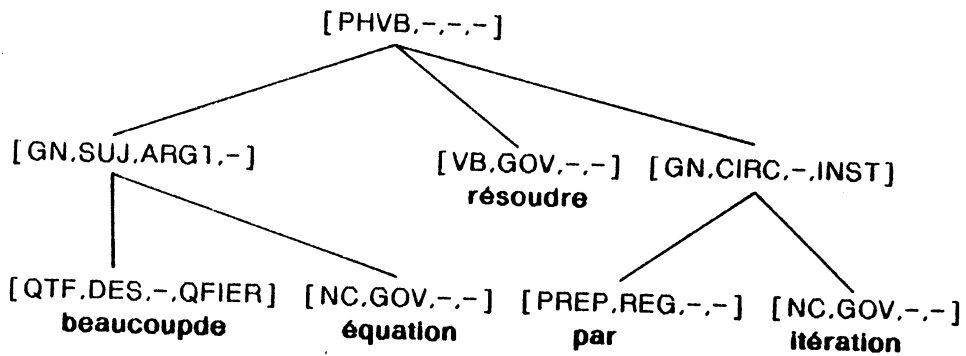


Figure III.14 Structure multiniveau associée à (3).

(4) il se résoud beaucoup d'équations par itération

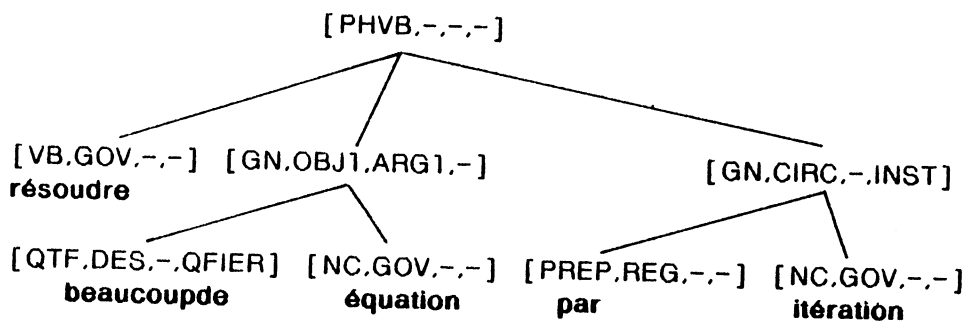


Figure III.15 Structure multiniveau associée à (4).

Par des algorithmes simples, on peut déduire une structure pour chacune de ces quatre arborescences, pour chacun des niveaux choisis.

Les paraphrases (1) et (3) sont rassemblées sous une même structure au niveau (assez peu élevé) des fonctions syntaxiques, cette structure (générée à partir des figures III.12 ou III.14 indifféremment) est la suivante:

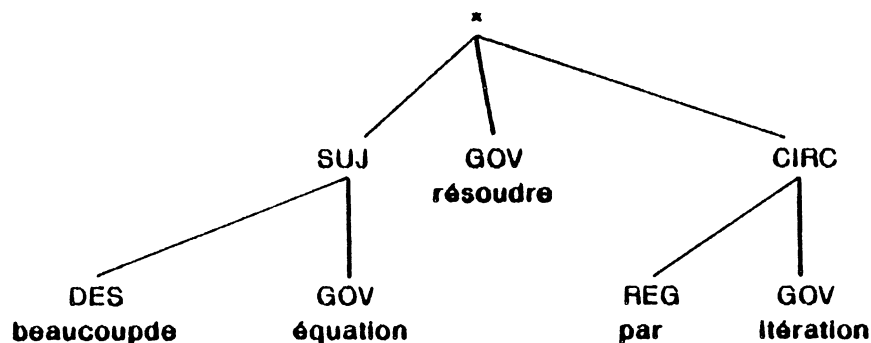


Figure III.16 Structure représentative des phrases (1) et (3) au niveau des fonctions syntaxiques.

Ces phrases reconnues comme équivalentes au niveau des fonctions syntaxiques (bien plus, dans notre cas, elles seront reconnues comme équivalentes syntaxiquement après vérification de la concordance des attributs syntaxiques, tels que le nombre, le temps, la voix, etc.), le seront au niveau plus ambitieux des relations logiques et sémantiques.

Pour reconnaître (4) en tant que paraphrase de (1) et de (3), il nous faudra aller jusqu'à ce niveau des relations logiques et sémantiques, où (1), (3) et (4) engendrent une structure commune (à l'ordre près, qui est alors indifférent).

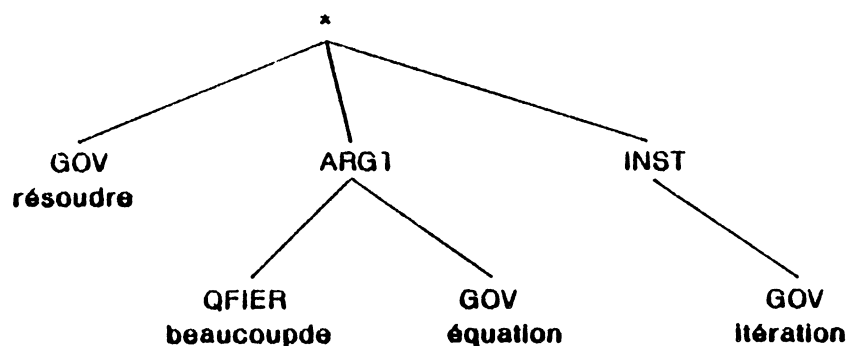


Figure III.17 Structure représentative des phrases (1), (3) et (4) au niveau des relations logiques et sémantiques

Pour finir, il faut reconnaître (2) comme une paraphrasage de (1), (3) et (4). Au niveau des relations logiques et sémantiques, la structure déduite de la figure III.13 est la suivante:

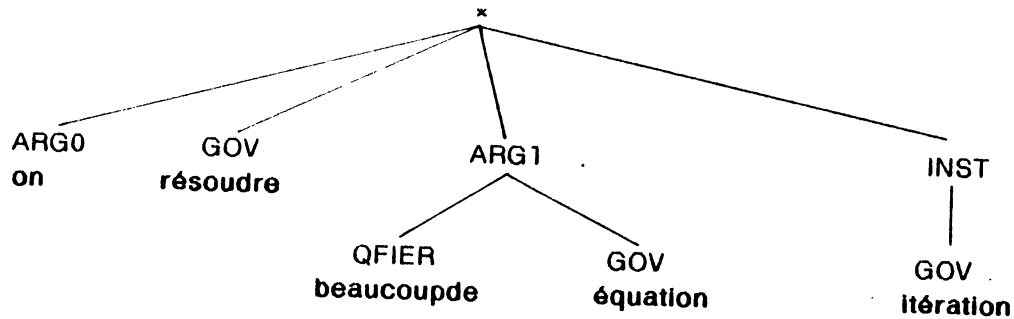


Figure III.18 Structure représentative de la phrase (2) au niveau des relations logiques et sémantiques.

Cette structure III.18 exprime la présence d'un premier argument pour le gouverneur (d'unité lexicale résoudre), cette structure profonde sera reconnue comme équivalente dans la mesure où le premier argument est un représentant indéfini. On peut exprimer cette équivalence par le schéma suivant:

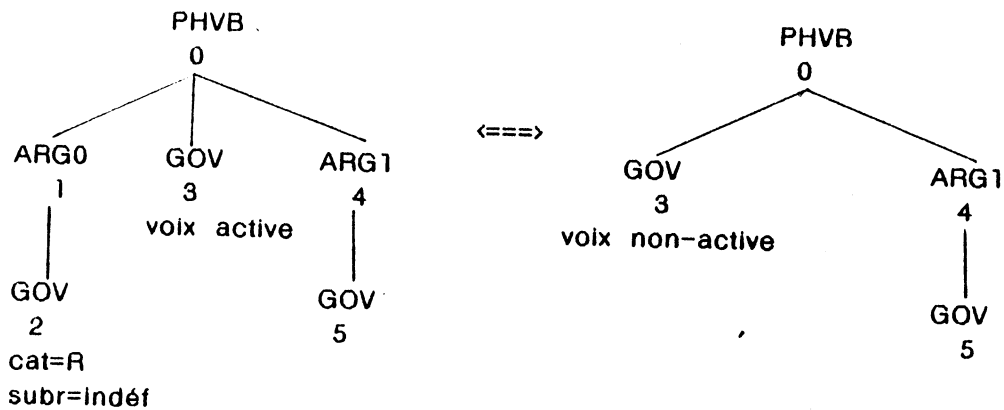


Figure III.19

Cet exemple montre bien que chaque niveau d'interprétation rend possible le rassemblement sous une même structure d'un certain nombre de paraphrasages. Ensuite on traite les synonymies par un répertoire des notions équivalentes.

Il est très important en traduction automatique d'être capable de traiter le problème du paraphrasage, car il peut y avoir (comme c'est le cas dans notre exemple), à partir de plusieurs phrases en langue source, une seule traduction dans une langue cible. Il faudra donc être capable de les reconnaître comme équivalentes, pour être en mesure de leur donner une même traduction.

CHAPITRE III

CHOIX ET REPRESENTATION D'UN MODELE

"TEXTE \longleftrightarrow INTERPRETATION".

DEUXIEME PARTIE

REPRESENTATION D'UNE GRAMMAIRE STATIQUE.



2. REPRESENTATION D'UNE GRAMMAIRE STATIQUE.

INTRODUCTION AUX GRAMMAIRES STATIQUES.

Dans le cadre d'une traduction multilingue, telle que nous l'avons définie, s'effectuant en trois phases, les problèmes en analyse ne sont pas les mêmes que ceux en génération (ni que ceux en transfert) pour une langue donnée, mais ils se rapportent aux mêmes références linguistiques, d'où l'idée que nous développons ici, d'écrire des grammaires statiques -qui sont des grammaires de description- de la langue, dans le modèle multiniveau que nous avons défini.

Ces grammaires statiques, qui se veulent ignorantes des problèmes de stratégie liés à l'analyse aussi bien qu'à la génération, serviront par contre de bases à l'élaboration de grammaires dynamiques aussi bien en analyse qu'en génération, ainsi d'ailleurs qu'en transfert.

La grammaire statique d'une langue peut être décomposée en une grammaire statique morphologique qui décrit la composition des mots de la langue et une grammaire statique structurale. De loin la plus compliquée, cette dernière aboutit à la représentation multiniveau des phrases que l'on peut construire dans la langue.

La grammaire statique structurale établit une correspondance entre les chaînes de mots de la langue et les arborescences à multiples niveaux d'interprétation qui leur sont associées.

A une chaîne X donnée correspond le plus souvent (voir l'exemple précédemment développé §.III.1.3.5.) une et une seule structure multiniveau, Y génératrice de structures différentes pour chacun des niveaux d'interprétation choisis. Dans notre cas, Y sera génératrice de trois structures: Y1 (au niveau des classes syntagmatiques), Y2 (au niveau des fonctions syntaxiques) et Y3 (au niveau des relations logico-sémantiques). Nous avons vu aussi, qu'à une structure Y3 peut correspondre plusieurs structures Y2 et que, de la même façon, à chaque Y2 peut correspondre plusieurs structures Y1, à chacune étant associée une chaîne X (aux contraintes morphologiques près).

Une grammaire statique est une grammaire de description de la langue, description qui sera établie à partir de l'exploration des chaînes de mots qui constituent la langue; elle exprime une correspondance entre ces chaînes de mots et la structure multiniveau qui leur est associée, cette structure étant elle-même génératrice de différentes structures pour chacun des niveaux choisis. Nous remarquons que si en analyse la chaîne X se verra associer la structure Y unique qui permettra de générer Y1, Y2, Y3; en génération, à partir de Y3 on pourra déduire divers choix de Y2, auxquels choix correspondront divers Y1 (donc diverses X) -c'est ce que montre clairement notre exemple traitant des phrases (1) à (4)- et donc à une structure correspondront plusieurs chaînes satisfaisantes. Notre approche statique de la langue, favorise donc, par son principe qui est la correspondance entre une chaîne et une structure, son utilisation dans l'élaboration de grammaires dynamiques d'analyse, au détriment des grammaires dynamiques de génération. Elle n'en est pas moins pour ces grammaires dynamiques une étape, de description de la langue et des phénomènes linguistiques qui s'y rattachent, indispensable à leur élaboration dès qu'elles sont de taille importante.

L'écriture d'une grammaire statique pour une langue donnée nécessite:

- (1) d'étudier des textes dans cette langue pour identifier les phénomènes rencontrés (structures des phrases, des groupes syntagmatiques, ambiguïtés lexicales, ambiguïtés de structures, etc.);
- (2) de disposer d'un formalisme de description approprié;
- (3) de disposer d'une méthode de réalisation qui permette l'accroissement progressif de cette grammaire.

Notons que de façon évidente, le répertoire des étiquettes se rapportant au niveau des relations logiques et sémantiques (niveau Y3) est imposé a priori, puisque ces étiquettes seront communes à toutes les grammaires (de toutes les langues). Par contre, ce sera la grammaire statique qui déterminera le choix des étiquettes des niveaux intermédiaires, et c'est, en outre, en établissant cette grammaire, que l'on définit les divers attributs lexicaux et leur organisation (valences syntaxiques, valences sémantiques, traits sémantiques, etc.); toutefois, la répartition de ces informations est une question d'organisation des dictionnaires. En conséquence, la première tâche à réaliser, dans l'élaboration d'une grammaire statique, consiste à établir le répertoire des étiquettes. Ce répertoire doit être ouvert aux extensions qu'exige le développement ultérieur de la grammaire.

On trouvera en annexe un exemple de jeu de variables, adapté à une grammaire expérimentale du français.

2.1. AUTRES TRAVAUX EFFECTUES DANS CETTE OPTIQUE DE DESCRIPTION STATIQUE DES LIENS ENTRE LE TEXTE ET L'INTERPRETATION.

Le formalisme grammatical établi par M. Kay [KAY.79] [KAY.82a], doit fonctionner en tant que partie d'un modèle de production et de compréhension du langage.

Il cherche à décrire à la fois le sens et l'expression (chaîne de mots). Une description qui ne contient pas, à la fois, l'un et l'autre, est qualifiée d'incomplète.

Une grammaire d'unification peut être vue comme un transducteur dont l'entrée est une description plus ou moins incomplète et dont la sortie est une ou plusieurs descriptions complètes en accord avec la théorie. Ce transducteur vérifie en outre que l'entrée est un objet grammatical valide.

Le locuteur "idéel" fournit au processus une certaine représentation profonde, le travail du processus est de compléter cette image partielle de la phrase par la réalisation des syntagmes et des mots appropriés.

L'auditeur "idéel" a une description complète de la suite des mots, mais il désire la description des syntagmes et la signification du tout pour compléter cette image.

Un auditeur plus réaliste commence avec une image incluant des mots imparfaitement entendus et certaines notions sur ce qui est dit et il désire des détails à la fois au niveau du sens et de la chaîne de mots.

Dans tous les cas le processus consiste à appliquer la grammaire à la description incomplète pour obtenir une description complète (ou une détection de non-grammaticalité).

Ces grammaires sont des grammaires de description qui cherchent à représenter le lien entre une représentation profonde et une représentation de surface. Elles peuvent être compilées aussi bien en transducteur de génération qu'en transducteur d'analyse. La compilation de ces grammaires en analyse est un problème que Martin Kay traite dans un article récent [KAY.82b].

Citons aussi les travaux de Mel'tchuk et Zholkovskij, qui ont présenté leur fameux "modèle sens-texte". Il se sont intéressés de façon approfondie aux paraphrasages et ont fait une étude très détaillée des dérivations. Leurs articles [ME & ZH.70] [ME&ZH.72] présentent une description statique des règles du paraphrasage au niveau des SLS (Structures Logico-syntaxiques).

Une grammaire statique doit être une grammaire de description de la langue établie indépendamment de toute stratégie d'utilisation, et qui relie entre eux plusieurs niveaux d'interprétation. Or, dans la plupart des formalismes de description des langues naturelles, la frontière entre la description et l'implémentation n'est pas toujours claire, et très peu mettent directement en rapport plusieurs niveaux d'interprétation.

2.2. ETUDE D'UN FORMALISME DE REPRESENTATION.

Le formalisme utilisé pour établir les grammaires statiques structurales des langues, permet l'écriture de "planches". Chacune de ces planches se rapporte:

- soit à la construction de groupes syntagmatiques (groupes nominaux, propositions relatives, participiales, etc.);
- soit à des phénomènes qui intéressent divers groupes syntagmatiques (comparatives, interrogations indirectes....);
- soit enfin à la mise en évidence de phénomènes d'ambiguïtés (nécessaire pour une utilisation dans des grammaires dynamiques d'analyse) ou de choix de structure (pour la génération).

Une planche de grammaire statique établit une correspondance entre une chaîne (ou une famille de chaînes) et une structure à multiples niveaux d'interprétation. Cette structure est une arborescence décorée dont la décoration est fonction des informations contenues par les éléments de la chaîne.

Rappelons que les planches de grammaire statique (planches de description de la langue) serviront de base à l'écriture de grammaires dynamiques de la langue:

-grammaire dynamique d'analyse, on leur adjoint alors des planches qui mettent en évidence les ambiguïtés;

-grammaire dynamique de génération qui nécessitent des planches mettant en évidence les choix de construction;

-enfin grammaires dynamiques de transfert pour lesquelles les planches feront apparaître l'aspect contrastif entre la langue source et la langue cible, en termes de structure et d'étiquettes concernant soit les niveaux d'interprétation superficiels (Y1, Y2), soit l'actualisation des énoncés. Chaque planche est alors une correspondance de schémas entre des structures interfaces partielles jugées équivalentes, elle consiste donc en l'écriture de deux schémas arborescents avec les transferts d'étiquettes.

Les planches, telles que nous les avons envisagées, outre la partie "entête" que nous décrirons par la suite, sont établies en trois zones.

La première zone montre la structure arborescente associée à une famille de chaînes; chacune de ces chaînes peut être composée indifféremment de mots (cas des mots outils tels que "qui", "que", "dont"....), de classes syntaxiques de mots (nom commun, verbe conjugué préposition, etc.), et/ou de groupes syntagmatiques (groupes adjectival, phrase infinitive....) ou autres types de groupes (comparatives....) définis dans d'autres planches.

La deuxième zone exprime les contraintes d'existence et de non-existence d'un élément de chaîne en fonction de la présence, de l'absence ou de conditions sur la décoration d'autres éléments ainsi que les contraintes sur les attributs portés par les éléments de chaîne (contraintes propres et contraintes inter-sommets).

La troisième zone exprime les correspondances de décoration sur les différents sommets de la structure; on décomposera cette zone en deux parties, une partie correspondance directe de décoration et une partie correspondance liée à des contraintes sur les sommets.

Une contrainte intrinsèque à ce formalisme est celle de couverture maximale. En effet, dans son utilisation, la chaîne décrite par la planche est la plus grande chaîne possible entrant en correspondance avec l'une des chaînes de la famille de chaînes décrite dans la planche.

2.2.1. FORMALISME D'ECRITURE DE LA PREMIERE ZONE.

Dans cette zone s'exprime donc visuellement la **correspondance chaîne-structure** qui est le but premier de l'établissement d'une planche de grammaire statique.

Les chaînes d'éléments traitées sont des **chaînes d'éléments connexes**, composées indifféremment d'éléments qui sont des **mots**, des **classes syntaxiques** de mots, des racines de **groupes syntagmatiques** ou racines d'autres types de groupes définis dans d'autres planches. Ces racines représentent alors des sous-chaînes qui elles mêmes peuvent contenir des racines de groupes déjà décrits, etc... Le fait de pouvoir faire référence à un groupe décrit dans une planche à l'intérieur d'une autre planche induit:

(1) la notion de **hiérarchie** entre les planches, puisqu'il faudra qu'une certaine planche ait été décrite pour qu'une autre planche qui l'utilise, puisse l'être;

(2) la possibilité de retrouver la description référée.

Nous reviendrons sur l'établissement de cette hiérarchie qui sera un travail préliminaire à l'établissement de toute grammaire statique et sur les problèmes soulevés par les références.

Les structures associées aux chaînes et les sous-structures référées sont des structures et des sous-structures que l'on décrit et que l'on réfère dans leur totalité; c'est-à-dire que dans nos planches, tout ce qui est présent doit être clairement explicité.

Cette zone nécessite la description d'un formalisme de représentation:

- des chaînes;
- des structures;
- des références à des structures définies dans d'autres planches;
- des correspondances entre les chaînes et les structures associées.

2.2.1.1. FORMALISME D'ECRITURE DES CHAINES.

Dans le but de rendre la grammaire d'une langue la moins volumineuse possible, nous voulons limiter au maximum le nombre de planches et par là même, multiplier le nombre de chaînes décrites à l'intérieur d'une même planche.

Le formalisme d'écriture des chaînes doit donc couvrir une famille de chaînes. Cette famille sera composée d'une suite d'éléments connexes dont certains seront facultatifs, certains itérés, certains seront de diverses catégories possibles et/ou de diverses classes, etc. A ce propos, nous essaierons de ne pas trop limiter les possibilités, de façon à donner au linguiste qui écrira les grammaires un outil qui lui permette de recouvrir un

maximum de types de chaînes par une même planche.

Chaque élément de chaîne sera marqué par un signe particulier (par exemple "x"). Pour rendre possible la référence à un élément de chaîne à l'intérieur de la planche, on lui associe un numéro unique.

Soit 7 le numéro affecté à un élément de chaîne.

-Cet élément peut être facultatif, il sera alors noté [7]

-S'il est obligatoire ce sera 7

-S'il doit être itéré au moins une fois ce sera 7⁺

-S'il peut être itéré et peut être absent, 7* (qui correspond à [7]⁺)

(+ et * sont les notations classiques des expressions régulières).

Nous pouvons éventuellement étendre ces conventions à des suites d'éléments connexes et exprimer par exemples la notion suivante: "8 est facultatif, 9 aussi, mais si 8 est présent, 9 l'est aussi" par le parenthésage suivant: [[8]9]. Cette notation permet en outre, de réduire la taille de la zone suivante, en supprimant certaines vérifications sur des incompatibilités de présences.

Exemple de chaîne:

x	x	x	x	x _*	x	x _*
[2]	[3]	[4]	[5]	6	7	8

Dans cette chaîne seul 7 est obligatoire.

Figure III.20

De façon évidente il faut que les chaînes vérifient un certain nombre de contraintes, pour être reconnues comme appartenant à la famille de chaînes décrite. Contraintes de catégorie syntaxique (CAT) et de sous-catégorie syntaxique correspondante (SUBCAT), contraintes de classe syntagmatique (K), contraintes d'unité lexicale (UL) puisque la chaîne peut éventuellement contenir un mot (cas des mots outils tels que "qui", "que", "dont",...), éventuellement autres contraintes jugées nécessaires par le linguiste pour une planche particulière (contrainte de type pour les planches traitant du noyau verbal par exemple, qui permettra de mettre immédiatement en évidences les modaux, les auxiliaires, ...).

Pour des raisons de commodité, on peut imposer sur un même élément de chaîne plusieurs sortes de contraintes. On réduit ainsi le nombre de cas cernés. On liera ces contraintes entre elles: par un "et" (par exemple, figure III.21, imposer une UL et une cat/subcat à un même élément) ou par un "ou" (par exemple, figure III.22, élément devant vérifier une certaine classe syntagmatique ou une certaine catégorie syntaxique; éléments de diverses sous-catégories possibles, figure III.23).

Nous pouvons d'autre part introduire l'utilisation de l'opérateur "¬" (opérateur de négation) sur ces contraintes (figure III.24).

Exemples:

x
7
"should" et v/vb

Où 7 est le modal "should" sous une forme conjuguée.

Figure III.21

x
[5]
GCARD ou a/card

5 est facultatif et peut être soit un groupe cardinal, soit un adjoind de sous-catégorie d'adjoind: cardinal

Figure III.22

x
[9]
v/vb ou inf

ou encore

x
[9]
v/vb ou v/inf

qui représentent la possibilité de donner deux choix possibles d'éléments pour une même place dans la chaîne. Dans les deux cas on a affaire à un verbe qui est indifféremment un Infinitif ou une forme conjuguée.

Figure III.23

x
9
v/-inf

exprime que le sommet 9 est un verbe qui n'est pas à l'Infinitif.

x
9
-v/inf

exprime que le sommet 9 n'est pas un verbe à l'infinitif.

Figure III.24

Exemple de famille de chaînes:

(ici, chaînes représentatives d'un certain nombre de groupes nominaux simples -c'est-à-dire ne contenant que des groupes de "hiérarchie" moins élevée: groupes adjectivaux et groupes cardinaux- en français).

	x	x	x	x	x _x	x	x _x
	[2]	[3]	[4]	[5]	6	7	8
K.				GCARD	GADJ		GADJ
cat/subcat.	prep	a/adv	d/-			n/nc	

UL...

Dans ce cas particulier, 4 est un déclcteur de sous-catégorie quelconque.

Figure III.25

Un élément de chaîne qui est une classe syntagmatique K_i, représente une arborescence qui est une arborescence référée, c'est-à-dire une arborescence dont une description est faite à l'intérieur d'une autre planche, arborescence dont la racine est K_i. Les planches qu'il est possible de référer de cette façon seront indiquées dans la partie entête de la planche qui les utilise.

2.2.1.2. FORMALISME DE CONSTRUCTION DES STRUCTURES.

La structure construite est une structure arborescente multiniveau telle que nous l'avons précédemment décrite, dont le niveau immédiatement accessible est celui des classes syntagmatiques Y₁. Dans cette première zone, ce niveau est donc présent, ainsi que, sur certains éléments de chaîne, les attributs relatifs au niveau des fonctions syntaxiques Y₂ qui peuvent parfois découler directement de la position des éléments dans la chaîne et donc être déductibles de la structure syntaxique. Nous aurons donc dans cette première zone: une chaîne qui représente en fait une famille de chaînes dont on connaît classe, catégorie, unité lexicale et éventuellement autres types de renseignements lexicographiques sur les éléments, à laquelle on associe une structure décorée par des étiquettes de classes syntagmatiques et, éventuellement, de fonctions syntaxiques.

La structure associée à la chaîne exemple du paragraphe précédent est la structure d'un groupe nominal simple (c'est-à-dire ne comprenant que des groupes déclarés de hiérarchie moins élevée que lui-même; dans notre exemple, ces groupes seront des groupes adjectivaux et des groupes nominaux). Elle sera représentée de la façon suivante:

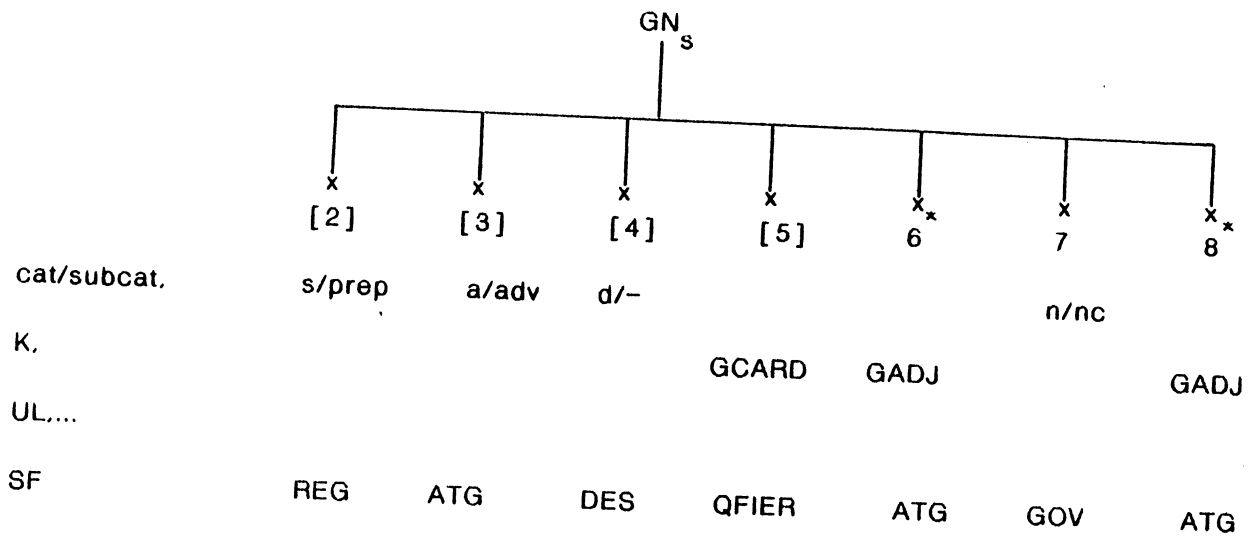


Figure III.26

Les étiquettes représentatives du niveau des relations logico-sémantiques Y3 seront établies en fonction des attributs portés par les noeuds de la structure.

Que la structure associée à une famille de chaînes soit la structure descriptive d'un groupe syntagmatique ou une structure inter-classe, elle peut dans tous les cas faire références à d'autres groupes syntagmatiques ou à d'autres structures déjà décrites et nécessiter la connaissance des éléments constitutifs de ces groupes référés.

Il faut donc, pour traiter tous ces cas, se donner la possibilité de référer à la structure décrite à l'intérieur d'une autre planche autrement que par l'appel de sa racine (comme nous l'avons fait dans l'exemple précédent qui décrit un groupe nominal simple absorbant des structures de GADJ et GCARD déjà décrits par le seul appel des racines de ces structures "GADJ" et "GCARD").

Rappelons que nous avons décidé de représenter toute structure décrite et donc toute sous-structure référée, dans sa totalité.

Lors d'une référence, le groupe référé, qui peut donc être un groupe syntagmatique ou tout autre groupe, peut être représenté par sa racine (portant le nom du groupe), dominant son gouverneur (s'il en a un), ou tout autre dépendant. Cet élément étant encadré par ce que nous appellerons son contexte gauche CG et son contexte droit CD, ceux-ci pouvant éventuellement être vides.

Par exemple le groupe nominal précédemment défini pourra être référé de la façon suivante:

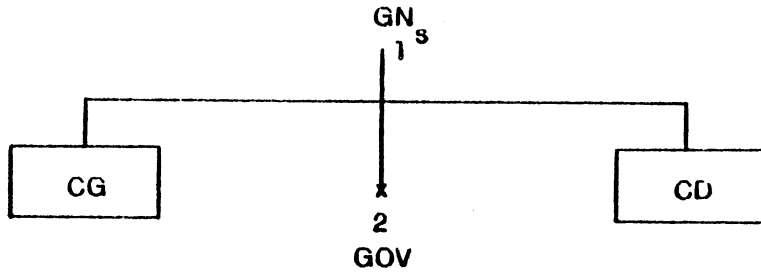


Figure III.27

Cette structure est une structure totale, tout ce qui est à gauche du gouverneur dans le groupe nominal simple référé, est contenu dans le contexte gauche; de la même façon, tout ce qui est à droite du gouverneur est dans le contexte droit. De plus, le contexte gauche détermine la limite gauche du groupe référé, le contexte droit détermine sa limite droite. Dans ce cas la référence permet d'accéder au gouverneur du groupe référé -c'est le besoin le plus fréquent- et à ses attributs, ce gouverneur est repéré par la contrainte de fonction syntaxique de gouverneur. De même, tout autre sommet peut être mis en évidence par la vérification de valeurs particulières sur certaines de ses d'attributs.

Nous étendrons cette notion de contexte à celle de contexte intermédiaire CI qui permettra de mettre en évidence un élément particulier à l'intérieur du groupe référé.

