Analyse syntaxique automatique du français écrit : applications à l'indexation automatique

Geneviève Lallich-Boidin

UNIVERSITE
DES SCIENCES SOCIALES
DE GRENOBLE

ECOLE NATIONALE SUPERIEURE
DES MINES DE SAINT-ETIENNE

THESE DE DOCTORAT

présentée par

Geneviève LALLICH-BOIDIN

Spécialité : Informatique en Sciences Sociales

Analyse syntaxique automatique du français

Applications à l'indexation automatique

Soutenue à Grenoble le 8 Octobre 1986, devant la commission d'examen :

M. Jacques **ROUAULT** , *Professeur à l'Université Grenoble II*

M. Alain **BERRENDONNER**, *Professeur à l'Université de Fribourg*

M. Jacques **COURTIN**, *Professeur à l'Université Grenoble I*

M. Michaël **GRIFFITHS**, *Professeur à l'Université d'Aix-Marseille III*

M. Bernard **PEROCHE**, *Professeur à l'Ecole des Mines de Saint-Etienne*

UNIVERSITE
DES SCIENCES SOCIALES
DE GRENOBLE

ECOLE NATIONALE SUPERIEURE
DES MINES DE SAINT-ETIENNE

THESE DE DOCTORAT

présentée par

Geneviève LALLICH-BOIDIN

Spécialité : Informatique en Sciences Sociales

Analyse syntaxique automatique du français

Applications à l'indexation automatique

Soutenue à Grenoble le 8 Octobre 1986, devant la commission d'examen :

M. Jacques **ROUAULT** , *Professeur à l'Université Grenoble II*

M. Alain **BERRENDONNER**, *Professeur à l'Université de Fribourg*

M. Jacques **COURTIN**, *Professeur à l'Université Grenoble I*

M. Michaël **GRIFFITHS**, *Professeur à l'Université d'Aix-Marseille III*

M. Bernard **PEROCHE**, *Professeur à l'Ecole des Mines de Saint-Etienne*

REMERCIEMENTS

Qu'il me soit permis ici d'exprimer à tous ceux qui ont accepté de juger ce travail ainsi qu'à tous ceux qui m'ont aidée à le mener à terme, mes plus vifs remerciements.

A Jacques Rouault qui préside ce jury. C'est à lui que je dois d'avoir vécu cette grande aventure. Il a suivi ce travail de sa genèse à son terme et j'ai toujours trouvé auprès de lui et de l'équipe qui l'entoure, des conseils scientifiques éclairés et des encouragements sans réserve.

A Alain Berrendonner qui, avec patience et non moins d'humour, a tenté de me faire comprendre le fonctionnement d'une langue que je croyais connaître.

A Bernard Péroche qui m'a permis de terminer cette thèse dans d'excellentes conditions morales et matérielles et qui, de plus, a accepté de lire très attentivement ce travail et de le juger.

A Michael Griffiths pour l'intérêt qu'il a porté à la lecture de cette thèse et pour sa présence aujourd'hui.

A Jacques Courtin pour avoir accepté de participer au jury et apporter ainsi toute sa compétence dans le domaine que nous traitons.

A tous mes amis de l'équipe SYDOG, Georges Antoniadis, Yolla Polity, Carmen Rodriguez, Ramon Alvarez, Ghassan Kallas, Dominique Maret, Ghassan Kahwati avec lesquels travailler est toujours un plaisir.

A tous les membres du C.R.I.S.S. et plus particulièrement Jean-Pierre Fauché et Gérard Henneron qui m'ont tirée de quelques pas délicats.

A tous mes copains du Département Informatique de l'Ecole des Mines de Saint-Etienne qui m'ont toujours soutenue dans cette rude entreprise, et ont toujours supporté mes absences, même lorsque j'étais présente : Jean-Jacques, Sega, Sabine, Paul-André, Florence, Jacques, Jonathan, Ehoud, François, Clément, Roland, Marie-Line, Antoine, Christian...

A Stéphane sans le soutien quotidien duquel ce travail n'aurait pu aboutir.

A Chloé, Anaïs et Gaspard dont les rires et les pleurs m'ont permis de garder les pieds sur terre.

A **groff**, formateur de texte conçu par Jean-Jacques Girardot, pour la réalisation matérielle de ce rapport.

Et enfin, à ma brave voiture qui m'a inlassablement transportée entre Saint-Etienne et Grenoble.

SOMMAIRE

Introduction

"Ainsi les idées, dont j'usais précédemment pour me figurer un cheval que je n'avais pas encore vu, étaient de purs signes, comme les empreintes sur la neige étaient des signes de l'idée de cheval : et on use des signes et des signes de signes dans le seul cas où les choses nous font défaut."

Umberto ECO, "Le nom de la rose"
Paris, Grasset, 1982.

Ce travail s'inscrit dans un projet beaucoup plus large mené par le groupe de recherche SYDO (SYstèmes DOcumentaires). Ce groupe réunit informaticiens, linguistes et documentalistes associés aux équipes de recherche suivantes :

le Centre de Recherches Linguistiques et Sémiologiques de l'Université Lyon II, dirigé par M. Le Guern,

le Département de Linguistique Française de l'Université de Fribourg (Suisse), animé par A. Berrendonner,

le Laboratoire d'Informatique Documentaire de l'Université Claude Bernard (Lyon I) sous la responsabilité de M. Bouché,

le CRISS, Centre de Recherche en Informatique et Sciences Sociales de l'Université des Sciences Sociales de Grenoble, au sein duquel l'équipe SYDOG (SYDO Grenoble) travaille sous la direction de J. Rouault.

L'objectif du projet SYDO est de construire un système documentaire automatisé à partir des données que sont d'une part le texte des documents et d'autre part les questions des utilisateurs. Les documents textuels et les questions ont en commun de s'exprimer à travers la langue naturelle. C'est pourquoi ce projet s'insère dans le domaine du traitement automatique des langues naturelles.

Ce domaine est immense puisqu'il englobe reconnaissance et génération d'une ou plusieurs langues sous forme écrite ou orale. Le cadre du projet est de dimension plus restreinte. En effet, il ne prend en compte qu'une seule langue, la langue française. Cette limitation peut sembler soit peu réaliste dans la mesure où la plupart des fonds documentaires sont multilingues, soit encore simplificatrice puisque l'on affronte les difficultés d'une seule langue. En fait, cette limitation est dictée par la conviction qui est la nôtre et qui s'exprime ainsi :

> "la langue est liée, doublement, aux habitudes de pensée car si elle est bien le véhicule des idées, ce n'est qu'en les contraignant et en leur imposant à son tour sa structure". [J. ROUAULT, linguistique automatique et informatique documentaire, colloque Franco-Anglais, Déc. 1984]

Cette conviction remet en cause toutes les tentatives d'élaboration de schéma conceptuel universel et par la-même, toute transposition d'un concept d'une langue dans une autre au moyen d'un tel outil. En contrepartie, elle nous permet de formuler l'hypothèse suivante : une étude approfondie du fonctionnement de la langue est un moyen d'accès aux idées véhiculées par cette langue. Or, si le traitement automatique des langues naturelles a su franchir le cap de l'analyse syntaxique grâce aux travaux de Chomsky, il bute toujours inexorablement sur les niveaux sémantique et pragmatique [H.L. DREYFUS, 1984]. C'est donc par une analyse approfondie du fonctionnement d'une langue, en l'occurrence le français, que nous tenterons d'en approcher les idées contenues. De plus, la nature textuelle des documents traités nous permet de limiter notre étude à celle du français écrit.

L'application visée est la réalisation d'un système documentaire. Classiquement, un tel système se présente comme une chaîne de traitements au sein de laquelle convergent documents et questions. Chaque document du fonds est soumis à l'opération d'indexation. L'indexation consiste à appréhender le contenu d'un document, puis à en retenir les notions principales, et enfin à exprimer ces notions à l'aide de descripteurs.

> "un descripteur est un mot ou un groupe de mots retenus dans un thesaurus et choisi parmi un ensemble de mots équivalents pour représenter sans ambiguïté une notion contenue dans un document ou dans une demande de recherche documentaire" [norme AFNOR, NF Z 47-100, 1981]

> "un thesaurus est un ensemble de termes (descripteurs ou non descripteurs) et de relations qui précisent leur environnement sémantique" [idem]

D'un autre côté, les questions des utilisateurs formulées en langue naturelle, sont transcrites sous forme d'une expression booléenne dont les opérandes sont des descripteurs. Un système documentaire classique compare l'expression de la question aux images des documents et extrait du fonds les références de tous les documents sélectionnés.

Ce schéma séquentiel d'un système documentaire classique : indexation, stockage, interrogation, peut être décomposé différemment. En effet, la transcription d'un document et celle d'une question en termes de descripteurs sont deux opérations intellectuelles que l'on peut isoler des processus de stockage et de tri qui eux sont très mécaniques, et donc aisément

automatisables. L'automatisation des systèmes documentaires bute à l'heure actuelle, sur les problèmes posés par l'automatisation de l'indexation et de l'interrogation.

Jusqu'alors, de nombreuses méthodes d'indexation et de recherche automatisées ont été élaborées. Les premières reposent sur des modèles statistiques et probabilistes [ROUAULT, 1986, ch. 3 et 4], [VAN RIJSBERGEN, 1979]. Ces méthodes qui donnent des résultats satisfaisants sur la langue anglaise, sont beaucoup moins performantes sur le français. C'est pourquoi, plusieurs logiciels à finalité documentaire ont été réalisés pour traiter notre langue. Les logiciels que nous avons choisi de présenter ici, sont ceux dont nous avons eu connaissance au travers des congrès et des publications. Ils ont en commun d'être conçus pour traiter des textes rédigés en français et de ce fait d'inclure des traitements linguistiques.

Les logiciels PIAFDOC et PIAFPS

Le logiciel PIAF [COURTIN, 1977] est un logiciel interactif et modulaire d'analyse morpho-syntaxique de la langue naturelle. Il a été appliqué à divers domaines parmi lesquels la documentation avec les versions PIAFDOC [GRANDJEAN, VEILLON, 1980] et PIAFPS [MERLE, 1982]. Ces logiciels ont une fonction d'aide à l'indexation et requièrent de ce fait l'intervention d'un opérateur ; ils doivent donc être exécutés en temps réel.

Le logiciel PIAFDOC est une adaptation de l'analyseur morphologique (transducteur d'états finis) à la base de données textuelles BIPA de la Documentation Française. L'analyseur s'appuie sur un dictionnaire contenant l'ensemble des bases, des flexions, des mots composés et des locutions. A l'issue de l'analyse morphologique, le texte est segmenté en mots (simples ou composés), à chaque mot valide étant associé une ou plusieurs analyses possibles. Pour l'application documentaire, le dictionnaire est enrichi de renseignements tels que : mot vide, synonyme préférentiel, mot composé. Ainsi les formes du texte sont-elles normalisées :

- les noms et adjectifs sont représentés par leur base
- les verbes et participes par la forme infinitive
- les mots composés sont localisés
- les mots vides sont éliminés
- les synonymes préférentiels sont substitués aux mots équivalents.

Lorsqu'une même forme accepte plus d'une analyse possible, le programme s'interrompt ; or la plupart des interventions consistent à lever des homographies, c'est-à-dire à choisir, pour une même forme, entre plusieurs catégories grammaticales ; les interventions pour des polysémies sont plus rares. D'où la version PIAFPS de PIAFDOC qui contient en plus un module de levée d'homographies ; ce module repose sur un automate d'états finis conduit par un ensemble de règles linguistiques. Ces règles peuvent être modifiées en fonction des cas rencontrés et des corpus analysés.

En résumé les logiciels PIAFDOC et PIAFPS sont des logiciels d'aide à l'indexation automatique, et aussi à la recherche documentaire, car une question est soumise à un traitement similaire. Ils sont adaptables au corpus traité puique le contenu du dictionnaire ainsi que les règles peuvent être modifiés au moyen d'éditeurs. Ils sont soumis à des contraintes de temps d'exécution puisqu'ils opèrent en temps réel : c'est la raison pour laquelle ils ne

procèdent pas à l'analyse syntaxique des textes, bien que les logiciels adéquats existent au sein de PIAF. Enfin, ils s'intègrent dans l'architecture classique des systèmes documentaires.

Le groupe "Systèmes intelligents de recherche d'informations" du Laboratoire Génie Informatique de l'IMAG.

Cette équipe a conçu un système documentaire intégrant à l'heure actuelle la constitution d'un thesaurus et l'indexation automatique. La phase d'interrogation est en cours de réalisation. Ce logiciel est expérimenté dans le cadre du projet Concerto, sur un corpus de "Normes d'exploitation et de fonctionnement des autocommutateurs" (NEF).

La première phase consiste à construire un thesaurus à partir de l'ensemble du corpus [BRUANDET, 1985]. Pour ce faire, le corpus est d'abord soumis à une analyse morphologique sommaire donnant pour chaque forme, sa catégorie morphologique et son représentant privilégié (terme). Puis, partant de l'hypothèse que deux termes souvent voisins dans un texte sont sémantiquement proches, on établit une mesure qui rende compte de la force de liaison de chaque couple de termes à partir de leur proximité dans le texte, de leur catégorie et de leur nombre de cooccurrences. D'où un graphe dont on extrait les sous-graphes complets maximaux ou cliques. Les sommets d'une même clique sont des termes fortement liés au sens de la mesure définie. Une clique est interprétée comme un concept important du corpus, ses sommets sont les termes composant ce concept dans le texte. De plus un même sommet peut appartenir à plusieurs cliques, donc l'ensemble des cliques est une représentation des concepts du corpus et de leurs relations sémantiques.

L'indexation des textes du même corpus intervient ensuite [CHIARAMELLA, KERKOUBA, 1984]. Elle tient compte de la structuration des documents, et distingue donc, par exemple, l'indexation des titres de celle des paragraphes de texte. L'indexation des textes consiste à détecter les syntagmes nominaux et à trouver pour chacun d'eux la plus grande partie incluse dans une clique. On met alors en correspondance des cliques de termes et des syntagmes réalisés en surface. Les syntagmes représentés par une ou plusieurs cliques sont retenus comme mots clés, les autres rejetés.

La procédure de recherche, telle qu'elle est prévue [BRUANDET, CHIARAMELLA, KERKOUBA, 1985], consiste à aider l'utilisateur à formuler sa requête, sous forme d'une expression booléenne de mots clés, puis à retrouver en utilisant le thesaurus, les parties de documents répondant à la requête.

Ce projet accorde une part essentielle aux traitements statistiques ; les traitements linguistiques servent à améliorer le rendement du modèle statistique ; ils restent cependant assez sommaires afin d'éviter la production d'ambiguïtés. Il est actuellement testé sur un corpus en grandeur réelle : l'analyse des résultats obtenus permettra alors de juger de l'efficacité d'un tel système.

Le logiciel SPIRIT [FLUHR, 1982, 1984]

SPIRIT est un logiciel documentaire qui inclut indexation automatique des documents textuels et recherche dans une base de données à partir d'une question formulée en langue naturelle. Il associe traitements linguistiques et statistiques.

Les traitements linguistiques ont pour objet :

1 de résoudre les synonymies, c'est-à-dire de détecter les différentes formes d'une même unité de sens pour les remplacer par un représentant unique.

2 de lever les homographies : distinguer deux formes identiques qui recouvrent des unités de sens distinctes.

3 de relever les relations de dépendance syntagmatique qui lient les mots dans une phrase.

4 de compresser l'information

La réalisation de ces objectifs met en jeu différents niveaux d'analyse linguistique : morphologie flexionnelle et dérivationnelle, syntaxe, sémantique et pragmatique, ainsi qu'un dictionnaire important (250.000 entrées).

Les traitements statistiques opèrent sur les textes et les questions, après le traitement linguistique. Ils classent les textes en fonction de leur degré de pertinence à une question.

Ce logiciel, dont on connaît surtout les fonctionnalités externes, est d'un grand intérêt : en effet, il est opérationnel et donc validé sur un nombre de documents significatif. En celà, les conclusions de C. FLUHR sont à prendre en compte :

> "Les différences d'efficacité entre les techniques (statistiques) sont minimes. Elles sont en fait très faibles vis-à-vis de l'influence de la qualité des traitements linguistiques sur la statistique. En effet, la non-reconnaissance des synonymes et des homographes perturbent beaucoup la statistique."
>
> [C. FLUHR, Bulletin du CID, no5, 1982]

Un autre intérêt du logiciel SPIRIT est de proposer une alternative partielle aux opérateurs booléens : les relations de dépendance syntagmatiques à la fois plus précises et plus proches du fonctionnement de la langue naturelle.