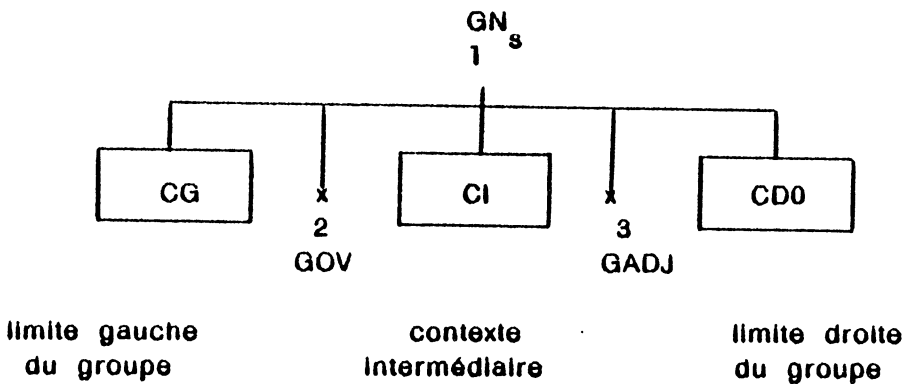


Figure III.28

Un contexte gauche (respectivement droit) dont on exige qu'il soit vide sera noté CG0 (respectivement CD0). De ce fait, dans notre exemple, le GADJ mis en évidence, est le dernier à droite du groupe référé.

Un contexte intermédiaire vide peut être noté CIO ou indifféremment ne pas être exprimé dans la structure, car, puisque celle-ci est totalement décrite, la juxtaposition de deux éléments sous-entend leur connexité (tout ce qui fait partie du groupe référé entre CG et CD, doit être exprimé; de même, tout ce qui est exprimé entre CG et CD, doit faire partie du groupe référé).

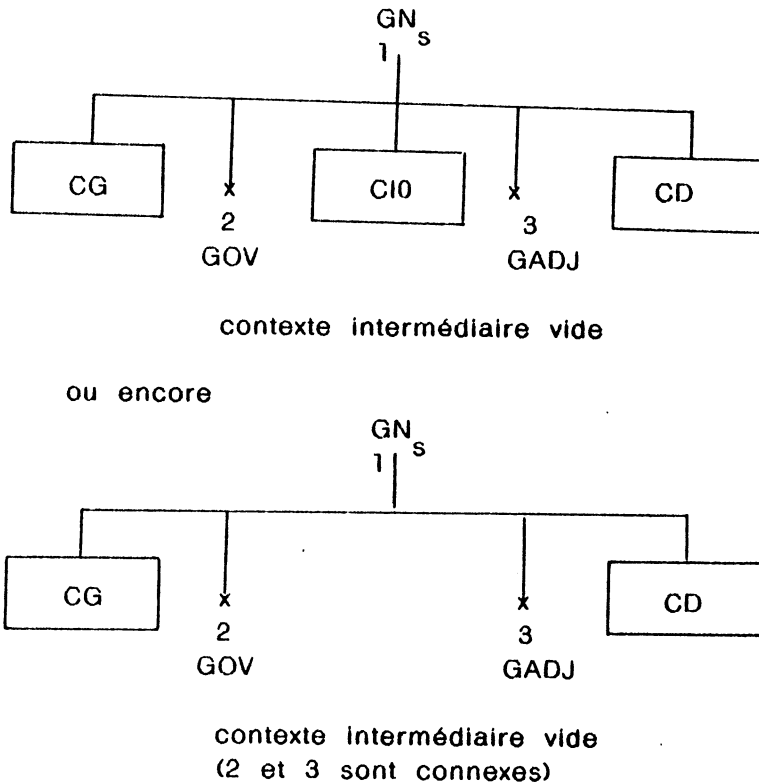


Figure III.29

Le groupe adjectival mis en évidence dans ces deux représentations est le premier à droite du gouverneur.

Le schéma suivant exigerait un groupe nominal simple n'ayant qu'un seul groupe adjectival à droite du gouverneur.

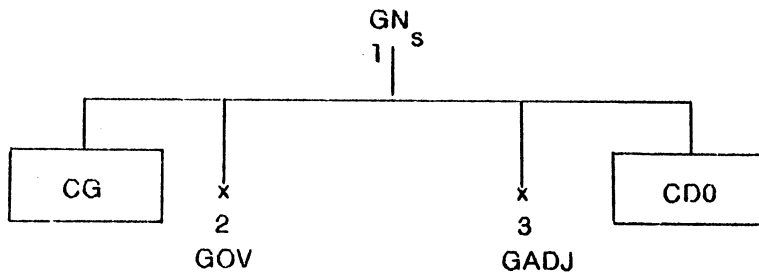


Figure III.30

Dans le cas le plus général, le noeud auquel on veut accéder est placé entre Contexte Gauche et Contexte Droit afin d'être isolé. Si sa localisation ne se satisfait pas des seuls contextes et contraintes sur les attributs du sommet recherché, elle peut se faire après localisation d'autres noeuds qui serviront de points de repère et après utilisation éventuelle d'un Contexte intermédiaire. Quoiqu'il en soit, pour mettre un élément en évidence les contextes gauches et droits (éventuellement vides) seront toujours spécifiés.

Cette possibilité d'utilisation d'une planche à l'intérieur d'une autre (ou d'autres) planche, et la "hiérarchie" sur les planches que cette possibilité induit, est une réponse au phénomène de hiérarchie naturelle entre les classes syntagmatiques et les constructions faites à partir de ces classes: GN nécessitant la connaissance des GADJ, comparatives exigeant celle de GN et GADJ, etc. Nous reviendrons sur ce caractère "bottom-up" du modèle linguistique.

Nous allons définir notre formalisme en ce qui concerne le problème des références à d'autres structures décrites, à l'intérieur d'une nouvelle planche. Pour cela, nous allons partir de cas assez particuliers qui sont généralement de représentation plus aisée, pour traiter ensuite des représentations de plus en plus complexes, utilisables pour des références dans des cas plus généraux.

Insertion simple d'une structure référée

Si le groupe lors de la référence conserve sa racine et sa structure, il n'y aura pas de problème pour le localiser. Au regard de la nouvelle planche, on saura directement quel type de sous-structures rassemble cette référence et donc, si on se donne les moyens de retrouver les planches qu'il est possible de référer, on saura alors quelles sous-structures représente cette référence.

Exemple commenté:

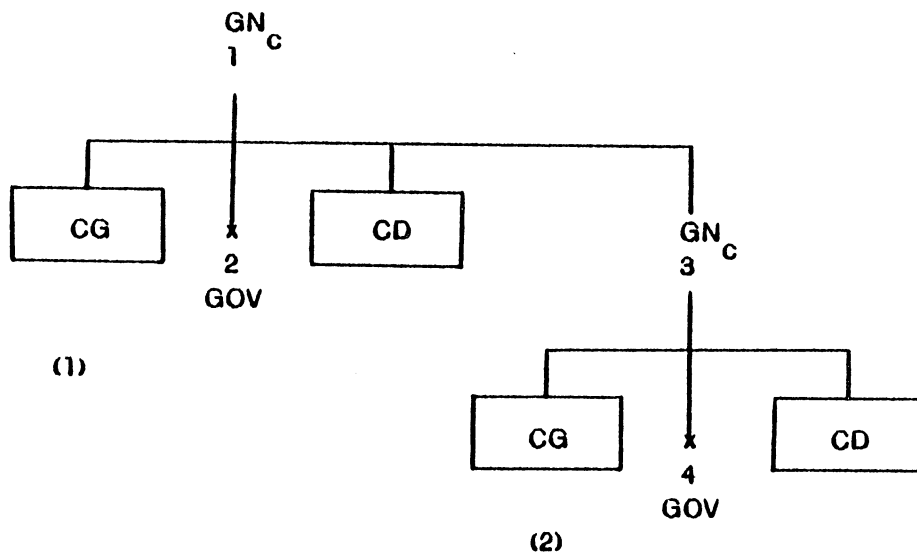


Figure III.31

Ce schéma décrit la structure d'un groupe nominal complexe qui utilise les descriptions déjà faites de deux types de groupes syntagmatiques (1) et (2) qui sont tous deux des groupes nominaux complexes, pris parmi une liste de groupes nominaux complexes décrits, accessibles pour cette planche. Les groupes référés en (1) et (2) sont limités totalement à gauche et à droite par

leur Contexte Gauche et leur Contexte Droit. Dans ce cas, le groupe (2) est donc rattaché à droite (du Contexte Droit) du groupe (1), il n'y a pas d'autre interprétation possible, et donc aucune source de confusion.

Pour que l'outil de description des structures référées soit plus puissant, il doit permettre:

(1) soit l'absorption de la racine d'un groupe référé et donc sa "disparition" dans la nouvelle description, sans pour cela perdre de renseignements sur l'identité des planches référées par elle et, en conservant les descendants de la racine "absorbée"; soit le changement d'identité pour la racine du groupe référé;

(2) la modification d'une structure référée lors de son intégration à une nouvelle structure plus complexe. Cette modification peut être de deux types: changement de la répartition des éléments à l'intérieur de cette structure référée ou insertion dans cette structure de nouveaux éléments (notons que ceux-ci ne peuvent alors provenir que d'un déplacement dans la chaîne d'entrée); modification de cette structure référée au moment de son insertion dans la nouvelle structure.

Cas de changement d'identité du groupe référé

Lorsqu'il y a seulement regroupement de deux racines en une seule ou changement d'identité de la racine, on citera les deux racines (l'absorbante et l'absorbée ou la nouvelle et l'ancienne) et on marquera leur fusion par une accolade.

G1 }
G2 }

Le G1 "absorbe" le G2.

Figure III.32

Exemple: description d'un groupe adjectival simple à partir de la description du groupe adjectival élémentaire par absorption à gauche du groupe adverbial ou de l'adverbe.

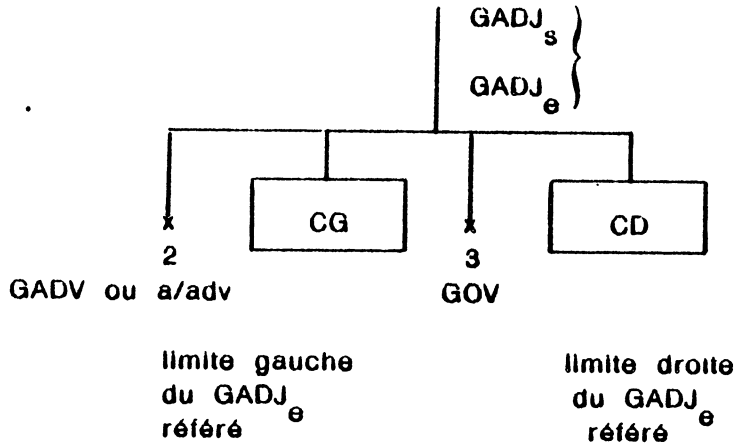


Figure III.33

Cas de changement de structure du groupe référé.

Le formalisme de représentation des structures référées doit permettre en outre de modifier la structure du groupe référé pour une répartition différente des descendants de la racine de cette structure.

Comme il nous faut absolument conserver la structure du groupe référé (qui nous permettra d'utiliser cette référence) et connaître la nouvelle structure, notre planche devra exprimer sur un même schéma ces deux structures.

Exemple de changement de répartition des descendants (ici insertion d'un nouvel élément qui ne peut provenir que du déplacement d'un élément dans la chaîne ou, tout simplement, d'une création, car G2 représente la structure associée à une chaîne d'éléments connexes exprimée par la référence: "CG 3 CD").

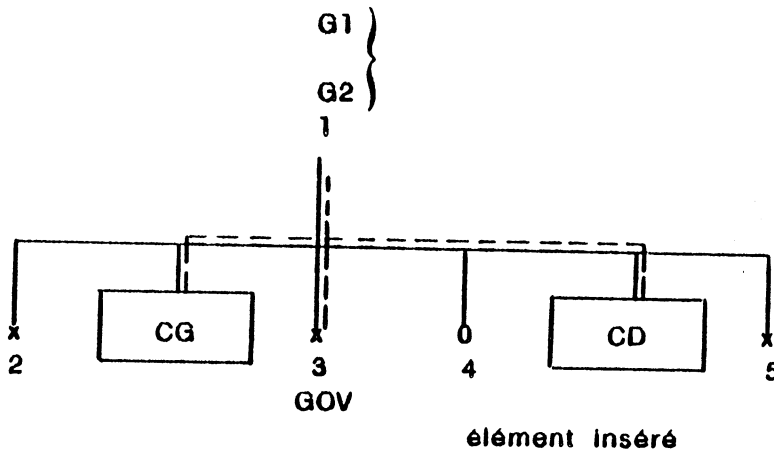


Figure III.34

Sur cet exemple apparaissent clairement les deux structures, l'une référée, l'autre créée. La structure référée sera exprimée en pointillés.

Un certain nombre d'exemples qui utilisent cette double structure, ont déjà été développés. On peut voir à ce sujet la troisième partie de ce chapitre et tout particulièrement, les planches SF07 et SF08.

On peut envisager de représenter aussi des changements plus complexes de structure, par cette méthode. Entre autres, le changement de structure suivant:

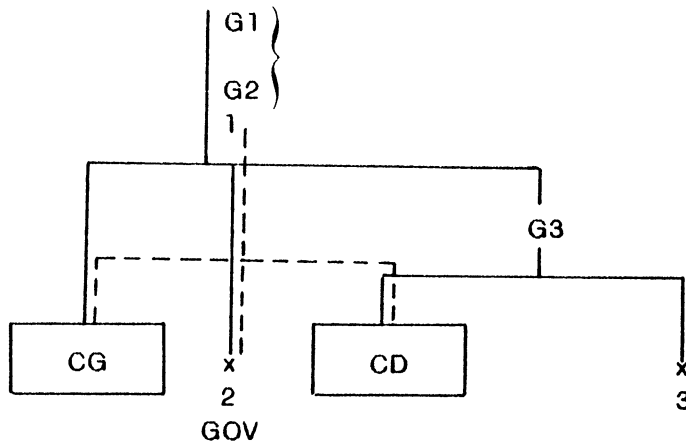


Figure III.35

Nous pensons couvrir, par ce formalisme de représentation des références, la plus grande partie des problèmes de référence qui se poseront aux linguistes lors de l'écriture des grammaires statiques des langues.

Remarquons que l'exemple simple de référence traité en premier lieu, n'est de façon évidente que l'assimilation des deux structures (référée et nouvelle) en une seule et des deux racines en une seule lorsque celles-ci coïncident totalement. Nous aurions pu aborder le problème de façon plus rigoureuse, en partant du cas le plus général pour aboutir au cas le plus particulier. Nous avons, au contraire, traité les cas les plus particuliers pour leur plus grande simplicité et surtout leur plus grande fréquence.

2.2.1.3. CORRESPONDANCES CHAINES-STRUCTURES.

Nous avons vu comment associer une structure arborescente à une chaîne. Parfois, cette association peut poser des problèmes dus à une non-correspondance entre les éléments de la chaîne et les feuilles de la structure. Les problèmes les plus fréquemment rencontrés sont:

- (1) Élément appartenant à la chaîne, n'appartenant pas à la structure.;

- (2) Elément appartenant à la structure mais n'appartenant pas à la chaîne;
- (3) Elément ne se trouvant pas à la même place dans la chaîne et la structure.

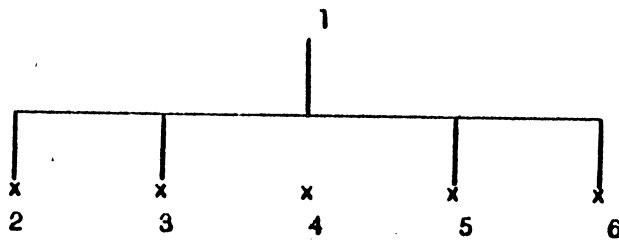
Le formalisme doit permettre une représentation particulière de chacun de ces problèmes. Ils devront être visualisés sur le schéma de façon à être immédiatement repérés pour ne pas être sources de confusions.

Elément présent dans la chaîne et absent dans la structure.

Ce cas de "disparition" se présentera de façon très fréquente au cours de l'élaboration des grammaires statiques (cas des prépositions fortement régies, du "to" de la forme infinitive anglaise, des auxiliaires et des modaux, etc.).

Puisque ce sommet est présent dans la chaîne, nous marquerons sa présence; puisqu'il est absent dans la structure, nous ne le relierons pas à cette structure.

Représentation:



Le sommet 4 présent dans la chaîne est absent dans la structure

Figure III.36

Elément présent dans la structure et absent dans la chaîne.

Cette absence dans la chaîne sera marquée par l'absence de "x" ou mieux par la présence d'un symbole particulier tel que "0". L'élément sera alors normalement relié à la structure. Ce cas se présentera fréquemment; il permettra notamment, de traiter les problèmes d'élisions.

Représentation:

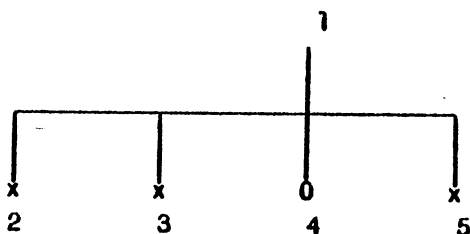
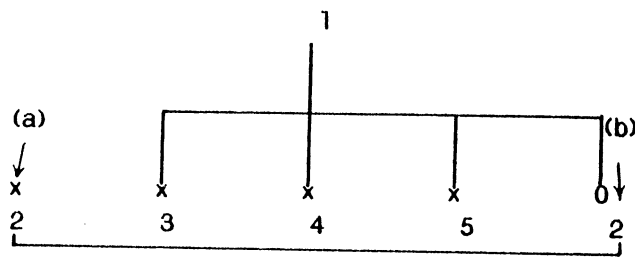


Figure III.37

Cas de rupture chaîne-structure.

On a besoin de rétablir le bon ordre dans les éléments de la chaîne pour pouvoir lui associer une structure. On veut toutefois conserver l'ordre des éléments de la chaîne.

Représentation:



L'élément présent en (a) dans la chaîne se trouve en (b) dans la structure.

Figure III.38

Cela est tout-à-fait compatible avec le formalisme adopté dans les deux premiers cas. Le noeud présent dans la chaîne et absent dans la structure est marqué dans la chaîne (par "x") et n'est pas relié à la structure; le noeud absent dans la chaîne et présent dans la structure est marqué comme absent dans la chaîne (par "0") et est relié à la structure. L'unité de concept entre 2 et 6 est exprimée clairement par un lien.

2.2.1.4. REPRESENTATION DU CONTEXTE.

Il est nécessaire que notre formalisme permette de consulter un certain nombre d'éléments de contexte (à gauche comme à droite du groupe construit) avant d'établir certaines correspondances entre les chaînes et les structures ou avant d'établir certaines valeurs d'attributs. Nous traiterons le cas du contexte droit, le formalisme applicable au contexte gauche s'en déduira de façon évidente.

Plusieurs cas peuvent se présenter:

(1) on a besoin de l'élément suivant dans la chaîne en tant que contexte. (Par exemple un groupe nominal ne peut absorber le groupe nominal qui le suit que si celui-ci ne peut lui-même absorber un éventuel groupe nominal successeur, il faudra donc, pour décrire le groupe nominal complexe constitué de deux groupes nominaux successifs, s'intéresser au groupe successeur);

(2) on a besoin de vérifier qu'il existe un élément de contexte droit, répondant à un critère de choix précis dans la suite de la chaîne;

(3) on a besoin de vérifier que tous les point du contexte droit vérifient un certain critère, ou qu'aucun point ne vérifie un critère;

(4) on a besoin de tester un point particulier du contexte droit;

(5) on a besoin de tester le dernier point de contexte droit.

Pour traiter tous ces cas, on peut considérer le contexte comme une liste d'éléments (prise à un certain niveau par rapport à la structure définie dans la planche, ce niveau est généralement le niveau immédiatement supérieur), et traiter les références au contexte comme des références à un ou à des éléments pris dans cette liste d'éléments.

Les éléments de la liste d'éléments de contexte seront reliés à la structure décrite, par des pointillés.

Le contexte est considéré comme référé dans sa totalité, seuls les éléments qui dans ce contexte nous intéressent seront explicités.

Pour mettre en évidence un (ou des) point particulier, vérifiant une certaine condition, dans la liste d'éléments de contexte ainsi disponible, on associe à cette liste d'éléments les opérateurs "+" et "*" définis pour les chaînes et les quantificateurs universels et existentiels "∀" et "∃" auxquels on donne les significations habituelles et qui représenteront une condition sur la liste d'éléments de contexte, dans une structure du type suivant:

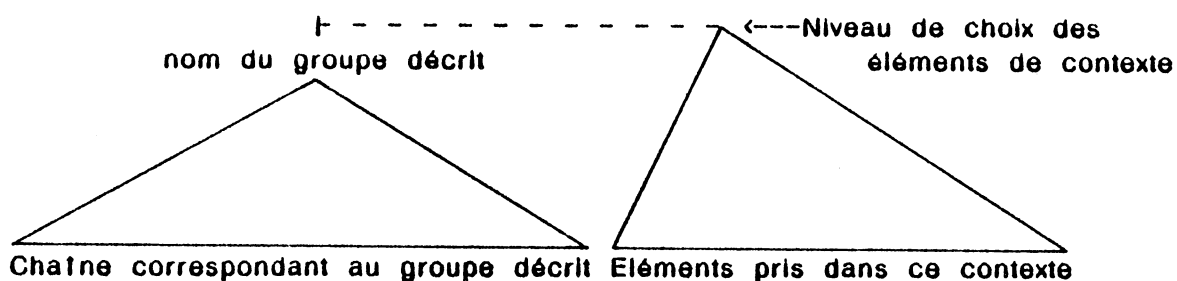


Figure III.39

Nous Imposerons aussi, que les éléments de la liste des éléments de contexte, cités soient des éléments connexes (pris à partir de la droite pour le contexte gauche et à partir de la gauche pour le contexte droit). De ce fait, on pourra tester un élément particulier par ses prédécesseurs ou ses successeurs.

Enfin, la fin du contexte pourra en cas de besoin être extrimée par "*".

Exemples de localisation d'éléments dans le contexte:

"Il existe un élément pris dans la liste d'éléments de contexte qui vérifie le critère C1".

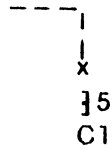


Figure III.40

"Tous les éléments de la liste d'éléments de contexte vérifient le critère C2".

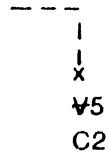


Figure III.41

"Aucun élément de la liste d'éléments de contexte ne vérifie C3".

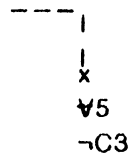


Figure III.42

"Le premier élément de contexte vérifie C4".

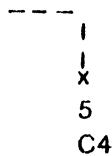
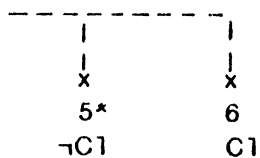
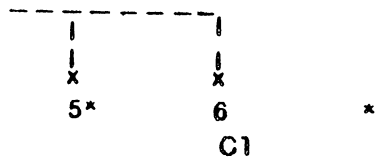


Figure III.43

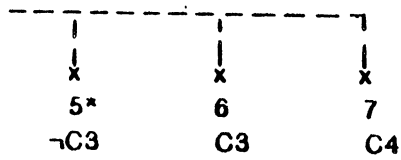
Mise en évidence de points particuliers à l'intérieur du contexte:



met en évidence le premier élément de contexte droit vérifiant C1.



permet de vérifier que le dernier point de contexte vérifie C1.



met en évidence un élément vérifiant C4 suivant immédiatement le premier élément vérifiant C3.

Figure III.34

On peut de cette façon localiser très précisément un élément de contexte en multipliant les conditions sur ses voisins.

2.2.2. FORMALISME D'ECRITURE DE LA DEUXIEME ZONE.

Cette zone exprime les contraintes d'existence et de non-existence d'un élément de chaîne en fonction de la présence, de l'absence ou de conditions sur la décoration d'autres éléments, ainsi que les contraintes sur les attributs portés par les éléments de chaîne: contraintes propres et contraintes inter-sommets. Elle permet donc de sélectionner les chaînes étudiées dans les planches à partir des contraintes portées par les éléments de chaîne (éventuellement, contraintes portées par des éléments de contexte). Ce sera donc dans cette zone: les contraintes propres autres que celles présentées en zone 1 et toutes les contraintes inter-sommets qui elles, ne peuvent être exprimées en zone 1.

Contraintes d'existence.

Nous aurons, en premier lieu, besoin de vérifier la présence ou l'absence d'un élément de chaîne (ou de structure dans le cas d'une référence à une structure déjà décrite).

Soit 7 cet élément.

L'absence de 7 s'écrira $\neg 7$.
Sa présence s'exprimera par 7.

Liste des opérateurs nécessaires à l'écriture des contraintes.

- E- égalité
- NE- non-égalité
- I- intersection
- U- union
- CONT- contenance: $Y\text{-CONT-}x$, Y contient x
- NCONT- non-contenance
- DANS- contenance: x est contenu dans Y
- NDANS- non-contenance
- ET- et logique
- OU- ou logique (non-exclusif)
- OUE- ou exclusif
- IMP- implication
- SSI- équivalence logique

On associe à ces opérateurs parenthésage et hiérarchie classiques.

Variables et procédures.

Nous nous autorisons à utiliser des variables et des procédures qui seront définies dans un glossaire. Le nom de procédure sera précédé de "\$" lors de son appel. On pourra leur appliquer l'opérateur unaire "-" (qui signifiera la non-vérification de la procédure).

Exemples:

\$AGRNUM (x,y,z): agrément en nombre entre x, y et z.
\$AGRVL1 (x,y): Agrément en valence 1 entre x et y.

Notion d'ensemble.

Nous associerons aux opérateurs déjà définis la notion d'ensemble, exprimée simplement par une liste de valeurs mise entre crochets.

Exemple:

SUBD(3) -DANS- {ARTD,ARTI,DEM}
Qui demande à l'élément "3" d'être de sous-catégorie de déicteur: article défini ou indéfini, ou encore démonstratif.

Problèmes des références aux éléments de chaîne absents.

Lors de l'établissement des contraintes sur les attributs, Un élément testé peut être absent. Or, dans l'établissement de la grammaire statique, toute référence aux attributs portés par un élément supposait la présence de cet élément. Par exemple, l'écriture du test d'agrément en nombre entre un nom x, un article y et un adjectif z en français (\$AGRNUM (x,y,z)) supposait la présence des trois éléments x, y et z; de même, vérifier que CAT(7)-E-t supposait l'existence de 7.

Nous définissons une valeur d'absence: NIL et une valeur indéfinie, qui sera la valeur prise par un test simple sur un élément absent.

Ainsi nous comprendrons "\$AGRNUM(x,y,z)" comme "\$AGRNUM(x,y,NIL)" si z est absent et "CAT(3) -E- x" prendra la valeur indéfinie si le sommet 3 n'est pas présent dans la chaîne.

Pour traiter les problèmes dus à la présence de valeurs indéfinies dans les contraintes, nous établissons une logique à trois variables: "V", "F", "I".

La valeur indéfinie, "I", vérifie les relations logiques suivantes:

Table de "¬X".

X	¬X
V	F
F	V
I	I

donc, $\neg I = I$

Table de "X -ET- Y".

X \ Y	V	F	I
V	V	F	I
F	F	F	F
I	I	F	I

Table de "X -OU- Y".

X \ Y	V	F	I
V	V	V	V
F	V	F	I
I	V	I	I

Dans cette logique la commutativité du "OU" et du "ET" est conservée.

Table de "X -IMP- Y"

X \ Y	V	F	I
V	V	F	I
F	V	V	V
I	V	I	I

Table de "X -SSI- Y".

X \ Y	V	F	I
V	V	F	I
F	F	V	I
I	I	I	I

Exemples:

(3) CAT(3) -E- x -IMP- \neg 7

Si 3 n'est pas dans la chaîne, le test CAT(3) -E- x est indéfini et donc l'implication est vrai si \neg 7 est vrai, indéfinie sinon.

(4) CAT(3) -E- x -IMP- (CAT(7) -E- y -OU- SUBCAT(3) -E- z)

Les cas possibles sont:

"I -IMP- I -OU- I", si 3 et 7 sont absent, le résultat est I.

"V -IMP- I -OU- V" (qui est V) ou "V -IMP- I -OU- F" (indéterminé),

"F -IMP- I -OU- V" (V) ou "F -IMP- I -OU- F" (V), si 7 seul est absent.

"I -IMP- V -OU- I" (V), "I -IMP- F -OU- I" (I), si 3 seul est absent.

(5) CAT(3) -E- x -IMP- (CAT(7) -E- y -ET- SUBCAT(3) -E- z)

Les cas possibles sont:

"I -IMP- I -ET- I" (I), "I -IMP- V -ET- I" (V),

"I -IMP- F -ET- I" (I) si 7 est absent;

"V -IMP- I -ET- V" (I), "V -IMP- I -ET- F" (F),

"F -IMP- I -ET- V" (V), "F -IMP- I -ET- F" (V) si 3 est absent.

(6) (CAT(3) -E- x -ET- 7) -IMP- \neg 5

Ici, si 3 est absent, les cas possibles seront:

"I -ET- V -IMP- V" (V),

"I -ET- V -IMP- F" (I),

"I -ET- F -IMP- V" (F),

"I -ET- F -IMP- F" (V).

Toute non-vérification (valeur de vérité "F") d'une contrainte en zone 2 entraîne la non-validité de la chaîne associée pour cette planche et donc qu'il n'y a pas correspondance entre cette chaîne et la structure décrite.

2.2.3. FORMALISME D'ECRITURE DE LA TROISIEME ZONE.

La troisième zone exprime des correspondances de décoration sur les différents sommets de la structure. Nous décomposerons cette zone en deux parties, une partie pour les correspondances directes et une partie pour les correspondances liées à des contraintes sur les sommets. Il s'agit ici de mettre en place la décoration portée par la structure. Cette mise en place se fait donc, dans la première partie, directement; elle est alors une déduction immédiate de la structure. Dans la deuxième partie, elle s'établit en fonction de contraintes vérifiées par les attributs portés par les éléments de la structure. L'essentiel du travail, dans ce dernier cas, porte sur l'établissement des relations logiques et sémantiques attribuées aux divers éléments de la structure.

Dans cette zone, nous reconduirons les opérateurs définis dans la deuxième zone ainsi que les notions de variables et de procédures. Les contraintes portant sur des sommets absents dans la structure seront traitées de la même façon.

Les correspondances directes s'expriment de façon simple, par l'opérateur "-E-" (par exemple: NBR(1) -E- NBR(2)).

Les correspondances liées à des contraintes s'expriment par un (ou plusieurs) graphe(s) établi(s) suivant les relations existant entre: d'une part, une série de contraintes et d'autre part, une série de correspondances dans la décoration.