Le projet SYDO, comme les logiciels PIAF et SPIRIT, accorde la priorité aux traitements linguistiques. Cependant, ces deux derniers logiciels font intervenir des connotations sémantiques dans le dictionnaire, connotations utilisées, par exemple, pour détecter les synonymes ; or, les traits sémantiques attachés à une forme sont très dépendants du contexte, ils ne sont donc pas toujours transposables d'un contexte dans un autre, et réduisent ainsi le domaine d'application du logiciel. C'est en cela que le projet SYDO est original : l'analyse morpho-syntaxique d'un texte est une voie d'accès à son contenu informatif, de ce fait les

traitements linguistiques sont applicables à des textes scientifiques indépendamment du domaine traité. Les informations de nature sémantique interviendront ultérieurement, si besoin est, et leur portée sera limitée à un domaine précis.

Le sujet du travail présenté ici est l'analyse syntaxique des syntagmes nominaux, à des fins d'indexation. Le premier chapitre présentera l'ensemble des étapes préalables à l'analyse syntaxique des syntagmes nominaux, et déjà réalisées par l'équipe de recherche : ces étapes nous conduisent du texte brut à l'ensemble des syntagmes nominaux maximaux du texte.

Le deuxième chapitre est une présentation des analyseurs hors contexte existants. Il nous a paru en effet nécessaire de faire le point sur les outils informatiques adaptés à notre problème et de donner les raisons pour lesquelles l'algorithme d'Earley a été retenu.

Le troisième chapitre montre la nécessité d'adopter une stratégie d'analyse fondée sur des données linguistiques. En effet, le fonctionnement de la langue n'est pas purement combinatoire ; il obéit à des règles plus strictes qui permettent d'éliminer un grand nombre de solutions proposées par un analyseur de type combinatoire, comme celui d'Earley.

Le quatrième chapitre est axé sur les outils linguistiques nécessaires à la mise en oeuvre de la stratégie d'analyse : présentation d'une grammaire hors-contexte du syntagme nominal, définition des indicateurs de structure, aspects linguistiques de la stratégie et de ses limites.

Puis le cinquième chapitre est consacré aux aspects informatiques. Il décrit les fichiers et les adaptations de l'algorithme d'Earley à la stratégie d'analyse.

Enfin, le dernier chapitre traite des résultats obtenus par l'analyseur, sur un corpus de résumés d'articles scientifiques et propose, à partir de ces résultats, une représentation du contenu des résumés.

Chapitre 1

LE PROJET SYDO

Deux idées-forces de ce modèle, qui reviendront tout au long de ce chapitre comme un leitmotiv s'expriment en ces termes :

Elimination des ambiguïtés

La langue naturelle est intrinsèquement ambiguë. Nous l'admettons, et ne lèverons jamais toutes les ambiguïtés. Cependant, le traitement automatique nous impose d'être vigilants. En effet, si une étape du traitement crée des ambiguïtés, la suivante y ajoutera les siennes. Et comme, un système automatique ne peut choisir entre plusieurs solutions linguistiquement admissibles, il les conserve toutes. Ainsi de quelques ambiguïtés peuvent naître de très nombreuses solutions. La vigilance requise s'exercera sur deux points : d'une part, en veillant lors de chaque étape à ne pas créer des ambiguïtés superflues, d'autre part en éliminant dès que possible, les solutions parasites, c'est-à-dire au moment où l'on dispose de l'information pour ce faire. On espère ainsi éviter la création de solutions parasites, que l'on ne pourrait éliminer dans une phase ultérieure, les informations nécessaires ayant disparu.

Régularisation

Le fonctionnement de la langue naturelle respecte le plus souvent des règles peu nombreuses. Néanmoins, toute règle érigée appelle en contrepartie une liste d'exceptions. Ces exceptions peuvent à leur tour nécessiter d'autres règles qui auront un domaine d'application très restreint ; elles auront cependant le même statut que les règles à haut rendement. Le risque est alors grand de se trouver face à beaucoup de règles, parmi lesquelles quelques

unes seront souvent utilisées, et les autres, très nombreuses, quasiment jamais. Outre le gaspillage d'espace et de temps, cette solution présente un inconvénient majeur : des règles trop nombreuses ont des effets difficilement contrôlables et peuvent alors engendrer des solutions parasites.

Pour éviter cet écueil, la solution retenue pour traiter exceptions et cas particuliers, est de les rapporter au cas général par un procédé de régularisation, matérialisé par une seule règle ayant un domaine d'application bien défini. Les exceptions, ainsi régularisées, rejoindront pour la suite du traitement, le cas général.

Ces deux idées-forces nous serviront de pivot pour esquisser les fonctions de chacune des étapes préalables à l'analyse syntaxique. Nous ne reviendrons pas ici sur tous les aspects de chacune des étapes car ils ont déjà fait l'objet de plusieurs publications : aspects méthodologiques [ROUAULT 1983 et 1986], aspects linguistiques [BERRENDONNER, 1983], aspects informatiques [ANTONIADIS, 1984 et GALIOTOU, 1983].

1. SAISIE DU TEXTE ET PRETRAITEMENT MORPHOGRAPHIQUE

Puisque l'objet traité est du texte écrit en français, la première phase a trait à la saisie de ce texte. La saisie s'effectue à l'aide d'un éditeur qui, outre les fonctions classiques de gestion de fichiers, prétraite la graphie du texte dans le double but annoncé :

1.1. ELIMINATION DES AMBIGUITES

L'éditeur de textes impose des conventions de saisie [ANTONIADIS, 1984], afin que chaque forme (mot ou ponctuation) soit séparée des formes voisines par un blanc. L'éditeur permet de plus :

la distinction du point-abréviation et du point-ponctuation : le premier est accolé à la lettre qu'il suit. Le second est cerné par des espaces.

```
"M. le Préfet a inauguré le stade . "
```

la prise en compte des signes diacritiques pour distinguer des mots voisins comme : élève et élevé.

1.2. REGULARISATION

Le même éditeur prétraite le texte à des fins de régularisation. Voici quelques exemples :

1.2.1. Les majuscules

Les caractères majuscules marquant un début de phrase, sont remplacés par le même caractère minuscule. Cette substitution se justifie dans la mesure où dans une phase ultérieure, l'analyse morphologique, il y a consultation d'un dictionnaire. Ce dictionnaire, pour des raisons d'espace et de cohérence, ne contiendra pour chaque mot, qu'une seule

entrée , celle correspondant au premier caractère minuscule.

Cette substitution n'est pas automatique car les noms propres, les abréviations débutent par une majuscule qui est alors matérialisée par un signe spécial, "\", (M devient \m).

1.2.2. Les formes élidées

Pour des raisons de syllabation, notre langue admet des formes comme : "j' ", "l' ", "s' ", "t' ", "n' ", "m' ", "c' ", "d' ". On les remplacera par leur forme complète quand elle est unique. Ainsi :

```
"j' " -> "je" "t' " -> "te" "n' " -> "ne"
"m' " -> "me" "c' " -> "ce" "d' " -> "de".
```

Sinon on conservera la forme élidée :

```
"l' " représente "le" ou "la",
"s' " représente "se" ou "si"
```

sauf dans le contexte suivant :

```
"s'il" -> "si il"    "s'ils" -> "si ils".
```

1.2.3. Consonnes géminées et quelques autres séquences de lettres

Pour simplifier l'analyse morphologique flexionnelle des noms, adjectifs et verbes, les substitutions suivantes seront systématiquement effectuées.

```
ll -> l*        ss -> s*        tt -> t*
nn -> n*        ch -> c*        v -> f*
uë  -> u*e      gui -> g*i      gua -> g*a
gue -> g*e      guo -> g*o
gea -> g_a      geo -> g_o
```

Grâce à ces substitutions, les adjectifs : bon, cruel et ambigu suivent le même modèle de flexions [, s , *e , *es], car

```
   bon        bon-s       bon-*e       bon-*es
 cruel      cruel-s     cruel-*e     cruel-*es
ambigu     ambigu-s    ambigu-*e    ambigu-*es
```

Il en est de même pour les verbes placer et nager. Notons ici que la cédille du c se note c_. Ainsi

```
plac-e    plac-es    plac-_ons    plac-ez    plac-ent
 nag-e     nag-es     nag-_ons     nag-ez     nag-ent
```

Ces substitutions de chaînes de caractères, qui du texte initial :

```
" la matière sèche du couvert végétal "
```

donne l'image suivante :

```
"la matie're se'c*e du couf*ert f*e'ge'tal "
```

ne se justifient que par rapport aux traitements ultérieurs.

2. PRETRAITEMENT MORPHOSYNTAXIQUE : L'ECLATEMENT DES AMALGAMES

Ce prétraitement s'effectue sur certaines formes du texte de surface, dites amalgames. Les amalgames sont soit de nature orthographique ("lesquels", "pourquoi"), soit de nature morphologique (aux, des). Il leur substitue une séquence de formes équivalentes, d'usage plus courant, et donc régularise le texte. Simultanément, il associe à ces dernières des informations de nature morphologique et syntaxique, induites par la transformation. En effet, ces informations sont présentes au moment de l'éclatement, et seulement à cet instant. Leur omission autoriserait les traitements suivants à créer des ambiguïtés sur des formes non ambiguës et donc des solutions parasites. Ici, élimination d'ambiguïtés et régularisation s'associent dans une même opération.

Soit, par exemple, l'amalgame "lesquels". On lui substituera les deux formes "les"+"quels". Mais on sait aussi que "les" est un déterminant (et non une particule préverbale), et l'on sait encore que "les" est de genre masculin (et non féminin). Alors, "lesquels" se réécrira : ("les", D [MAS]) + "quels").

La liste des amalgames et de leur décomposition est donnée dans [BERRENDONNER, 1983-b]. L'algorithme de recherche des amalgames et de leur substitution utilisé par G. ANTONIADIS est une variante de celui de AHO et CORASICK [1975].

3. L'ANALYSE MORPHOLOGIQUE

Il s'agit ici de l'analyse morphologique flexionnelle, c'est-à-dire, de celle qui, à partir d'une forme du texte, en donne la base (ou entrée dans le dictionnaire) et les flexions. Ce découpage d'une forme en base et flexion, permet d'inférer sur cette forme, des renseignements sur son comportement linguistique. L'analyse morphologique s'inscrit dans le processus de traitement de la langue naturelle. Elle associe à chaque forme du texte prétraité, une catégorie et des variables - ou plusieurs catégories en cas d'ambiguïté. Elle n'est qu'une étape du processus global et ne se justifie que par lui ; les résultats qu'elle fournit doivent donc faciliter les opérations ultérieures, tout d'abord l'analyse syntaxique.

3.1. LES ASPECTS LINGUISTIQUES

3.1.1. Les catégories

Elles représentent les grandes classes à l'intérieur desquelles les formes peuvent être réparties, en fonction de leur rôle syntaxique. Ce sont :

F : les noms et adjectifs
V : les verbes
D : les déterminants
Y : les particules préverbales
C : les conjonctions de coordination
Q : les conjonctions de subordination
W : les adverbes
T : les ponctuations
P : les prépositions
G : les négations
H : les prophrases : "oui", "si".

Ces catégories sont définies sur des critères distributionnels en accord avec les principes énoncés : régularisation et élimination des ambiguïtés. En voici une illustration.

3.1.1.1. F regroupe les noms et les adjectifs.

On opposera noms et adjectifs sur les critères distributionnels suivants :

X est un nom si la construction "c'est un X" est possible,
et si la construction "il est X " ne l'est pas.

X est un adjectif si la construction "il est X" est possible,
et si la construction "c'est un X " ne l'est pas.

X est nom ou adjectif si les deux constructions sont simultanément acceptées.

Ainsi "fauteuil" est un nom, "épouvantable" un adjectif, mais "artiste" est à la fois nom et adjectif.

Si par contre, on avait opté pour deux catégories distinctes, "nom" et "adjectif", alors chaque mot du type "artiste" appartiendrait à chacune de ces catégories. D'où la création d'une ambiguïté que l'analyse syntaxique ne pourrait pas toujours résoudre.

(1) "l'artiste a interprété une fugue de Bach"
(2) "l'artiste peintre a décoré cette salle"

Le mot "artiste" dans l'expression (1) se comporte syntaxiquement comme un nom, alors que dans l'expression (2), la syntaxe seule ne nous permet pas de déterminer quel est d'"artiste" ou de "peintre" le mot qui joue le rôle de nom. Il est donc inutile d'alourdir la syntaxe en

cherchant à résoudre une indétermination qui ne peut être levée qu'au niveau discursif.

En accord avec le principe de régularisation, la catégorie F regroupe noms et adjectifs, et une variable NA de sous-catégorisation de F se verra attribuer suivant les cas les valeurs, NOM, ADJ ou NAN (non déterminé).

3.1.1.2. Les pronoms personnels

A. BERRENDONNER distingue parmi les pronoms personnels, les pronoms toniques qui constituent à eux-seuls un syntagme nominal, des pronoms clitiques. Les premiers, comme "eux", "lui", "elle", seront catégorisés F. Les seconds, qui entrent dans l'une des constructions suivantes :

```
"Ne V __ pas ? "  comme "je", "tu", "il" , "on"...
"Ne __ V pas "    comme "le", "lui", "leur", "y"...
```

où V représente un verbe, seront catégorisés Y, c'est-à-dire pronoms préverbaux.

Il est vrai que nombreux sont les pronoms personnels qui entrent, a priori, dans les deux catégories F et Y, donc nombreuses sont les ambiguïtés issues de cette démarche. Cependant, ces deux catégories ont des comportements syntaxiques très différents. De ce fait, cette ambiguïté est aisée à lever puisque les contextes d'occurrence de F et Y sont distincts. De plus, la non-distinction de ces comportements entraînerait une perte d'informations linguistiques.

3.1.2. Les variables

3.1.2.1. Leur rôle

Lors de l'analyse morphologique, chaque forme se voit attribuer au moins une catégorie morphologique qui lui confère un rôle syntaxique. Ces catégories autorisent une analyse syntaxique grossière, mais ne reflètent qu'une faible part de l'information portée :
- par les formes : genre, nombre...
- par le lexique : animation, propriétés rectionnelles.

Cette information complémentaire sera portée par des variables dont on trouvera ci-dessous un sous-ensemble. La liste exhaustive est dans [BERRENDONNER, 1983-a]. Ces variables permettent de faire une analyse morphologique plus fine que les seules catégories et donc une analyse syntaxique plus pertinente. Ainsi, suivant le degré de finesse souhaité, on pourra soit négliger les variables, soit les prendre en compte dans leur totalité, soit encore adopter toute solution intermédiaire.

3.1.2.2. Définition de quelques variables

La variable NA, type nominal, affecte la catégorie F et prend ses valeurs dans {NOM, NAN, ADJ} en fonction des critères définis au paragraphe 3.1.1.1.

La variable NN, sous-type nominal, affecte les mots marqués F(NOM). Elle prend ses valeurs dans {COM, PRP, PRO} pour indiquer respectivement qu'un nom est commun, propre ou pronominal.

La variable PA, participe, affecte les mots de la catégorie F et prend pour valeur PPA pour un participe passé, PPR pour un participe présent.

La variable GR, genre, affecte les catégories F, D, et Y. Elle prend ses valeurs dans {MAS, FEM, GRN} suivant que le genre est masculin, féminin ou non marqué.

La variable NB, nombre, affecte les catégories F, D, Y et V. Elle prend ses valeurs dans {SNG, PLU, NBN} suivant que le nombre est singulier, pluriel ou non marqué.

La variable DQ marque certains mots de la catégorie F. Elle prend pour valeur :

- DVB si ce mot est un nom d'action déverbal
- DAJ si ce mot est un nom déadjectival
- AGE si ce mot est un nom dérivé d'un verbe à valeur d'agent.

La variable VW marque les adverbes (W). Elle a pour valeur :

- AVJ pour les adverbes modifieurs d'adjectifs
- PRO pour les adverbes anaphoriques
- TAM pour les adverbes de temps, aspect ou mode
- QUA pour les adverbes de quantité qui syntaxiquement peuvent se comporter, peuvent introduire un syntagme nominal, ou encore, s'y substituer.