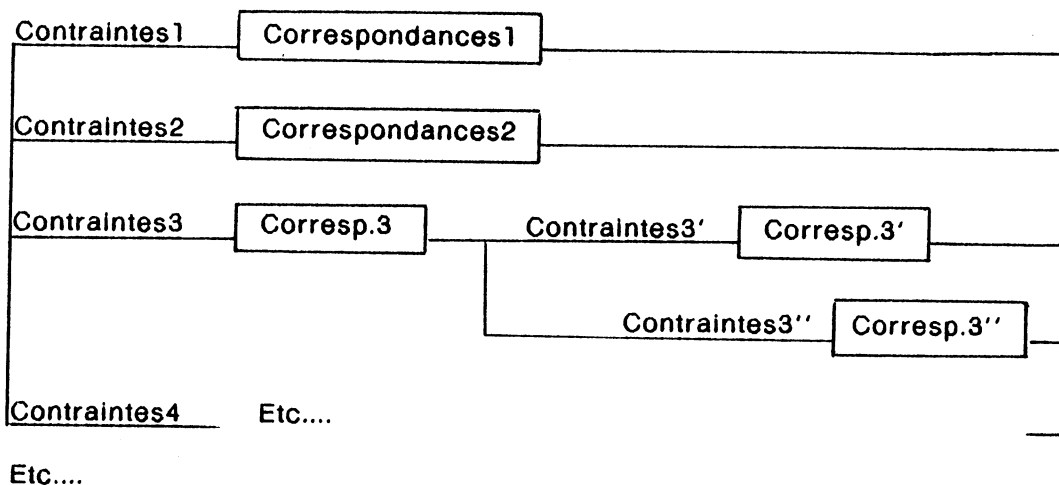


Figure III.46

Dans ce graphe les correspondances de décoration sur les sommets sont fonction de certaines valeurs d'étiquettes sur les structures.

Dans une utilisation en analyse ce graphe verrait certaines valeurs de contraintes vérifiées, l'analyse devrait alors établir les correspondances associées.

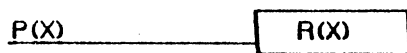
Ce graphe pourrait alors se lire comme un graphe d'affectations conditionnelles (de haut en bas et de gauche à droite) où, ce qui deviendrait des arcs serait les conditions et où les blocs contiendraient les affectations découlant de la vérification de ces conditions.

Dans une utilisation dans un modèle de génération, ce sont les correspondances qui seront ou non vérifiées. La génération devra établir les contraintes associées aux correspondances vérifiées.

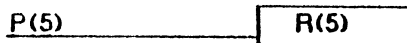
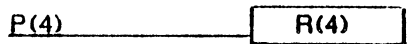
Pour des raisons pratiques nous nous autorisons à utiliser deux macro-écritures. Nous simplifierons de cette façon les graphes en regroupant un certain nombre de correspondances en un seul bloc et un certain nombre de contraintes en un seul arc.

(1) $X = 3 \text{ -OU- } 4 \text{ -OU- } 5$.

Dans ce cas le graphe



représente les graphes



Il s'agit là, de donner à X plusieurs valeurs de sommets. Chacune des références à X dans un graphe est en fait trois références distinctes. De ce fait chacune de ces références permet de n'écrire qu'un graphe au lieu de trois. On peut voir une utilisation de cette écriture dans la troisième partie de ce chapitre: planche n.SF25.

(2) $X \text{ -E- } x / y / z$.

Le graphe suivant



représente en fait le graphe



Il s'agit là, de la possibilité de faire un choix dans un éventail de correspondances parallèlement à un choix dans un éventail de contraintes possibles. (Voir planche n.MF08).

2.2.4. ARTICULATION ENTRE LES PLANCHES.

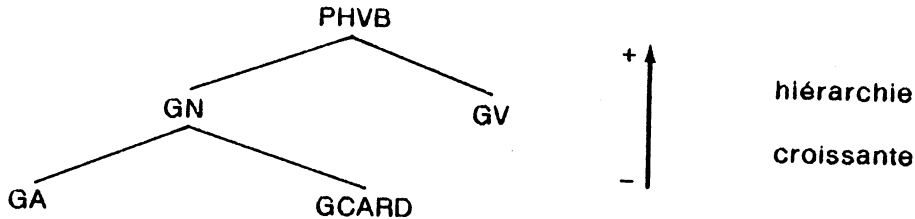
Nous avons déjà traité le problème des références, dans une planche, à des groupes décrits à l'intérieur d'autres planches, et nous avons parlé de la notion de hiérarchie entre les planches que cette possibilité de référence induit.

Cette hiérarchie entre les planches statiques est en fait le reflet de la hiérarchie naturelle entre les classes syntagmatiques et les constructions faites à partir de ces classes (Groupe Nominal nécessitant la connaissance des Groupes Adjectivaux, comparatives exigeant celle des Groupes Nominaux et des Groupes Adjectivaux...).

En ce qui concerne les planches relatives à la construction des groupes syntagmatiques, la démarche est la suivante:

Les classes syntagmatiques sont ordonnées de façon à établir en premier lieu des constructions "élémentaires", puis des constructions "simples".

Pour ces constructions, on peut établir par exemple la hiérarchie suivante:



- Où PHVB est la classe: phrase gouvernée par un groupe verbal.
- Où GV est la classe: groupe verbal gouverné par un verbe conjugué.
- Où GN est la classe: groupe nominal gouverné par un nom.
- Où GA est la classe : groupe adjectival gouverné par un adjectif.
- Où GCARD est la classe: groupe cardinal.

Figure III.47

Les constructions simples imposent que toute classe ne peut dominer une classe de hiérarchie plus élevée, ni elle-même. Par exemple, d'après cette hiérarchie un GN simple ne peut dominer que des GA simples, GA élémentaires, GADV simples ou des GADV élémentaires.

L'établissement d'une grammaire statique nécessite donc l'établissement préalable de la hiérarchie adoptée pour la construction des groupes syntagmatiques. On peut, en effet, envisager d'autres groupes que ceux-ci (par exemple groupes déictiques), pour faciliter la description, groupes que l'on incorporera à la présente hiérarchie; on peut aussi changer cette hiérarchie.

Une fois répertoriées les constructions simples, on décrit les constructions complexes (ex: GN complexes), de manière récursive. En effet, un groupe adjectival peut dominer un autre groupe adjectival, mais aussi un groupe nominal, et un groupe nominal peut en dominer un autre, peut dominer une phrase (relative, participiale, infinitive, etc.).

Exemple:

"Un exercice utile à la résolution très rapide du problème par le candidat".

On décrit les groupes élémentaires:

(Les gouverneurs sont soulignés).

GA : utile, rapide.

GN_e^e: exercice, résolution, problème, candidat.

Puis les groupes simples:

GA : utile, très rapide.

GN_s^s: un exercice utile, à la résolution très rapide, du problème, par le candidat.

Enfin les groupes complexes:

GN : résolution très rapide du problème par le candidat.

GA_c^c: utile à la résolution très rapide du problème par le candidat.

GN_c^c: un exercice utile à la résolution très rapide du problème par le candidat.

2.2.5. PARTIE ENTETE D'UNE PLANCHE STATIQUE.

La partie entête d'une planche statique doit signifier clairement:

- (1) la langue qu'elle décrit;
- (2) la grammaire statique à laquelle elle appartient;
- (3) le type général de groupe décrit par la planche.
(ex: GN_C, comparative, GN.....);
- (4) le cas particulier que traite^s le groupe décrit.
(ex: GN_C coordination de GN_C, GV forme passive, etc.);
- (5) le numéro d'ordre de la planche dans l'écriture de la grammaire statique.
- (6) les numéros des planches qu'elle réfère.

D'autre part, elle doit laisser des traces de son utilisation:

- (7) date d'écriture et source de cette planche;
- (8) nom du linguiste qui l'a établie;
- (9) modifications éventuelles, noms des linguistes et dates;
- (10) grammaires dynamiques utilisant cette planche et dates.

2.2.6. EXEMPLES DE PLANCHES STATIQUES.

Nous exposerons ici deux exemples dans une grammaire simplifiée du français.

2.2.6.1. EXEMPLE 1.

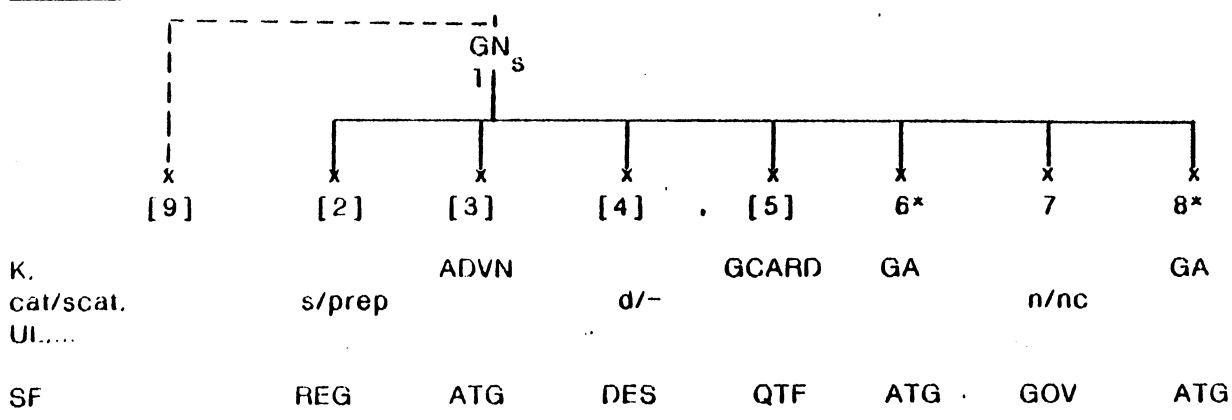
Langue décrite: français
 Grammaire statique: Fexemple
 Planche n°: 17

Type: GROUPE NOMINAL (SIMPLE)

Cas traité: Groupe nominal gouverné par un nom commun
 Planches référées: 8.9.10 (GA)
 15 (coordination de GA_s)

Etablie le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(2)	dans:
le:(3)	dans:

ZONE 1



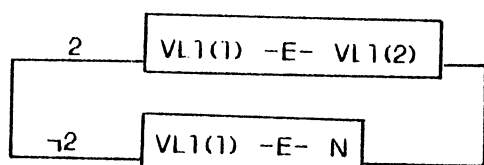
ZONE 2

3 -IMP- 4 -OU- 5
 (9 -ET- 2) -IMP- (4 -OU- 5)
 *com: si 1 n'est pas en tête de phrase.
 ‡AGR.GNR.NBR(4.5.6.7.8)
 *com: agrément en genre et en nombre entre les sommets 4.5.6.7 et 8.

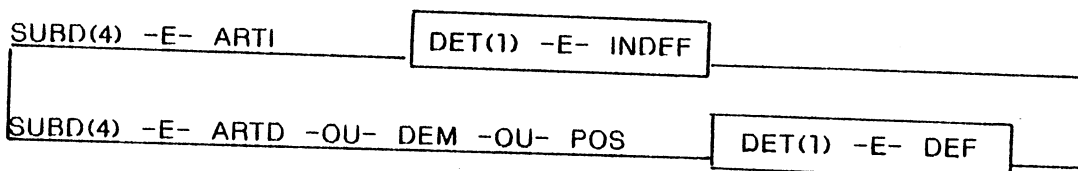
ZONE 3

GNR.NBR(1) -E- GNR.NBR(4) -ET- GNR.NBR(5) -ET- GNR.NBR(6) -ET-
GNR.NBR(7) -ET- GNR.NBR(8)

*com: en analyse résolution des ambiguïtés sur le groupe nominal.

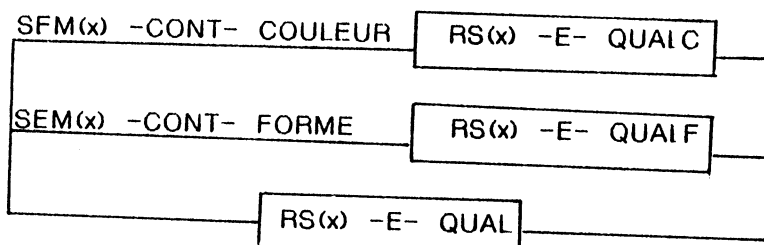


*com: correspondances valence d'état du GN, valence de la prep.



*com: valeur de définition du GN en fonction du d.

x = 6 -OU- 8



*com: relations sémantiques des GA en fonctions des attributs portés par ces GA.

2.2.6.2. EXEMPLE 2.

Langue décrite: français
 Grammaire statique: Fexemple
 Planche n: 25

Type: GROUPE NOMINAL (COMPLEXE)

Cas traité: groupe nominal dominant un autre groupe nominal

Planches référées: 17,18,19,20 (GN_C)

21 (coordination de GN)^s

23,24,25 (GN_C)

Etablie

le: par:

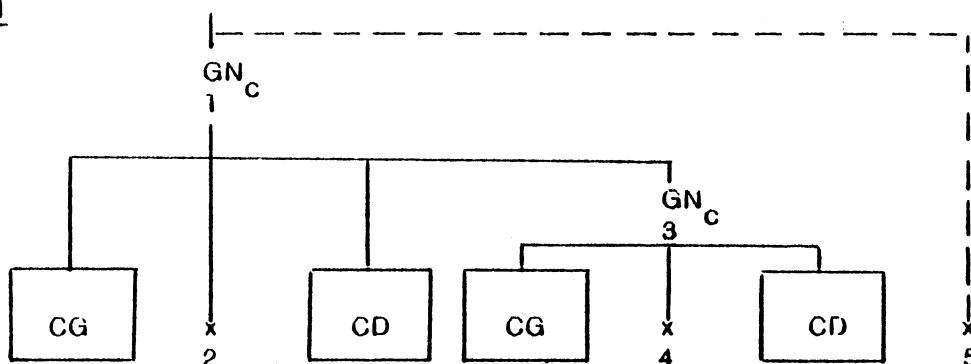
Modifiée

le:(1) par:
 le:(2) par:
 le:(3) par:

Intégrée

le:(1) par:
 le:(2) par:
 le:(3) par:

ZONE 1



UL...

SF

GOV

GOV

ZONE 2

K(5) -NE- (GN -OU- GA -OU- PHPART) -OU- \$TFSTINC(3,5)

*com: 3 ne peut dominer 5.

\$AGR.VL11(2,3) -OU- \$AGR.VL21(2,3)

-OU-

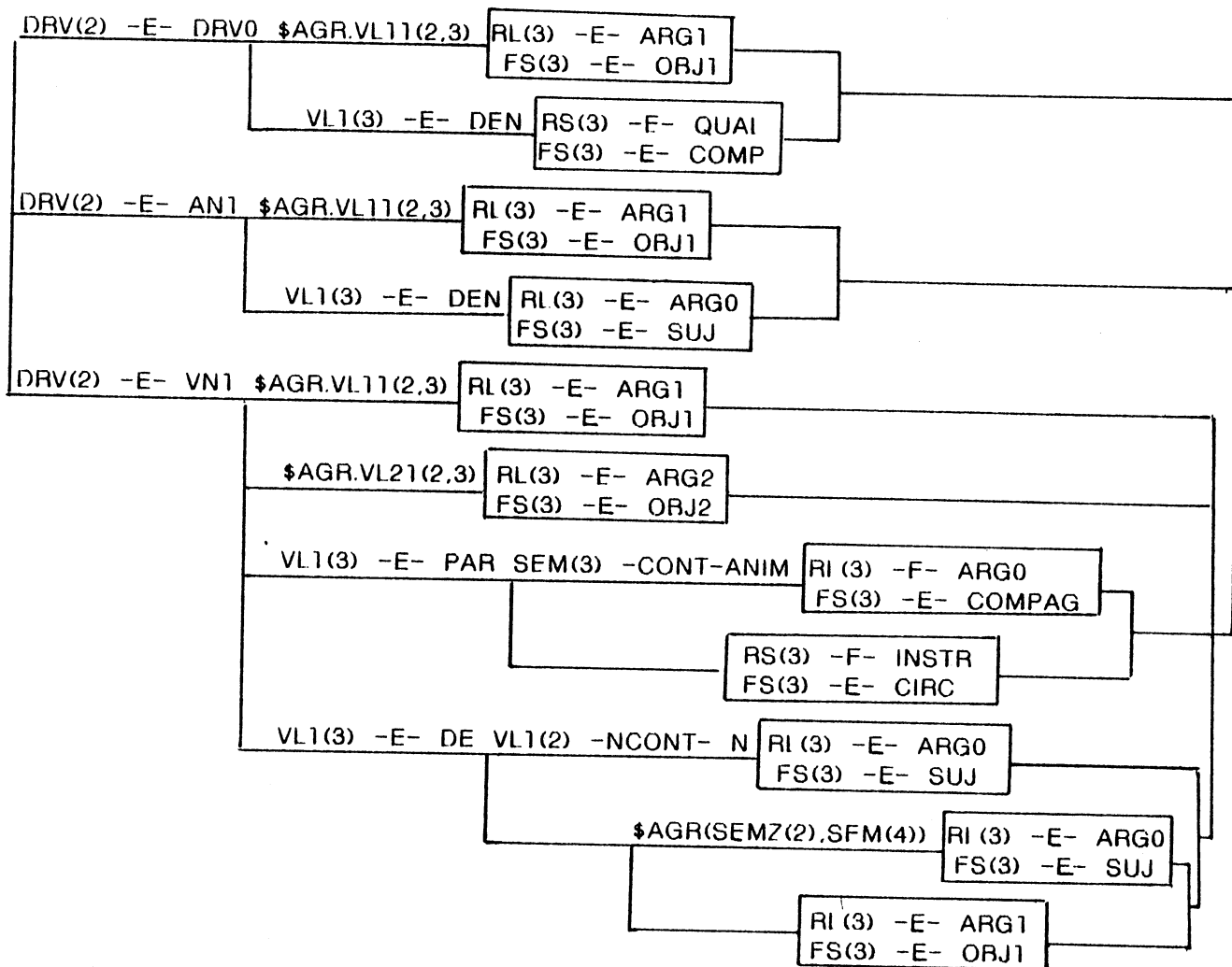
DRV(2) -E- VN1 -ET- VL1(2) -CONT- N -FT- VL1(3) -E- PAR

-OU-

VL1(3) -E- DEN

*com: la valence de rectlon du gouverneur du groupe référé 2 est satisfaite par le groupe nominal référé 3, ou le gouverneur de 2 est un nom d'action dérivé d'un verbe transitif et le groupe nominal 3 est régi par la préposition PAR, ou le groupe nominal 3 est régi par la préposition de.

ZONE 3



*COM: états des relations logiques et sémantiques et des fonctions syntaxiques sur le groupes nominal complexe en fonctions des attributs portés par les sommets des groupes référés.

2.2.7. GENERALISATION

2.2.7.1. UTILISATION DE CE FORMALISME POUR ECRIRE DES GRAMMAIRES STATIQUES EN MORPHOLOGIE.

La grammaire statique morphologique décrit la composition des mots. Elle correspond au choix des unités élémentaires des mots, telles que préfixes, bases, suffixes, désinences, ainsi qu'à la structure des mots composés (ULMCP), faisant apparaître ou non des symboles de liaison entre leurs composantes.

Dans cette grammaire statique, il s'agit de répertorier les chaînes d'unités élémentaires et/ou d'unités déjà construites et de donner la structure arborescente associée, qui est la structure d'une "ULOCC". La racine porte l'étiquette "ULOCC", dominant soit une "ULMCP" (si l'on a affaire à un mot composé), soit une feuille. Les planches statiques décrites seront donc du même type que celle définies pour la structure, mais elles seront plus simples, puisqu'elles se contenteront de distribuer les valeurs des attributs lexicaux qui se trouvent dans les dictionnaires en fonction de la segmentation choisie.

Exemples de planches statiques morphologiques du français.

Nous choisirons nos exemples dans une grammaire statique expérimentale du français.

Exemple 1.

La planche statique morphologique présentée dans cet exemple traite des mots ayant une base nominale, tels que noms, dérivés adjectivaux de noms, dérivés adverbiaux de noms.

-le mot "nationale" correspondra à la chaîne:

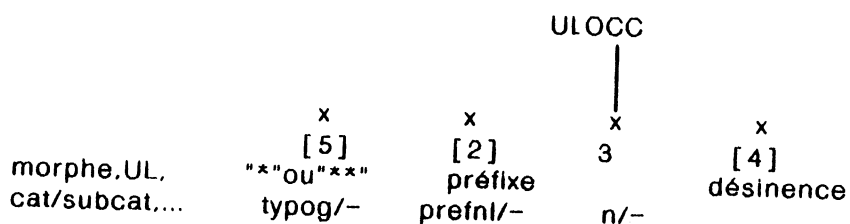
	x	x
	3	4
nation		ale

où CAT(3) -E- N, SUBN(3) -E- NC, POTD (3) -E- {NA, NAM}
(la potentialité de dérivation de 3 est nom-ad] ou nom-adv)
DRV(4) -E- NA, CAT(4) -E- A, SUBA(4) -E- ADJ

Langue décrite: français
 Grammaire statique: M.F.EXP
 Planche n°: MF01
 TYPE: BASE NOMINALE
 Cas traité: nom, adjectif dérivé, adverbe dérivé.
 Planches référées: aucune

Ftable le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(2)	dans:
le:(3)	dans:

ZONE I



ZONE II

\$NON-FORME(3) -IMP- 4
 com: la désinence est obligatoire si la base n'est pas une forme.
 \$AGRD(3,4)
 com: agrément en système de désinence entre 2, 3, et 4.
 \$AGRPD(2,3,4)
 com: agrément en système de préfixe et désinence entre 2, 3 et 4.

ZONE III

UL(0) -E- U LOCC com: établissement de l'U LOCC.
 NBR(3) -E- NBR(4) com: le nombre vient de la désinence.
 TYPOG(3) -E- TYPOG(1) com: la typographie viendra de 1 (si présent).
 NEG(3) -E- NEG(2) com: la négation vient du préfixe de négation (non-lexicalisé).

DRV(4) -NE- DRV0

DRV(3) -E- DRV(4)
 GNR(3) -E- GNR(4)
 CAT(3) -E- CAT(4)
 \$SUBCAT(3) -E- \$SUBCAT(4)

¬\$GNRIM(3) -ET- 4

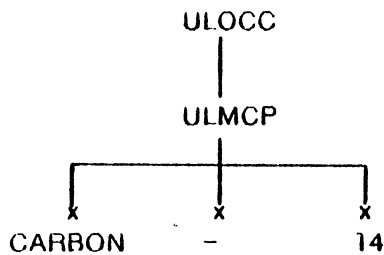
GNR(3) -E- GNR(4)

com: si le genre de 3 n'est pas immuable, il prend celui de 4.

Exemple 2.

Cet exemple traite des mots composés "nom-nom" ou "nom-cardinal", où le cardinal est écrit en chiffres.

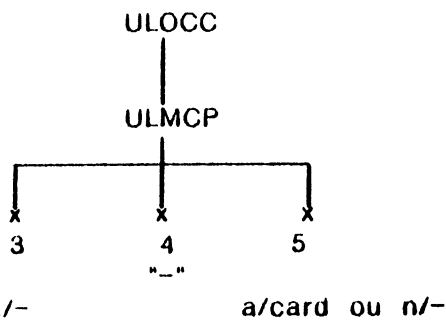
Par exemple le mot "Carbone-14", correspond à la chaîne et à la structure:



Langue décrite: français
 Grammaire statique: M.F.EXP
 Planche n°: MF05
 TYPE: MOTS COMPOSES
 Cas traité: nom + nom, nom + card
 Planches référées: MF01, MF02, MF03, MF04

Etablie le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

CAT(5) -E- CARD -IMP- TN(5) -E- chiffre
 com: les cardinaux considérés sont écrits en chiffres.

ZONE III

2.2.7.2. UTILISATION DE CE FORMALISME POUR ECRIRE DES GRAMMAIRES STATIQUES EN TRANSFERT.

Nous avons vu que les planches relatives au transfert devaient faire apparaître l'aspect contrastif entre la langue source et la langue cible en termes de structures et d'étiquettes concernant soit les niveaux d'interprétation superficiels, soit l'actualisation des énoncés.

Chaque planche est une correspondance de schémas entre des structures intermédiaires partielles jugées équivalentes. Le formalisme d'écriture de ces planches se borne donc à l'écriture de deux schémas arborescents avec les correspondance d'étiquettes. La grammaire statique de transfert est donc une étude contrastive entre deux langues. Une planche de grammaire statique sera constituée de trois zones: une zone schéma pour la langue source; une zone schéma correspondant dans la langue cible; un zone correspondance d'étiquettes.

CHAPITRE III

CHOIX ET REPRESENTATION D'UN MODELE

"TEXTE \longleftrightarrow INTERPRETATION"

TROISIEME PARTIE

APPLICATION

EXTRAIT D'UNE GRAMMAIRE STATIQUE EXPERIMENTALE

DU FRANCAIS



3. EXTRAITS D'UNE GRAMMAIRE STATIQUE EXPERIMENTALE DU FRANCAIS.

Nous n'avons pas l'ambition de présenter ici un modèle exhaustif de description de la langue française, nous voulons seulement montrer comment notre formalisme permettra cette description en présentant quelques exemples de planches statiques choisies pour leur représentativité des divers problèmes qui se poseront au linguiste dans ce travail.

Nous donnerons un court extrait des attributs lexicaux utilisés dans ces planches statiques établies à titre d'exemple, et la hiérarchie respectée dans leur élaboration.

3.1. ATTRIBUTS LEXICAUX.

Nous avons vu qu'il était important d'établir la liste des attributs susceptibles d'entrer dans la décoration des noeuds des structures décrites dans nos planches statiques. Parmi ces attributs, certains nous intéressent tout particulièrement. Ce sont ceux liés aux différents niveaux d'interprétation. Nous avons vu que les attributs représentatifs des relations logiques et sémantiques sont des attributs universels, c'est-à-dire qu'ils sont communs à toutes les langues et sont établis une fois pour toutes. Tous les autres attributs sont des attributs propres à une langue donnée.

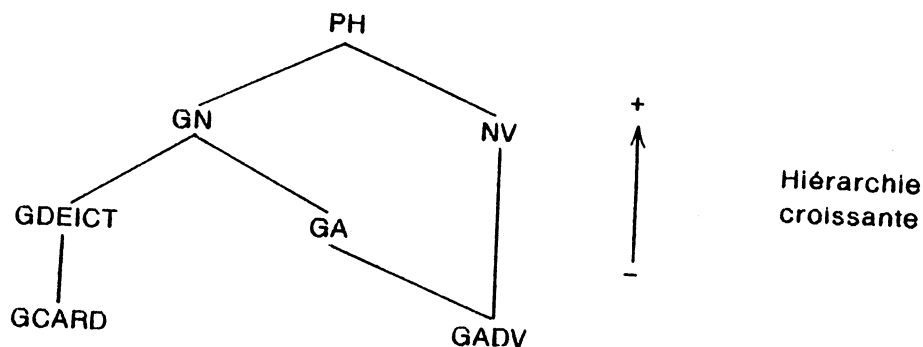
On trouvera en annexe (annexe 1), l'extrait qui concerne la langue française dans l'étude expérimentale que nous ferons.

3.2. HIERARCHIE ADOPTEE.

Les classes syntagmatiques établies pour cette analyse expérimentale du français sont les suivantes:

- GA groupe agjectival
- GN groupe nominal
- GADV groupe adverbial
- GCARD groupe cardinal
- GDEICT groupe déictique
- NV noyau verbal
- PHINF phrase infinitive
- PHVB phrase verbale
- PHPART phrase participiale
- PHREL phrase relative
- PHSUB phrase subordonnée

Sur ces classes syntagmatiques, on adoptera la hiérarchie suivante:



Les références à des groupes déjà décrits à l'intérieur des groupes simples ne pourront se faire que selon cette hiérarchie. Par exemple, un GNsimple ne pourra référer qu'à un GAsimple, à un GDEICT, à un GADV ou à un GCARD.

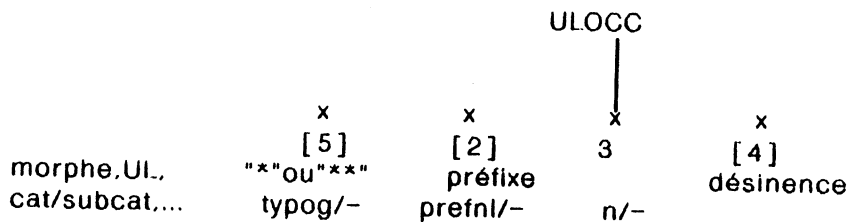
Rappelons que l'on décrira d'abord les groupes élémentaires (nous les choisirons sans référence), puis les groupes simples, selon la présente hiérarchie, enfin les groupes complexes, de façon récursive.

3.3 QUELQUES PLANCHES MORPHOLOGIQUES.

Langue décrite: français
 Grammaire statique: M.F.EXP
 Planche n°: MF01
 TYPE: BASE NOMINALE
 Cas traité: nom, adjectif dérivé, adverbe dérivé.
 Planches référées: aucune

Ftable le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(2)	dans:
le:(3)	dans:

ZONE I



ZONE II

\$NON-FORME(3) -IMP- 4
 com: la désinence est obligatoire si la base n'est pas une forme.
 \$AGRD(3,4)
 com: agrément en système de désinence entre 2, 3, et 4.
 \$AGRPD(2,3,4)
 com: agrément en système de préfixe et désinence entre 2, 3 et 4.

ZONE III

UL(0) -E- ULOCC com: établissement de l'ULOCC.
 NBR(3) -E- NBR(4) com: le nombre vient de la désinence.
 TYPOG(3) -E- TYPOG(1) com: la typographie viendra de 1 (si présent).
 NEG(3) -E- NEG(2) com: la négation vient du préfixe de négation (non-lexicalisé).

DRV(4) -NE- DRV0

DRV(3) -E- DRV(4)
 GNR(3) -E- GNR(4)
 CAT(3) -E- CAT(4)
 \$SUBCAT(3) -E- \$SUBCAT(4)

→\$GNRIM(3) -ET- 4

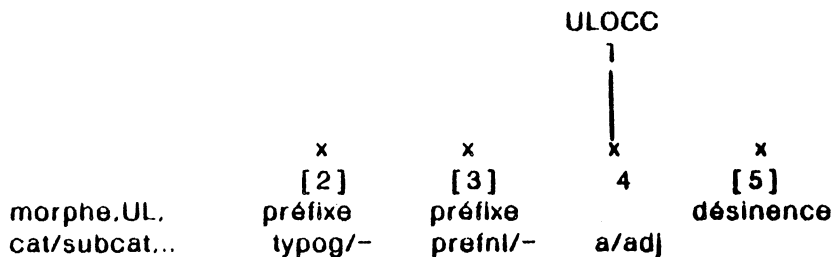
GNR(3) -E- GNR(4)

com: si le genre de 3 n'est pas Immuable, il prend celui de 4.

Langue décrite: français
 Grammaire statique: M.F.EXP
 Planche n°: MF02
 TYPE: BASE ADJECTIVALE
 Cas traité: adjectif préfixé, adverbe dérivé et nom dérivé
 Planches référées: aucune

Etablie le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

§NON-FORME(4) -IMP- 5
 com: la désinence est obligatoire si la base n'est pas une forme.
 §AGRPD(3,4,5)
 com: agrément en système de préfixe et désinence entre 3, 4 et 5.
 § AGRD (4,5)
 com: agrément en système de désinence entre 4 et 5.

ZONE III

DRV(4) -E- DRV(5)
 TYPOG(4) -E- TYPOG(2)
 GNR(4) -E- GNR(5)
 NBR(4) -E- NBR(5)
 NEG(4) -E- NEG(3)
 com: les valeurs de dérivation, genre et nombre proviennent de la désinence, la typographie de 2, et la valeur de négation du préfixe non lexicalisé de 3.

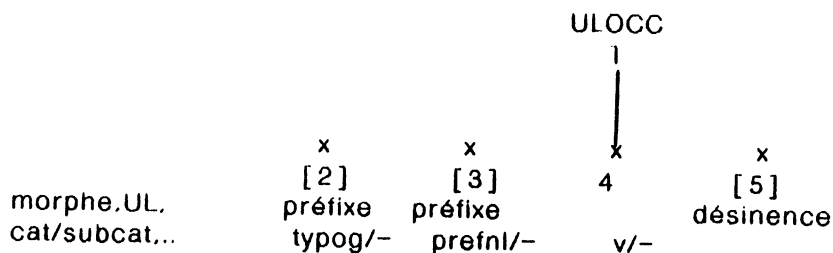
DRV(5) -NE- DRV0

CAT(4) -E- CAT(5)
 §SUBCAT(4) -E- §SUBCAT(5)

Langue décrite: français
 Grammaire statique: M.F.EXP
 Planche n°: MF03
 TYPE: BASE VERBALE
 Cas traité: formes régulières des verbes conjugués, formes participiales et dérivations du verbe.
 Planches référées: aucune

Etablie le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

\$NON-FORME(4) -IMP- 5
 \$AGRPD(3,4,5)
 \$AGRD(4,5)

ZONE III

DRV(4) -E- DRV(5)
 NBR(4) -E- NBR(5)
 NFG(4) -E- NEG(3)
 TYPOG(4) -E- TYPOG(2)
 SUBV(4) -E- SUBV(5)
 TEMPS(4) -E- TEMPS(5)
 PFRS(4) -E- PERS(5)

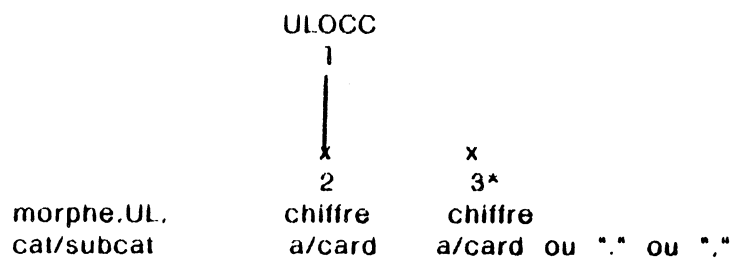
GNR(5) -NE- GNR0

GNR(4) -E- GNR(5)

Langue décrite: français
Grammaire statique: M.F.EXP
Planche n°: MF04
TYPE: CARDINAUX
Cas traité: cardinaux en chiffres
Planches référées: aucune

Établie le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

TYPOG(2) -E- digit
TYPOG(2) -E- digit

ZONE III

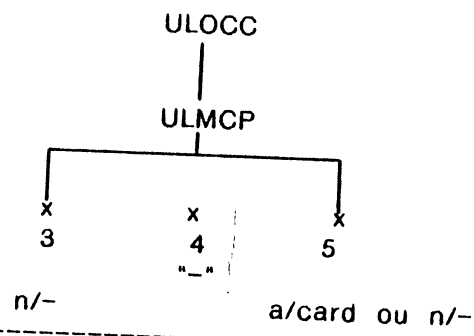
3

UL(2) -E- forme complète
NBR(2) -E- PLU

Langue décrite: français
Grammaire statique: M.F.EXP
Planche n°: MF05
TYPE: MOTS COMPOSES
Cas traité: nom + nom, nom + card
Planches référées: MF01, MF02, MF03, MF04

Etablie le:		par:
Modifiée	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



morphe.UL,
cat/subcat...

ZONE II

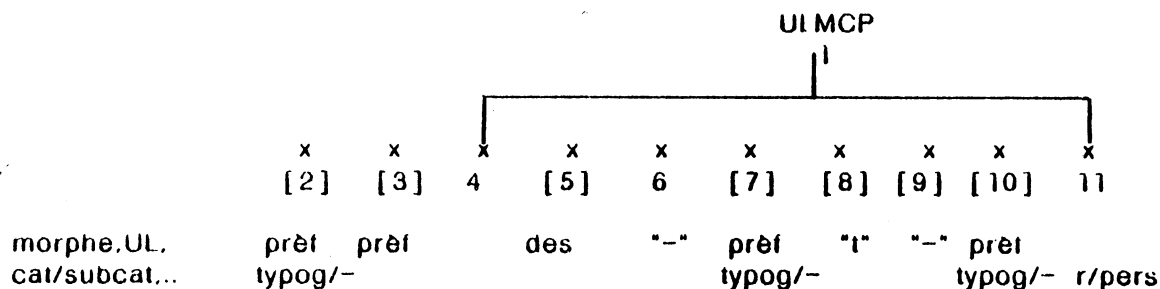
CAT(5) -E- CARD -IMP- TN(5) -E- chiffre
com: les cardinaux considérés sont écrits en chiffres.

ZONE III

Langue décrite: français
 Grammaire statique: M.F.EXP
 Planche n°: MF06
 TYPE: VERBE INTERROGATIF
 Cas traité: va-t-il, peut-il, a-t-il
 Planches référées: aucune

Etablie le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(1)	dans:
le:(3)	dans:

ZONE I



ZONE II

\$NON-FORME (4) -IMP- 5
 \$INV(4) -IMP- -5
 (DES(5) -E- "e" -OU- "a") -OU- (DES(5) -E- DES0 -ET- \$term-en-e-ou-a)
 -SSI- 8 -ET- 9
 com: si le verbe se termine par un "a" ou un "e" l'insertion de "-I-" est obligatoire.
 7 -IMP- 2 -ET- 10 -ET- CAP(2) -E- TCAP -ET- CAP(10) -E- TCAP
 com: remarque: si on a un "T", tout doit être en majuscules.
 (5 -ET- \$AGRNUM(5,11)) -OU- (~7 -ET- \$AGRNUM(4,11))
 com: agrément entre pronom et verbe en nombre.
 CAP(7) -ET- CAP(2) -ET- CAP(10) -NE- CAP0
 com: les préfixes 2, 7 et 10 sont de capitales.
 DES(5) -E- CONJ
 com: 5 est une désinence de conjugaison de verbe.
 \$AGRD(4,5)
 \$AGRPD(3,4,5)
 UL(3) -E- UL0

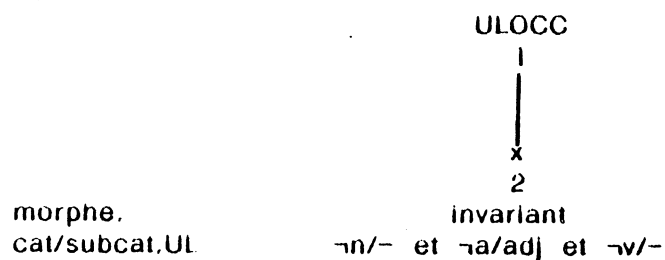
ZONE III

TVB(6) -E- INT
com: interrogatif.
NBR(4) -E- NBR(5)
NEG(4) -E- NEG(3)
TM(4) -E- TM(5)
CAP(4) -E- CAP(2)
CAP(8) -E- CAP(7)
CAP(11) -E- CAP(10)

Langue décrite: français
Grammaire statique: M.F.EXP
Planche n°: MF07
TYPE: INVARIANT
Cas traité: autre que nom, adjectif et verbe.
Planches référées: aucune

Etablie le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

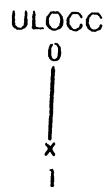
(CAT(2) -NE-N) -ET- (CAT(2) -NE- V)
-ET- (CAT(2) -NE- A -OU- SUBA(2) -NE- ADJ)

ZONE III

Langue décrite: français
 Grammaire statique: M.F.EXP
 Planche n°: MF08(introduction)
 TYPE: CARDINAL
 Cas traité: cardinal simple (alpha)
 Planches référées: aucune

Etablie le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(1)	dans:
le:(3)	dans:

ZONE I



morphe.
 cat/subcat

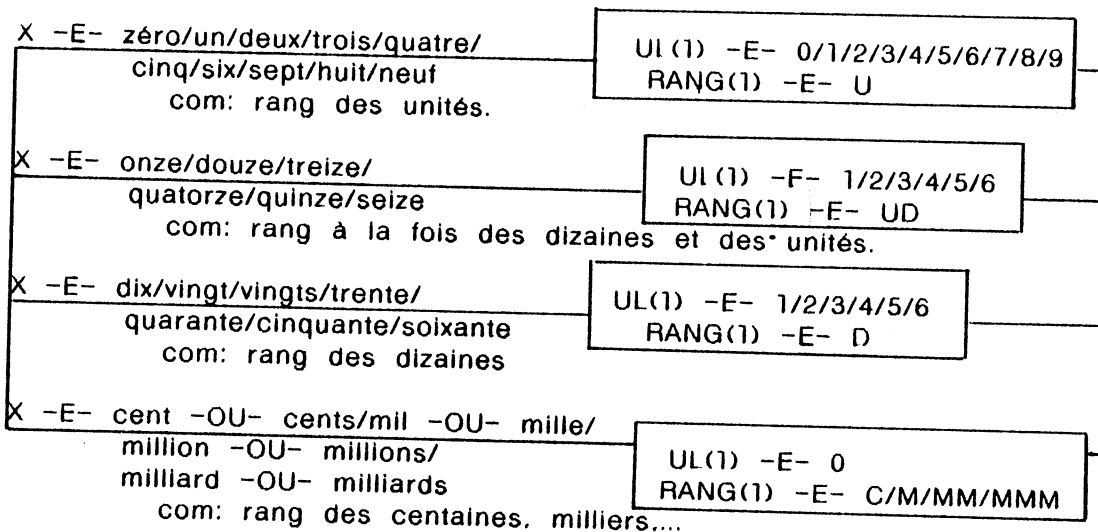
a/card

ZONE II

TYPOG(1) -NE- digit
 com: 1 est un cardinal écrit en lettres.

ZONE III

X=\$CHAINE(1)



com: ces renseignements sont fournis par les dictionnaires.

Langue décrite: français

Grammaire statique: M.F.EXP

Planche n^o: MF09

TYPF: CARDINAL COMPOSE

Cas traité: quatre-vingts, quatre-vingt-dix, soixante-dix, dix-huit, vingt-sept, quatre-vingt-dix-neuf, etc...

Planches référées: aucune

Fiablie le:

par:

Modifiée

le:(1)

par:

le:(2)

par:

le:(3)

par:

Intégrée

le:(1)

dans:

le:(1)

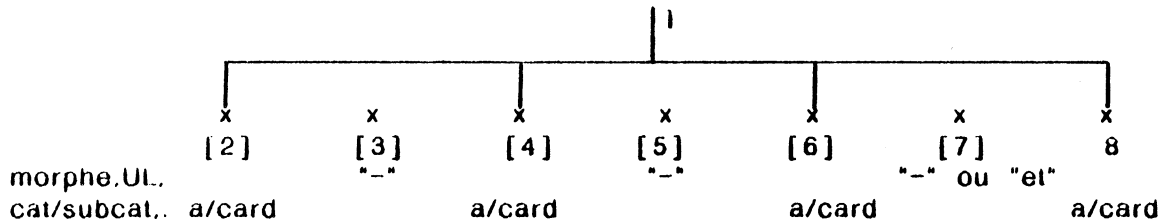
dans:

le:(3)

dans:

ZONE I

ULMCP



ZONE II:

2 -IMP- 3

3 -IMP- 4

4 -IMP- 5

5 -IMP- 6

6 -IMP- 7

com: cadrage à gauche.

\$ULRG(2) -F- 4,U

\$ULRG(4) -E- 4,U -OU- 2,D -OU- 6,D

\$ULRG(6) -E- 4,U -OU- 1,D -OU- 2,D -OU- 6,D

\$ULRG(8) -E- (1 -OU- 2 -OU- 3 -OU- 4 -OU- 5 -OU- 6).(U -OU- UD)

-OU- (7 -OU- 8 -OU- 9),U

-OU- (1 -OU- 2),D

com: la procédure \$ULRG donne l'unité et le rang du cardinal suivant les conventions exprimées dans la planche MF08.

2 -IMP- \$ULRG(4) -E- 2,D

-ET- \$ULRG(6) -E- 1,D

-ET- \$ULRG(8) -E- (7 -OU- 8 -OU- 9),U

(-2 -ET- 4) -IMP- [\$ULRG(4) -E- 4,U
-ET- \$ULRG(6) -E- 2,D
-ET- (\$ULRG(8) -E- (1 -OU- 2 -OU- 3 -OU- 4
-OU- 5 -OU- 6),(U -OU- UD)
-OU- (7 -OU- 8 -OU- 9),U
-OU- (1 -OU- 2),D)]
-OU- [\$ULRG(4) -E- 6,D
-ET- \$ULRG(6) -E- 1,D
-ET- \$ULRG(8) -E- (7 -OU- 8 -OU- 9),U]

(-4 -ET- 6) -IMP- [\$ULRG(6) -E- 4,U
-ET- \$ULRG(6) -E- 2,D]
-OU- [\$ULRG(6) -E- 1,D
-ET- \$ULRG(8) -E- (7 -OU- 8 -OU- 9),U]
-OU- [\$ULRG(6) -E- 6,D
-ET- \$ULRG(8) -E- 1,D]
-OU- [\$ULRG(6) -E- (6 -OU- 3 -OU- 4 -OU- 5 -OU- 6),D
-ET- \$ULRG(8) -E- (2 -OU- 3 -OU- 4 -OU- 5
-OU- 6 -OU- 7 -OU- 8 -OU- 9),U]

com: traitement de la compatibilité des éléments de chaîne.

ZONE III

Langue décrite: français
Grammaire statique: M.F.EXP
Planche n°: MF10
TYPE: ORDINAL
Cas traité: adjectif et adverbe.
Planches référées: MF08, MF09

Ftable le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I

	U LOCC	
	x	x
	2	3
morphe,		désinence
cat/subcat...	a/card	

ZONE II

com: la base cardinale en 2 ne doit pas être terminée par un "e".

ZONE III

NBR(2) -E- NBR(3)

CAT(2) -E- CAT(3)

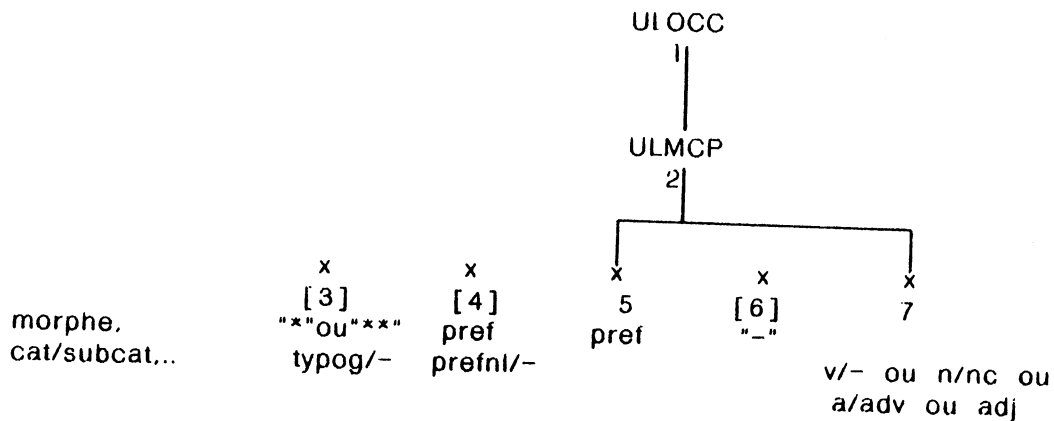
\$(SUBCAT(2) -E- \$(SUBCAT(3)

com: nombre et catégorie proviennent de la désinence.

Langue décrite: français
 Grammaire statique: M.F.EXP
 Planche n°: MF11
 TYPE: PREFIXAGE
 Cas traité: trois types de préfixes sont envisagés
 Planches référées: MF01, MF02, MF03

Ftable le:		par:
Modifiée	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

UL(5) -NE- UL0
 TYPOG(3) -NE- TYPOG0
 CAT(4) -E- PREFNL

ZONE III

TYPOG(5) -E- TYPOG(3)
 NEG(5) -E- NEG(4)

Procédures utilisées

\$NON-FORME(x) vérifie que x n'est pas une forme.

\$AGRDES(x,y) vérifie l'agrément en système de désinence entre la racine x et la désinence y.

\$AGRPD(x,y,z) vérifie l'agrément en système de préfixe et désinence entre le préfixe x, la racine y et sa désinence z.

\$GNRIM(x) vérifie que 3 est de genre immuable.

\$INV(x) vérifie que 4 est invariant.

\$AGRNUM(x,y) agrément en nombre entre x et y.

\$AGRGN(x,y) agrément en genre entre x et y.

\$ULRG(2) fournit: "valeur d'Unité Lexicale" "." "valeur de rang".

\$SUBCAT(x) donne la sous-catégorie correspondant à la catégorie de x.

3.4. QUELQUES PLANCHES STRUCTURALES.

Langue décrite: français
Grammaire statique: S.F.EXP
Planche n°: SF01

TYPE: GROUPE ADVERBIAL élémentaire

Cas traité: adverbe simple
Planches référées: aucune

Etablie le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(1)	dans:
le:(3)	dans:

ZONE I

	GADV
	0 ^e
	x
	1
cat/subcat	a/adv
SF	GOV

ZONE II

ZONE III

VAR(0) -E- VAR(1)
DEG(0) -E- DEG(1)

Langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: SF02

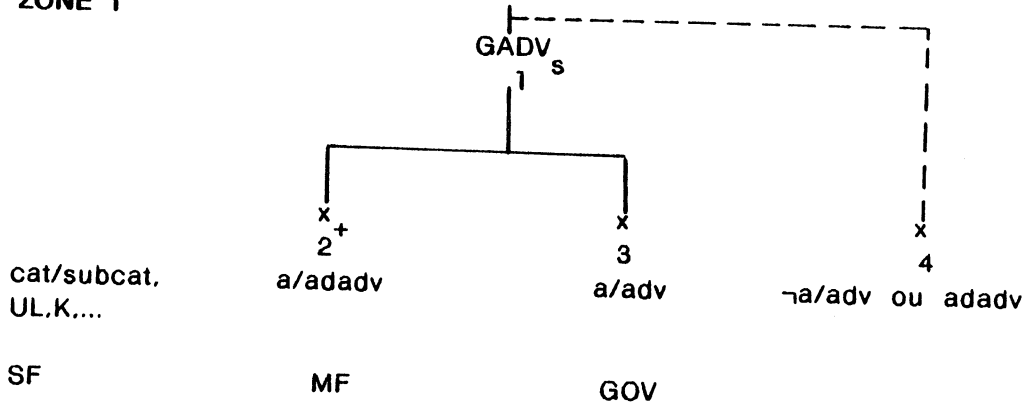
TYPE: GROUPE ADVERVIAL simple

Cas traité: séquence d'adverbes (au moins deux)

Planches référées: aucune

Etablie le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

NEG(2) -E- NEGS0

com: on ne veut pas de négation syntaxique: jamais, point, pas.

DEG(3) -NE- DEG0 -IMP- DEG(2) -E- DEG0

ZONE III

VAR(1) -E- VAR(3)

K(1) -E- GADV

FS(1) -E- MF

FS(2) -E- MF

FS(3) -E- GOV

SUBA(2) -E- ADADV

DEG(3) -E- DEG(0)

DEG(1) -E- DEG(2)

Langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: SF05

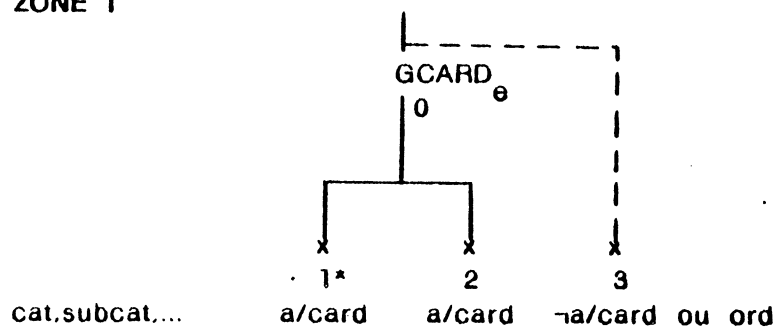
TYPE: GROUPE CARDINAL élémentaire.

Cas traité: construit à partir d'une succession de cardinaux

Planches référées: aucune.

Etablie le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



SF

GOV

ZONE II

ZONE III

VAR(0) -E- VAR(2)

SF(2) -E- GOV

Langue décrite: français
Grammaire statique: S.F.EXP
Planche n°: SF06

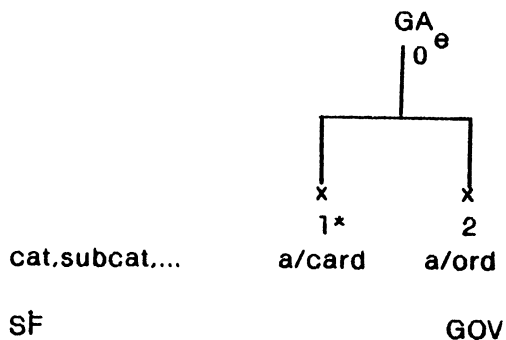
TYPE: GROUPE ADJECTIVAL élémentaire

Cas traité: GA construit à partir d'un ordinal

Planches référées: aucune.

Etablie le:		par:
Modifiée		
le:(1)		par:
le:(2)		par:
le:(3)		par:
Intégrée		
le:(1)		dans:
le:(1)		dans:
le:(3)		dans:

ZONE I



ZONE II

ZONE III

VAR(0) -E- VAR(2)
SF(2) -E- GOV

Langue décrite: français

Grammaire statique: S.F.EXP

Planche n°: SF07

TYPE: GROUPE CARDINAL (ADJECTIVAL) simple

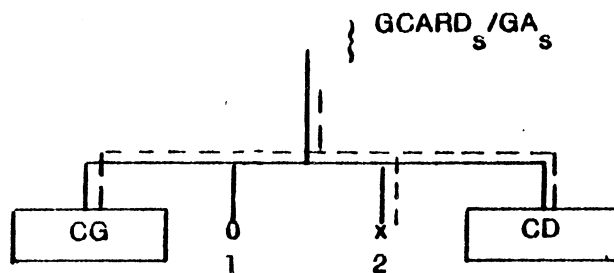
Cas traité: représentation d'un cardinal (ordinal) contracté

Planches référencées: (GCARDe) 5

(GAe) 6

Etablie le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(1)	dans:
le:(3)	dans:

ZONE I



cat/subcat.

a/card ou ord

ZONE II

RANG(2) -E- UD

com: traitement de onze, douze, seize.

ZONE III

UL(1) -E- 1

RANG(1) -E- D

RANG(2) -E- U

com: onze est décomposé en dix un, douze en dix deux, etc.

CAT(1) -E- A

‡SUBCAT(1) -E- CARD

Langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: SF08

TYPE: GROUPE CARDINAL (ADJECTIVAL) simple

Cas traité: représentation des "vigésimaux"

Planches référées: (GCARD-GA) 7

(GCARD) 5

(GA) 6

Etablie le:

par:

Modifiée

le:(1)

par:

le:(2)

par:

le:(3)

par:

Intégrée

le:(1)

dans:

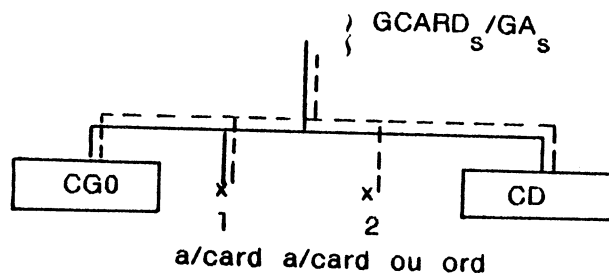
le:(1)

dans:

le:(3)

dans:

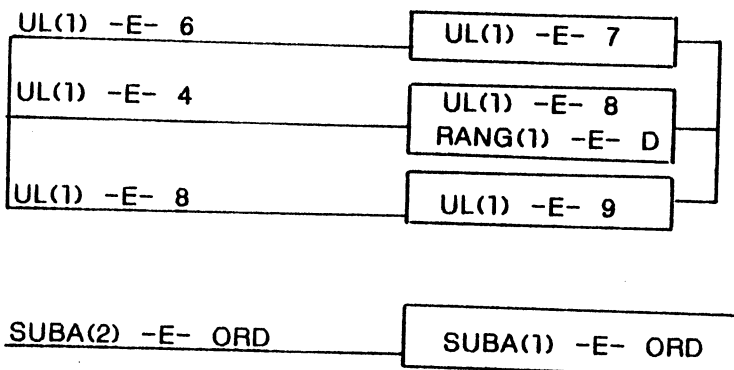
ZONE I



ZONE II

RANG(2) -E- D -ET- (RANG(1) -E- D -OU- UL(1) -E- 4)
 com: soixante-dix, quatre-vingts, quatre-vingt-dix.

ZONE III



Langue décrite: français
Grammaire statique: S.F.EXP
Planche n°: SF09

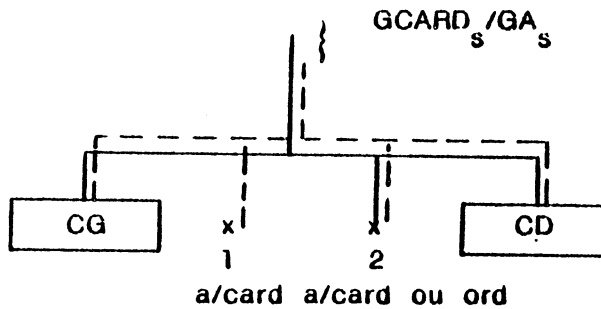
TYPE: GROUPE CARDINAL (ORDINAL) simple.

Cas traité: cardinal (C.M.MM.MMM) multiplié par un card (U)

Planches référées: (GCARD-GA) 7,8.
(GCARD_e) 5.
(GA_e) 6.

Etablie le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(1)	dans:
le:(3)	dans:

ZONE I



ZONE II

RANG(2) -E- C -OU- M -OU- MM -OU- MMM
RANG(1) -E- U
com: cas de trois cents, deux mille, etc...

ZONE III

UL(2) -E- UL(1)

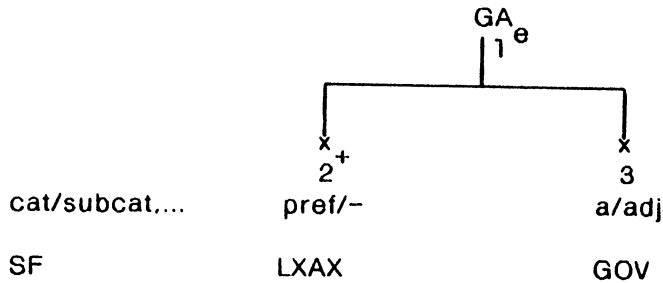
langue décrite: français
Grammaire statique: S.F.EXP
Planche n°: SF13

TYPE: GROUPE ADJECTIVAL élémentaire

Cas traité: préfixation des adjectifs
Planches référées: aucune

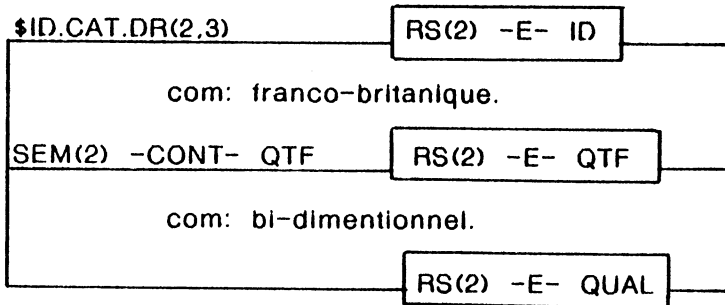
Ftable le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

ZONE III



Langue décrite: français
Grammaire statique: S.F.EXP
Planche n°: SF14

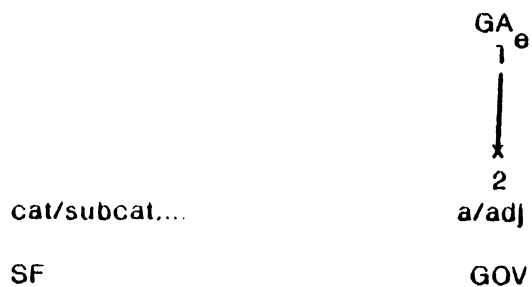
TYPE: GROUPE ADJECTIVAL élémentaire

Cat. traité: construit à partir d'un adjectif

Planches référées: aucune

Établie le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

ZONE III

K(1) -E- GA
VAR(1) -E- VAR(2)

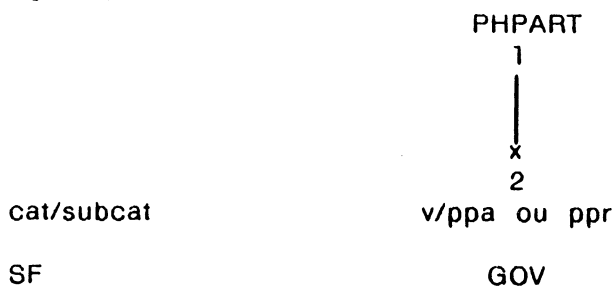
Langue décrite: français
Grammaire statique: S.F.EXP
Planche n°: SF15

TYPE: PARTICIPIALE élémentaire

Cas traité: participe seul
Planches référées: aucune

Ftable le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

ZONE III

K(1) -E- PHPART
VAR(1) -E- VAR(2)

Langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: SF16

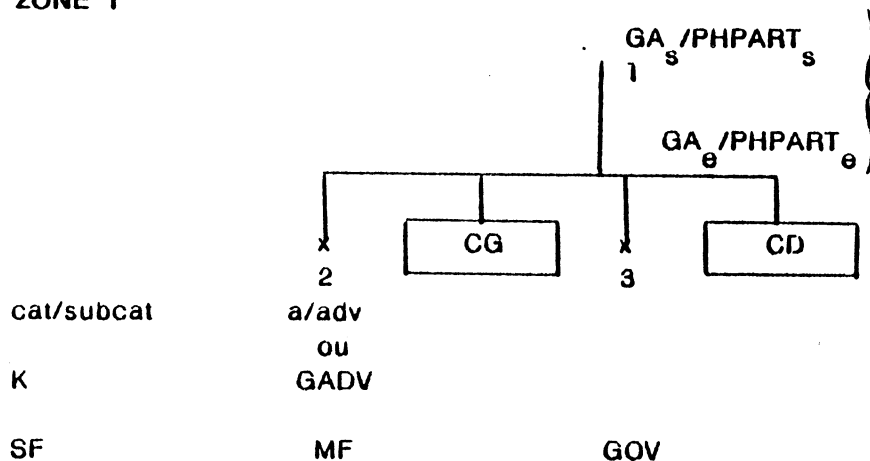
TYPE: GROUPE ADJECTIVAL (PARTICIPIAL) simple

Cas traité: construit à partir d'un groupe élémentaire et d'un groupe adverbial

Planches référées: (GA) 13.14.
 (GADV) 1.2.9.10.
 (PHPART) 15.

Etablie le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(1)	dans:
le:(3)	dans:

ZONE I



ZONE II

ZONE III

com: les attributs sont conservés de GA_e à GA_s (et pour PHPART).