"beaucoup l'ont vu"
"il mange peu de beurre"

La variable NU, quantification, affecte les déterminants (D), et prend ses valeurs dans {NUM, DEF, NNU}.

Un déterminant est marqué DEF s'il ne peut jouer seul le rôle d'un syntagme nominal.

* "mon a vu le chat" (*)

Il en est ainsi de : "le", "mon", "cet"...

Un déterminant est marqué NUM s'il peut jouer à lui seul le rôle d'un syntagme nominal et si, précédé d'un déterminant marqué DEF, il est constituant d'un déterminant.

"j'en ai vu deux"
"les deux chats"

Il en est ainsi de : "deux", "trois"...

Un déterminant est marqué NNU, s'il n'est ni DEF ni NUM, et donc s'il peut jouer à lui seul le rôle d'un syntagme nominal et s'il ne peut suivre un déterminant marqué DEF pour constituer un déterminant plus complet.

(*) L'astérisque qui précède une expression, indique que cette expression n'est pas attestée par la langue.

```
"j'en ai vu plusieurs"
* "les plusieurs ont vu le chat"
```

Il en est ainsi de : "plusieurs", "un", "certains"...

Les variables précédentes jouent un rôle primordial dans l'analyse syntaxique du syntagme nominal. Par la suite, nous aurons l'occasion d'en rencontrer d'autres que nous définirons au moment opportun.

3.2. LES ASPECTS INFORMATIQUES

A une forme du texte prétraité, l'analyseur morphologique associe une catégorie et des variables. Les informations dont il dispose, sont :

un dictionnaire : il contient les bases, les flexions (base + flexions = forme), et pour chaque base, son modèle de comportement morphologique, ainsi que des valeurs de variables lexicales comme DQ, NA...

une grammaire régulière dont les règles régissent l'association des bases et des flexions.

une liste de modèles qui portent les informations morphologiques de toutes les bases qu'ils représentent.

L'analyseur morphologique est un automate d'états finis. En effet, les régularisations effectuées sur les formes du texte par le prétraitement morphographique permettent d'établir un modèle d'analyse morphologique régulier.

4. LEVEES D'AMBIGUITES MORPHOLOGIQUES

A chaque forme, considérée isolément, l'analyseur morphologique attribue une ou plusieurs catégories. Il s'agit maintenant à l'aide du contexte, de choisir parmi les catégories, les seules possibles. Deux méthodes de levée d'ambiguïtés sont envisagées : une méthode statistique et une méthode linguistique. A l'heure actuelle, seule la première a fait l'objet d'une étude approfondie et d'une mise en oeuvre informatique [KALLAS, 1986]. La seconde est encore à l'état de projet. Elle serait plus cohérente, puisque fondée sur un modèle linguistique, avec l'ensemble du projet. Lorsqu'elle verra le jour, elle pourra utiliser des résultats acquis par la méthode statistique.

4.1. LA METHODE STATISTIQUE

Partant de l'hypothèse que la catégorie d'une forme dépend du contexte dans lequel elle apparaît, un modèle statistique adapté est le modèle markovien. La matrice de passage est construite à partir d'un échantillon de phrases du corpus à analyser, les ambiguïtés de cet échantillon étant levées à la main. Puis cette matrice est utilisée pour l'ensemble du corpus. Elle attribue à chacune des solutions proposées pour une forme, une probabilité. Le choix de la catégorie se portera sur la ou les solutions les plus probables. Une telle méthode est

efficace, son inconvénient réside dans le fait qu'elle peut imposer des solutions erronées que l'on ne peut rectifier par la suite.

4.2. LA METHODE LINGUISTIQUE

Il s'agit ici de lever les ambiguïtés sur des critères linguistiques, si possible indépendants du corpus traité, à l'aide de règles contextuelles. Lorsqu'une forme ambiguë est isolée au milieu de formes non ambiguës, il est facile de trouver la règle de réduction :

```
(D ou Y) F --> D F
```

Parfois cependant, on doit faire face à des zones d'ambiguïtés.

```
Y (Y ou D)  (F ou Y)  (V ou F)
je  la          lui      livre
```

Dans ce cas particulier, on constate l'attribution judicieuse de la catégorie Y à "je", pilier qui à lui seul permet de lire la séquence comme : Y Y Y V. On peut donc lever les ambiguïtés, mais quelles sont les règles adaptées ? Soit, on se fixe la règle globale

```
Y (Y ou D)  (F ou Y)  (F ou V) --> Y Y Y V
```

mais alors à chaque combinaison possible et peu probable, correspond une règle différente. D'où un grand nombre de règles. Soit, on se donne des règles à plus haut rendement comme :

```
Y (Y ou D ou F) --> Y Y
Y (F ou V) --> Y V
```

Dans ce cas, les règles seront appliquées au cours de la lecture du texte, celle-ci s'effectuant de gauche à droite, et de droite à gauche. jusqu'à ce qu'un passage ne lève aucune nouvelle ambiguïté. La deuxième solution offre plus de cohérence avec le modèle linguistique. Mais dans ce cas, les règles de levées d'ambiguïtés doivent être très bien choisies, afin de ne pas engendrer de solutions fausses. C'est pourquoi, la méthode linguistique n'est pas encore effective. Le choix des règles sera fait à partir d'un comptage des ambiguïtés et de leur contexte d'occurrence. L'objectif visé ici, n'est pas de résoudre toutes les ambiguïtés, mais seulement celles auxquelles on peut apporter une solution sûre.

Quel que soit le procédé de levée d'ambiguïtés adopté, linguistique ou statistique, les analyses morphologiques parasites ne seront jamais totalement éliminées à ce stade. Cela est inhérent au caractère de la langue. Nous espérons cependant que les ambiguïtés qui persisteront, seront peu nombreuses, et que, de plus, toutes les solutions proposées seront compatibles, pour une syntaxe grossière, avec leur contexte. En effet, la levée des ambiguïtés s'effectue d'après leur contexte d'apparition. Or les successions de formes dans un texte sont régies par la syntaxe. Si donc, le module de levée d'ambiguïtés ne peut choisir entre plusieurs catégories, c'est que chacune d'elles a sa place dans la séquence, et donc que chacune des séquences composée à partir de l'une des catégories est syntaxiquement correcte, pour une analyse syntaxique grossière ne prenant en compte que les catégories et

non les variables. Ce n'est donc qu'une analyse syntaxique plus fine, voire une analyse sémantique qui pourra dans le meilleur des cas résoudre les ambiguïtés persistantes.

En résumé, la levée des ambiguïtés met en jeu leur contexte, et donc des informations syntaxiques de façon implicite. Elle prépare donc doublement le terrain en vue de l'analyse syntaxique proprement dite parce qu'elle réduit le nombre de séquences à analyser ultérieurement, et ce, sur des critères syntaxiques, donc de façon cohérente avec la suite.

5. DELIMITATION DES SYNTAGMES MINIMAUX

A l'issue de la levée d'ambiguïtés, nous considérons à nouveau la succession des catégories morphologiques, car elle apporte une information intéressante sur la structure syntaxique sous-jacente. Nous entrons ici dans le domaine de la présyntaxe, car sans procéder à l'analyse syntaxique, il s'agit cependant de délimiter des syntagmes.

Un syntagme X considéré comme une suite de mots peut être structuré ainsi $[AYB]_X$, où Y est le centre du syntagme, A représente les formes pré-centrales et B les formes post-centrales. La position pré-centrale ne peut être occupée que par des formes dites faibles : prépositions, déterminants, adjectifs antéposables. La position post-centrale est occupée par des formes fortes : verbes, noms, autres adjectifs. Le centre du syntagme est occupé par une forme forte, la première lorsqu'on le parcourt de gauche à droite.

Chacune des formes, exceptées celles des catégories Q, T et C déjà porteuses d'informations sur les frontières de syntagme, seront étiquetées par l'intermédiaire d'une variable FF prenant les valeurs, FOR pour les formes fortes, FAI pour les faibles et FFN pour les autres. La valeur pour une forme de la variable FF est très liée à la catégorie morphologique à laquelle cette forme appartient. Ainsi :

FF = FOR	pour les verbes pour les noms et les adjectifs sauf les adjectifs antéposables pour les adverbes sauf les modifieurs d'adjectifs (très, si) pour les corrélats de négation (pas, jamais, point)
FF = FAI	pour les déterminants pour les prépositions sauf dessus, dedans... pour les préverbaux pour les particules "ne" pour les adjectifs antéposés (bel, beau)
FF = FFN	pour les adjectifs antéposables (petit)

A l'issue de ce marquage, il est aisé de délimiter les syntagmes minimaux, c'est-à-dire ceux qui n'en contiennent pas d'autres : si une forme forte est suivie d'une forme faible (la lecture se faisant toujours de gauche à droite) alors une frontière de syntagme passe entre ces deux formes. L'efficacité de cet algorithme dépend du nombre de formes marquées FFN.

Exemple :

"La méthode qui consiste à identifier une frontière de syntagme entre deux mots x et y chaque fois que l'on aura une suite (x,FOR)(y,FAI) donne de bons résultats, mais n'est pas d'une efficacité absolue. Il peut en effet se trouver que les positions précentrales d'un syntagme soient inoccupées : le syntagme, qui ne contient alors aucune forme faible initiale se trouvera intégré par l'analyse à celui qui précède . Le plus irritant est que cette situation se produit souvent à l'initiale des syntagmes verbaux , qui constitue la frontière de syntagme majeure de bien des phrases." [in Berrendonner p. 49]

A l'issue de l'analyse morphologique, on obtient le nouveau texte suivant :

La[D, FAI] méthode[F, FOR] que[Q] %[F, FOR] consiste[V, FOR] à[P, FAI] identifier[V, FOR] une[D, FAI] frontière[F, FOR] de[P, FAI] syntagme[F, FOR] entre[P, FAI] deux[D, FAI] mots[F, FOR] x[F, FOR] et[C] y[F, FOR] chaque[D, FAI] fois[F, FOR] que[Q] on[Y, FAI] aura[V, FOR] une[D, FAI] suite[F, FOR] (x,FOR)(y,FAI)[F, FOR] donne[V, FOR] de[P, FAI] bons[F, FAI] résultats[F, FOR] ,[T] mais[C] ne[G, FAI] est[V, FOR] pas[G,FOR] de[P, FAI] une[D, FAI] efficacité[F, FOR] absolue[F, FOR]
.[T] Il[Y, FAI] peut[V, FOR] (en+effet)[W] se[Y, FAI] trouver[V, FOR] que[Q] les[D, FAI] positions[F, FOR] précentrales[F, FOR] de[P, FAI] un[D, FAI] syntagme[F, FOR] soient[V, FOR] inoccupées[F, FOR] : le[D, FAI] syntagme[F, FOR] ,[T] que[Q] %[F, FOR] contient[V, FOR] alors[W] pas[G, FOR] une[D, FAI] forme[F, FOR] faible[F, FOR] initiale[F, FOR] se[Y, FAI] trouvera[V, FOR] intégré[F,FOR] par[P, FAI] l'[D, FAI] analyse[F, FOR] à[P, FAI] cet[D, FAI] lui[F, FOR] que[Q] %[F,FOR] précède[V, FOR] .[T] Le[D, FAI] plus[W, FAI] irritant[F, FOR] est[V, FOR] que[Q] cette[D, FAI] situation[F, FOR] se[Y, FAI] produit[V, FOR] souvent[W,FOR] à[P, FAI] l'[D, FAI] initiale[F, FOR] de les[D, FAI] syntagmes[F, FOR] verbaux[F, FOR] ,[T] que[Q] %[F, FOR] constitue[V, FOR] la[D, FAI] frontière[F, FOR] de[P, FAI] syntagme[F, FOR] majeure[F, FOR] de[P, FAI] bien[F, FOR] de[P,FAI] les[D, FAI] phrases[F, FOR] .[T]

L'algorithme de découpage proposé donne le résultat suivant où chaque passage à la ligne correspond à une frontière de syntagme :

La[D, FAI] méthode[F, FOR]
que[Q]
%[F, FOR] consiste[V, FOR]
à[P, FAI] identifier[V, FOR]

une[D, FAI] frontière[F, FOR]
de[P, FAI] syntagme[F, FOR]
entre[P, FAI] deux[D, FAI] mots[F, FOR] x[F, FOR]
et[C]
y[F, FOR]
chaque[D, FAI] fois[F, FOR]
que[Q]
on[Y, FAI] aura[V, FOR]
une[D, FAI] suite[F, FOR] (x,FOR)(y,FAI)[F, FOR] donne[V, FOR]
de[P, FAI] bons[F, FAI] résultats[F, FOR]
,[T] mais[C]
ne[G, FAI] est[V, FOR] pas[G,FOR]
de[P, FAI] une[D, FAI] efficacité[F, FOR] absolue[F, FOR]
.[T]
Il[Y, FAI] peut[V, FOR] (en+effet)[W, FOR]
se[Y, FAI] trouver[V, FOR]
que[Q]
les[D, FAI] positions[F, FOR] précentrales[F, FOR]
de[P, FAI] un[D, FAI] syntagme[F, FOR] soient[V, FOR] inoccupées[F, FOR]
:[T]
le[D, FAI] syntagme[F, FOR]
,[T] que[Q]
%[F, FOR] contient[V, FOR] alors[W, FOR] pas[G, FOR]
une[D, FAI] forme[F, FOR] faible[F, FOR] initiale[F, FOR]
se[Y, FAI] trouvera[V, FOR] intégré[F,FOR]
par[P, FAI] l'[D, FAI] analyse[F, FOR]
à[P, FAI] cet[D, FAI] lui[F, FOR]
que[Q]
%[F,FOR] précède[V, FOR]
.[T]
Le[D, FAI] plus[W, FAI] irritant[F, FOR] est[V, FOR]
que[Q]
cette[D, FAI] situation[F, FOR]
se[Y, FAI] produit[V, FOR] souvent[W,FOR]
à[P, FAI] l'[D, FAI] initiale[F, FOR]
de[P, FAI] les[D, FAI] syntagmes[F, FOR] verbaux[F, FOR]
,[T] que[Q]
%[F, FOR] constitue[V, FOR]
la[D, FAI] frontière[F, FOR]
de[P, FAI] syntagme[F, FOR] majeure[F, FOR]
de[P, FAI] bien[F, FOR]
de[P,FAI] les[D, FAI] phrases[F, FOR]
.[T]

Voici, sur un texte pris au hasard, le résultat de la délimitation des syntagmes. On constate que, dans l'ensemble, la méthode est efficace ; cependant, on relèvera quelques inexactitudes :

```
une[D, FAI] suite[F, FOR] (x,FOR)(y,FAI)[F, FOR] donne[V, FOR]
de[P, FAI] un[D, FAI] syntagme[F, FOR] soient[V, FOR] inoccupées[F, FOR]
Le[D, FAI] plus[W, FAI] irritant[F, FOR] est[V, FOR]
de[P, FAI] syntagme[F, FOR] majeure[F, FOR]
```

Les trois premières sont une illustration du texte qui se poursuit ainsi :

> " Il est en effet fréquent qu'un syntagme verbal commence directement par le verbe, c'est-à-dire par son centre. On pourrait traiter ce cas particulier fréquent, en posant une règle spécifique, qui placerait une frontière devant tout verbe précédé d'une forme forte."

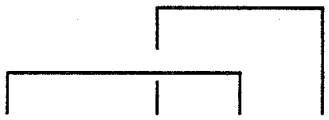
Le découpage alors obtenu, n'est pas encore parfait parce que certains syntagmes sont imbriqués comme "la frontière de syntagme majeure". Mais, l'objectif visé par la délimitation des syntagmes minimaux n'est pas d'obtenir un résultat très fin. Ce procédé doit plutôt être perçu comme un moyen très simple de progresser rapidement dans le processus d'analyse. En effet, les données (la valeur de la variable FF, pour chacune des formes) sont issues de l'analyse morphologique ; de plus, l'algorithme est trivial. Les résultats sont intéressants puisque connaissant les frontières d'un syntagme, la catégorie morphologique du centre de syntagme, nous en précise la nature : syntagme verbal, nominal, adjectival, adverbial... Ces résultats sont précieux pour lever quelques ambiguïtés de nature morphologique.