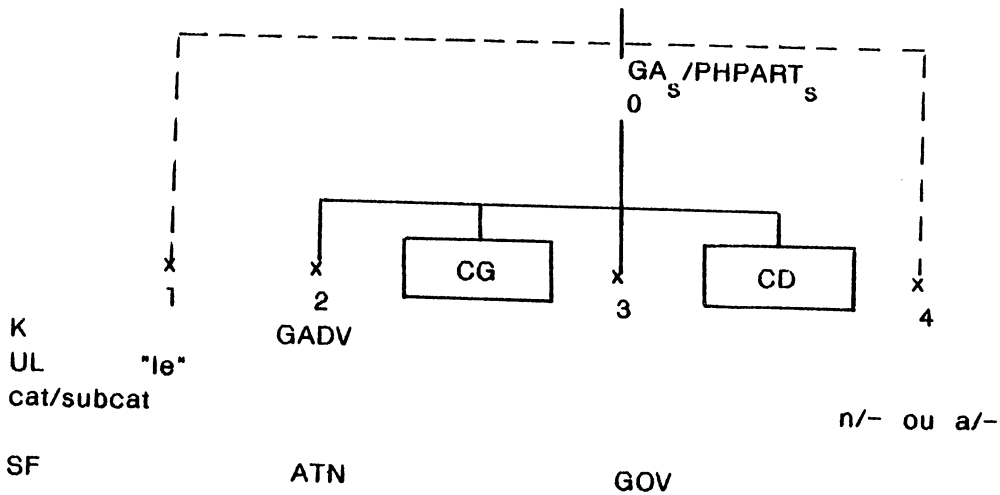
SF(2) -E- MF
 SF(3) -E- GOV

Langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: SF18

TYPE: GROUPE ADJECTIVAL (PARTICIPIAL) simple
 Cas traité: "le" + "plus" ("moins") + adj (part)
 Planches référées: (GA) 13,14,16.
 (GADV) 1,2,9,10.
 (PHPART) 15,16.

Etablie le:		par:
Modifiée	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

K(0) -E- GA OU PHPART
 DEG(2) -NE- DEG0

ZONE III

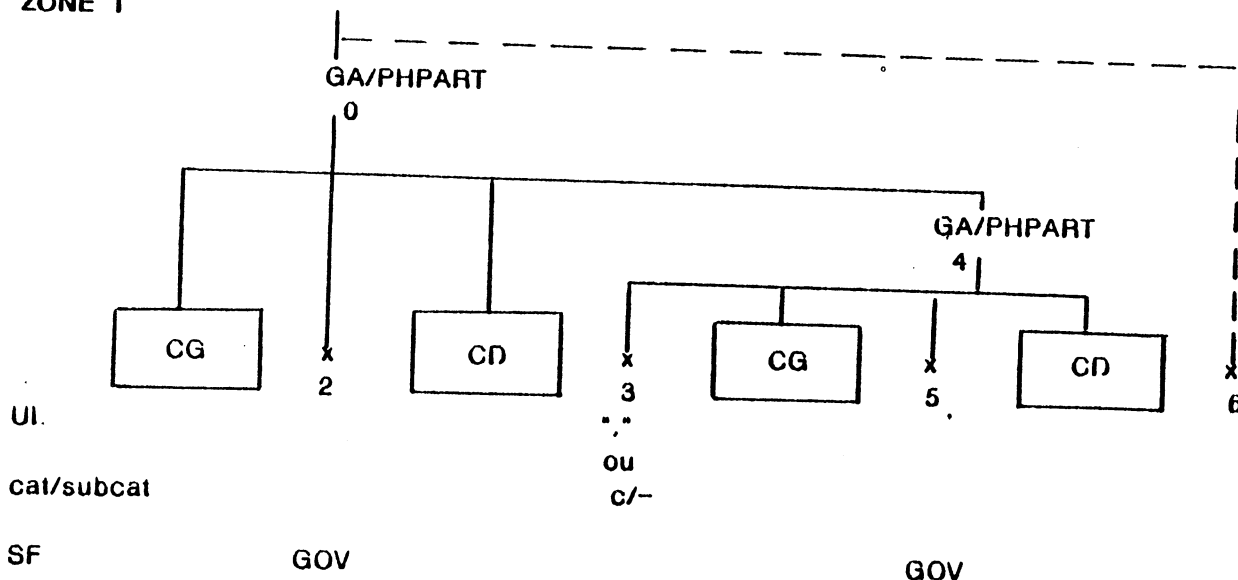
DEG(0) -E- SUP
 SF(2) -E- ATN

Langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: SF20

TYPE: GROUPE ADJECTIVAL (PARTICIPIAL) simple
 Cas traité: COORDINATION de groupes simples
 Planches référées: (GA) 1.2.6.7.8.9.13.14.16.18.20.
 (PHPART) 15.16.18.20.

Etablie le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(1)	dans:
le:(3)	dans:

ZONE I



ZONE II

K(0) -E- K(4)
 \$TSCOOF(1,4)
 com: test de coordination forte entre 1 et 4 (identité de K, CAT.SURCAT)
 (UL(6) -NE-*,*) -ET- (CAT(6) -NE-C).
 \$AGR.NB.GR(1,4)

ZONE III

SF(3) -E- COORD
 RL(3) -E- ID

Langue décrite: français
Grammaire statique: S.F.EXP
Planche n°: SF21

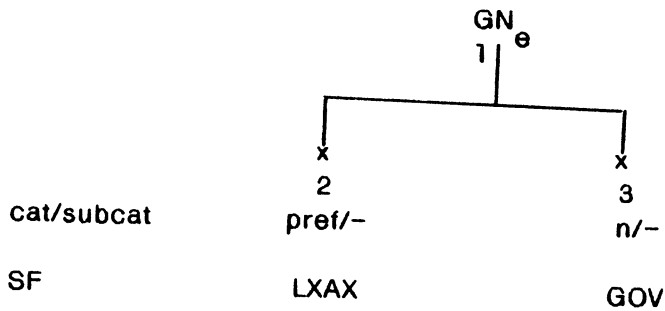
TYPE: GROUPE NOMINAL élémentaire

Cas traité: préfixage d'un nom

Planches référées: aucune

Etablie le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I

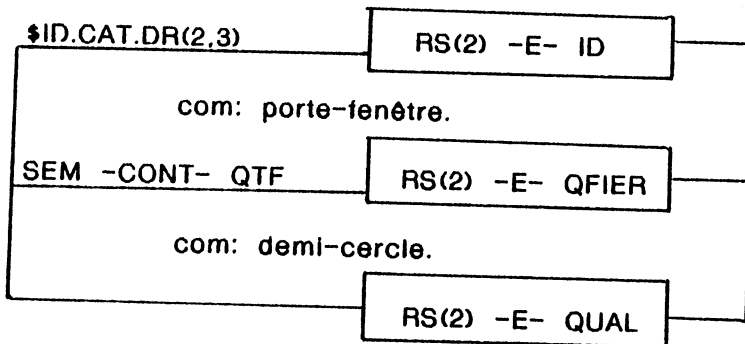


ZONE II

NEG(2) -E- NEG0
TYPOG(2) -E- TYPOG0

ZONE III

VAR(1) -E- VAR(3)



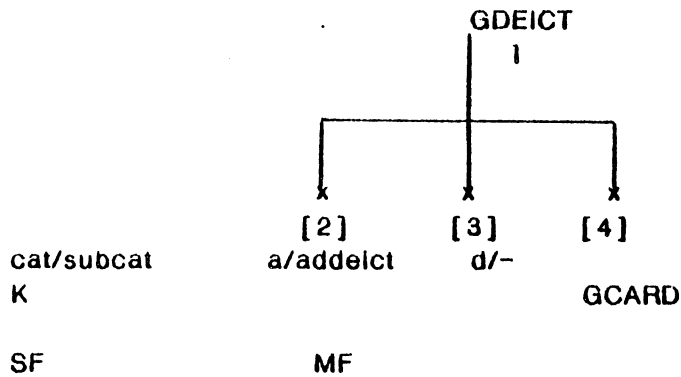
Langue décrite: français
Grammaire statique: S.F.EXP
Planche n°: ST23

TYPE: GROUPE DEICTIQUE

Cas traité:
Planches référées: (GCARD) 5,7,8,9.

Etablie le:		par:
Modifiée		
le:(1)		par:
le:(2)		par:
le:(3)		par:
Intégrée		
le:(1)		dans:
le:(1)		dans:
le:(3)		dans:

ZONE I

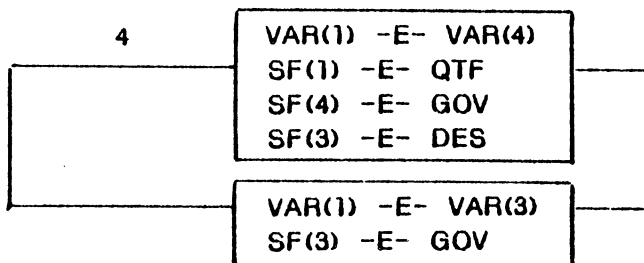


ZONE II

-3 -IMP- 4
2 -IMP- 3 -OU- 4
\$AGR.NB(3,4)

ZONE III

K(1) -E- GDEICT



DET(1) -E- DET(3)
GNR(1) -E- \$INTER.GN(3,4)
NBR(1) -E- \$INTER.NB(3,4)

Langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: SF25

TYPE: GROUPE NOMINAL simple.

Cas traité: gouverné par un nom commun

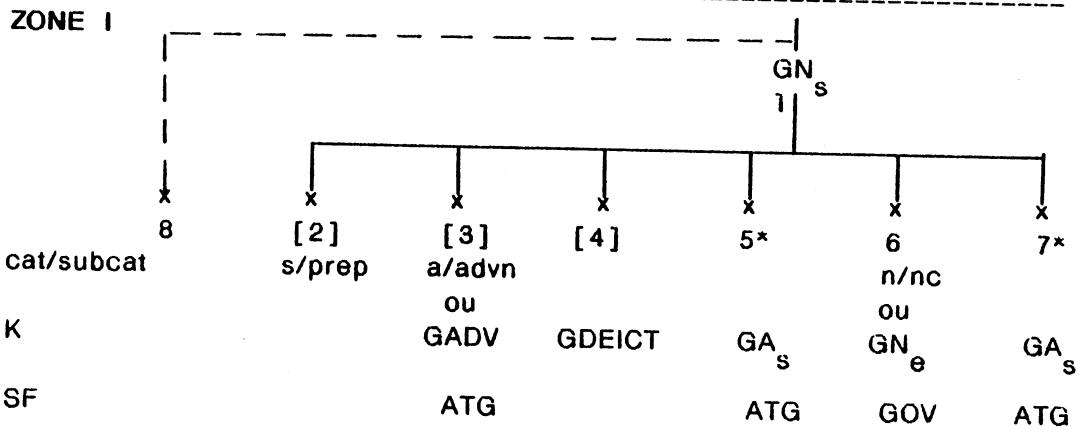
Planches référées: (GN_e) 21.

(GADV) 1,2,9,10.

(GDEICT) 23,24.

(GAs) 6 à 9,13,14,16 à 20.

Etablie le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:



ZONE II

8 -imp- (-2 -imp- 4)

3 -IMP- 4

\$AGR.GN.NB(4,5,6,7)

2 -IMP- VALETAT(1) -E- N

ZONE III

GNR(1) -E- \$INTERGN(4,5,6)

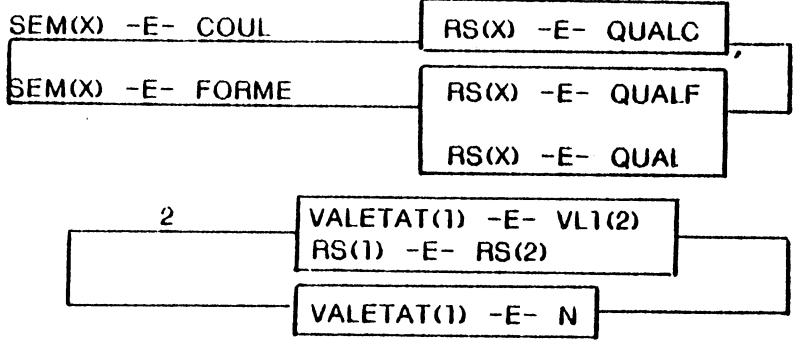
NBR(1) -E- \$INTERNB(4,5,6)

DET(1) -E- DET(4)

DRV(6) -NE-DRV0

VL1(1) -E- VL1(6)
VL2(1) -E- VL2(6)

X = 5 -OU- 7



langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: SF26

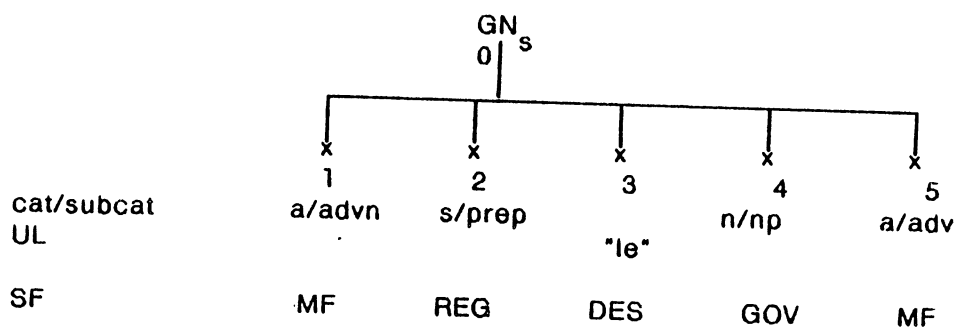
TYPE: GROUPE NOMINAL simple

Cas traité: gouverné par un nom propre

Planches référées: aucune

Fiable	le:	par:
Modifiée	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I

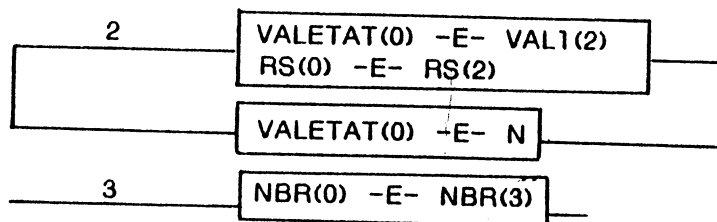


ZONE II

1 -IMP- -5
 \$AGR.GN.NB(3,4)

ZONE III

VAR(0) -E- VAR(4)
 K(0) -E- GN
 com: cat. subcat. sem, abst. concr. être. sexe. forme. gnr. ...



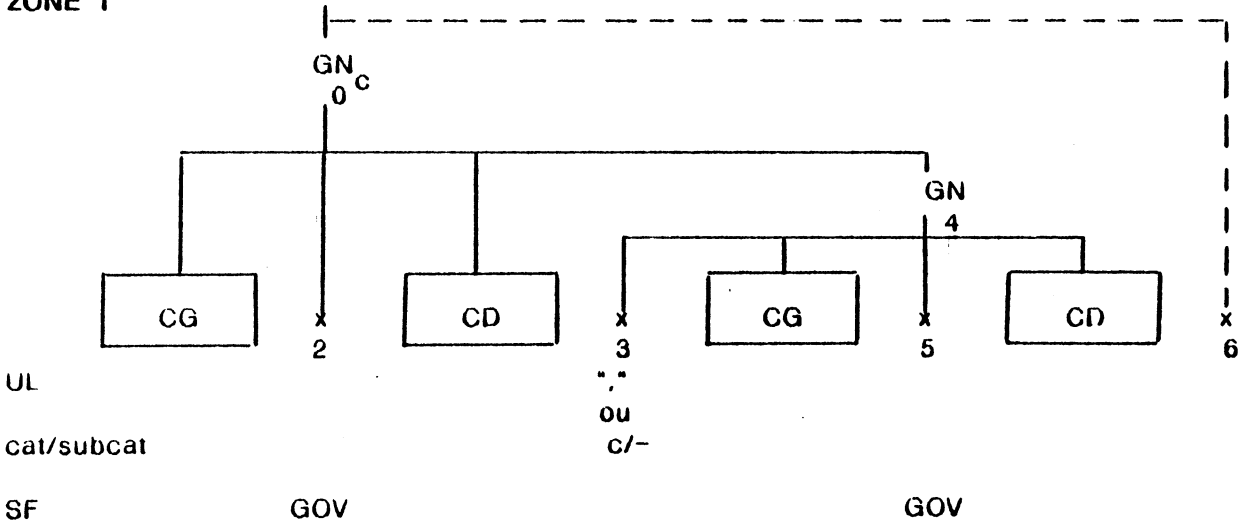
Langue décrite: français
Grammaire statique: S.F.EXP
Planche n°: ST30

TYPE: GROUPE NOMINAL complexe

Cas traité: coordination de GN
Planches référées: (GN^s) 25 à^s 29.
(GN^s_C) 30.

Etablie le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(1)	dans:
le:(3)	dans:

ZONE I



ZONE II

§TSCOOF(1,4)
com: test de coordination forte entre 1 et 4 (identité de K, CAT.SURCAT)
UL(6) -NE- "." -ET- CAT(6) -NE- C
com: 6 ne peut être coordonné à 4.
§TESTINCOMP(4,6)
com: 4 ne peut dominer 6.
§AGR.NB.GR(1,4)

ZONE III

SF(3) -E- COORD
RL(3) -E- ID

Langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: SF31

TYPE: GROUPE NOMINAL complexe

Cas traité: groupe nominal dominant un autre groupe nominal

Planches référées: (GN^s) 25 à 29.

(GN^s_C) 30,31.

Etablie

le: par:

Modifiée

le:(1) par:

le:(2) par:

le:(3) par:

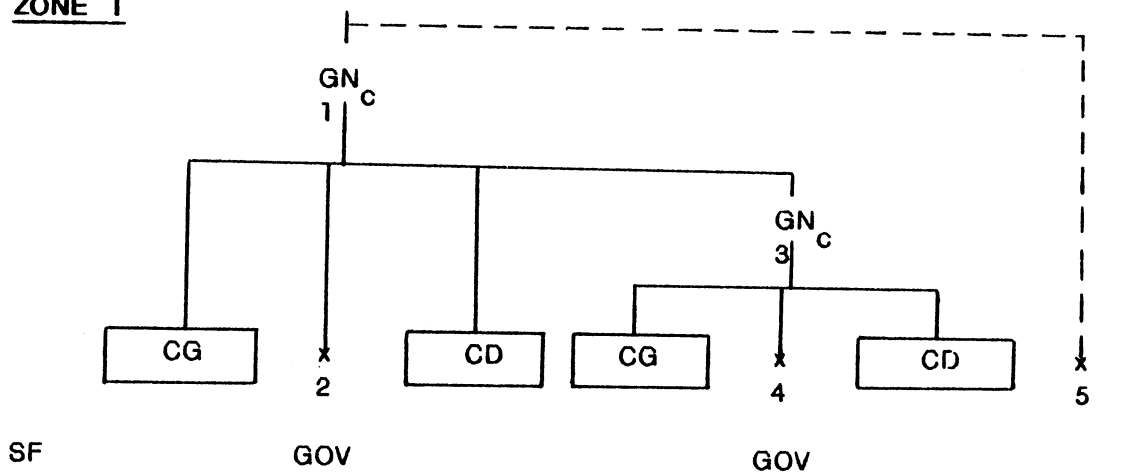
Intégrée

le:(1) par:

le:(2) par:

le:(3) par:

ZONE 1



ZONE 2

K(5) -NE- (GN -OU- GA -OU- PHPART)
 -OU-

\$TESTINCOMP(3,5)

com: 3 ne peut dominer 5.

\$AGR.VL11(2,3) -OU- \$AGR.VL21(2,3)

-OU-

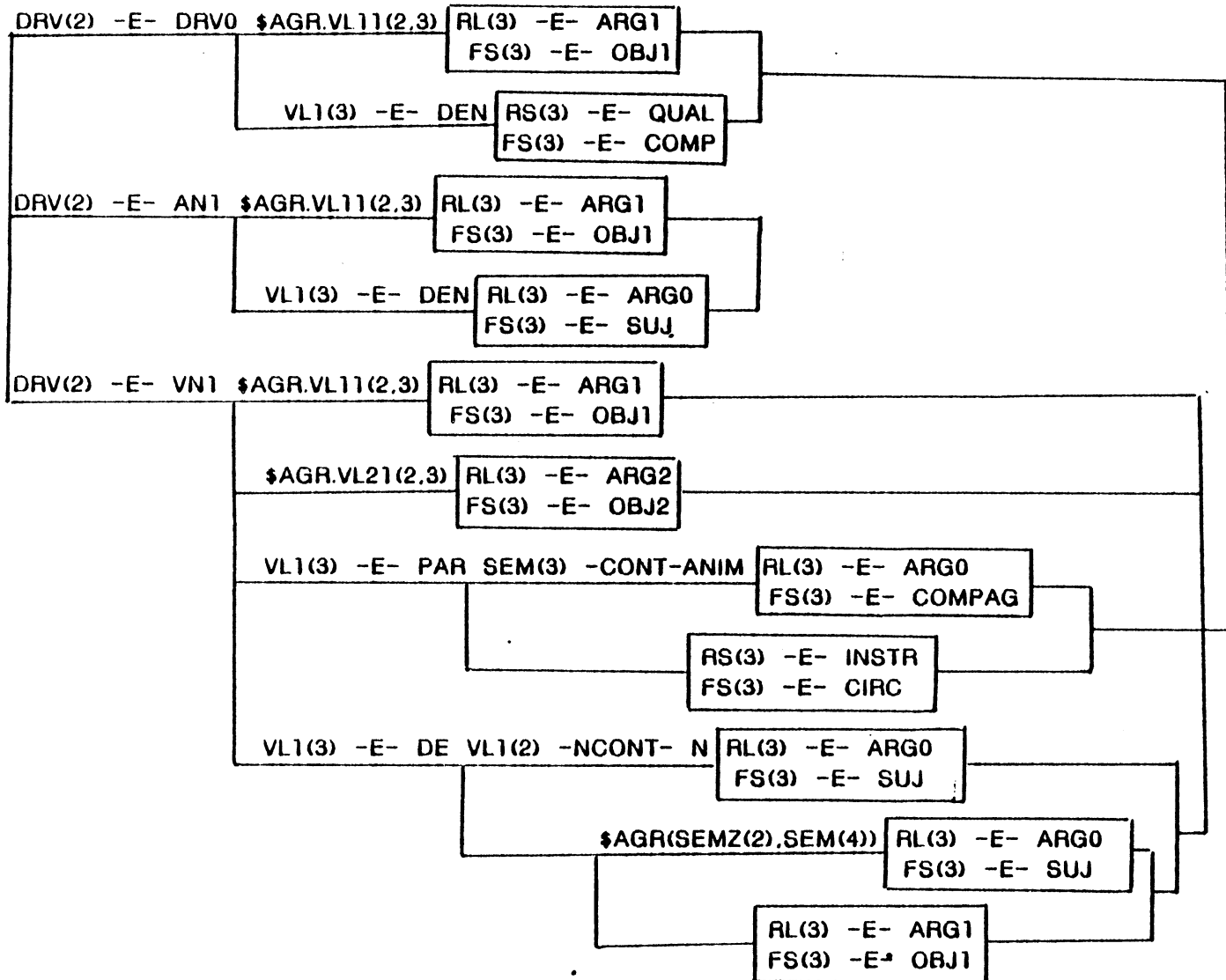
DRV(2) -E- VN1 -ET- VL1(2) -CONT- N -ET- VL1(3) -E- PAR

-OU-

VL1(3) -E- DEN

*com: la valence de rection du gouverneur du groupe référé 2 est satisfaite par le groupe nominal référé 3, ou le gouverneur de 2 est un nom d'action dérivé d'un verbe transitif et le groupe nominal 3 est régi par la préposition PAR, ou le groupe nominal 3 est régi par la préposition de.

ZONE 3



*COM: états des relations logiques et sémantiques et des fonctions syntaxiques sur le groupes nominal complexe en fonctions des attributs portés par les sommets des groupes référés.

Langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: ST32

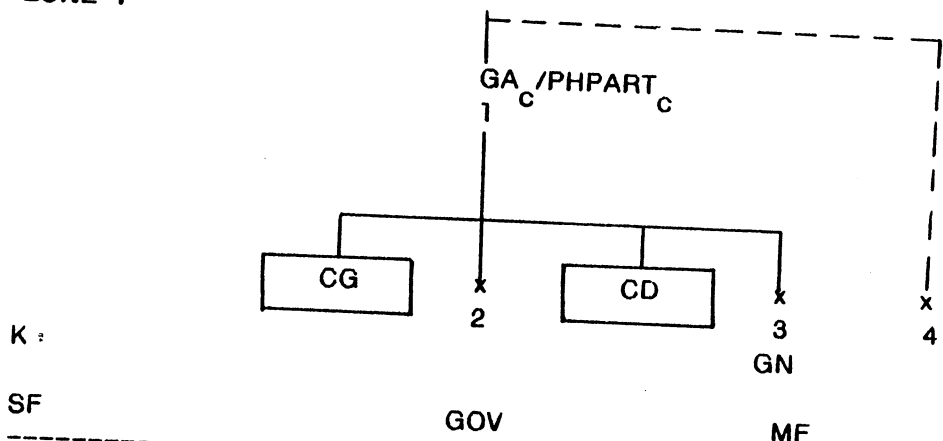
TYPE: GROUPE ADJECTIVAL (PARTICIPIAL) complexe

Cas traité: construit à partir d'un groupe simple et d'un groupe nominal absorbé.

Planches référées: (GAs) 6, 11, 13, 14.
 (GNc) 30, 31.
 (GNs) 25 à 29.
 (GAc) 32.

Etablie le:		par:
Modifiée	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I



ZONE II

K(4) -NE- (GN -OU- GA -OU- PHPART) -OU- \$TESTINCOMP(3,4)
 com: 3 ne peut dominer 4.

SUBV(2) -E- PPA

-ET- VAL1(3) -E- PARN

-ET- VAL1(2) -CONT- N

-ET- VSUJ(3) -E- VSUJ0

-OU-

\$AGVL11(2,3)

-ET- (VAL(3) -NE- N -OU- SUBV(2) -E- PPR)

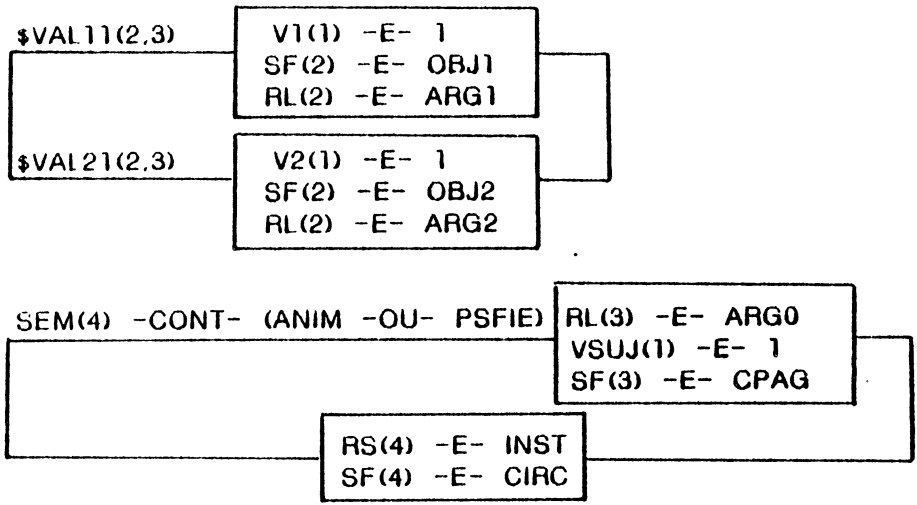
-ET- V1(1) -E- V10

-OU-

\$AGVAL21(2,3)

-ET- (VAL1(3) -NE- N -ET-V1(1) -E- V10)

ZONE III



Langue décrite: français
Grammaire statique: S.F.EXP
Planche n°: ST40

TYPE: NOYAU VERBAL 1

Cas traité: noyau à un élément
Planches référées: aucune

Etablie le:		par:
Modifiée		
	le:(1)	par:
	le:(2)	par:
	le:(3)	par:
Intégrée		
	le:(1)	dans:
	le:(1)	dans:
	le:(3)	dans:

ZONE I

	NV
	1
	x
	2
cat/subcat	v/vb
SF	GOV

ZONE II

ZONE III

Langue décrite: français
Grammaire statique: S.F.EXP
Planche n°: ST41

TYPE: NOYAU VERBAL 2

Cas traité: "avoir" (ou "être") + ppa (ou ppr).

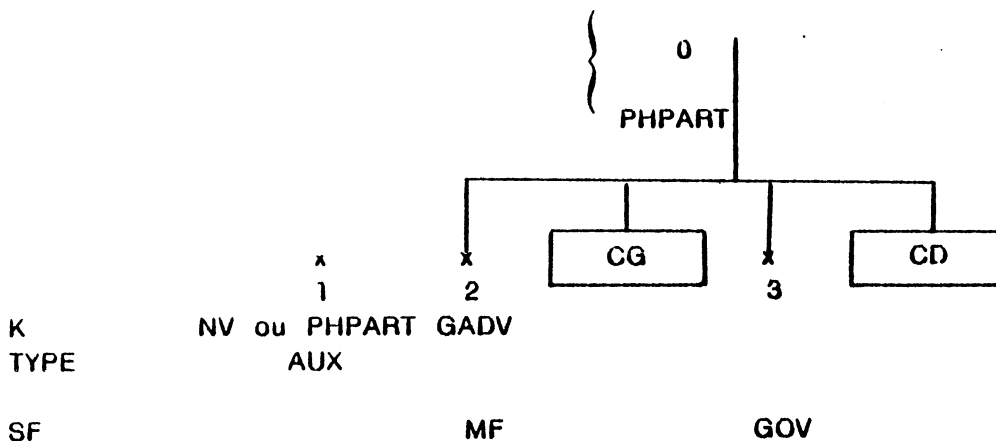
Planches référées: (PHPART) 15 à 20.

(GADV) 1.2.9.10.

(NV) 40.

Etablie le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(1)	dans:
le:(3)	dans:

ZONE I

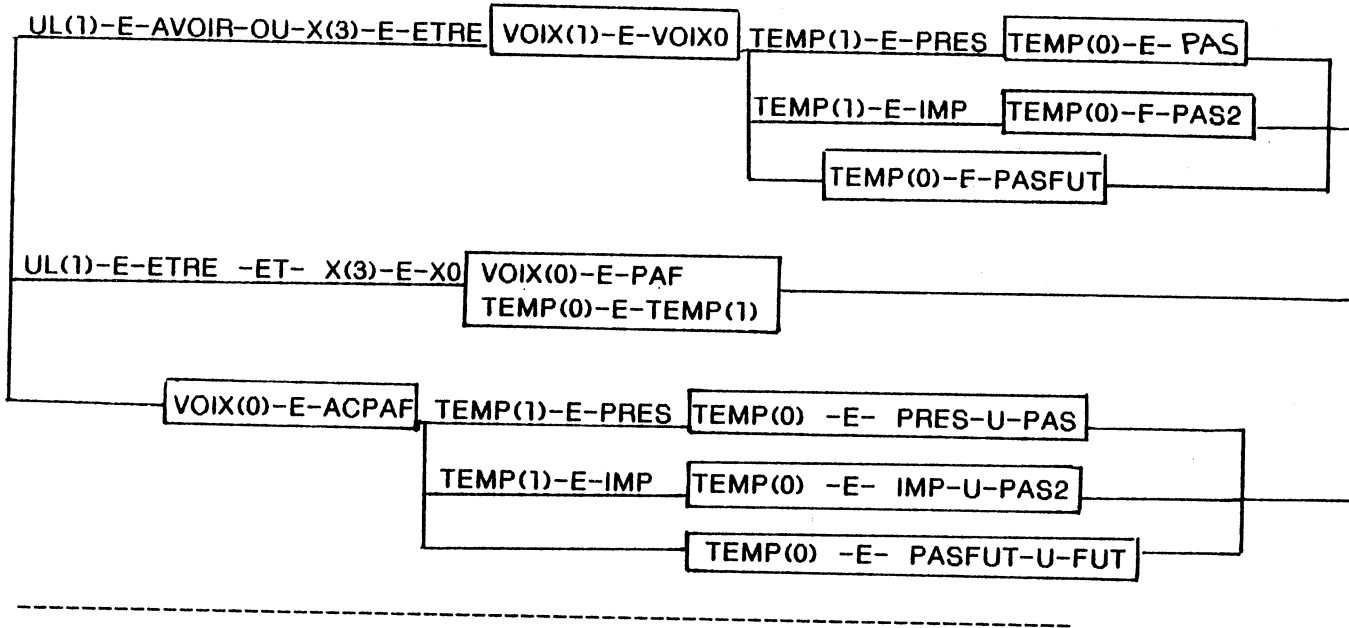


ZONE II

SUBV(3) -E- PPA'
SUBV(1) -NE- PPA
TYPE(1) -E- AUX

ZONE III

K(0) -E- K(1)
MODE(0) -E- MODE(1)
NUM(0) -E- NUM(1)
SUBV(0) -E- SUBV(1)



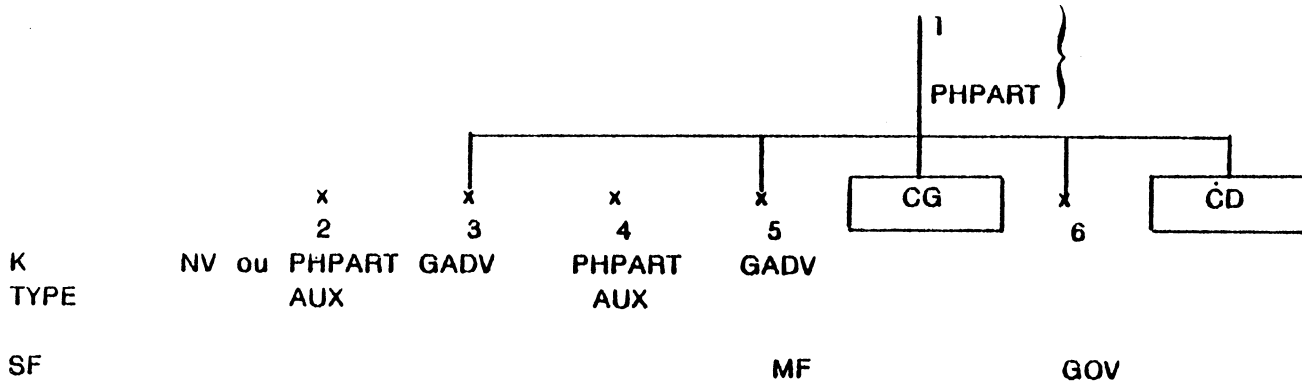
Langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: ST42

TYPE: NOYAU VERBAL 3

Cas traité: "avoir" + "être" + ppa.
 Planches référées: (PHPART) 15 à 20.
 (GADV) 1,2,9,10.
 (NV) 40.

Etablie le:		par:
Modifiée		
le:(1)		par:
le:(2)		par:
le:(3)		par:
Intégrée		
le:(1)		dans:
le:(1)		dans:
le:(3)		dans:

ZONE I

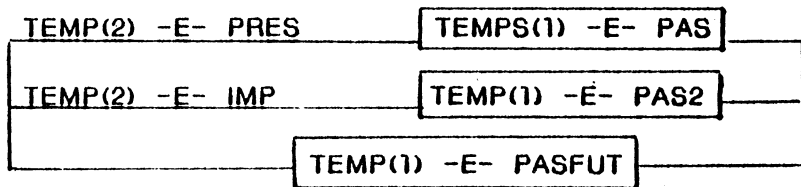


ZONE II

UL(2) -E- AVOIR
 UL(4) -E- ETRE
 SUBV(4) -E- PPA
 SUBV(2) -NE- PPA

ZONE III

K(1) -E- K(2)
 MODE(1) -E- MODE(2)
 NUM(1) -E- NUM(2)
 SUBV(1) -E- SUBV(2)



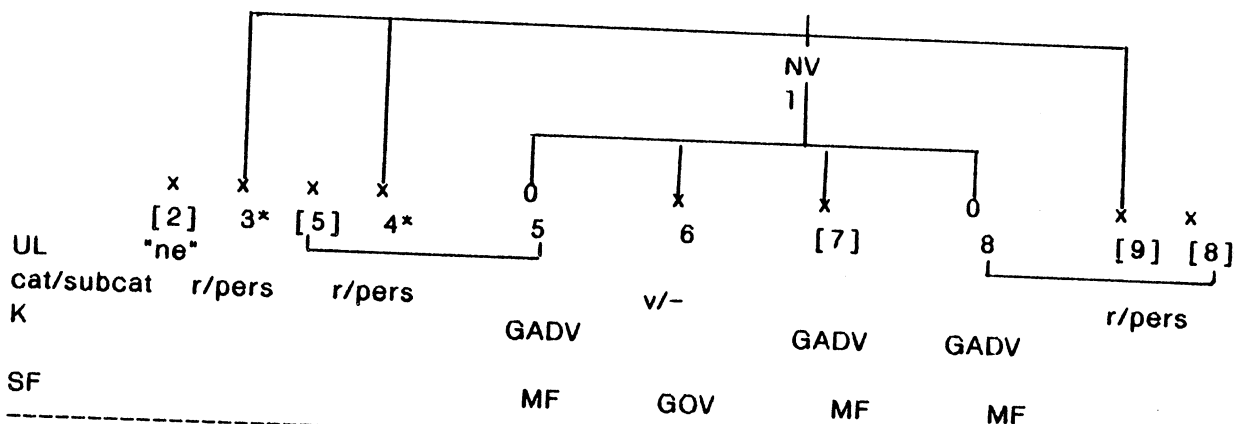
Langue décrite: français
 Grammaire statique: S.F.EXP
 Planche n°: SF43

TYPE: NOYAU VERBAL 4

Cas traité: temps simple
 Planches référées: aucune

Etablie le:	par:
Modifiée	
le:(1)	par:
le:(2)	par:
le:(3)	par:
Intégrée	
le:(1)	dans:
le:(1)	dans:
le:(3)	dans:

ZONE I



ZONE II

- NEGS(5) -E- 1
- NEGS(7) -E- 1
- NEG(8) -E- NEGS0
- 2 -SSI- (5 -OU- 7)
- 5 -IMP- (-7 -ET- SUBV(5) -CONT-INF)
- SUBV(6) -E- INF) -IMP- 5
- ROLE(3) -NE- SUBJ
- ROLE(4) -NE- SUBJ

ZONE III

- VAR(1) -E- VAR(6)
- K(1) -E- NV

2 NEGS(1) -E- 1

- VOX(1) -E- ACT
- COMPOSI(1) -E- SIMPLE
- PROP(1) -E- PROP(6)

Procédures utilisées

- \$AGR.NB(x,y) agrément en nombre entre x et y.
- \$AGR.GN(x,y) agrément en genre entre x et y.
- \$AGR.GN.NB(x,y) agrément en genre et en nombre entre x et y.
- \$SUBCAT(x) donne la sous-catégorie correspondant à la catégorie de x.
- \$ID.CAT.DR(x,y) vérifie l'identité de catégorie et de dérivation de x et y.
- \$TS.COOF(x,y) test de coordination forte entre x et y.
- \$INTERGN(x,y,...) calcule le genre intersection pour x, y, ...
- \$INTERNB(x,y,...) calcule le nombre intersection pour x, y, ...
- \$TESTINCOMP(x,y) vérifie le non-agrément en valences entre x et y (que x ne peut dominer y).
- \$AGRVL1(x,y) vérifie l'agrément en valence 1 entre x et y.
- \$AGRVL2(x,y) vérifie l'agrément entre la valence 2 de x et la valence 1 de y.



CHAPITRE III

CHOIX ET REPRESENTATION D'UN MODELE

"TEXTE \longleftrightarrow INTERPRETATION"

QUATRIEME PARTIE

A PROPOS DES GRAMMAIRES STATIQUES

ET DE LEUR UTILISATION DYNAMIQUE



4. A PROPOS DES GRAMMAIRES STATIQUES ET DE LEUR UTILISATION DYNAMIQUE.

Les grammaires dynamiques sont construites à partir des grammaires statiques qui leur servent de référence.

Ces grammaires constituent avec l'apport des informations trouvées dans les dictionnaires, les modèles calculables par ordinateur, des phases d'analyse, de transfert et de génération.

Ces grammaires dynamiques ne sont pas réversibles. En effet, la grammaire structurale d'une langue est distincte de la génération structurale de cette même langue, bien que toutes deux aient pour référence la même grammaire statique, qui est une description de cette langue.

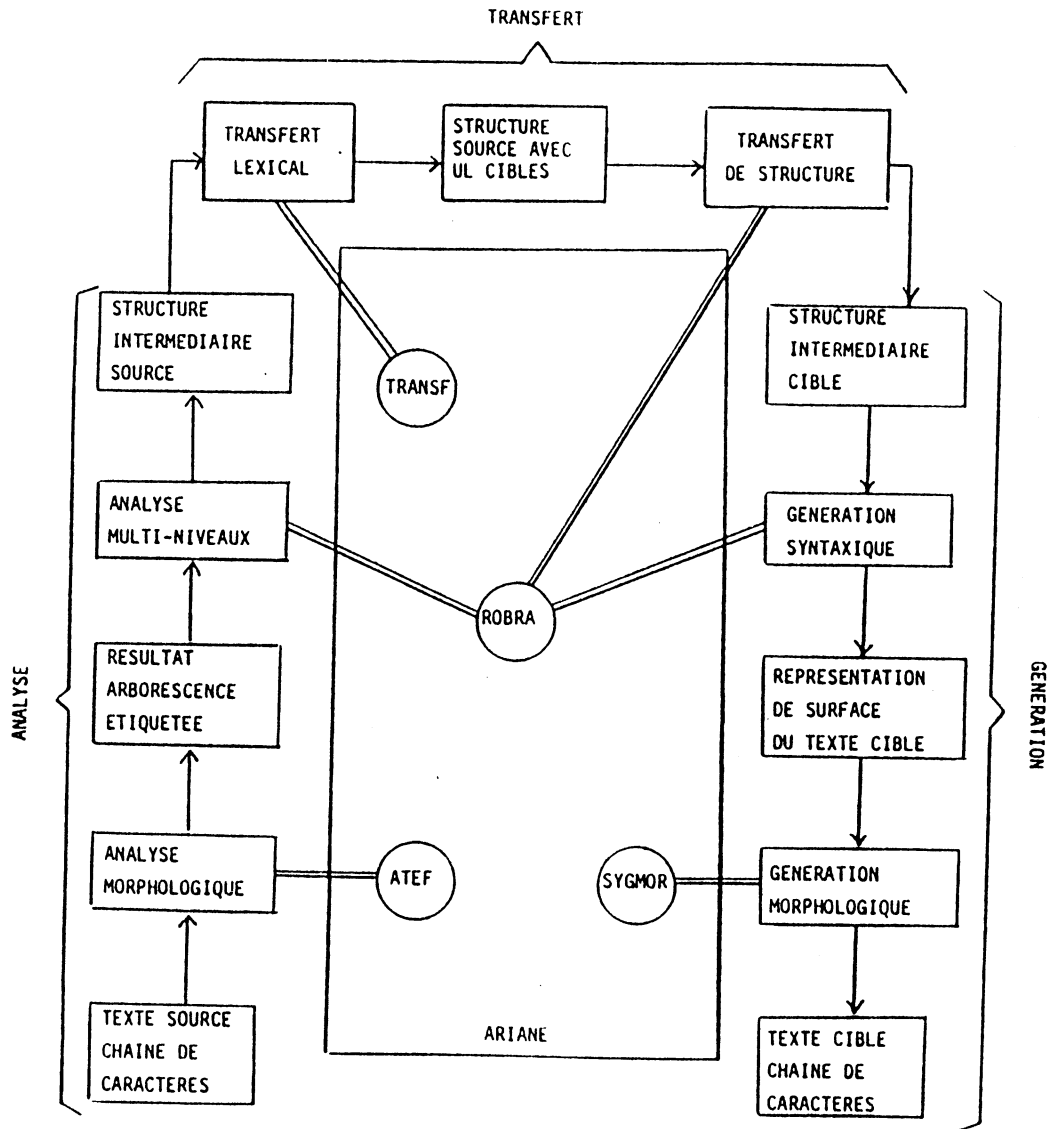
Les grammaires dynamiques de transfert sont elles, beaucoup plus réduites et ne posent pas de problèmes de programmation heuristique.

Les grammaires d'analyse et de génération se décomposent en grammaire dynamique morphologique et grammaire dynamique structurale.

En outre, toutes ces grammaires sont écrites au moyen d'un langage et dans le cadre de certains algorithmes.

En Ariane-78 ce sera:

- ATEF [CH&GU&QU.78] [GETA.82] pour l'analyse morphologique.
- SYGMOR [GETA.82] [JAEG.78] pour la génération morphologique.
- TRANSF [GETA.82] pour le transfert morphologique.
- ROBRA [BO&GU&QU.78] [GETA.82] dont nous avons déjà parlé (chapitre II.3.3) pour l'analyse, le transfert et la génération structurales.



Les composantes d'Ariane-78 qui implémentent les quatre modèles algorithmiques différents, apparaissent encadrées et sont rattachées par des doubles traits à des rectangles correspondant aux données linguistiques écrites dans le métalangage associé pour l'étape indiquée. Les flèches simples indiquent le flot de contrôle.

Figure III.48

En Ariane-78.5 on propose un nouveau schéma de traduction, où FXPANS [GUIL.83] est une généralisation de TRANSF, utilisable plus largement. Il a deux fonctions: transfert lexical (TL) et complément lexical (AX, TX, GX, AY, TY, GY).

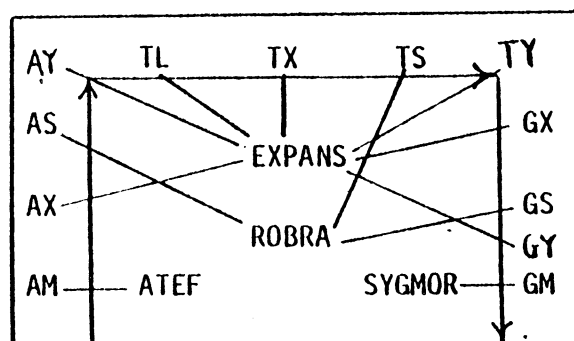


Figure 3.49

En Ariane-X, système destiné à remplacer Ariane-78.5, les algorithmes seront TETHYS, PRIAPE et d'autres.

D'une manière générale, ces grammaires sont des systèmes transformationnels organisés en modules et chaque module comporte un ensemble de règles et de paramètres, de guidage de l'algorithme.

4.1. GRAMMAIRES DYNAMIQUES D'ANALYSE.

L'analyse morphologique est réalisée au moyen d'un modèle d'états-finis "étendu". La grammaire dynamique est donnée par un graphe classique de modèle d'états-finis où la démarche heuristique est présentée sous forme de commentaire.

En analyse structurale, il s'agit de construire la structure intermédiaire à partir du résultat de l'analyse morphologique. Pour cela, on élabore une stratégie d'analyse en utilisant les planches de la grammaire statique structurale et des planches mettant en évidence les ambiguïtés.

On distingue trois types d'architectures de modules en analyse structurale: l'architecture des modules de construction; l'architecture des modules de résolution d'ambiguïtés et l'architecture des modules de contrôle et de correction.

Les modules de construction sont établis en vue d'obtenir la structure intermédiaire multi-niveau. La démarche heuristique pour aboutir à ce résultat est guidée par les considérations suivantes:

-Les constructions se font en recherchant d'abord les groupes syntagmatiques élémentaires, puis les groupes simples et enfin les groupes complexes.

-Dans les phases initiales, on ne construit que des structures sûres (sans ambiguïté); aussi, dès le début, des modules de résolution d'ambiguïtés de classes lexicales (nom-verbe, article-pronom, etc.) interfèrent avec les modules de construction.

-On itère les processus précédents jusqu'à ce qu'on arrive à la solution ou bien jusqu'à la rencontre de phénomènes d'ambiguïtés (lexicales ou de structure) qui obligent à essayer une construction incertaine.

Les modules concernés établissent cette construction en marquant sur la structure le risque encouru. C'est le rôle des modules de contrôle de comparer, après certaines étapes de l'analyse, les structures obtenues à celles que la grammaire statique prévoit d'obtenir. En cas de désaccord, les marques laissées par les modules de construction permettent aux modules de contrôle de détecter la cause d'erreur et d'aiguiller vers un module de correction approprié ou bien d'engager le processus de retour-arrière (back-tracking).

-Enfin, les tests qui orientent la succession des modules interrogent le texte (phrases par phrases); on a ainsi le moyen de guider l'analyse par le texte lui-même et d'éviter l'emploi de modules inutiles lorsque l'on rencontre des phrases simples.

L'architecture des modules de construction et celle des modules de résolution d'ambiguïtés ont été largement explorées, les recherches à venir devront se pencher tout particulièrement sur l'architecture des modules de contrôle et de correction.

Pour illustrer, quelque peu, le travail pris en charge par les modules de correction, voici un exemple de correction automatique d'erreur de construction, qui est tiré de l'analyse de l'anglais.

Dans la phrase: "For the first two million years on earth men lived by eating the plants and the wild animals", l'absence de virgule entre "earth" et "men" fait prendre le risque de construire le groupe nominal "on earth men"; on a donc la construction partielle provisoire:

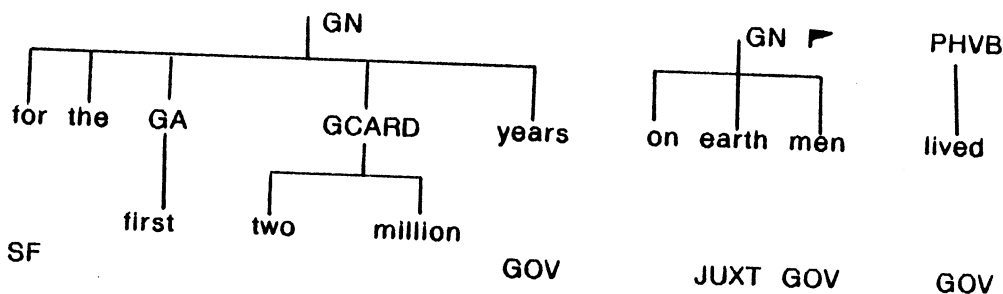


Figure III.50a

On cherche alors un sujet syntaxique pour le verbe "lived" (gouverneur de la phrase en attente d'être construite complètement) qui en exige un. On ne le trouve pas. Le groupe nominal

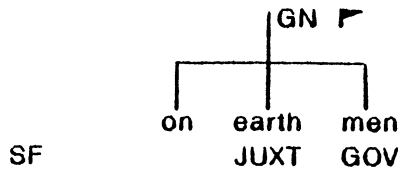


Figure III.50b

est porteur de la marque "▶" introduite par la règle qui a effectué la juxtaposition de "earth" et de "men".

Cette construction peut empêcher de trouver un sujet lorsque comme ici elle a été faite à tort. La correction doit "casser" le groupe nominal en deux groupes nominaux

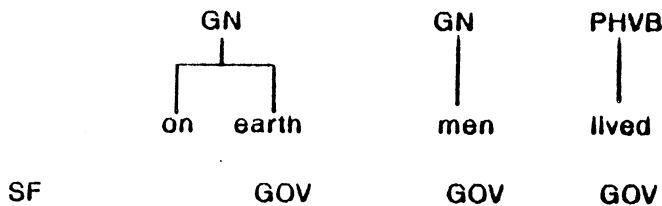


Figure III.50c

Le groupe nominal gouverné par "men" devient alors un sujet possible pour "lived".

A propos de l'organisation de l'analyse structurale, on trouvera en annexe une stratégie globale d'analyse de l'anglais, résultat d'un travail effectué dans le cadre d'une traduction de l'anglais vers le malais.

4.2. GRAMMAIRES DYNAMIQUES DE TRANSFERT.

Fondée sur les études contrastives entre langue source et langue cible (au niveau de la structure intermédiaire) qui sont répertoriées dans la grammaire statique, la grammaire dynamique se borne à écrire les règles et les modules correspondant à ces phénomènes. Cette grammaire doit donc, le cas échéant, substituer à la structure intermédiaire issue de l'analyse et du transfert lexical, une autre structure qui aurait été celle que l'on aurait obtenue en analysant la traduction dans la langue cible.

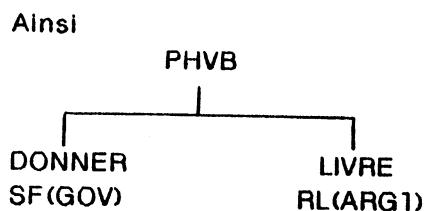
4.3. GRAMMAIRE DYNAMIQUE DE GENERATION.

Dans la phase de génération structurale, on peut avoir un choix assez important de constructions syntaxiques. On va donc trouver dans ces grammaires une organisation des modules articulée de la manière suivante:

- architecture des modules de construction;
- architecture des modules de sélection.

La stratégie consiste à partir de la racine d'une phrase (ou d'un titre) pour laquelle on impose la valeur de classe syntagmatique, phrase avec verbe conjugué (ou groupe nominal). On calcule alors, en fonction des relations logiques et sémantiques des dépendants de ce sommet, les nouvelles fonctions syntaxiques que l'on doit attribuer à ces dépendants. Le choix éventuel entre plusieurs fonctions syntaxiques pour réaliser la même relation logique est d'une part réduit par les contraintes lexicales dans la langue cible et d'autre part guidé par les priorités indiquées dans les modules de sélection.

Par exemple, une phrase comportant simplement le prédicat verbal et son deuxième argument peut être énoncée, aussi bien en anglais qu'en français, soit par voix passive (la relation logique de deuxième argument est alors réalisée par la fonction syntaxique de sujet), soit par voix active (en créant d'une part un sujet formel ou indéfini et d'autre part une fonction syntaxique d'objet pour le deuxième argument).



où RL(ARG1) indique que "livre" est relié à "donner" par la relation logique ARG1 de deuxième argument.

Figure III.51

En attribuant à "livre" la fonction syntaxique "SUJET", on impose le passif et on aura par la suite: "le livre a été donné".

En lui attribuant la fonction syntaxique "OBJET-1", on impose la voix active et on aura: "on a donné le livre".

Si en français le gouverneur n'est pas transitif direct, la construction à la voix passive est interdite.

Par contre, une phrase comportant un prédicat verbal, son deuxième et son troisième argument, peut parfois en anglais être énoncée par deux types de

voix passives (les relations logiques de deuxième ou troisième argument pouvant indifféremment être réalisées par la fonction syntaxique de sujet).

"John has been given a book"

"A book has been given to John"

En français le troisième argument ne peut jamais être réalisé par la fonction syntaxique de sujet.

Une fois les fonctions syntaxiques déterminées, on calcule (et on choisit) pour ces mêmes sommets, les classes syntagmatiques qui se réalisent. La démarche est analogue à la précédente. Là encore, le choix peut être limité par les contraintes lexicales et sera guidé par le "style" que l'on a programmé en priorité (ce style peut être, uniquement, d'essayer de "coller" le plus possible au texte source).

La construction morphologique relève d'un modèle simple (modèle des langages d'états-finis) et à ce niveau on ne prend pas en considération les choix de forme des mots (ils sont de toutes façons en nombre illimité; en français: paie-payé, assieds-asseois, etc.); on fait un choix arbitraire.



CHAPITRE IV

CONCLUSION



Une grammaire statique telle que nous l'avons présentée est un outil de description de la langue. Les grammaires traditionnelles n'étant adaptées ni à nos problèmes, ni à nos besoins, il devenait nécessaire de définir un formalisme permettant de représenter la langue et les phénomènes linguistiques qui s'y rattachent, cette représentation devant être utilisable pour élaborer des grammaires dynamiques dans notre modèle à multiples niveaux d'interprétation.

Nos planches statiques, qui établissent une correspondance entre une famille de chaînes et une structure, sont utilisables aussi bien en analyse qu'en génération.

En analyse, les recherches devront à l'avenir se pencher tout particulièrement sur l'architecture des modules de contrôle et de correction.

En génération, le problème sera d'améliorer la lisibilité des planches statiques de construction par l'adjonction de modules de sélection qui permettront d'établir des priorités de choix de fonctions syntaxiques (lorsque les possibilités de choix sont multiples), pour réaliser une relation logique.

Il faudra, à l'avenir, regarder de plus près, comment adapter le formalisme, à l'écriture de grammaires statiques en transfert; ces grammaires devant être représentatives des correspondances entre deux langues, correspondances établies indépendamment du choix de la cible et de la source dans le couple de langues considéré.

Chacune de nos planches statiques établit une correspondance entre "un morceau de chaîne" appartenant à la langue et une construction. Nous avons vu que ce morceau de chaîne peut lui-même faire référence à un autre morceau de chaîne de la langue et ainsi de suite. Un des problèmes principaux, lié à ces grammaires statiques, est la maîtrise de l'aspect combinatoire induit par notre formalisme de description. Pour lutter contre cela, nous nous sommes donnés deux armes. La première est la contrainte de couverture maximum, qui imposera le choix de la plus grande chaîne possible, entrant en concordance avec la famille de chaîne décrite. La seconde est celle de hiérarchie dans l'établissement des différentes planches qui permet de maîtriser la combinatoire inhérente aux langues naturelles elles-mêmes.

Nous pourrions envisager, par la suite, d'implémenter la construction automatique de la totalité des chaînes décrites par nos planches statiques. Dans ces chaînes, les références seraient remplacées par les chaînes référées elles-mêmes, de façon à visualiser la grammaire statique dans sa totalité. Ceci nous donnerait une vision globale (mais sans doute complexe), de toutes les chaînes décrites par notre modèle, et nous permettrait de vérifier la cohérence de nos grammaires statiques, ou encore, par la suite, de vérifier la cohérence entre une grammaire statique et les grammaires dynamiques qui en sont issues.



ANNEXE I

ATTRIBUTS LEXICAUX DU FRANCAIS

EXTRAIT



Attributs relatifs aux différents niveaux d'interprétation.

Relations logiques: RS

ARG0 argument 0
ARG1 argument 1
ARG2 argument 2
TRL0 transfert de relation (propriétés) sur arg0
TRL1 transfert de relation (propriétés) sur arg1
ID identité de propriétés
SMARG semi-argument lexeme de particularisation

Relations sémantiques: RL

CAUSE cause d'origine
SOURCE matière, point de départ (lieu et temps)
EMIT agent, expérimentateur, propriétaire
COND condition
AIM but en lieu et temps
FINAL finalité d'une action
RECEP receveur ou bénéficiaire
RESULT objet transformé
CONSEQ action ou état résultant
TOPIC relatif à
TOY tout objet subissant l'action
INST instrument, objet physique
ACCOMP accompagnement
ANALOG analogie
CONCESS concess (heureux malgré sa pauvreté)
QFIER quantification
QOBJ objet quantifié
PROX proximité
LOCAL localisation en lieu et temps
QUAL qualification

Fonctions syntaxiques: FS

SURJ sujet réel
OBJ1 premier complément d'objet
OBJ2 deuxième complément d'objet
FSURJ sujet formel
ATSUBJ attribut du sujet
ATOBJ attribut de l'objet
ATG attribut du gouverneur
MF modificateur
DES désignation
CIRC circonstanciel
CPAG complément d'agent
COMPN complément de nom
COORD coordination
REG régisseur de groupe (prep ou conj. de subord.)
GOV gouverneur
AUX auxiliaire de mode, de temps
JUXT juxtaposition

APP apposition
INC inclusion
LXAX lexème auxiliaire
COMPAR second élément d'une comparaison

Classes syntagmatiques: K

GA groupe adjectival
GADV groupe adverbial
GN groupe nominal
GDEICT groupe déictique
NV noyau verbal
PHINF phrase infinitive
PHVB phrase verbale
PHPART phrase participiale
PHREL phrase relative
PHSUB phrase subordonnée

Catégories syntaxiques: CAT

V verbe
N nom
A adjectif
D déicteur
R représentant
S subordonnant
C coordonnateur
P ponctuation
PREF préfixe libre
PREFNL préfixe non lexicalisé
INC mot inconnu
NA forme non alphabétique
PART particule

Sous-catégories morpho-syntaxiques

Sous-catégorie d'adjoints: SUBA

ADJ adjectif
ADV adverbe
ADDEICT adjectif de déicteur
ADVN adjectif de nom
ADCL adverbe de clause
QTF quantificateur
CARD adjectif cardinal
ORD adjectif ordinal

Sous-catégorie de déicteur: SUBD

ART article
DEM démonstratif
QTF quantificateur
POS possessif
REL relatif
INT Interrogatif

Sous-catégorie de nom: SUBN

NC nom commun

NP nom propre

Sous-catégorie de représentant: SUBR

DEM démonstratif

QTF quantificateur

POS pronom possessif

REL pronom relatif

PERS pronom personnel

Sous-catégorie de verbe: SUBV

VB verbe conjugué

INF infinitif

PPA participe passé

PPR participe présent

Autres attributs utilisés, classés par ordre alphabétique.