Exemple : "le " toujours marqué faible (sauf au sein de tournures impératives, peu probables dans notre cas) sera analysé Y dans un syntagme verbal ou D dans un syntagme nominal.

Ces résultats sont de plus indispensables à la phase de traitement suivante : le regroupement des morphèmes discontinus.

6. LE TRAITEMENT DES MORPHEMES DISCONTINUS.

La présence dans la langue française de morphèmes discontinus - les négations ne...pas, ne...jamais, les temps composés des verbes - sont un handicap majeur pour un traitement automatique. Leur existence rompt en effet la régularité du texte à analyser, et de ce fait nous contraindrait à opter pour un modèle linguistique plus complexe où les structures syntaxiques ne seraient plus des arborescences.



je ne lui avais pas donne l'adresse

Pour éviter cet écueil, A. BERRENDONNER [1983-b] propose les régularisations suivantes :

6.1. CAS DE LA NEGATION

La négation est marquée dans une phrase par la cooccurrence de constituants disjoints : la particule "ne", et un ou plusieurs corrélats (guère, pas, point, nul, jamais...). Le plus souvent, la particule "ne" ne porte que sur un corrélat.

"je ne l'ai pas vu"
"je ne l'ai vu nulle part"
"personne ne l'a vu"

Alors, la présence simultanée dans la phrase de la particule "ne" et d'un corrélat indique qu'il y a négation, la négation portant suivant le cas sur le verbe ou sur un syntagme nominal. Une phrase peut aussi contenir plusieurs corrélats négatifs associés à une unique particule "ne". Cette particule porte alors sur chacun des corrélats.

"je ne l'ai jamais vu à aucun endroit"

A partir de ces remarques, on peut proposer pour résoudre le cas de la négation, la démarche suivante :

- la rencontre d'une particule "ne" dans une phrase, active une procédure
- cette procédure distribue la particule "ne" sur chacun des corrélats négatifs de la phrase
- les corrélats négatifs sont régularisés : "aucun" est remplacé par "pas un", "jamais" par "pas une fois" et l'on obtient pour l'exemple précédent :

" je l'ai vu (ne+pas) une fois (ne+pas) un endroit"

- lors de l'analyse syntaxique l'amalgame "ne+pas" ne sera pas considéré comme un constituant mais l'information sera transmise sur le syntagme qui le contient au moyen d'un trait NEG.

6.2. CAS DES VERBES CONJUGUES AUX TEMPS COMPOSES

A. BERRENDONNER propose dans ce cas, une idée séduisante qui consiste à ramener le participe passé devant l'auxiliaire. Ces deux formes accolées peuvent alors être soumises à la même analyse morphologique que les formes simples des verbes.

"je ne lui avais pas donné l'adresse"
devient
"je ne lui donné-avais pas l'adresse"

Cette forme s'analyse alors :

```
donn     /     é-av     /     ai   /   s
 |             |             |       |
 |             |             |       |
radical      accompli       passé  personne
```

Ce procédé de régularisation ne peut s'appliquer sans précaution. En effet, toutes les cooccurrences d'un auxiliaire et d'un participe passé (analysé ici F[adj, ppa]) ne sont pas des temps composés de verbes. Avec l'auxiliaire être on trouve des participes qui fonctionnent comme des adjectifs :

"il est cuit"

ou encore des formes passives :

"il est surpris par sa maigreur".

Ainsi, si le verbe auxiliaire est une forme du verbe "être", il est nécessaire de s'assurer avant tout regroupement que le participe passé dérive d'un verbe qui se compose avec le verbe "être" :

"il est venu".

Le cas de l'auxiliaire avoir est plus simple puisqu'il n'entre pas dans la composition de formes passives simples, et ne sert non plus de support aux adjectifs attributs. Cependant, beaucoup de participes passés jouent le rôle d'adjectifs dans un syntagme nominal. Ainsi lorsque dans la même phrase, l'on a un tel adjectif et un auxiliaire, cette régularisation peut avoir des effets fâcheux : " il a un langage fleuri ".

On se trouve une nouvelle fois devant la nécessité de repérer les frontières de syntagme avant de procéder à une régularisation, et donc d'effectuer préalablement une analyse morphologique qui attribuera dans un premier temps à l'auxiliaire, la catégorie V[AUX] et au participe passé F[PPA]. Après regroupement de ces deux formes, un complément d'analyse morphologique portant sur le verbe sera nécessaire. Ainsi :

"avais donné"

sera d'abord analysé

"avais, avoir V[AUX]" et "donné, donné F[PPA]"

puis, après regroupement, l'analyse définitive sera

"donné-avais, donner V"

Le regroupement des morphèmes discontinus s'opère sur un texte analysé morphologiquement sans ambiguïtés. Le nouveau texte obtenu présente un double intérêt :

1 il prépare l'analyse morphologique des verbes aux temps composés ; donc ce texte devra à nouveau être soumis à l'analyseur morphologique.

2 il régularise le texte pour préparer l'étape ultérieure : l'analyse syntaxique.

Ainsi, à l'issue du regroupement deux voies s'ouvrent : soit un retour à l'analyse morphologique, soit une poursuite de l'analyse par passage à l'analyse syntaxique.

7. L'EXTRACTION DES GENERATEURS

Considérons maintenant le texte issu des traitements précédents : à chaque mot du texte a été assignée une catégorie unique. L'objectif visé par notre application est l'analyse syntaxique des syntagmes nominaux. Or avant de procéder à leur analyse, il est nécessaire de les extraire. Comme un syntagme est défini par la syntaxe, il faudrait en toute logique procéder à l'analyse syntaxique du texte pour en extraire les syntagmes nominaux. L'analyse du texte intégral étant beaucoup plus complexe et donc coûteuse que celle des seuls syntagmes nominaux, il nous faut opérer autrement, et trouver un moyen de repérer les séquences de texte contenant les syntagmes nominaux. Plus précisément, ce ne sont pas toutes les séquences de texte qui contiennent un syntagme nominal que nous retiendrons (car en cas d'inclusion d'un syntagme nominal dans un autre, il y aurait redondance) ; mais ce sont les séquences qui contiennent les "syntagmes nominaux maximaux". Nous définirons un "syntagme nominal maximal" comme étant un syntagme nominal non inclus dans un autre.

Sans le recours à l'analyse syntaxique du texte plein, il est utopique de penser repérer exactement les syntagmes nominaux maximaux. Aussi, le groupe SYDO a-t-il défini la notion de générateurs [G. ANTONIADIS, 1983].

7.1. LES GENERATEURS : DEFINITIONS

Un **ensemble de catégories morphologiques pertinentes** est un ensemble dont les éléments sont les catégories morphologiques pouvant entrer dans la composition d'un syntagme nominal.

Une **expression régulière pertinente à structure fixe** est une expression régulière formée de catégories morphologiques pertinentes et dont la structure a été prédéfinie.

Une **expression régulière pertinente à structure libre** est une expression régulière formée de catégories morphologiques pertinentes de structure quelconque.

Une **expression régulière pertinente** est une suite d'expressions régulières pertinentes à structure fixe et/ou d'expressions régulières à structure libre.

Un **générateur** est une suite maximale d'unités lexicales qui a une structure d'expression régulière pertinente.

7.1.1. Extraction de générateurs

Le texte apparaît comme une séquence d'unités lexicales, i.e. de triplets : forme, catégorie morphologique, liste de variables. Etant donnés un ensemble de catégories pertinentes et un ensemble d'expressions régulières pertinentes, l'algorithme d'extraction de générateurs fournit :

(1) les occurrences dans le texte des générateurs maximaux, c'est-à-dire, des générateurs tels qu'il n'existe pas d'autres générateurs les englobant.

(2) l'ensemble des générateurs contenus dans chacun des générateurs maximaux.

Remarque

Si l'ensemble d'expressions régulières, donnée de l'algorithme, est réduit à des expressions régulières fixes ou cadres, alors le résultat de l'algorithme est l'ensemble des occurrences de ces cadres dans le texte.

7.1.2. Pertinence des résultats

Cet algorithme réalisé par G. Antoniadis sur la base de l'algorithme de AHO et CORASICK a des avantages indéniables. Tout d'abord, il s'exécute sur un texte analysé morphologiquement. Ainsi, il permet d'extraire les segments de texte contenant les syntagmes nominaux. Ces segments seuls seront soumis à l'analyse syntaxique. De ce fait, c'est un outil rapide et efficace. De plus, il est cohérent avec l'hypothèse faite pour l'indexation automatique : l'information est contenue dans les syntagmes nominaux.

Cependant, il présente quelques inconvénients inhérents au fait qu'il n'exploite que les résultats de l'analyse morphologique. Le premier est issu du choix des catégories morphologiques pertinentes. En effet, une catégorie est soit interdite, soit admise. Considérons, par exemple, la catégorie "ponctuation", notée T : le point ne sera jamais élément de syntagme nominal alors que la virgule peut l'être, en cas de coordination de substantifs. Que l'on décide d'exclure ou d'admettre cette catégorie, le résultat ne sera jamais parfait. On pourrait remédier à cet ennui en créant des catégories *ad hoc*, mais alors on renierait un principe de base. Cette faiblesse de l'algorithme provient du fait que le modèle linguistique a été défini sur des objectifs plus larges que la seule extraction des générateurs qui, elle ne représente qu'un outil ponctuel vers l'indexation automatique. Un deuxième inconvénient se révélera dans l'étape ultérieure, celle de l'analyse syntaxique. En effet, une fois un générateur extrait, on ne dispose plus du contexte dans lequel il était situé. Or, un générateur, n'étant qu'une séquence de catégories morphologiques, peut regrouper plusieurs syntagmes nominaux régis par une forme de catégorie exclue (verbe, par exemple). Dans ce cas, une analyse syntaxique de ces générateurs engendrera des solutions parasites que l'on ne pourra éliminer, l'information nécessaire n'étant plus disponible.

" elle ne conduit pas dans [toutes les circonstances à une protection efficace de l'hôte]".

Le générateur est ici délimité par les crochets. Il contient deux syntagmes nominaux indépendants, puisque le verbe "conduit" régit un syntagme prépositionnel en "à". Mais après extraction du générateur, cette information est perdue et de ce fait, des solutions parasites sont engendrées par les étapes ultérieures.

La méthode d'extraction des générateurs n'est pas pour autant condamnée. On ne peut cependant l'utiliser sans être conscient de ses faiblesses. Aussi esquisserons-nous une autre solution, fondée sur la segmentation d'un texte analysé morphologiquement.

7.2. EXTRACTION DE SYNTAGMES NOMINAUX PAR SEGMENTATIONS SUCCESSIVES DU TEXTE

La segmentation du texte analysé morphologiquement s'opère en deux temps. La première étape consiste à délimiter les propositions, la seconde les syntagmes à l'intérieur des propositions.

La segmentation du texte en propositions repose sur l'algorithme MAEGAARD et SPANG HANSSEN [1973]. Le fonctionnement de cet algorithme se résume ainsi. A chaque forme analysée est associé un numéro de classe. Les classes sont constituées de la façon suivante :

1. les subordonnants
2. les conjonctions "et", "mais, "ou"
3. la ponctuation ","
4. les formes n'appartenant à aucune autre classe
5. les verbes finis dont le sujet n'est pas un pronom personnel, et les particules préverbales
6. les ponctuations fortes : " . ; ! ? "
7. la conjonction "car"
8. les verbes finis dont le sujet est un pronom personnel, et les particules préverbales

On obtient donc à partir du texte, une séquence de numéros de classe. Cette séquence est soumise à un automate qui détecte le début et la fin des propositions, celles-ci pouvant être éventuellement imbriquées. Cet automate exploite des informations linguistiques comme :

une proposition ne peut contenir qu'un verbe fini,
un subordonnant indique le début d'une proposition subordonnée,
la fin d'une proposition subordonnée se situe immédiatement à gauche du verbe de la proposition dominante...

Cet algorithme résout, dans la plupart des cas, la segmentation d'un texte en proposition. A l'issue de ce traitement, la deuxième étape peut débuter. Au sein de chaque proposition, on repère le verbe, première forme forte du syntagme verbal. On a alors accès à son comportement syntaxique, ce qui permet d'isoler les syntagmes régis. Ce procédé d'extraction des syntagmes nominaux est plus délicat à mettre en oeuvre. Il est beaucoup plus coûteux que le précédent, mais il devrait permettre d'éliminer des solutions parasites.

Conclusion

Avant que l'analyse syntaxique ne rentre en jeu, le texte initial est soumis aux traitements étudiés dans ce chapitre, traitements dont l'enchaînement est indiqué par la figure 1.1. Par la suite, nous supposerons que ces traitements conduisent à un texte analysé morphologiquement sans ambiguïtés ni erreurs afin de ne pas oblitérer les résultats de l'analyse syntaxique. Cette position est quelque peu idéaliste, car, dans la réalité, certaines ambiguïtés morphologiques ne pourront être résolues que par l'analyse syntaxique. Mais elle nous autorise à concevoir, dans une première phase, l'analyse syntaxique comme une étape consécutive à l'analyse morphologique.

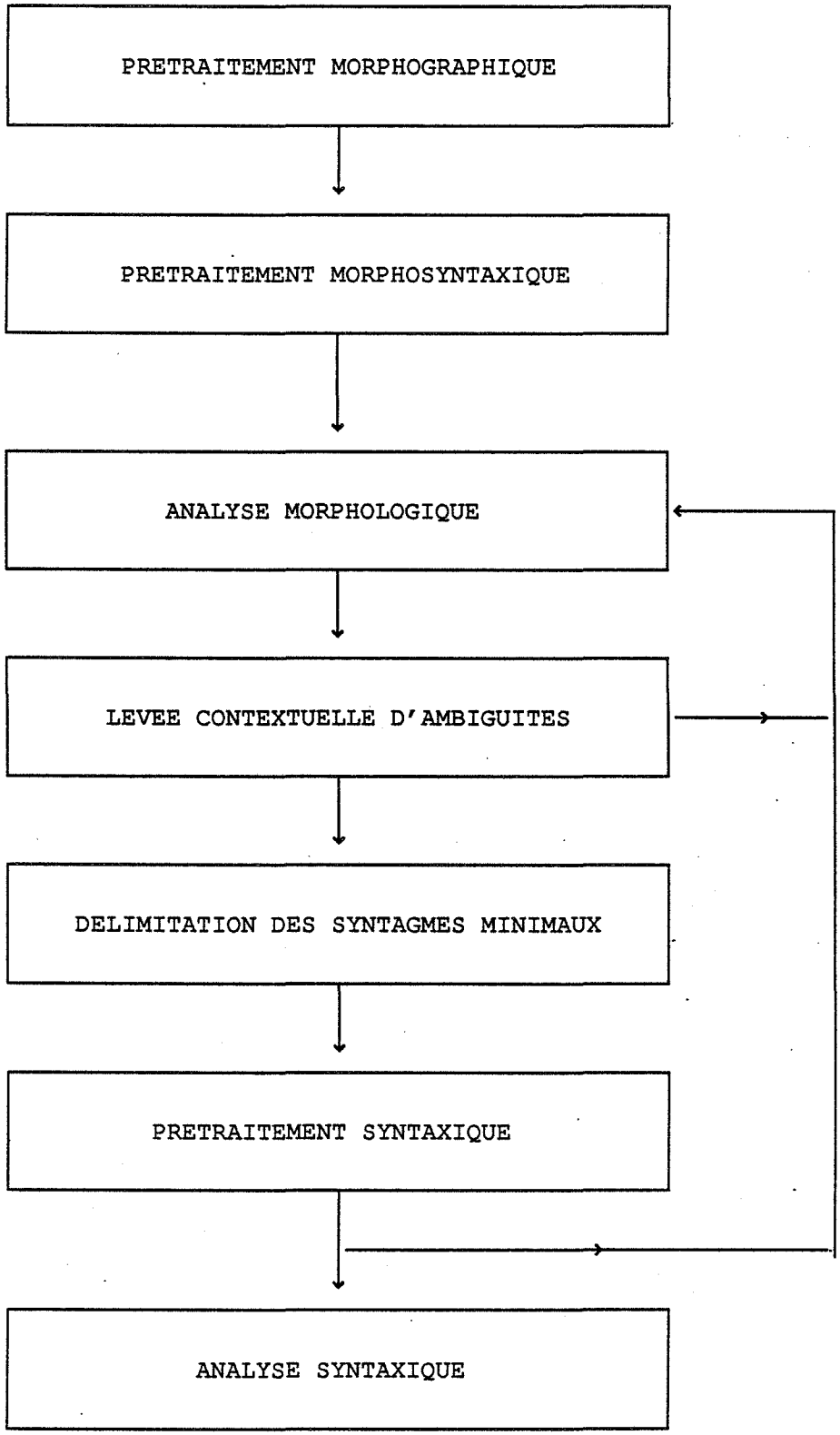

PRETRAITEMENT MORPHOGRAPHIQUE
PRETRAITEMENT MORPHOSYNTAXIQUE
ANALYSE MORPHOLOGIQUE
LEVEE CONTEXTUELLE D'AMBIGUITES
DELIMITATION DES SYNTAGMES MINIMAUX
PRETRAITEMENT SYNTAXIQUE
ANALYSE SYNTAXIQUE

FIGURE 1.1.