AUX auxiliaire de conjugaison

ETRE
AVOIR

DEG degré

COMP comparatif
SUP superlatif
CONS consécutif
REL relatif

DRV dérivation

VN construction
NA cérealier
AN précision
VNAG constructeur
AM utillement
AP néo
VAA constructif
VPA remarquable
VAAN agressivité
VPAN constructibilité
NP ferro
PREPP anti (de contre)

ETRE type d'être animé

ANIMAL
HUMAIN
PERSNF personifié

GNR genre grammatical

MAS masculin
FEM féminin

MODE mode (subv=vb)

IND indicatif
SUBJ subjonctif
IMP impératif
CDL conditionnel

NBR nombre

SG singulier
PL pluriel

NEG négation

OUI

SEM classes sémantiques principales

ESPACE localisation spatiale
TEMPS localisation temporelle

ETRE être animé
CONCRET objet physique ou matière
ABSTR état ou action ou notion

SEM0 classe sémantique de l'argument 0
SEM1 classe sémantique de l'argument 1
SEM2 classe sémantique de l'argument 2

SEXE genre sémantique
MAS masculin
FEM féminin

TEMPS temps
PRES présent
PAS passé
FUT futur
IPF imparfait

TYPE type de prédicat
AUX auxiliaire de conjugaison
MODAL modal ou assimilé (pouvoir, devoir...)
COP copule (verbe introduisant attribut du sujet)
ATROB verbe pouvant introduire un attribut de l'objet
RX verbe obligatoirement réfléchi
VIMP verbe impersonnel
COMP verbe pouvant être à la base d'un prédicat composé
NORMAL autre comportement syntaxique

TYPOG typographie
MAJ majuscule
TMAJ tout en majuscules
DIGIT nombre
ABR abréviation

VAL0 valence du sujet (voix active)
N Jean mange une pomme
I boire permet de survivre
Q qu'in chante si bien étonne

VAL1 valence de l'objet 1 d'un prédicat
N souder x
AN penser à x
AVECN compter avec x
DANSN s'infiltrer dans x
DEN dépendre de x
PARN commencer par x
POURN passe pour x
SURN enquêter sur x
I vouloir faire
AI tenir à participer
DEI venir de partir

PARI finir par accepter
POURI insister pour obtenir
Q penser que
AQ tenir à ce que
DEQ venir de ce que

VAL2 valence de l'objet 2 du prédicat
idem à VAL1

VOX voix (subv=vb.ppr.inf)
ACT active
PAF passive
RXPAF réflexive passive
IMPER impersonnelle

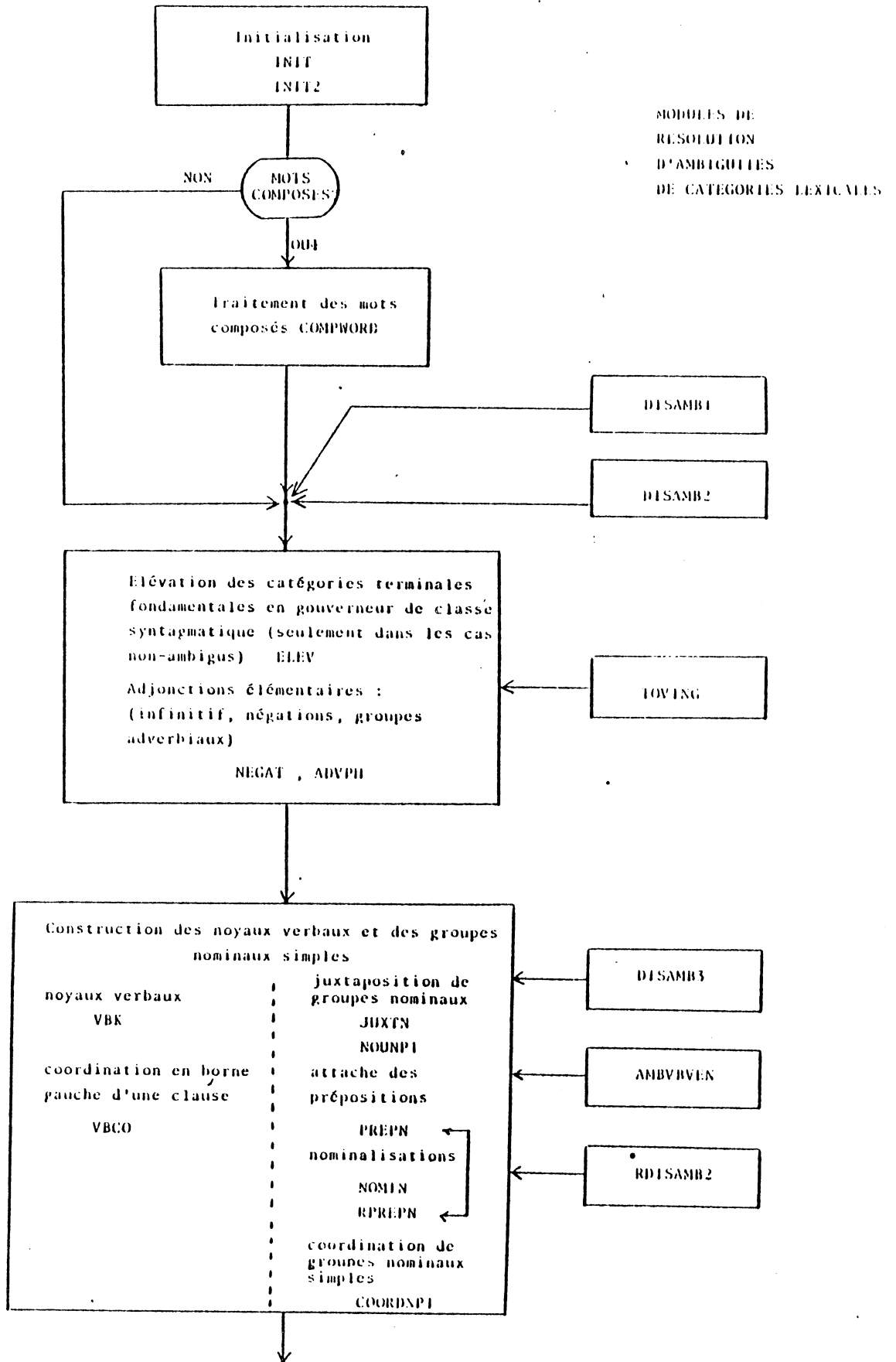
ANNEXE II

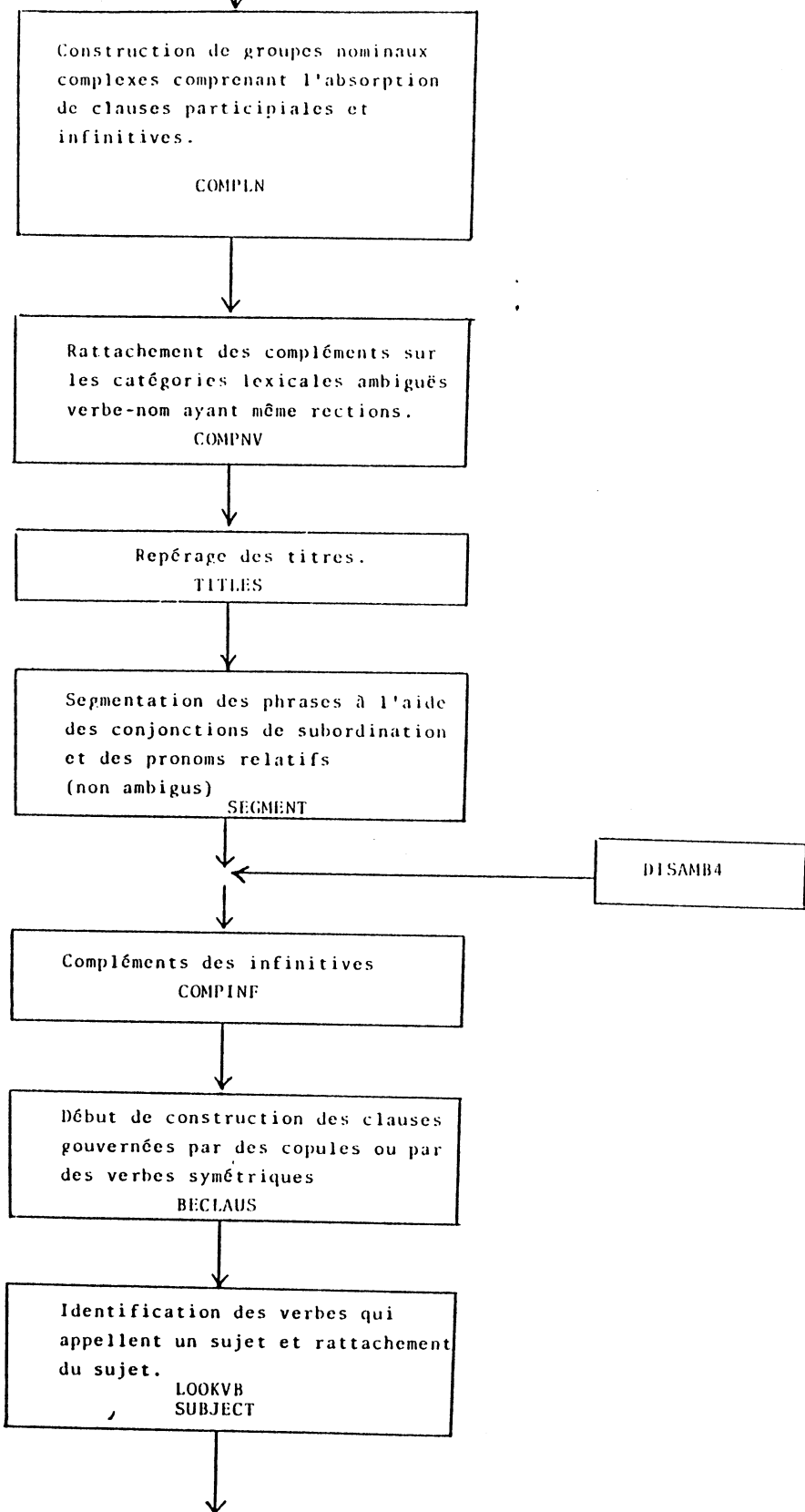
STRATEGIE GLOBALE DU MODELE D'ANALYSE DE L'ANGLAIS

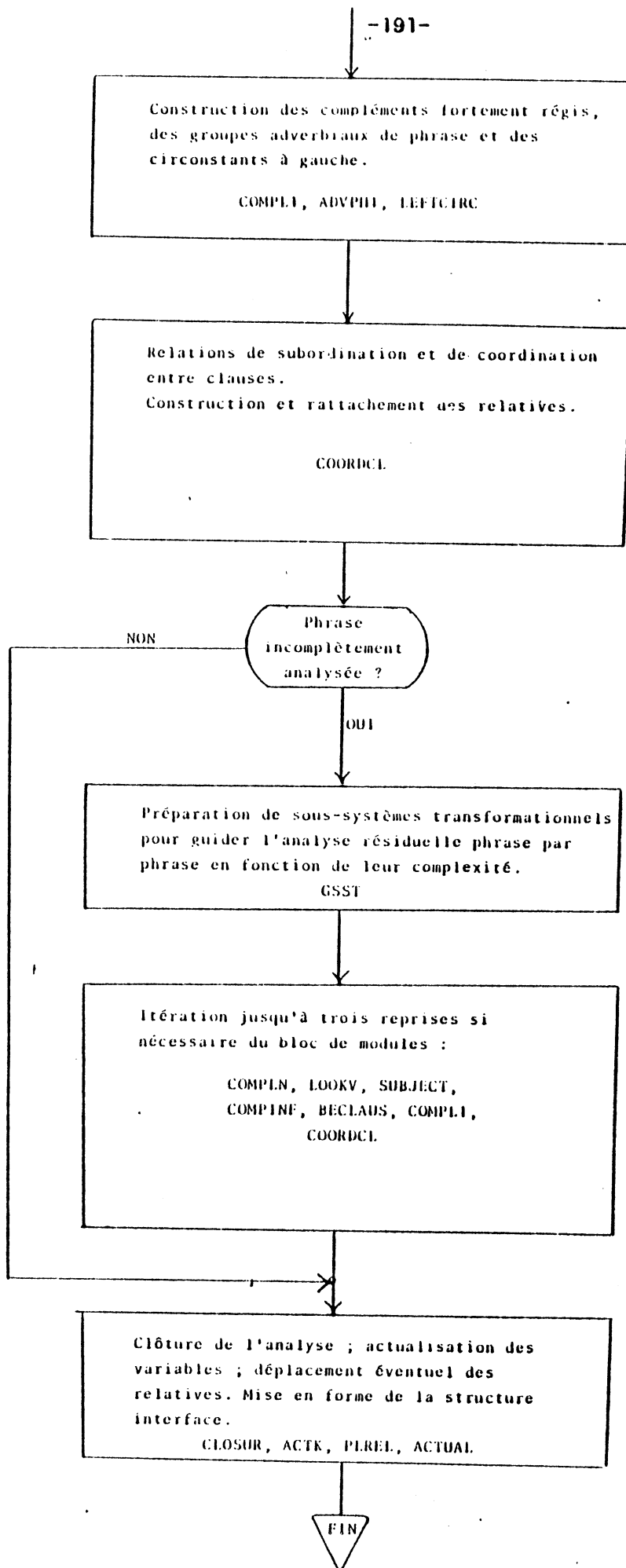
ORGANISATION DES MODULES

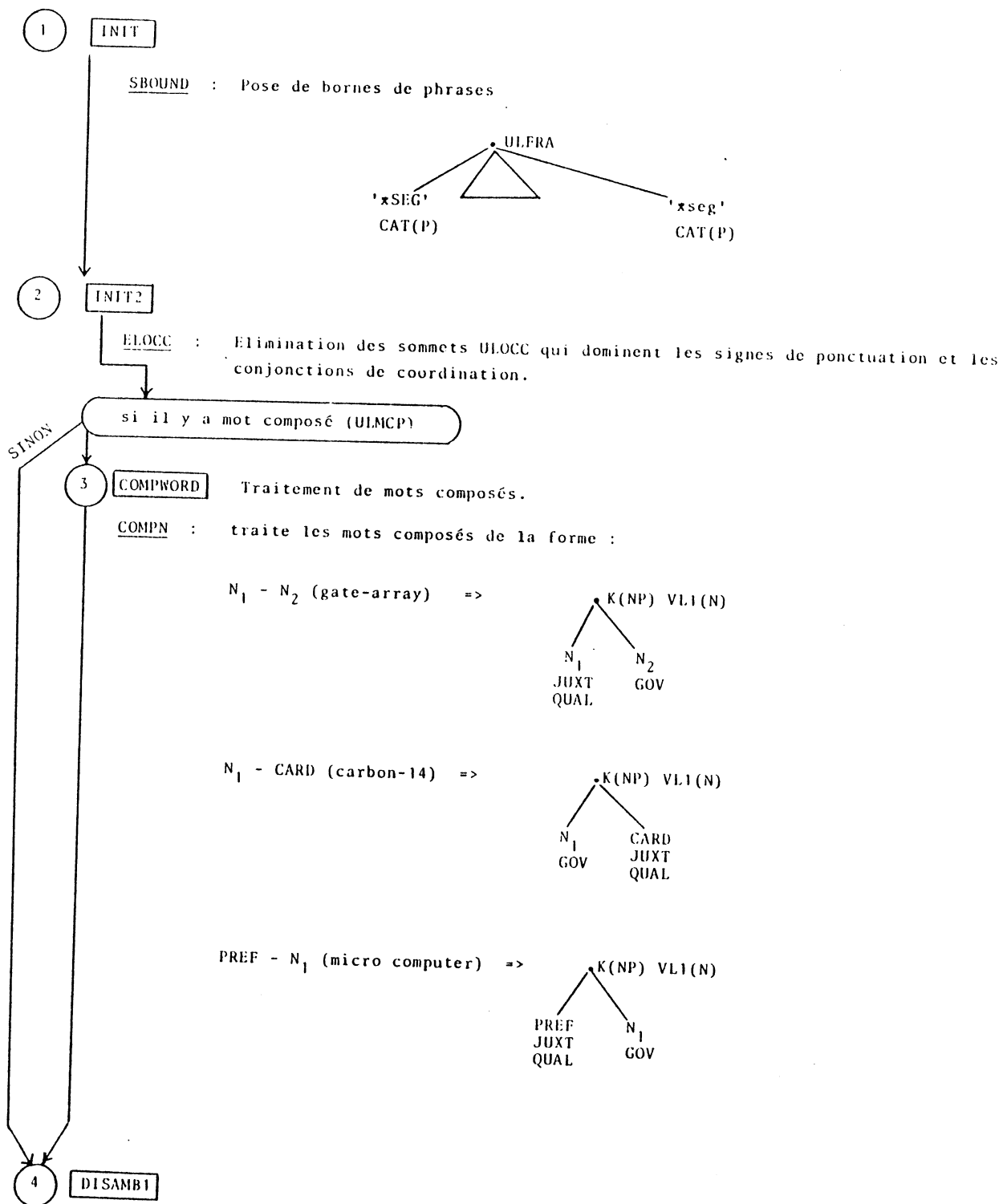
Cette analyse, extraite d'un document GETA [VA&DA.82] est le résultat du travail effectué en collaboration avec l'Université de Malaisie à Pénang dans le cadre de recherches faites sur la traduction de l'anglais vers le malais.











4

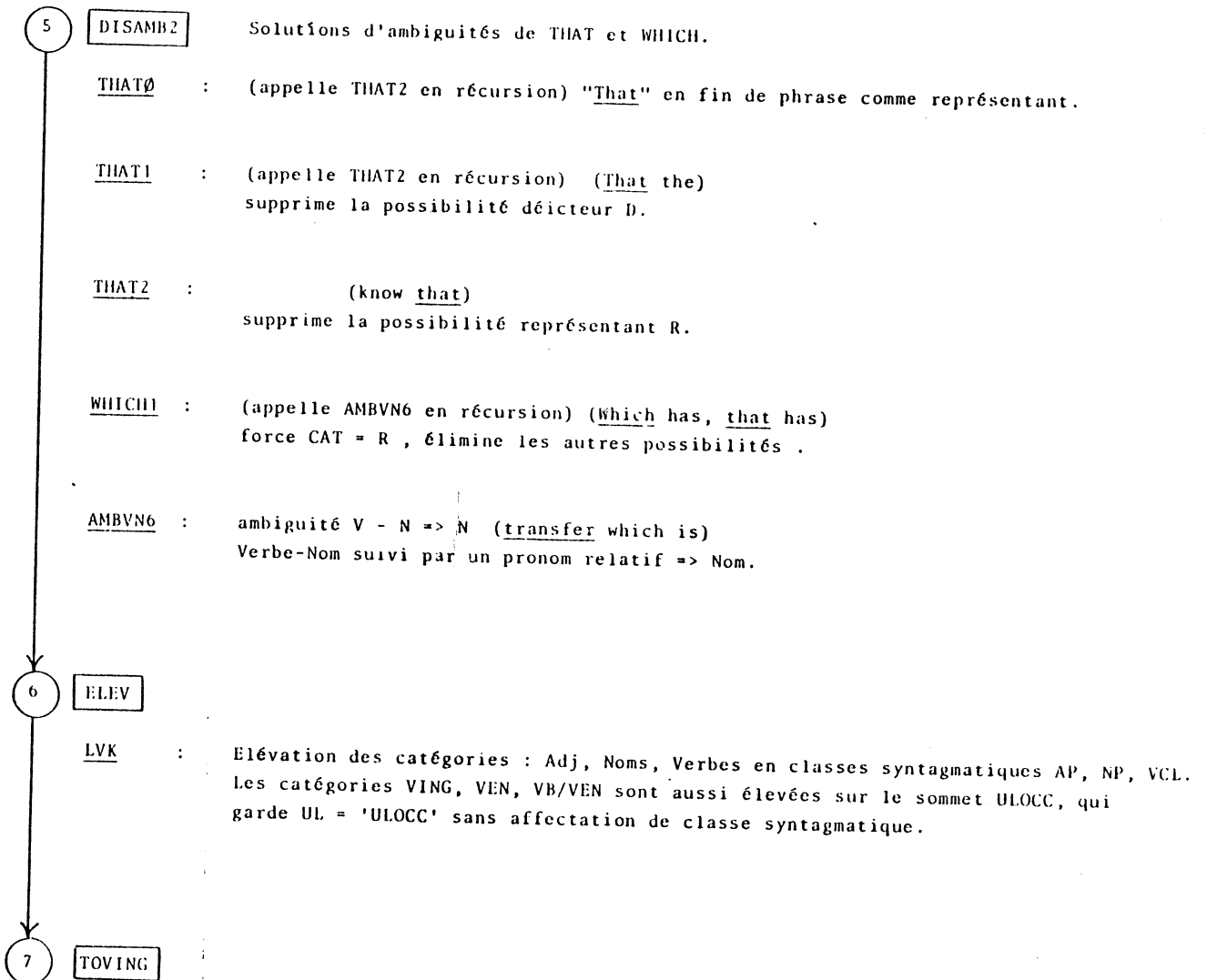
DISAMB1

Première phase de solutions d'ambiguïtés de catégorie syntaxique.

- AMBVN1 : ambiguïté V - N => N (The heat, this type)
 Adj Adj
résoud aussi l'ambiguïté éventuelle sur le détecteur.
- AMBVN2 : ambiguïté V - N => V (They heat, this means)
résoud aussi l'ambiguïté éventuelle sur le représentant,
ne s'applique pas aux formes verbales ambiguës VB - VEN (followed).
- AMBVN3 : ambiguïté V - N => N (of heat)
 Adj Adj
ne s'applique pas avec la préposition to.
- AMBVN4 : ambiguïté V - N => N (a safety screen)
contexte gauche d'un article indéfini suivi d'un nom abstrait de propriété.
- VBVENO : ambiguïté VB/VEN => VEN : (the proposed method).
- AMBRD1 : ambiguïté R - D => R (to another).
- AMBRD2 : ambiguïté R - D => R (This was).
- AMBPAD : ambiguïté P - AD (of about)
(préposition-adjoint => adjoint).

5

DISAMB2



7 TOVING

Solutions d'ambiguïtés de VING
Construction de l'infinitif.

PVING : (. giving a , for making the)
Préposition ou ponctuation + VING + déicteur
donne une PARTCL/ING (participiale en ing).

DETINGP : (the manufacturing of)
Déicteur + VING + préposition ou verbe
donne un Nom.

PRINGVB : (the aim of designing is)
Préposition + VING + Verbe ou ponctuation
donne un Nom.

TOVB : (to incorporate)
Construction de l'infinitif.
Élévation d'une PARTCL/INF ; verrouillage du sujet
(SLOCK = 2) ; valence d'état VLI = 1.
Le sommet TO disparaît.

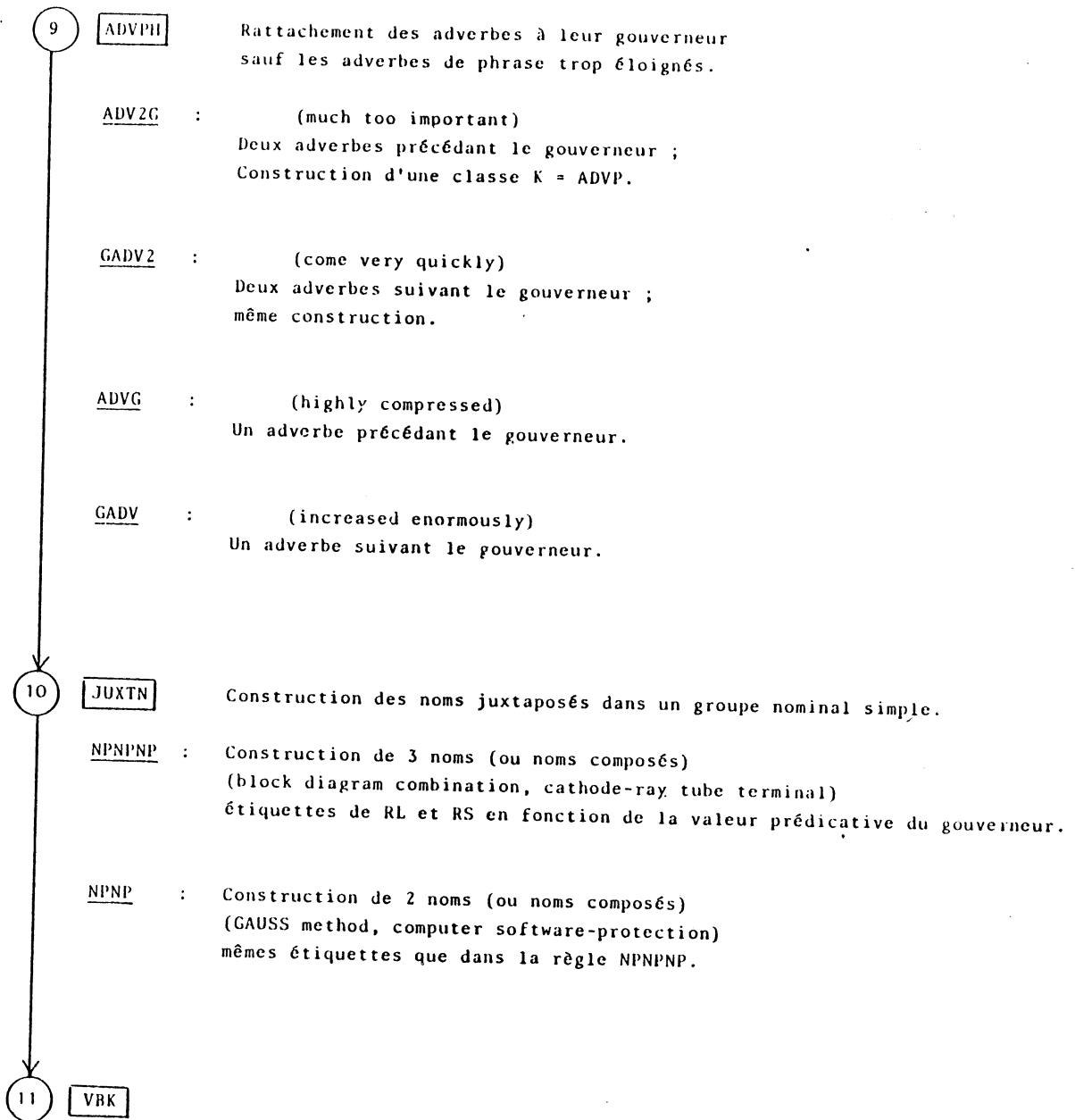
8 NEGAT

Construction des négations analytiques.

NEG1 : (do not make)
Construction avec DO ; les sommets DO et NOT disparaissent ;
la négation est portée en variable sur la clause.

NEG2 : (is not, have not, may not)
Construction avec copule ou auxiliaire ou modal.
Mêmes résultats que NEG1.

9 ADVPH



11

VBK

Construction du noyau verbal.

BEINGEN : aspect progressif : (are being added)

BEING : aspect progressif : (are doing)

MDHBVEN : Modal + have + been + VEN
(can have been done)

MDHVENA : Modal + have + VEN (active)
(could have reduced)

MDHVEN : Modal + be + VEN
(can be reduced)

MDVBA : Modal + VB (active)
(can combine)

HVEN : have + been + VEN
(have been performed)

HVEN : have + VEN
(having performed)

ISVEN : be + VEN
(was performed)

FUT : future : (shall do, will be)

12

DISAMB3

Nouvelle phase de solutions d'ambiguïtés.

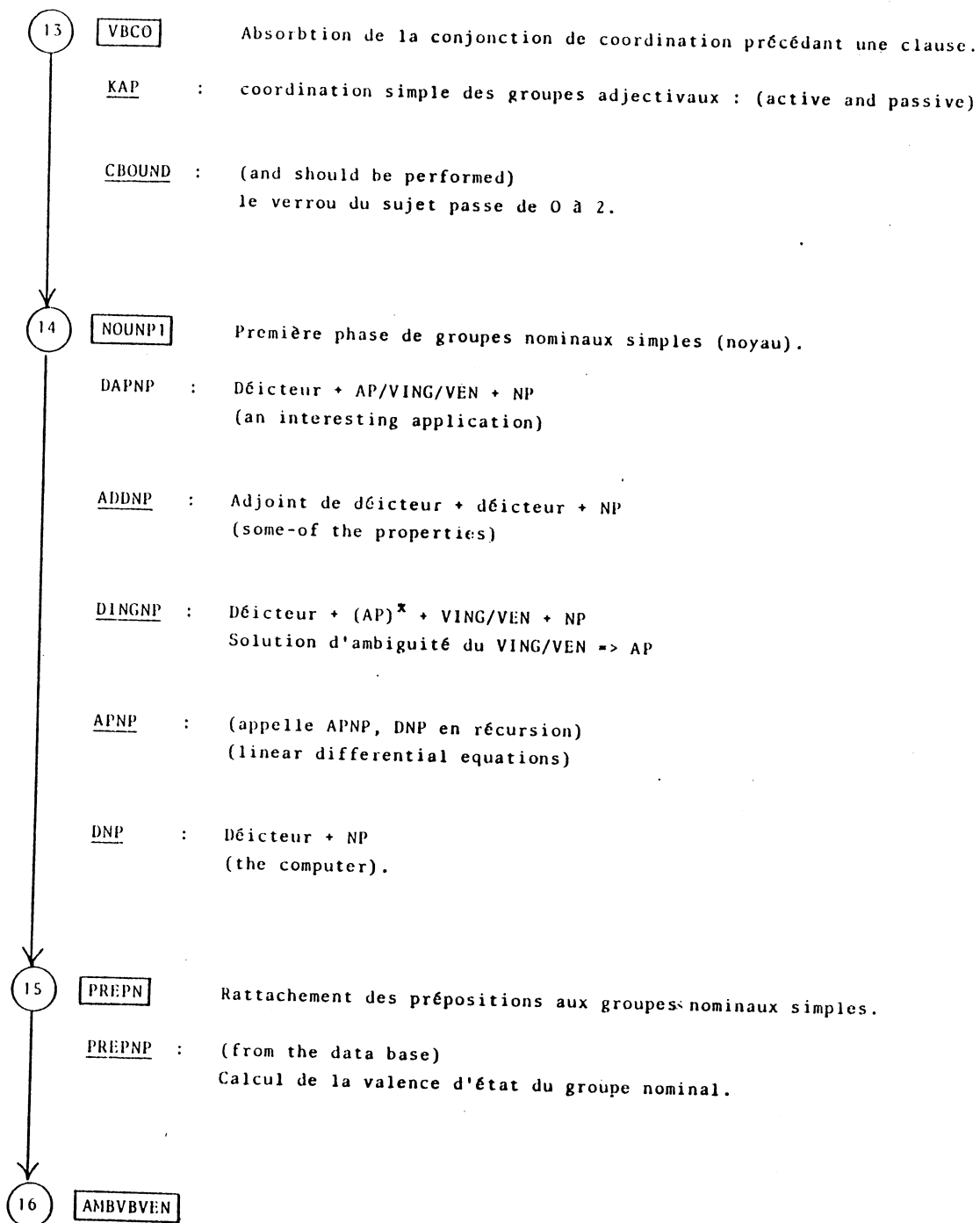
AVINGAP : ambiguïté de VING résolue en adjectif (AP)
(The interesting easy experiment)
valable aussi pour les formes en VEN.