Chapitre 2

ANALYSEURS SYNTAXIQUES POUR LES LANGAGES HORS CONTEXTE AMBIGUS *

Ce chapitre fait le point sur les algorithmes d'analyse syntaxique, pour langages hors contexte. Après quelques rappels sur la théorie des langages, les algorithmes de Cocke et d'Earley sont présentés de façon détaillée.

1. RAPPELS ET DEFINITIONS

1.1. GRAMMAIRES ET LANGAGES HORS CONTEXTE

Nous nous contentons de rappeler ici les définitions essentielles à la compréhension de ce qui suit. Les bases de la théorie des langages sont exposées dans [AHO & ULLMAN, 1972].

1 Soit une grammaire formelle G = (T, N, S, R) où T est le vocabulaire terminal, N le vocabulaire non terminal, ou auxiliaire, S l'axiome et R l'ensemble fini des règles.
On notera :

T union N = V (vocabulaire total)
T inter N = $\emptyset$
T^* est l'ensemble des mots sur T
V^* l'ensemble des mots sur V

Elle est réputée hors contexte si toutes les règles de R sont de la forme :

X ---> v où X est dans N et v dans V^*.

---

* Ce chapitre a été rédigé en collaboration avec J. ROUAULT [1986].

2 Un langage L est hors contexte s'il existe une grammaire hors contexte G qui l'engendre : L = L(G).

1.2. RECONNAISSANCE ET ANALYSE SYNTAXIQUES

Etant données une grammaire G et une chaîne x de T^*, la **reconnaissance** de x consiste à répondre par "oui" ou par "non" à la question de l'appartenance de x à L(G).

En outre, dans le cas où x appartient à L(G), **l'analyse syntaxique** produit une structure syntaxique que la grammaire G associe à x. Remarquons que, dans beaucoup de cas, la reconnaissance et la construction de la structure sont menées simultanément.

Les problèmes de la reconnaissance et de l'analyse syntaxique se posent évidemment pour tous les types de langages. Cependant, les analyseurs les plus courants concernent les langages hors contexte. Dans ce cas, en effet, la grammaire associe à chaque chaîne du langage une ou plusieurs structures syntaxiques qui sont des arborescences. Nous nous plaçons dans ce cas.

1.3. CARACTERISTIQUES DES ANALYSEURS

Etant donnée une grammaire G, un **accepteur** associé à G est un algorithme qui résout (en un nombre fini de pas) le problème de la reconnaissance d'une chaîne x de T*. Un **analyseur syntaxique,** lui, résout le problème de l'analyse syntaxique.

Les algorithmes peuvent être construits suivant des stratégies variées et, classiquement, on les opposera sur les critères suivants.

1.3.1. Premier critère : descendant / ascendant

Dans la démarche descendante, on cherche à reconstruire la chaîne x (à analyser) en partant de l'axiome ; on utilise donc les règles en remplaçant leur partie gauche par leur partie droite. Inversement, dans la démarche ascendante, on cherche à retrouver l'axiome à partir de la chaîne x ; les règles s'appliquent en remplaçant leur partie droite par leur partie gauche.

Remarque

Les deux stratégies ne sont pas duales l'une de l'autre. Elles ne requièrent pas les mêmes conditions d'application : pour que l'algorithme descendant se termine en un nombre fini de pas, pour toute chaîne x de T^*, il est nécessaire et suffisant que la grammaire G ne soit pas récursive à gauche. Par contre, l'algorithme ascendant se termine si, et seulement si, la grammaire ne contient ni cycles, ni règles dont la partie droite est la chaîne vide (règles de suppression de symboles de N).

1.3.2. Deuxième critère : général / particulier

Un algorithme général est capable de fonctionner sur toute grammaire hors contexte G, c'est-à-dire dont les parties droites des règles ne sont soumises à aucune restriction particulière. Par opposition aux algorithmes particuliers, qui supposent que les parties droites de

règles aient une forme particulière :
- dans la forme normale de GREIBACH les règles ont une partie droite de la forme aw où a appartient à T et w à N^* ;
- dans la forme normale de CHOMSKY les règles sont de l'un ou l'autre des deux types suivants

X ---> YZ (Y et Z appartenant à N)
X ---> a (a appartenant à T)

Remarque
Cette distinction paraît secondaire puisque toute grammaire hors contexte peut être transformée en une grammaire équivalente, soit sous forme normale de Greibach, soit sous forme normale de Chomsky. Cependant, il ne s'agit ici que d'équivalence faible : la grammaire d'origine et sa transformée engendrent le même langage mais associent des structures syntaxiques différentes à la même chaîne x. Comme une grammaire est liée à un modèle linguistique, la normaliser revient à transformer ce modèle, donc à produire des structures non conformes aux hypothèses de départ.

1.3.3. Troisième critère : simple / multiple

Ce critère est lié à la propriété de la grammaire G d'être ambiguë ou non. Nous disons qu'une grammaire G est ambiguë si elle associe à au moins une chaîne de L(G) plus d'une structure syntaxique.

Si la grammaire G n'est pas ambiguë, et si, pour une chaîne x de T^*, G produit une structure syntaxique, cette structure est unique. L'analyseur peut donc s'arrêter dès qu'il l'a trouvée (puisqu'il n'y en a pas d'autres) : nous qualifions de simple ce fonctionnement d'un analyseur.

Remarque
Une analyse simple n'est pas forcément déterministe puisqu'elle n'exclut pas les retours arrière pour arriver à la solution. Dans le cas d'une grammaire déterministe, on peut trouver une stratégie d'analyse simple et déterministe, c'est-à-dire sans retour arrière.

Si, par contre, la grammaire G est ambiguë, et si on choisit un algorithme simple, il faudra le modifier de façon à engendrer toutes les structures syntaxiques que la grammaire G associe à la chaîne x à analyser. Il ne peut donc s'arrêter que si on a exploré tous les chemins possibles. Il est, dans ce cas, plus opportun d'utiliser un algorithme multiple, c'est-à-dire qui construit simultanément toutes les structures possibles.

1.3.4. Quatrième critère : prédictif / combinatoire

Une langue naturelle étant intrinséquement ambiguë toute grammaire qui l'engendre est ambiguë : aucun analyseur déterministe ne peut être envisagé. Il est cependant possible d'échapper dans certains cas à l'exploration systématique de toutes les voies possibles (combinatoire totale) en mettant en oeuvre des processus prédictifs.

Cette prédiction se fait à partir des caractères à analyser de la chaîne d'entrée, dans le cas d'une analyse descendante, et à partir des symboles de N déjà obtenus, dans le cas d'une analyse ascendante.

1.3.5. Cinquième critère : mode déclaratif / mode procédural

La réalisation d'un analyseur passe par l'écriture d'un programme. Suivant les rapports qu'entretiennent le programme et la grammaire, on distinguera :

le mode déclaratif : lorsque la grammaire est une donnée externe au programme. Dans ce cas, la grammaire peut être modifiée (tout en restant dans le modèle hors contexte), sans remettre en cause l'analyseur. On obtient ainsi un bon outil pour tester des grammaires.

le mode procédural : l'analyseur est conçu pour une grammaire donnée, chaque règle étant l'objet d'une procédure. Alors toute modification de la grammaire entraîne une modification de l'analyseur. En contrepartie, l'exécution du programme est moins coûteuse. Le mode procédural sera donc préféré dès que la grammaire aura une forme définitive.

2. L'ALGORITHME DE COCKE

2.1. CARACTERISTIQUES

Relativement aux critères définis précédemment, cet algorithme se situe de la façon suivante :
- il est ascendant : partant de la chaîne, il cherche à remonter à l'axiome de la grammaire ;
- il est particulier : il suppose la grammaire mise sous forme normale de Chomsky ; il est important de rappeler que les modèles linguistiques utilisés habituellement produisent des grammaires qu'il est impossible de construire directement sous cette forme ; comme la transformation faisant passer à la forme normale de Chomsky dénature le modèle initial, nous avons ici une première limite à l'utilisation de cet algorithme ;
- il est multiple : il construit simultanément toutes les analyses que la grammaire G associe à la chaîne à analyser ; ces analyses sont représentées dans une table unique ;
- il est combinatoire : il ne met en oeuvre aucune méthode de prédiction destinée à éviter des solutions partielles qui sont vouées ultérieurement à l'échec.

2.2. NOTATIONS PARTICULIERES

La grammaire utilisée étant sous forme normale de Chomsky, elle est de la forme :

G = (T, N, S, R)

où l'ensemble R ne contient que des règles de l'un ou l'autre des deux types suivants

A ---> BC (A,B,C éléments de N)

A ---> a (a élément de T).

La chaîne à analyser est notée : x = x[1]x[2]...x[n].

On utilise une table triangulaire inférieure t dont les éléments sont notés t[i,j], i variant de 1 à n et j de 0 à n-i+1. Chaque élément t[i,j] est une liste de non terminaux. Cette table contiendra le résultat de chaque étape de la reconnaissance [Figure 2.1].

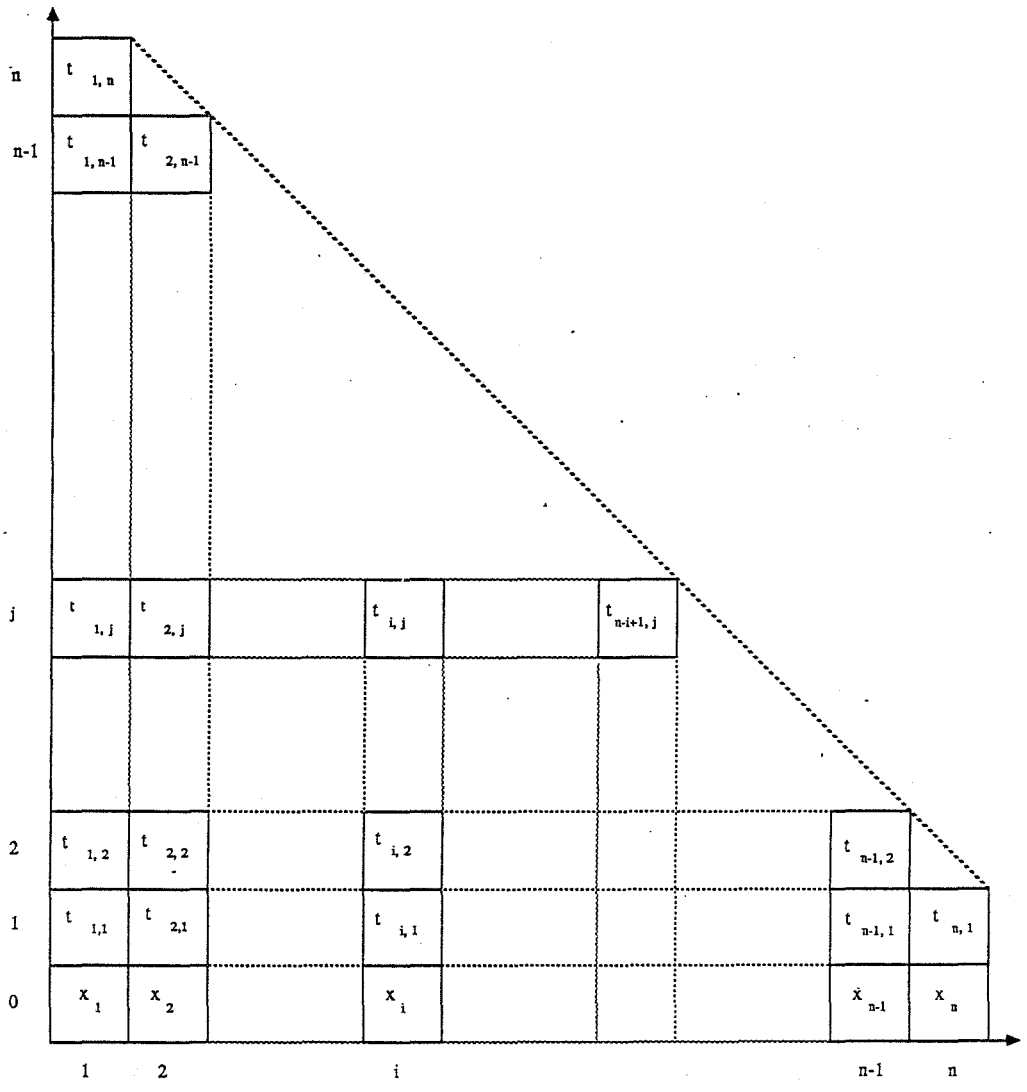


FIGURE 2.1.

2.3. L'ACCEPTEUR

2.3.1. Description du fonctionnement

calcul des t[i,1], pour i=1,2,...,n.

La ligne zéro (j=0) contient la chaîne à reconnaître ; l'élément x[i] occupe la case t[i,0]. Alors la case t[i,1] contiendra tous les non terminaux A tels que :

A ---> x[i] est une règle de R.

construction de la ligne j+1 à partir des lignes d'indices 1,2,...j :

L'indice i désigne le rang du caractère x[i] considéré, dans la chaîne d'entrée. La signification de l'indice j est la suivante : dans la chaîne d'entrée x, on considère la sous-chaîne de longueur j qui commence en x[i] :

u = x[i]x[i+1]...x[i+j-1].

La case t[i,j] du tableau contient les non terminaux qui, par application de règles de la grammaire, engendrent la chaîne terminale x ci-dessus. Comme un élément A de t[i,j] engendre u et que les règles de G sont binaires (forme normale de Chomsky), le processus qui fait passer de A à u débute forcément par application d'une règle A ---> BC [Figure 2.2]. Par ailleurs, on suppose la reconnaissance effectuée jusqu'à la ligne j-1; donc B et C appartiennent à des cases déjà construites de la table t. De plus, B engendre une partie :

u_1 = x[i]x[i+1]...x[i+k-1]

de u et C engendre la partie restante :

u_2 = x[i+k]...x[i+j-1]

(propriété des grammaires hors contexte).

La construction de la case t[i,j] s'en déduit : supposons construites les lignes 1,2,...,j-1 de la table t. Dans ces lignes on choisit deux cases s_1 et s_2 de la façon suivante : s_1 est le résultat de la reconnaissance d'un préfixe u_1 de u et s_2 celle du suffixe u_2 correspondant. La case s_1 est évidemment la case t[i,k]. La case s_2 est associée à la chaîne u_2, dont le premier caractère est le (i+k)ème dans x et le dernier, le (i+j-1)ème dans x. Donc s_2 est la case t[i+k,j-k].

Un élément de t[i,j] est donc la partie gauche A d'une règle :

A ---> BC

où B est à prendre dans t[i,k] et C dans t[i+k,j-k].

Comme on explore toutes les possibilités, il faut :

- pour deux cases ainsi choisies, prendre en considération toutes les règles possibles, B parcourant t[i,k] et, pour chaque B, C parcourant t[i+k,j-k] ;
- prendre en considération tous les couples de cases possibles, c'est-à-dire qu'il faut décomposer la chaîne u de toutes les manières possibles en un préfixe u_1 et le suffixe u_2 associé. Ceci revient évidemment à donner à k toutes les valeurs possibles de 1 à j-1.

Le processus se termine par la construction de la case t[1,n].

La chaîne x est reconnue (c'est-à-dire appartient à L(G)) si et seulement si S appartient à t[1,n].