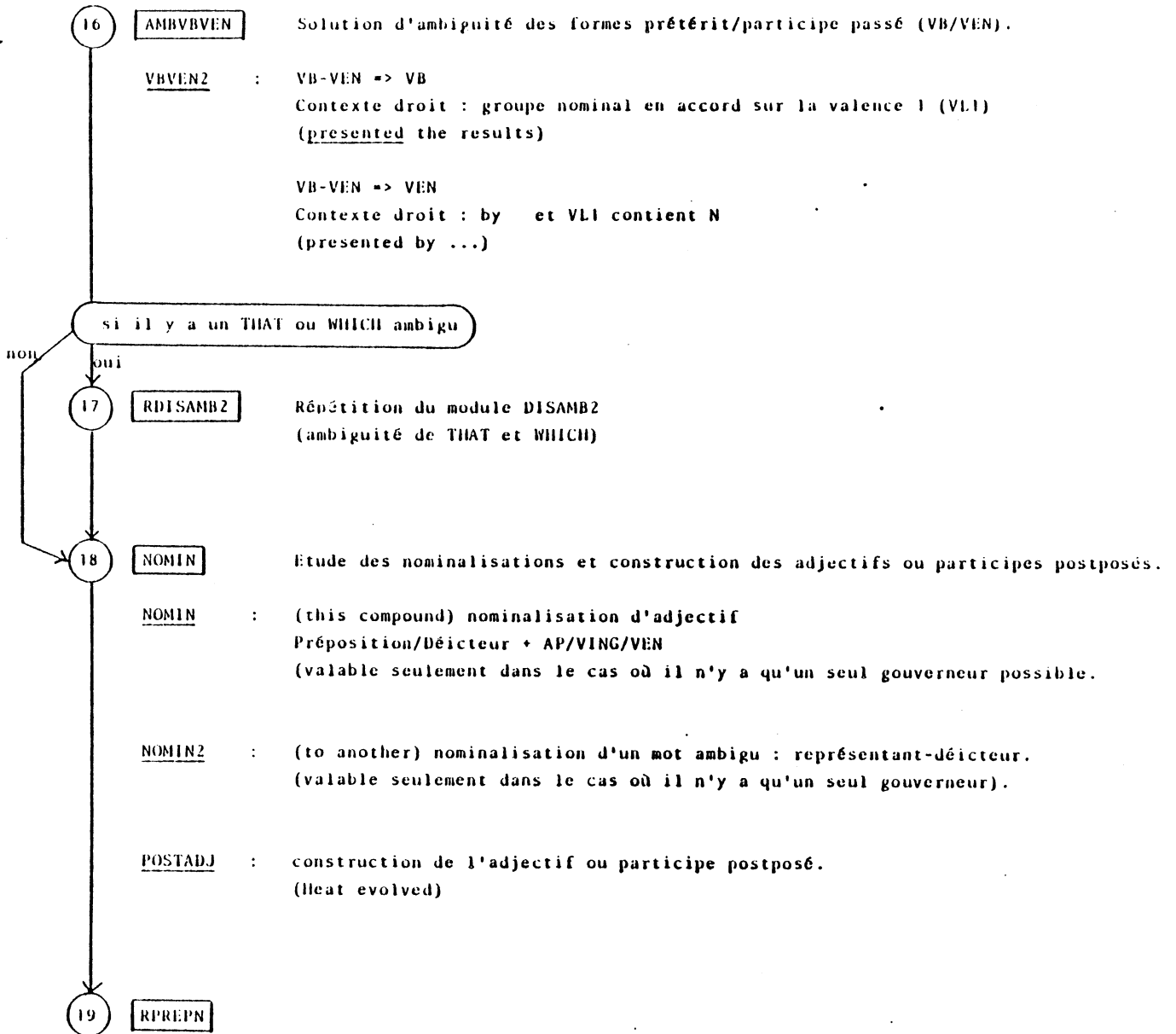
VBVEN1 : VB - VEN => VB
Contexte gauche¹: Représentant (which performed)
VB - VEN => VEN (K = AP)
Contexte gauche : verbe (produces used ...)
(VCL)

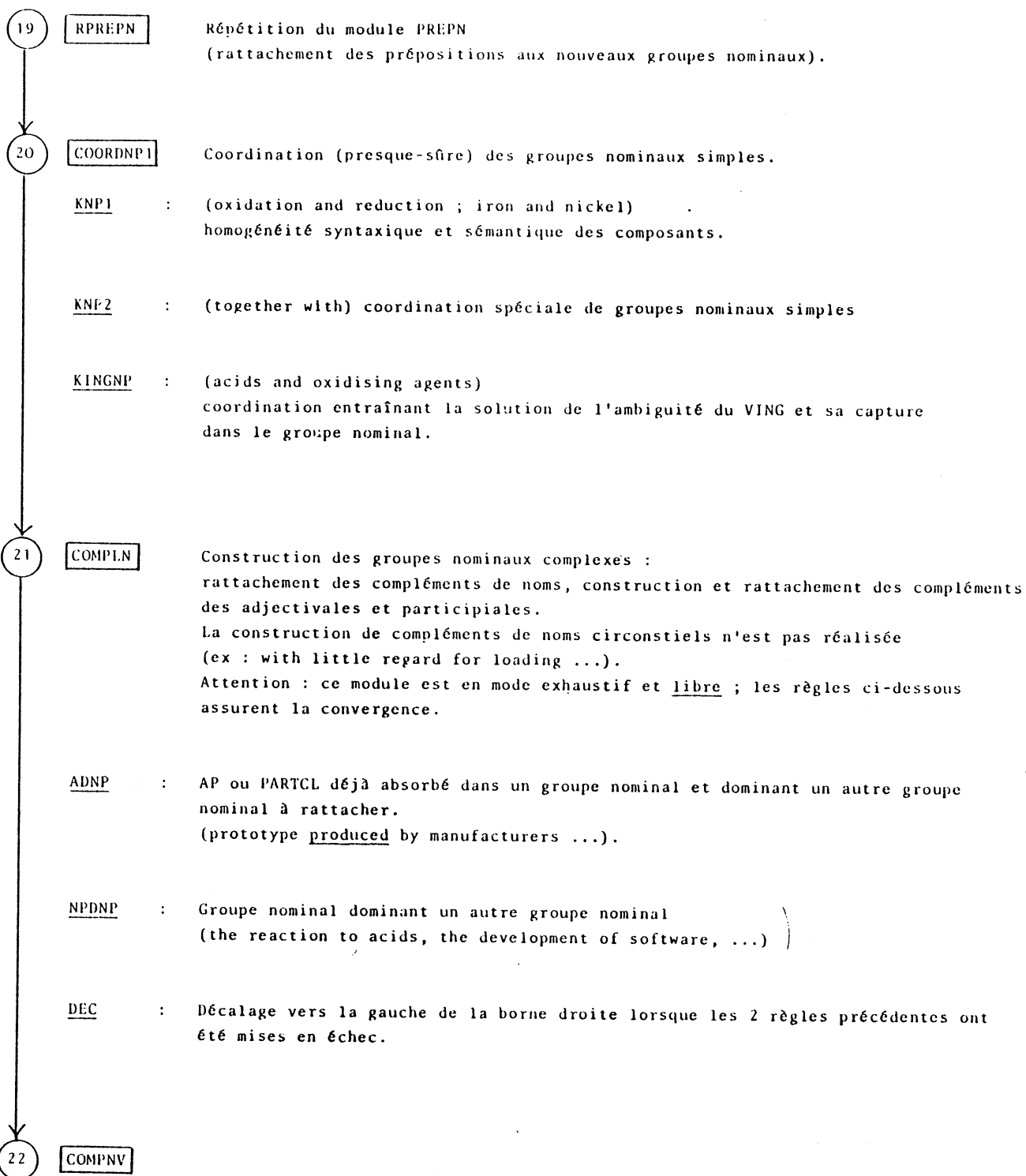
AMBVN7 : ambiguïté Verbe-Nom
(this new design, with adequate profit)
V - N => N - contexte :
Déicteur/préposition/punctuation + AP + V-N
donne un nom.

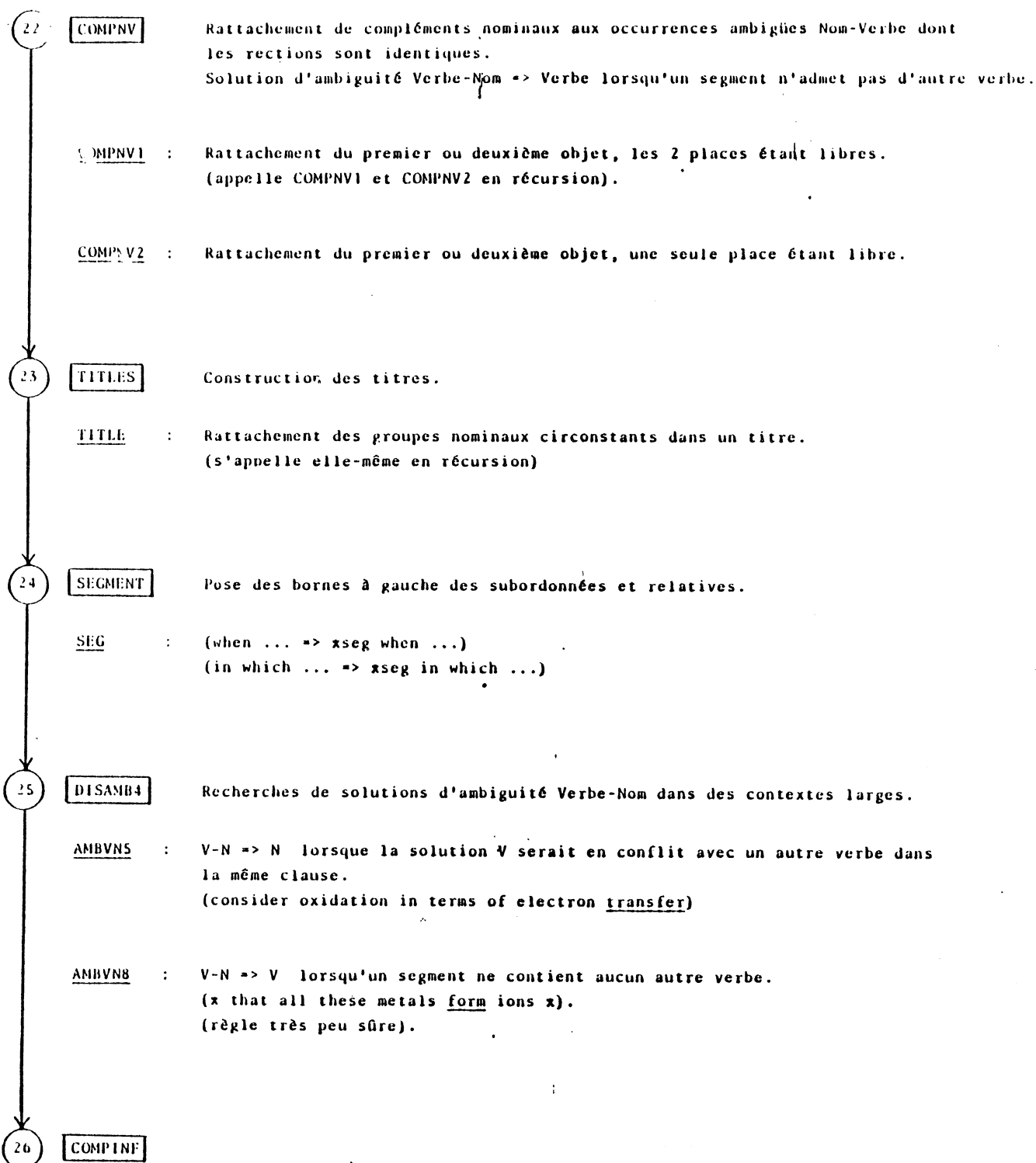
13

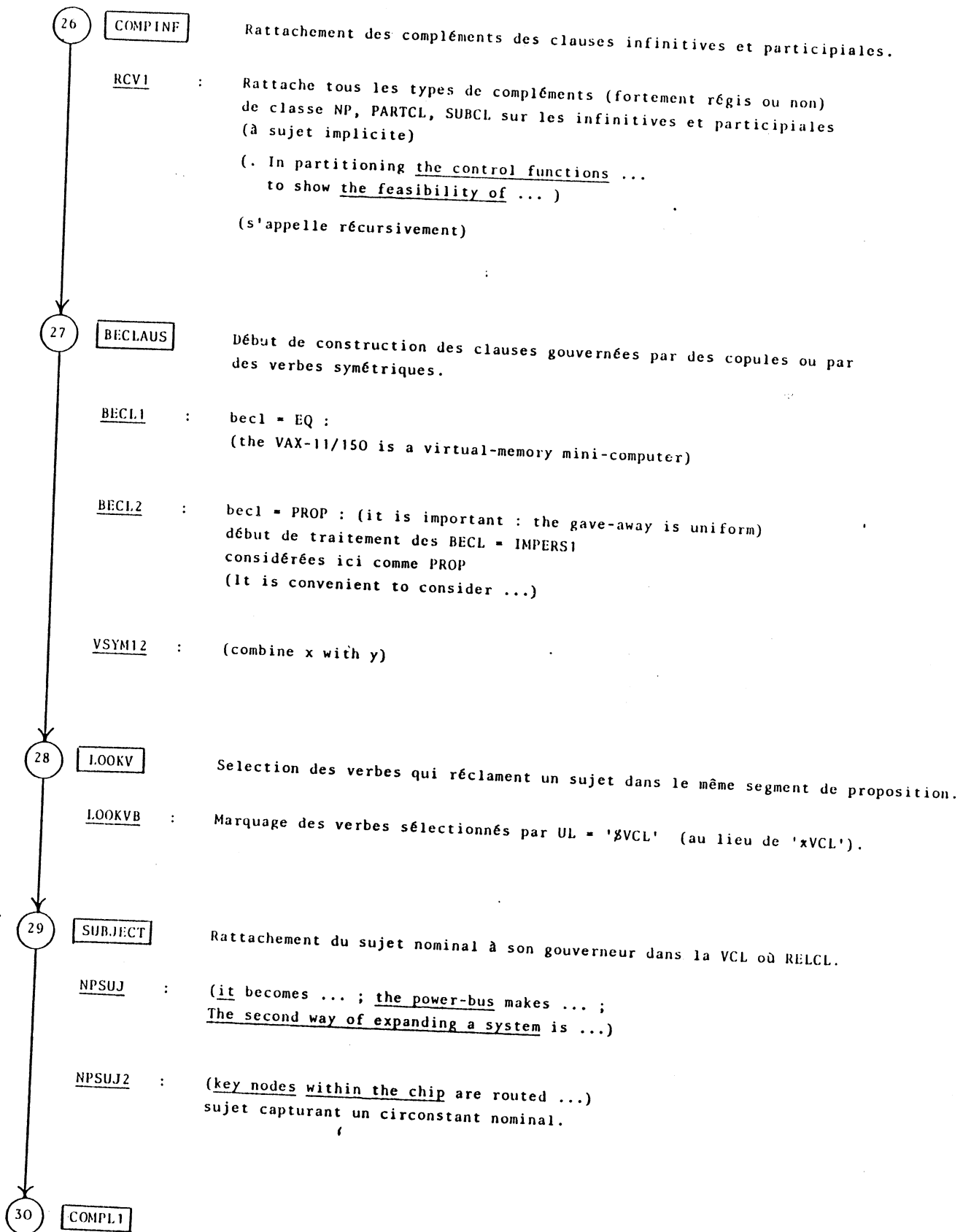
VBGO

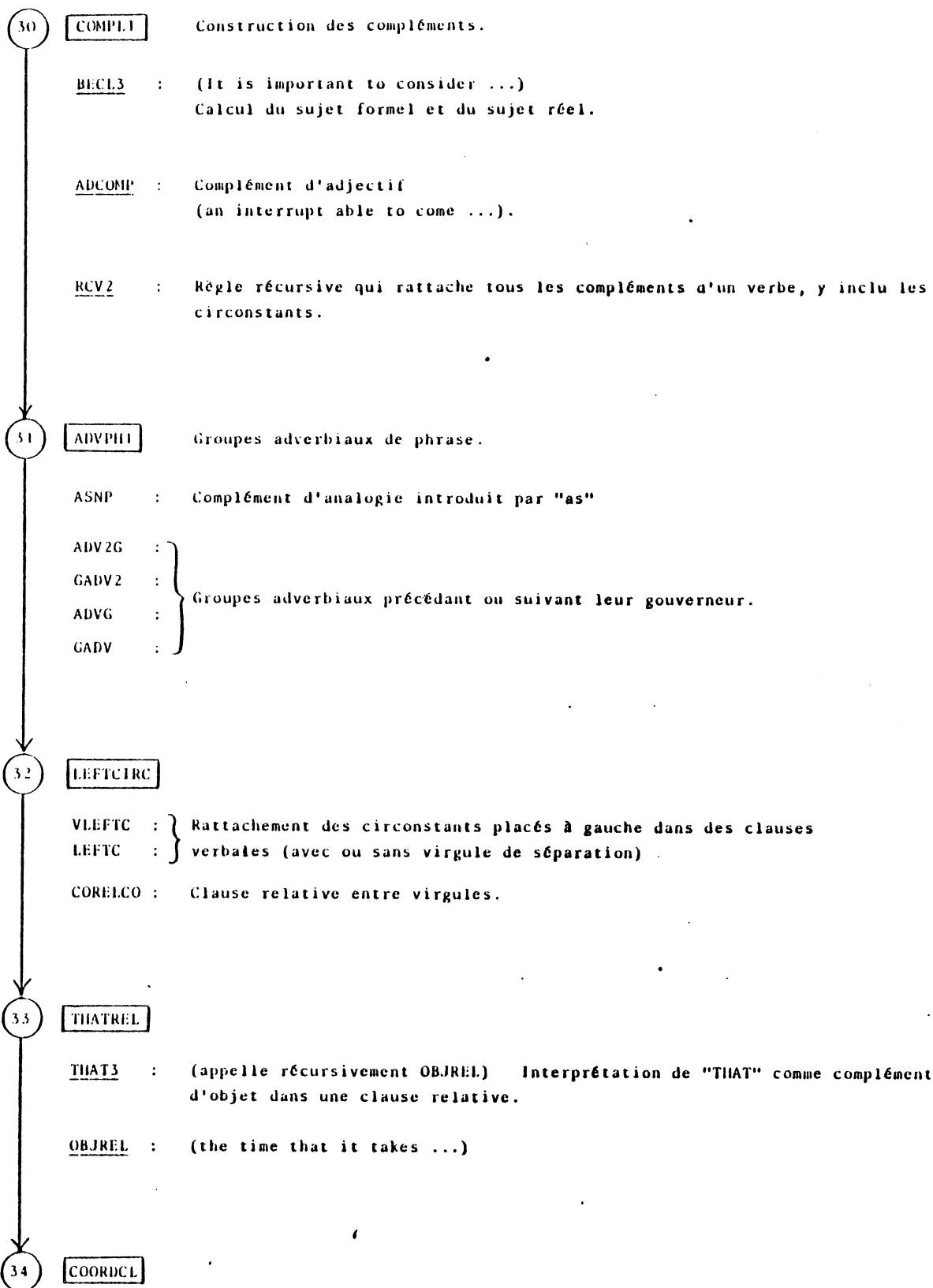












- 34
- COORDCL** : Coordination et subordination de clauses.
- LSUBCL** : Clause subordonnée à gauche de la principale
- GSUBCL** : Rattachement des relatives ou subordonnées par dépendant généralisé au plus bas de la structure
- SUBCL** : (appelle récursivement tout le module)
Ex : when the signals allow the decomposition which is ...
- SUBCL2** : Conjonction de clauses subordonnées en fin de phrase
- RELCL3** : Rattachement des clauses relatives (par dépendant généralisé, au plus bas de la structure)
- RELCL2** :
- RELCL** : Rattachement direct d'une clause relative
- KCL** : Coordination de clauses indépendantes.

- 35
- SST1** : Préparation appel et retour de sous-systèmes transformationnels pour séparer l'analyse phrase par phrase et ne procéder à la suite que pour les phrases compliquées qui n'ont pas encore une structure complète.
- SSTCALL** : (appelle récursivement la suite de modules débutant par RCOMPLN)
- SSTRET** : retour.

- 36
- CLOSURE**

36

CLOSURE

SST1 : Sous-systèmes transformationnels.

(35A)	RCOMPLN - COMPLN	(24)
(35B)	RCOMPINF - COMPINF	(26)
(35C)	RLOOKV - LOOKV	(28)
(35D)	RSUBJECT - SUBJECT	(29)
(35E)	RBECLAUS - BECLAUS	(27)
(35F)	RCOMPLI - COMPLI	(30)
(35G)	RADVPHI - ADVPHI	(31)
(35H)	RLEFT - LEFTCIRC	(32)
(35I)	RCOORDCL - COORDCL	(34)

Itération de l'enchaînement des modules mentionnés ci-contre

SST2 : Rappel éventuel d'un deuxième SST.

SSTCALL : Condition d'appel

SSTRET : retour

Itération identique à SST1.



37

CLOSURE

Clôture de l'analyse multiniveau.

CLOSURE : Effacement des bornes à gauche et à droite de chaque phrase ou titre.

38

ACTGOV

Actualisation des variables sur les gouverneurs.

ACTK : Actualisation du nombre, valences réalisées, verrous, temps, voix, aspects.

39

PIREL

Déplacement des clauses relatives en fonction de choix meilleurs.

RELDPL : Critère de l'accord en nombre.

40

ACTUAL

Actualisation des prépositions et des références.

RSPREP : Actualisation des relations sémantiques des circonstants

ERREG : Efface les répresseurs des groupes en place d'arguments de prédicats

REFPERS1 : Calcul des antécédents des pronoms personnels.



FIN



TABLE BIBLIOGRAPHIQUE



[AHO.73]

A.V.AHO (1973)
"CURRENTS IN THE THEORY OF COMPUTING."
PRENTICE-HALL, INC.

[AH&UL.72]

A.V.AHO & J.D.ULLMAN (1972)
"THE THEORY OF PARSING TRANSLATION AND COMPILING."
VOL.1: "PARSING."
PRENTICE-HALL, INC.

[BO&DA&GU&NE&VA.82]

C.BOITET & P.DAUM-FRAGA & JP.GUILBAUD & N.NEDOREJKINE
& B.VAUQUOIS (1982)
"DEFINITION D'UNE METHODE DE TRAVAIL D'EQUIPE LINGUISTIQUE."
DOCUMENT ESOPE.

[BO&GU&QU.78]

C.BOITET & P.GUILLAUME & M.QUEZEL-AMBRUNAZ (1978)
"MANIPULATION D'ARBORESCENCE ET PARALLELISME:
SYSTEME ROBRA."
COMMUNICATION COLING 78.

[CHAN.78]

CHANG MAY SEE (1978)
"COMPUTER SYSTEM AIDS IN NATURAL LANGUAGES DATA
PROCESSING WITH APPLICATION IN BAHASA MALAYSIA."
THESE UNIVERSITE DE MALYSIE, PENANG.

[CHAU.74]

J.CHAUCHE (1974)
"UN SYSTEME DE CONTROLE ET DE TRANSDUCTION
D'ARBORESCENCES: LE SYSTEME CETA."
GRENOBLE SEMINAIRE DE PROGRAMMATION, JUIN 74.

[CHAU.75]

J.CHAUCHE (1975)
"PRESENTATION DU SYSTEME CETA."
CETA, DOCUMENT GETA, GRENOBLE.

[CH&GU&QU.78]

J.CHAUCHE & P.GUILLAUME & M.QUEZEL-AMBRUNAZ (1978)
"LE SYSTEME ATEF."
DOCUMENT GETA.

[CHOM.57]

N.CHOMSKY (1957)
"SYNTACTIC STRUCTURES."
MOUTON-LA HAYES.

- [CHOM.65]
N.CHOMSKY (1965)
"ASPECT OF THE THEORY OF SYNTAX."
THE M.I.T. PRESS.
- [COLM.71]
A.COLMERAUER (1971)
"LES SYSTEMES-Q OU UN FORMALISME POUR ANALYSER ET
SYNTHETISER DES PHRASES SUR ORDINATEUR."
TAUM 71.
- [EARL.70]
J.EARLEY (1970)
"AN EFFICIENT CONTEXT-FREE PARSING ALGORITHM."
COMMUNICATION OF THE A.C.M., V.13, N.2, P.94-103.
- [GAIF.65]
H.GAIFMAN (1965)
"DEPENDENCY SYSTEMS AND PHASE STRUCTURE SYSTEMS."
INFORMATION AND CONTROL, V.8, N.3, P.304-337.
- [GETA.80]
DOCUMENT GETA (1980)
"ETUDE DE LA VALIDITE DU FORMALISME CHOISI POUR
REPRESENTER LA STRUCTURE INTERFACE."
CONTRAT CEE.
- [GETA.82]
DOCUMENT GETA (AVRIL 1982)
"LE POINT SUR ARIANE-78 DEBUT 82."
VOL.1: LE LOGICIEL.
VOL.2: LES APPLICATIONS LINGUISTIQUES.
- [GUIL.83]
P.GUILLAUME (1983)
"LE MODELE EXPANS."
DOCUMENT GETA.
- [HARR.78]
M.A.HARRISON (1978)
"INTRODUCTION TO FORMAL LANGUAGE THEORY."
ADDISON- WESLEY PUBLISHING COMPANY.
- [HAYS.67]
D.G.HAYS (1967)
"INTRODUCTION TO COMPUTATIONNAL LINGUISTICS."
AMERICAN ELSEVIER, NEW-YORK.
- [HO&UL.69]
J.E.HOPCROFT & J.D.ULLMAN (1969)
"FORMAL LANGUAGES AND THEIR RELATION TO AUTOMATA."
ADDISON WESLEY.

[JAEG.78]

D.JAEGER (1978)
"SYGMOR."
DOCUMENT GETA.

[KASA.65]

T.KASAMI (1965)
"AN EFFICIENT RECOGNITION AND SYNTAX ALGORITHM FOR
CONTEXT-FREE LANGUAGES."
SCI REP AFCRL -65-758.
AIR FORCE CAMBRIDGE RESEARCH LABORATORY, BEDFORD.

[KA&TO.69]

T.KASAMI & K.TORII (1969)
"A SYNTAX ANALYSIS PROCEDURE FOR UNAMBIGUOUS
CONTEXT-FREE GRAMMARS."
JOURNAL OF THE A.C.M., V.16, N.3, P.423-431.

[KAY.79]

M.KAY (1979)
"FUNCTIONNAL GRAMMARS."
XEROX PALO ALTO RESEARCH CENTER, USA
PROCEEDING AT THE FIFTH ANNUAL MEETING OF BERKEEY
LINGUISTIC SOCIETY.

[KAY.82a]

M.KAY (1982)
"UNIFICATION GRAMMAR."
XEROX PALO ALTO RESEARCH CENTER, USA

[KAY.82b]

M.KAY (1982)
"AN ALGORITHM FOR COMPILING PARSING TABLES FROM GRAMMARS."
XEROX PALO ALTO RESEARCH CENTER, USA.

[KULA.59]

O.KULAGINA (1959)
"MACHINE TRANSLATION FROM FRENCH INTO RUSSIAN."
PROBLEMS OF CYBERNETICS, V.3, V.4, PERGAMON PRESS.

[KU&ME.67]

O.S.KULAGINA & I.A.MEL'CHUK (1967)
"AUTOMATIC TRANSLATION: SOME THEORETICAL ASPECTS
AND THE DESIGN OF A TRANSLATION SYSTEM."
IN "MACHINE TRANSLATION", A.D.BOOTH, ED.
NORTH-HOLLAND.

[KU&VA.59]

O.KULAGINA & VAKULOVSKAYA (1959)
"EXPERIMENTAL TRANSLATION FROM FRENCH INTO RUSSIAN
USING A STRELLA COMPUTER."
PROBLEMS OF CYBERNETICS, V.2, PERGAMON PRESS.

[LAMB.66]

S.M.LAMB (1966)
"OUTLINE OF STRATIFICATIONAL GRAMMAR."
GEORGETOWN UNIVERSITY, SCHOOL OF LANGUAGES AND LINGUISTICS.

[LYON.70]

J.LYONS (1970)
"LINGUISTIQUE GENERALE: INTRODUCTION A LA LINGUISTIQUE
THEORIQUE."
LAROUSSE: "LANGUES ET LANGAGES."

[ME&ZH.70]

I.A.MEL'TCHUK & A.K.ZHOLKOVSKIJ (1970)
"SUR LA SYNTHÈSE SEMANTIQUE."
T.A.INFORMATION, V.2, 1970.

[ME&ZH.71]

I.A.MEL'TCHUK & A.K.ZHOLKOVSKIJ (1971)
"CONSTRUCTION D'UN MODELE ACTIF DE LA
LANGUE "SENS-TEXTE"."
DOC. DE LINGUISTIQUE QUANTIFICATIVE.DUNOD, 10.

[NIVA.72]

M.NIVAT (1972)
"LANGAGES ALGEBRIQUES SUR LE MAGMA LIBRE ET SEMANTIQUE
DES SCHEMAS DE PROGRAMMES."
EDITED BY M.NIVAT.
ED. NORTH-HOLLAND PUBLISHING COMPANY.

[PA&QU.68]

C.PAIR & A.QUERE (1968)
"DEFINITION ET ETUDE DES BILANGAGES REGULIERS."
INFORMATION AND CONTROL, V.13, N.6, P.565-593.

[SALO.73]

A.SALOMAA (1973)
"FORMAL LANGUAGES."
ACADEMIC PRESS.

[STEW.75a]

G.STEWART (1975)
"LE LANGAGE DE PROGRAMMATION REZO."
UNIVERSITE DE MONTREAL. RAPPORT DE STAGE TAUM.

[STEW.75b]

G.STEWART (1975)
"MANUEL D'UTILISATION DU LANGAGE DE PROGRAMMATION REZO."
UNIVERSITE DE MONTREAL. RAPPORT DE STAGE TAUM.

[VAUQ.69]

B.VAUQUOIS (1969)
"CALCULABILITE ET LANGAGES."
COURS U.S.M.G.

[VAUQ.75]

B.VAUQUOIS (1975)
"LA TRADUCTION AUTOMATIQUE A GRENOBLE."
DOCUMENT DE LINGUISTIQUE QUANTITATIVE, DUNOD.

[VAUQ.78]

B.VAUQUOIS (1978)
"DESCRIPTION DE LA STRUCTURE INTERMEDIAIRE."
COMMUNICATION PRESENTEE AU COLLOQUE DE LUXEMBOURG, AVRIL 78.

[VA&DA.82]

B.VAUQUOIS & P.DAUM FRAGA (FEVRIER 1982)
"ETUDE DE LA FAISABILITE D'UN SYSTEME DE TRADUCTION
AUTOMATISEE ANGLAIS-FRANCAIS. RAPPORT FINAL."
DOCUMENT GETA.

[WOOD.70]

W.WOODS (1970)
"TRANSITION NETWORKS GRAMMARS FOR NATURAL LANGUAGES
ANALYSIS."
COMMUNICATION OF THE A.C.M., V.13, N.10, P.591-606.

[WOOD.71]

W.WOODS (1971)
"AN EXPERIMENTAL PARSING SYSTEM FOR TRANSLATION
NETWORK GRAMMARS."
DANS "NATURAL LANGUAGE PROCESSING, COURANT COMPUTER
SCIENCE SYMPOSIUM 8." EDITED BY R.RUSTIN.
ACADEMIC PRESS.

[YOUN.67]

H.D.YOUNGER (1967)
"RECOGNITION AND PARSING OF CONTEXT-FREE LANGUAGES
IN TIME N³."
INFORMATION AND CONTROL, V.10, N.2, P.189-208.



AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974,

VU le rapport de présentation de Monsieur B. VAUQUOIS, Professeur

Mademoiselle CHAPPUY Sylviane

est autorisée à présenter une thèse en soutenance pour l'obtention du titre de
DOCTEUR de TROISIEME CYCLE, spécialité "Informatique".

Fait à Grenoble, le 13 juin 1983

Le Président de l'I.N.P.-G.

D. BLOCH
Président
de l'Institut National Polytechnique
de Grenoble

P.O. le Vice-Président,



THESE DE DOCTEUR DE 3EME CYCLE
INFORMATIQUE

Auteur: Sylviane CHAPPUY.

Titre: Formalisation de la description des niveaux d'interprétation des langues naturelles. Etude menée en vue de l'analyse et de la génération au moyen de transducteurs.

Date: 2 juillet 1983.

Mots clés: Grammaires formelles, Transducteurs, Traduction Automatisée, Grammaires Statiques.

Résumé: Présentation des problèmes liés à la traduction automatisée et des solutions apportées par les transducteurs. Etablissement d'un formalisme de représentation de la langue au moyen de grammaires statiques qui sont des grammaires de description dans un modèle multiniveau. Ces grammaires statiques sont présentées sous forme de planches établissant des correspondances entre les chaînes de la langues et les constructions à multiples niveaux d'interprétation associées.