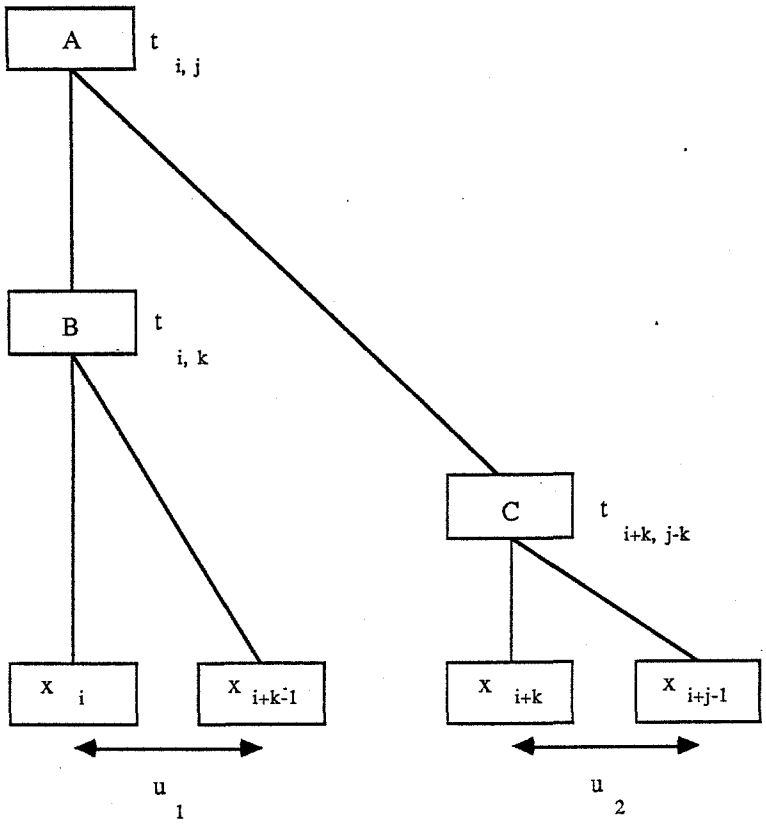


FIGURE 2.2

2.3.2. L'algorithme

L'ACCEPTEUR est donc défini par l'algorithme suivant : **programme ACCEPTEUR COCKE**

```
début

      initialiser les t[i,j] à vide

      pour i de 1 à n faire
            pour chaque règle A ---> x[i] faire
            t[i,1] := A concaténé à t[i,1].

      pour j de 2 à n faire
            pour i de 1 à n-j+1 faire
                  pour k de 1 à j-1 faire
                        pour chaque règle A ---> BC, où B
                        parcourt t[i,k] et, pour chaque B,
                        C parcourt t[i+k,j-k] faire
                        t[i,j] := t[i,j] union {A}

      si S appartient à t[1,n] imprimer "vrai"
      sinon imprimer "faux".
fin.
```

2.3.3. Exemple

Considérons la grammaire :

$$G_0 = (\{D, F, P\}, \{N, N', N'', A, SP\}, N'', R_0)$$

où R_0 est l'ensemble suivant de règles :

$$R_0 = \{$$
N'' ---> D N'
N' ---> N
N ---> A N
N ---> N SP
SP ---> P N''
A ---> F
N ---> F}

Par transformation, on obtient la grammaire suivante, équivalente à G et mise sous forme normale de Chomsky :

$$G = (\{D, F, P\}, \{N, N'', A, SP, D', P'\}, N'', R)$$

où R est le nouvel ensemble suivant de règles :

R = {
N'' ---> D' N [1]
N ---> A N [2]
N ---> N SP [3]
SP ---> P' N'' [4]
A ---> F [5]
N ---> F [6]
D' ---> D [7]
P' ---> P [8]}

Par ailleurs, on se donne la chaîne terminale suivante à analyser :

x = D F F P D F

La table d'analyse est représentée sur la figure 2.3. A titre d'exemple, indiquons comment les cases t[4,3] et t[2,5] sont calculées.

Calcul de la case t[4,3]

Un élément de cette case doit engendrer la sous-chaîne de x de longueur 3 et commençant au quatrième élément, soit la chaîne : P D F. La reconnaissance de cette chaîne peut être obtenue :

soit en prenant P comme préfixe et DF comme suffixe : alors P étant le quatrième élément de la chaîne x et sa longueur étant 1, la case t[4,1] contient le symbole qui engendre P, soit P'. Pour les mêmes raisons, t[5,2] contient l'élément N'' qui engendre D F. Or il existe une règle de R :

SP ---> P' N''

On affecte donc le non terminal SP à la case t[4,3].

soit en prenant P D comme préfixe et F comme suffixe, ce qui correspond aux cases t[4,2] et t[6,1]. Or la case t[4,2] est vide et la case t[6,1] contient N et A. La vacuité de t[4,2] entraîne que cette décomposition de P D F est vaine.

Finalement, t[4,3] ne contient que SP.

Calcul de la case t[2,5]

Cette case recouvre la sous-chaîne F F P D F ; les différentes décompositions donnent :

F*F P D F : les cases t[2,1] = {N,A} et t[3,4] = {N} ainsi que la règle N ---> A N permettent d'affecter à t[2,5] la valeur {N}.

F F*P D F : on considère alors les cases t[2,2] = {N} et t[4,3] = {SP} et la règle N ---> N SP. Ceci permet d'affecter à t[2,5] une nouvelle fois N (voir remarque ci-dessous).

F F P*D P : correspond à la case t[2,3], qui est vide, et à la case t[5,2], qui contient N''. Cette décomposition ne produira donc rien.

F F P D*F : correspond à la case vide t[2,4] et à la case t[6,1], qui contient N et A ; cette décomposition est elle-aussi improductive.

Finalement, la case t[2,5] contient N deux fois.

Remarque

Les cases de la table d'analyse contiennent en fait des LISTES de non terminaux. Dans le cadre de la reconnaissance, ces listes peuvent être assimilées à des ensembles ; donc, lorsque l'on obtient dans une case plus d'une fois le même non terminal, il suffit de l'affecter une seule fois. Par contre, comme on le verra dans le paragraphe suivant, lorsqu'on utilise cette même table pour l'analyse syntaxique, la structure de liste s'impose pour chacune des cases car deux non terminaux identiques correspondent à deux analyses différentes. Nous adopterons la structure de liste, puisqu'elle est plus générale.

6	N''					
5	∅	N				
4'	∅	∅	N			
3	N''	∅	∅	SP		
2	N''	N	∅	∅	N''	
1	D'	N,A	N,A	P'	D'	N,A
0	D	F	F	P	D	F
	1	2	3	4	5	6

FIGURE 2.3

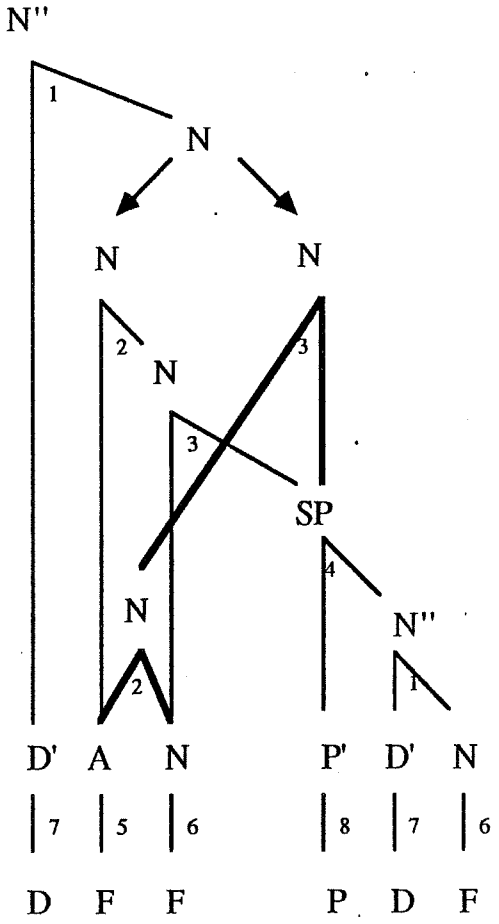
2.4. L'ANALYSEUR

2.4.1. Représentation des structures syntaxiques.

En tant qu'arborescence, une structure syntaxique peut être représentée comme un graphe [voir HOROWITZ, SAHNI -1976, ou AHO, HOPCROFT et ULLMAN -1974]. Cette structure peut être construite parallèlement au fonctionnement de l'accepteur ou déduite de la table d'analyse. La première solution permet de mener simultanément la reconnaissance et l'analyse pour la construction effective de l'arborescence ; cependant, en cas d'échec du processus de reconnaissance, on aura construit des structures partielles inutiles. La seconde solution remédie à ce défaut mais entraîne la répétition d'opérations semblables dans les phases de reconnaissance et d'impression de structures. En effet, dans cette dernière solution, l'arborescence n'est pas construite explicitement : on se contente d'exploiter la table d'analyse pour IMPRIMER une chaîne de caractères équivalente à la structure syntaxique cherchée (forme post-fixée, par exemple). En ce qui concerne l'algorithme de Cocke, nous adoptons cette dernière solution.

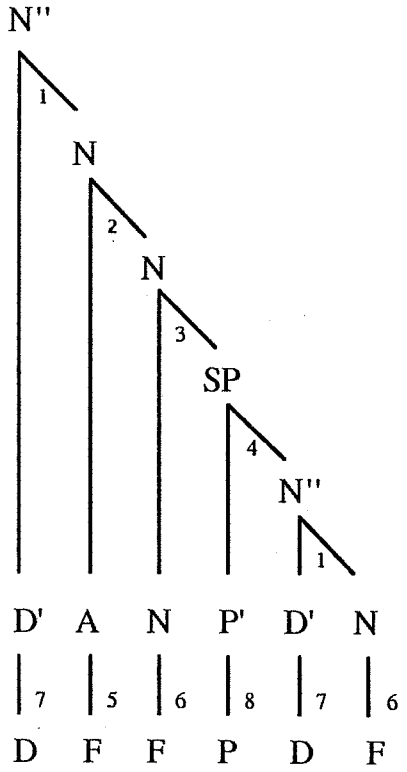
Du fait de l'ambiguïté, plusieurs structures syntaxiques peuvent être associées à une même chaîne d'entrée, chacune d'elles étant une arborescence. La table d'analyse rend compte de toutes ces structures : son exploitation va donc produire, non pas une structure, mais un ensemble de structures ayant des parties communes. L'objet obtenu, que nous appelons POLYSTRUCTURE, est en fait un di-graphe superposant la relation de descendance figurant dans une arborescence et la relation d'alternance obtenue de la façon suivante :
Si un noeud de l'arborescence peut être la racine de sous-structures concurrentes, ces racines sont reliées au noeud par la relation d'alternance. [Figure 2.4].

On imprimera une polystructure sous forme post-fixée : la relation d'alternance sera notée "v". Son utilisation imposera un parenthésage destiné à regrouper les structures alternantes d'un même noeud [Figure 2.5].

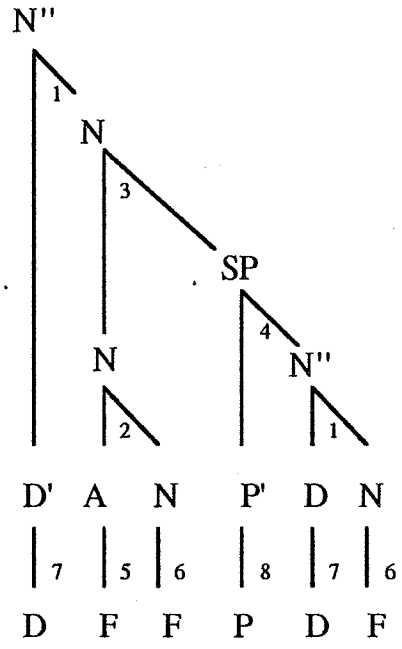


N" 1 D' 7 D N (3 N 2 A 5 F N 6 F SP 4 P' 8 P N" 1 D' 7 D N 6 F V 2 A 5 F N 3 N 6 F SP 4 P' 8 P N" 1 D' 7 D N 6 F)

FIGURE 2.4.



N" 1 D' 7 D N 2 A 5 F N 3 N 6 F SP 4 P' 8 P N" 1 D' 7 D N 6 F



N" 1 D' 7 D N 3 N 2 A 5 F N 6 F SP 4 P' 8 P N" 1 D' 7 D N 6 F

FIGURE 2.5

2.4.2. L'analyseur syntaxique de COCKE.

Le problème posé ici est celui de l'impression de la polystructure à partir de la table produite par l'accepteur. C'est un problème de parcours d'arbre binaire : une procédure récursive est bien adaptée. Cette procédure, nommée PARCOURS dans ce qui suit, a trois paramètres : deux entiers i et j définissant la sous-chaîne de x, de longueur j et commençant au ième caractère, soit :

u = x[i]x[i+1]...x[i+j-1] ;

le troisième paramètre est l'un des symboles A de la case t[i,j] de la table d'analyse, c'est-à-dire un non terminal tel que A ===>+ u.

La procédure PARCOURS(i,j,A) s'arrête si j arrive à la valeur 1 ; dans ce cas on imprime un terminal (application d'une règle terminale).

Si j>1, le fonctionnement de l'accepteur impose de chercher à appliquer une règle A ---> BC où B est à prendre dans t[i,k] et C dans t[i+k,j-k]. S'il existe une telle règle, l'appel PARCOURS(i,j,A) déclenchera les deux appels successifs PARCOURS(i,k,B) et PARCOURS(i+k,j-k,C).

Mais il peut exister plusieurs valeurs de k, chacune correspondant à une réécriture de A, donc engendrant deux nouveaux appels de la procédure PARCOURS.

La procédure PARCOURS utilise une procédure TERM dont les deux paramètres sont l'indice i d'un caractère terminal dans la chaîne x et un non terminal A ; la procédure TERM fournit le numéro p de la règle terminale A ---> x[i] et ajoute l'information p suivi de x[i] dans la représentation post-fixée de la structure cherchée.

procédure TERM(i,A)

```
début
      trouver le numéro p de la règle A ---> x[i]
      pf := "px[i] || pf
fin
```

La procédure PARCOURS est la suivante :

procédure PARCOURS(i, j, A)

```
début
    pf := "A" || pf
    si j=1 alors TERM(i,A)
    sinon début
            pf := "(" || pf
            pour k de 1 à j-1 faire
            début
                    pour chaque règle A ---> BC (de numéro p),
                    où B parcourt t[i,k]
                    et où, pour chaque B, C parcourt t[i+k,j-k] faire
                    début
                            pf := "p" || pf
                            PARCOURS(i,k,B)
                            PARCOURS(i+k,j-k,C)
                            pf := "v" || pf
                    fin
            si pf[0] = "v" alors pf[0] := ")"
            sinon pf[0] := nul
    fin
fin
```

Le processus d'impression est commandé par le programme :

programme IMPRESSION COCKE

```
début
    pf := ""
    PARCOURS(1,n,S)
fin.
```

Commentaires :

(1) la chaîne pf est la représentation post-fixée de la polystructure associée à la chaîne x ;
(2) pf[0] désigne le dernier caractère de la chaîne pf (voir la section 2 du chapitre 2) ; l'opérateur "||" désigne la concaténation de chaînes ;
(3) la procédure TERM(i,A) renvoie à la construction de la première ligne de la table d'analyse (application des règles terminales).

2.4.3. Exemple d'analyse.

Reprenons l'exemple traité pour l'accepteur (paragraphe 4). A partir de la table d'analyse [Figure 2.3], imprimons la polystructure associée à la chaîne :

x = D F F P D F

Débutons par l'appel de PARCOURS(1,6,N''), puisque nous souhaitons imprimer la polystructure associée à la chaîne x (qui débute en 1 et de longueur 6) et dont la racine est

l'axiome N''.

Nous donnons ci-dessous le début du travail du programme d'impression et nous renvoyons le lecteur aux figures 2.4 et 2.5 pour une illustration du fonctionnement de l'impression sur l'ensemble des valeurs.

PARCOURS (1, 6, N'') :

1. impression de N''

2. j > 1, alors impression de "(" et k = 1 on a t[1,1] = {D'} et t[2,5] = {N} ; la règle N'' ---> D' N s'applique. D'où l'impression de "1" et l'appel de PARCOURS (1,1,D') et de PARCOURS (2,5,N).

k = 2 les ensembles concernés sont t[1,2] = {N''} et t[3,4] = {N'}, mais il n'existe pas de règle N'' ---> N'' N'.

k = 3 de même, il n'existe pas de règle N'' ---> N'' SP, or les ensembles concernés sont t[1,3] = {N''} et t[4,3] = {SP}.

k = 4 la case t[1,4] étant vide, il n'y a pas de solution.

k = 5 c'est la case t[1,5] qui est vide.

2.5. VARIANTE DE YOUNGER

L'algorithme exposé précédemment paraît être le contemporain de ceux de YOUNGER et de KASAMI. L'algorithme de YOUNGER peut être considéré comme une version booléenne de celui de COCKE. Le résultat de l'accepteur est une table d'analyse de même forme que celle de COCKE. Mais dans ce dernier cas, les cases contiennent des ensembles de non terminaux, alors que dans la version de Younger, une case est un vecteur booléen défini de la façon suivante : Les non terminaux de la grammaire sont numérotés et représentés par leur numéro. Soit s la table d'analyse de Younger ; son élément (i,j) est un vecteur booléen à q composantes, q étant le nombre de non terminaux. On peut considérer que s est une matrice booléenne à trois indices : i et j ayant la même signification que dans l'algorithme de Cocke et m indiquant le numéro de la composante dans le vecteur booléen.

Ainsi, s[i,j,m] = 1 si et seulement si le non terminal A[m] appartient à t[i,j]. Les deux tables d'analyse s et t contiennent donc la même information. De plus, les algorithmes de construction de ces tables sont similaires : d'abord la construction de la sous-matrice pour j=1 (application des règles terminales) se fait par :

s[i, j, m] := 1 si A[m] ---> x[i] appartient à R.

Ensuite, la valeur de s[i,j,m], pour j>1, est la suivante :

union union s[i,k,m1] s[i+k,j-k,m2]

où la seconde union opère pour les valeurs de k comprises (bornes incluses) entre 1 et j-1, et où la première union opère pour tous les couples m1 et m2 tels que A[m] ---> A[m1] A[m2] est une règle de R (m1 et m2 sont évidemment compris entre 1 et q).

Il est facile de vérifier que ce mode de calcul de la table s correspond de façon duale à celui de la table t pour l'accepteur de Cocke.

3. ALGORITHME D'EARLEY

3.1. CARACTERISTIQUES

L'exposé qui suit, se fonde sur les travaux d'EARLEY [1968 et 1970] et le livre de AHO et ULLMAN [1972, pp.320-330] Cet analyseur est descendant car les structures sont construites en partant de l'axiome. Il est général car il s'applique à toute grammaire hors contexte. Il est multiple car il engendre toutes les structures syntaxiques en une seule passe. Enfin, il est prédictif puisqu'il peut n'engendrer que les structures compatibles avec les m symboles suivants dans la chaîne à analyser. Si m=0, la prédictivité n'intervient pas. Le plus souvent, on choisira , pour des raisons de performance, m=1. Cependant m peut être quelconque.

3.2. NOTATIONS ET DEFINITIONS PARTICULIERES.

3.2.1. La grammaire

Soit une grammaire hors contexte quelconque G=(T,N,S,R). Pour les besoins de l'algorithme :

(i) on adjoint à T, un nouveau terminal noté $, marquant la fin de la chaîne d'entrée.
(ii) on adjoint à N, un nouveau symbole PHI qui devient l'axiome.
(iii) on ajoute à R, la règle PHI --> S $.
(iv) chacune des règles de R est numérotée de 0 pour cette dernière à r.
(v) la longueur d'une règle est égale au nombre de symboles de sa partie droite.

Remarque

La seule restriction que nous imposerons à la grammaire est qu'elle ne contienne pas de règle engendrant la chaîne vide. En effet, il est peu probable que ce cas se présente dans les applications que nous traiterons. De plus, il rend l'algorithme beaucoup plus complexe. Comme d'une part, il existe un algorithme décidant en un nombre fini de pas, si une grammaire hors contexte engendre la chaîne vide et, d'autre part, comme toute grammaire hors contexte peut être transformée en une grammaire équivalente ne contenant pas de règles A ---> ε, sauf éventuellement S ---> ε, cette restriction peut être contournée.

3.2.2. La chaîne d'entrée.

Soit x = x[1] x[2]...x[n], la chaîne à analyser, et soit m la longueur de la sous-chaîne lue dans la chaîne d'entrée et sur laquelle repose la prédictivité de l'algorithme ; alors la chaîne analysée sera x[1] x[2]...x[n] x[n+1]...x[n+m+1] où x[n+k] = $ pour k de 1 à m+1.

3.2.3. Définition et calcul de l'ensemble des premiers de X.

Nous nous plaçons ici dans le cas restrictif où la prédiction sur la chaîne d'entrée se fait sur un seul symbole. Pour le cas général, se reporter à [AHO, ULLMAN, 1972, pp 300 et 357].

3.2.3.1. Définition de Prem(X).

Soit X un élément de V ; si X est dans T alors Prem(X) = {X}. Supposons maintenant que X appartienne à N. On peut définir Prem(X) de la façon récurrente suivante. Soit X --> Yu une règle de R où Y appartient à V et u, reste de la partie droite à V^*. Si Y est dans T, alors Y appartient à Prem(X) ; sinon, il existe une règle Y --> Zv dans R. Le cas de Z est identique à celui de Y : ou bien Z est terminal et il appartient à Prem(X), ou bien c'est un non terminal et nous poursuivons la récurrence. Celle-ci se termine forcément puisque R et V sont finis, et que la grammaire est supposée réduite, c'est-à-dire ne contenant pas de symboles improductifs. En définitive, Prem(X) est l'ensemble des terminaux qui peuvent débuter une chaîne engendrée par X.

3.2.3.2. Calcul de Prem(X).

Pour chaque X, on construit une suite d'ensembles F[i,X].

1 pour tous les X de T et pour i entier positif ou nul F[i,X] = {X}

2 pour les X appartenant à N, on construit d'abord F[0,X] comme l'ensemble des terminaux qui débutent les parties droites de règles dont X est partie gauche.

3 supposant construits les ensembles d'indices 0,1,...,i-1 pour X, on construit F[i,X] comme étant la réunion de F[i-1,X] et des F[i-1,Y] où Y est un symbole débutant la partie droite d'une règle dont X est partie gauche.

4 on s'arrête lorsque, pour tous les X de V, les ensembles F[i-1,X] et F[i,X] sont égaux.

3.2.4. Les structures de données

3.2.4.1. Le vecteur de listes

L'algorithme utilise un vecteur V dont les éléments sont notés V[i] pour i de 0 à n+1 ; chaque élément V[i] est une liste d'états ordonnée et sans répétition. Par conséquent, ajouter un état à V[i] consiste à insérer cet état en fin de liste après avoir vérifié qu'il n'existait pas déjà.

3.2.4.2. Les états

Un état est un quadruplet <r,k,j,a> où :

r est le numéro d'une règle de R.
k est un entier compris entre 0 et la longueur de la règle r.
j est un entier compris entre 0 et n+1
a est un élément de T^*, de longueur m, nombre de caractères de la prédiction.

Un état <r,k,j,a> où k est égal à la longueur de la règle r, est dit **état final.**

Autre notation

Soit A --> B[1] B[2]...B[l(r)] la règle r ; on représentera aussi un état <r,k,j,a> sous cette forme plus explicite :

[A ---> B[1]...B[k] * B[k+1]...B[l(r)] , j , a]

3.2.4.3. Interprétation

Point de vue statique

L'appartenance d'un état <r,k,j,a> à V[i] se traduit ainsi : soit A ---> B[1]B[2]...B[l(r)], la règle r de R, soit u la sous-chaîne de x débutant en x[j+1] et de longueur (i-j), l'entier k représente alors la partie B[1] B[2]...B[k] de la règle r qui engendre la chaîne terminale u. De plus, la partie B[k+1]...B[l(r)] de la règle r engendre une chaîne terminale v, non encore déterminée. Donc A partie gauche de la règle r engendre la chaîne terminale uv. L'élément a de T* est la sous-chaîne qui, dans x, doit figurer à droite de v pour que le processus d'analyse puisse se poursuivre. [Figure 2.6]

Remarque sur les indices i et j

La sous-chaîne u de x débutant en x[j+1] et de longueur (i-j), se note aussi x[j+1]...x[i]. Dans le cas où i et j sont égaux, il s'agit alors de la sous-chaîne débutant en x[i+1] et de longueur nulle. Cette sous-chaîne est vide, et i représente alors un point dans la chaîne. Comme G ne contient pas de règle engendrant la chaîne vide, aucun symbole ne peut engendrer la chaîne vide ; donc, dans le cas où i et j sont égaux, k vaut 0.

Point de vue dynamique

Le processus de reconnaissance se déroule par étapes successives, chaque étape conduit à la reconnaissance d'un caractère de la chaîne d'entrée, les caractères étant lus de gauche à droite. Une étape i débute après la reconnaissance du caractère x[i]. Elle consiste en deux opérations ; construction des structures partielles engendrant les sous-chaînes se terminant en x[i], et esquisse de toutes les dérivations qui peuvent engendrer une sous-chaîne commençant en x[i+1]. Parmi ces esquisses, celles qui sont compatibles avec x[i+1] entraînent la reconnaissance de ce symbole et donc le passage à l'étape suivante. Par contre, si aucune dérivation ne conduit à x[i+1], cela signifie que la sous-chaîne x[1]...x[i+1] n'est pas un préfixe d'un mot de L(G) ; la chaîne x est donc rejetée et l'algorithme s'arrête. L'étape i+1 ne peut débuter que si x[i+1] est reconnu et donc que V[i+1] n'est pas vide.

L'étape 0 consiste seulement en l'esquisse de toutes les dérivations issues de l'axiome PHI et se termine par la reconnaissance de x[1]. L'étape n, la dernière, doit conduire, si x est élément de L(G), à la construction de toutes les structures engendrées par S à partir de la chaîne d'entrée. Elle se termine par la reconnaissance de x[n+1] = $.

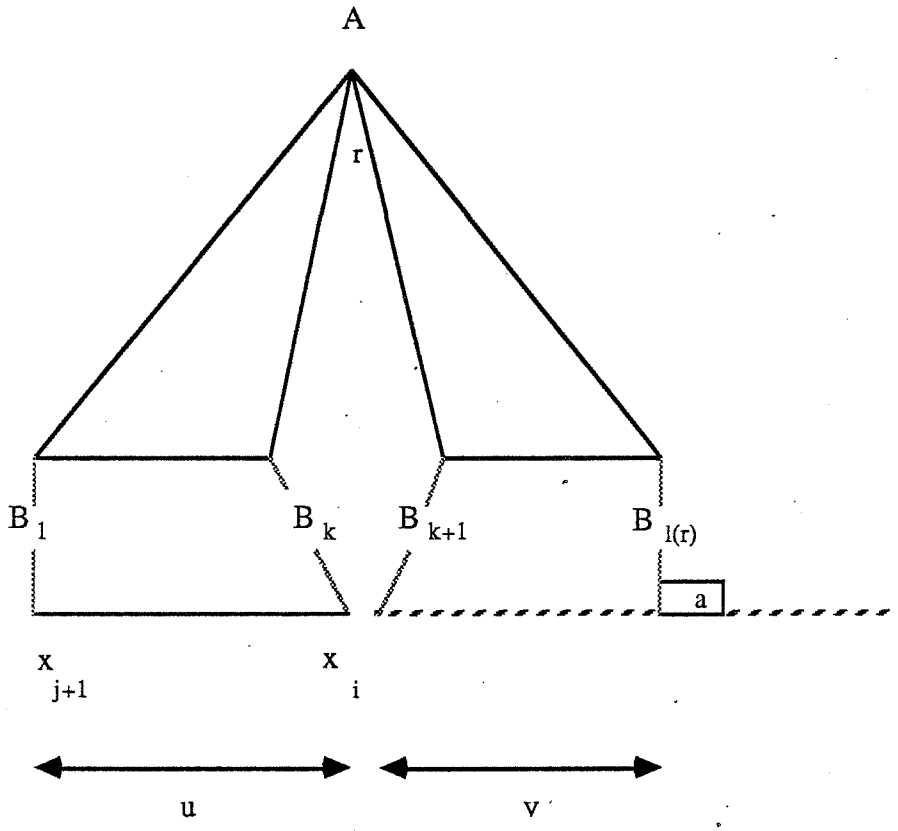


FIGURE 2.6

3.3. ACCEPTEUR

3.3.1. Description du fonctionnement pour m = 1

3.3.1.1. Initialisation

On affecte à V[0], l'état <0,0,0,$> où r = 0 puisque la règle est PHI ---> S $, k = 0 puisqu'aucun symbole de la règle 0 n'a été réécrit, j = 0 puisque le caractère de la chaîne à analyser est x[j+1] = x[1], mais qu'il ne l'est pas encore donc (i-j) = 0, a = $ puisque la chaîne à reconnaître est x $ $ et que PHI engendre S $.

3.3.1.2. Description de l'étape i

L'étape i débute à l'issue de la reconnaissance du caractère x[i], reconnaissance qui se traduit par la création d'un état dans V[i].

1 donc si V[i] est vide, la chaîne x est rejetée puisque x[i] ne peut être engendré à ce moment par la grammaire et l'algorithme s'arrête.

2 si i = n+1 alors si V[n+1] contient l'état <0,2,0,$>, ce qui signifie que PHI engendre la chaîne x[1]...x[n+1], x est élément de L(G) et l'algorithme s'arrête.

3 si i < n+1 et si V[i] n'est pas vide, pour chacun des états <r,k,j,a> de V[i], du premier jusqu'au dernier on procède :

soit à une dérivation (ou "predictor" d'après Earley), si le (k+1)ème symbole de la règle r est non terminal

soit à un balayage (ou "scanner") s'il est terminal

soit à une complétion (ou "completer") si k = l(r) et si a = x[i+1]

Dérivation à partir de l'état <r,k,j,a> de V[i], c'est-à-dire de l'état [A ---> B[1]...B[k]*B[k+1] B[k+2]...B[l(r)], j, a] de V[i].

Si le symbole B[k+1], situé à droite du point dans la règle r, est non terminal on considère toutes les règles de R dont ce symbole est partie gauche. Pour chacune de ces règles, on ajoute à V[i] des états construits de la façon suivante [figure 2.7] : soit r' l'une de ces règles ; elle s'écrit B[k+1] ---> C[1] C[2]...C[l(r')]. On lui associe des états <r',0,i,b>. L'application de la règle r déclenche donc, par l'intermédiaire de B[k+1], l'application de la règle r'. Au moment de la création d'un état, la règle r' n'a pas été encore appliquée. Donc le point dans la règle est à gauche de C[1], ce qui explique la valeur nulle du second élément de l'état. De plus, les explications du paragraphe 2.4.3 montrent que B[1]...B[k] engendre une partie de la chaîne x qui se termine en x[i]. Donc la sous-chaîne de x engendrée par B[k+1] commence en x[i+1], ce qui explique que le troisième élément de l'état est posé égal à i. En outre, comme le symbole B[k+1] n'a encore engendré aucun caractère de x, la longueur de la sous-chaîne de x engendrée par ce symbole est nulle (on est ici dans le cas vu précédemment où k=0 et i=j). Elle se termine donc en x[i]. C'est pourquoi, l'état est ajouté à

V[i].

Le dernier élément b de l'état est calculé ainsi : ce terminal b a pour but de renseigner sur le premier symbole de la chaîne engendré par l'application de la règle suivante. Dans notre cas, la règle suivante sera celle qui poursuit l'application de la règle r, une fois la règle r' entièrement exploitée. Donc b est un terminal, élément de Prem(B[k+2]).

En résumé, l'état <r,k,j,a> de V[i] , si B[k+1] est non terminal, engendre dans V[i] tous les états <r', 0, i, b> où r' représente tour à tour chacune des règles de R qui ont B[k+1] en partie gauche, et où b parcourt Prem(B[k+2]).

Remarque : les règles récursives à gauche.
Soit <r',0,i,b> un état de V[i] engendré par dérivation à partir d'un état <r,k,j,a> de V[i]. Si la règle r' est récursive à gauche et donc de la forme A ---> A C[2]...C[l(r')], l'état <r',0,i,b> est de la forme [A ---> *A C[2]...C[l(r')], i, b]. A est non terminal, donc par dérivation on engendre de nouveaux états <r',0,i,b'> dans V[i] où b' décrit Prem(C[2]). Puis ces nouveaux états engendrent à leur tour les mêmes états <r',0,i,b'> dans V[i]. Le processus amorcé est sans fin. Aussi pour éviter ces boucles infinies, l'algorithme n'ajoute-t-il un état dans une liste d'états que s'il n'existe pas déjà.

Balayage à partir de l'état <r,k,j,a> de V[i], c'est-à-dire de l'état [A ---> B[1]...B[k]*B[k+1] B[k+2]...B[l(r)], j, a] de V[i].

Le balayage ne s'applique que si le symbole B[k+1] situé à droite du point dans la règle est terminal. Comme la partie B[1]...B[k] de la règle r engendre une sous-chaîne de x se terminant en x[i], le balayage commence par la comparaison de B[k+1] et de x[i+1].

Premier cas : si B[k+1] est différent de x[i+1], il y a incompatibilité entre le symbole engendré par la grammaire et celui présent dans la chaîne. On est alors dans une impasse. On ne peut donc progresser ni dans la reconnaissance de la chaîne x ni dans l'application de la règle r. De ce fait, l'état <r,k,j,a> de V[i] n'engendre aucun état.

Deuxième cas : si B[k+1] et x[i+1] sont identiques, la règle r a engendré le symbole terminal x[i+1]. On ajoute alors l'état <r, k+1, j, a> soit [A ---> B[1]...B[k] B[k+1]*B[k+2]...B[l(r)], j, a] à V[i+1] pour indiquer que la partie B[1]...B[k+1] de r a engendré une sous-chaîne de x se terminant en x[i+1]. Cette chaîne commence en x[j] comme celle engendrée par B[1]...B[k]. Comme aucune autre règle n'a été appliquée, le terminal a, quatrième élément de l'état <r,k+1,j,a> de V[i+1], est le même que celui de l'état <r,k,j,a> de V[i] qui a provoqué le processus de balayage.

Complétion à partir de l'état <r,k,j,a> de V[i], où k = l(r), c'est-à-dire de l'état [A ---> B[1]...B[l(r)]*, j, a] de V[i].

La complétion s'applique en fait sous deux conditions :

1 celle déjà signalée k = l(r) ; elle signifie que la partie droite de la règle r a été complètement utilisée pour engendrer la sous-chaîne x[j+1]...x[i] de x. Autrement dit, cette sous-chaîne est engendrée par le non terminal A , partie gauche de r.

2 la deuxième condition impose qu'il y ait coïncidence entre le symbole a prédit et le symbole x[i+1], prochain symbole à reconnaître dans la chaîne d'entrée.

L'opération de complétion, elle-même, exploite le fait qu'une règle a été entièrement appliquée. Elle nous fait remonter à toutes les règles q à partir desquelles la règle r a été appliquée. De façon précise, ces règles q ne nous intéressent que si elles sont éléments d'états de V[j], puisque la sous-chaîne engendrée par la règle r débute en x[j+1]. Dans ces états de V[j], la règle q intervient sous la forme U ---> ...* A... [Figure 2.8]. Soit <q,k',j',a'> l'un de ces états. La complétion entraîne l'ajout de <q,k'+1,j',a'> à V[i] puisque l'on a engendré une sous-chaîne de x se terminant en x[i]. Tous les états <q,k',j',a'> de V[j] sont soumis à la même opération de complétion.

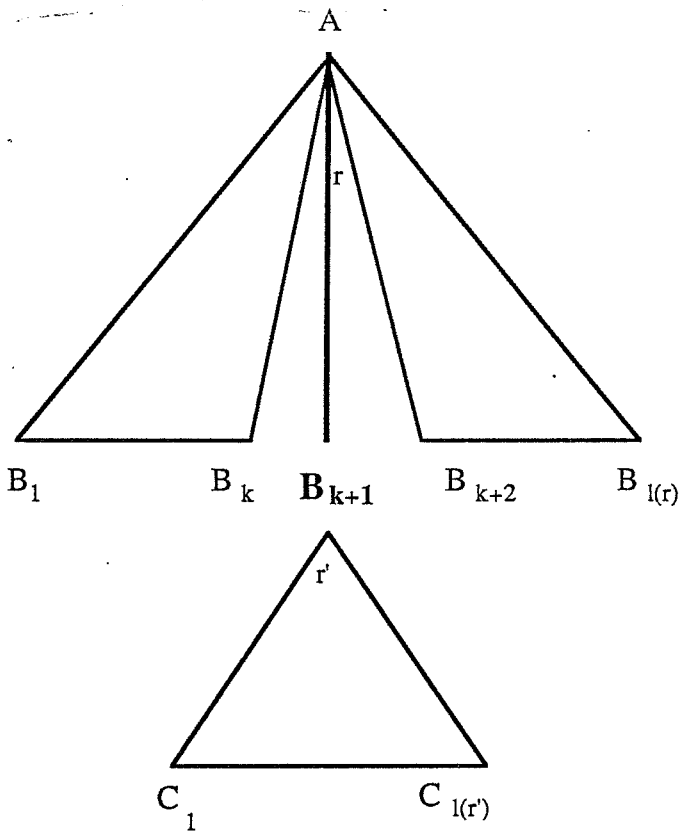


FIGURE 2.7

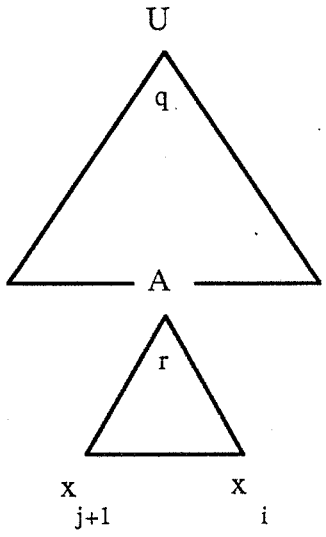


FIGURE 2.8

3.3.2. Algorithme d'Earley

Procédure reconnaissance;

```
début
 {1. initialisation}
 ajouter <0,0,0,$> à V[0] ;
 rejet := faux ;
 i:= 0;

 {2. corps de l'algorithme}

 tant que i<=n et non rejet faire
 début
      pour chaque état <r,k,j,a> de V[i] faire
      début
           si k<l(r) alors
                { dérivation}
                si B[k+1] est non terminal alors
                début
                     ajouter à V[i] tous les états <r',0,i,b> où :
                     - r' prend pour valeur les numéros de règle dont B[k+1]
                     est partie gauche.
                     - b parcourt Prem(B[k+2])
                fin
                { balayage }
                sinon si x[i+1] = B[k+1] alors ajouter <r,k+1,j,a> à V[i+1]
           sinon
                { completion }
                si a = x[i+1] alors
                     pour chacun des états <q,k',j',a'> de V[j] tels que :
                     le (k+1)ème symbole de la partie droitesoit égal à la part
                     ajouter à V[i] l'état <q,k'+1,j',a'>.
      finpour
      si V[i+1] est vide alors rejet := vrai sinon i:= i+1 ;
 fintant

 {3. sortie de l'algorithme}
 si i=n et <0,2,0,$> appartient à V[n+1]
      alors x est élément de L(G),
      sinon x est rejetée.
fin
```

3.3.3. Exemple

Reprenons l'exemple donné pour l'algorithme de Cocke, et choisissons m=1. D'après les conventions de l'algorithme d'Earley, la grammaire G_0 devient :

G = ({D,F,P,$}, {N,N',N'',A,SP,PHI}, PHI, R)

R = {
PHI ---> N'' $ [0]
N'' ---> D N' [1]
N' ---> N [2]
N ---> A N [3]
N ---> N SP [4]
SP ---> P N'' [5]
A ---> F [6]
N ---> F [7] }

Les ensembles des premiers de chacun des symboles de T et de N sont :

Prem(N) = Prem(N') = Prem(A) = Prem(F) = {F}
Prem(PHI) = Prem(N'') = Prem(D) = {D}
Prem(SP) = Prem(P) = {P}
Prem($) = {$}

La chaîne à analyser est x= D F F P D F $ $ où n = 6 ;

V[0] est initialisé avec
[PHI ---> * N'', 0, $].

N'' est non terminal donc par dérivation, on ajoute à V[0] l'état
[N'' ---> * D N', 0, $].

D est terminal, et x[1] = D, donc par balayage on ajoute à V[1] l'état
[N'' ---> D * N', 0, $].

L'étape 0 est terminée, car tous les états de V[0] ont été traités. V[1] n'est pas vide, l'étape 1 peut commencer. Dans le seul état de V[1], N' est non terminal. On ajoute donc, par dérivation, à V[1] l'état
[N' ---> * N, 1, $].

Ce dernier état engendre dans V[1] par dérivation les états
[N ---> * A N ,1, $] (1)
[N ---> * N SP , 1, $] (2)
[N ---> * F , 1, $] (3)

Par dérivation à partir de (1), on ajoute à V[1]
[A ---> * F, 1, F] (4)
puisque Prem(N) = {F}.

Par dérivation à partir de (2) on ajoute à V[1]

[N ---> * A N , 1, P] (5)
[N ---> * N SP , 1, P] (6)
[N ---> * F , 1, P] (7)

puisque Prem(SP) = {P}.

(3) et (4) engendrent dans V[2] par balayage

[N ---> F * , 1, $]
[A ---> F * , 1, F]

(5) engendre dans V[1] par dérivation l'état

[A ---> * F, 1, F]

identique à (4) ; on ne l'ajoute donc pas.

(6) engendre par dérivation les états

[N ---> * A N , 1, P]
[N ---> * N SP , 1, P]
[N ---> * F , 1, P]

qui existent déjà dans V[1].

(7) engendre dans V[2] par balayage

[N ---> F*, 1,P]

Tous les états de V[1] ont été examinés et à l'issue de l'étape 1, V[2] contient :

[N ---> F * , 1, $] (8)
[A ---> F * , 1, F] (9)
[N ---> F * , 1, P] (10)

(8) est un état final puisque le point dans la règle est en fin de règle, mais le caractère de prédiction $ n'est pas identique à x[3] qui vaut F. Donc la complétion n'agit pas sur (8).

(9) par complétion engendre les états

[N ---> A * N , 1, $] (11)
[N ---> A * N , 1, P] (12)

puisque dans V[1], les états (1) et (5) ont le symbole A à droite du point dans la règle.

Chacune des opérations ayant été illustrée, on donne maintenant les listes d'états engendrées pour l'analyse de la chaîne x.

V[0]

[PHI --> * N'', 0, $].
[N'' --> * D N', 0, $].

V[1]

[N'' --> D * N', 0, $].
[N' --> * N, 1, $].
[N --> * A N ,1, $]
[N --> * N SP , 1, $]
[N --> * F , 1, $]
[A --> * F, 1, F]
[N --> * A N , 1, P]
[N --> * N SP , 1, P]
[N --> * F , 1, P]

V[2]

[N --> F * , 1, $]
[A --> F * , 1, F]
[N --> N * , 1, P]
[N --> A * N , 1, $]
[N --> A * N , 1, P]
[N --> * A N , 2, $]
[N --> * N SP, 2, $]
[N --> * F , 2, $]
[N --> * A N , 2, P]
[N --> * N SP, 2, P]
[N --> * F, 2, P]
[A --> * F, 2, F]

V[3]

[N --> F * , 2, $]
[N --> F * , 2, P]
[A --> F * , 2, F]
[N --> A N * , 1, $]
[N --> A N * , 1, P]
[N --> N * SP, 2, $]
[N --> N * SP, 2, P]
[N' --> N *, 1, $]
[N --> N * SP, 1, $]
[N --> N * SP, 1, P]
[SP --> * P N'', 3, $]
[SP --> * P N'', 3, P]

V[4]

[SP --> P * N'', 3, $]
[SP --> P * N'', 3, P]

[N'' --> * D N', 4, $]
[N'' --> * D N', 4, P]

V[5]

[N'' --> D * N', 4, $]
[N'' --> D * N', 4, P]
[N' --> * N, 5, $]
[N --> * A N , 5, $]
[N --> * N SP, 5, $]
[N --> * F , 5, $]
[A --> * F , 5, F]
[N --> * A N , 5, P]
[N --> * N SP, 5, P]
[N --> * F , 5, P]
[N' --> * N, 5, P]

V[6]

[N --> F *, 5, $]
[A --> F *, 5, F]
[N --> F *, 5, P]
[N' --> N *, 5, $]
[N --> N * SP, 5, $]
[N --> N * SP, 5, P]
[N' --> N *, 5, P]
[N'' --> D N' *, 4, $]
[N'' --> D N' *, 4, P]
[SP --> P N'' *, 3, $]
[SP --> P N'' *, 3,P]
[SP --> * P N'', 6, $]
[SP --> * P N'', 6, P]
[N --> N SP *, 2, $]
[N --> N SP *, 2, P]
[N --> N SP *, 1, $]
[N --> N SP *, 1, P]
[N --> A N *, 1, $]
[N --> A N *, 1, P]
[N --> N * SP, 2, $]
[N --> N * SP, 2, P]
[N' --> N *, 1, $]
[N --> N * SP, 1, $]
[N --> N * SP, 1, P]
[N'' --> D N' *, 0, $]
[PHI --> N'' * $, 0, $]

V[7]

[PHI --> N'' $ *, 0, $]

3.4. ANALYSEUR

Comme pour l'analyseur de Cocke, le but est de construire une polystructure dont les noeuds sont des ensembles d'états de l'analyseur. Pour cela, deux méthodes sont possibles : construire la structure parallèlement au fonctionnement de l'accepteur, dans ce cas il n'y a pas enchaînement des algorithmes accepteur puis analyseur, mais substitution du second au premier ; ou alors construction de la structure à partir des listes d'états créées par l'accepteur. Nous étudierons ces deux méthodes.

3.4.1. Première méthode.

Cette méthode s'inspire de celle exposée par Earley dans [EARLEY 1968].

3.4.1.1. Notations et définitions particulières

La structure de données adoptée pour représenter une telle structure, s'inspire de la structure de données utilisée pour représenter les arbres quelconques en Pascal. Un noeud de la polystructure se représente ainsi :

```
pnoeud = ^tnoeud ;

tnoeud =    record
                    symbole de V
                    aine : pnoeud ;
                    frere : pnoeud
                    alter : pnoeud ;
            end ;
```

D'autre part, un état de l'analyseur est un quintuplet formé :

- du numéro d'identification d'une règle de la grammaire : r ;
- du point dans la règle : k ;
- du point dans la chaîne d'entrée : j ;
- du symbole de T attendu : a ;
- du noeud racine de la polystructure engendrée par l'état : P ;

3.4.1.2. Fonctionnement de l'analyseur

Il nous faut modifier l'algorithme d'Earley afin qu'il procède simultanément à la reconnaissance et à la construction de la structure. La construction de la structure s'opère à travers les modifications que subit la racine P de la polystructure, lors de sa création par dérivation, balayage ou complétion.

Dérivation :

un état créé par dérivation est affecté d'une racine vide.

Balayage :

Un état créé par balayage est de la forme

A --> B[1]...B[k+1]*B[k+2]...B[l(r)] ;

Il provient d'un état dont la racine P1 a pour aîne B[1], B[1] ayant pour frère B[2],..., B[k-1] ayant pour frère B[k] et de la réécriture de B[k+1] en x[i+1] symbole terminal de la chaîne d'entrée. La racine du nouvel état s'obtient donc ainsi : on crée un noeud P2 qui contient le symbole x[i+1], c'est donc une feuille. On juxtapose P1 et P2 : B[k] a maintenant pour frère P2, et enfin on crée une racine P qui a pour aîné P1 juxtaposé à P2 ; cette racine P est la racine du nouvel état créé par balayage.

Complétion

Un état créé par complétion provient :

d'un état courant de la forme B[k+1] --> C[1]...C[l(r')]*,
et d'un état candidat de la forme A --> B[1]...B[k]*B[k+1]...B[l(r)].

L'état candidat a pour racine P1, l'état courant a pour racine P2. On obtient la racine P du nouvel état en juxtaposant l'aîné de P1 à P2, et en créant une nouvelle racine P qui a pour aîné P1 juxtaposé à P2. Si le nouvel état n'existe pas on le crée en lui affectant la racine P, s'il existe déjà alors il a une racine P' et P devient un alternant de P'.

3.4.1.3. L'algorithme

Pour construire la polystructure on utilise les trois procédures suivantes : juxtaposer, enraciner et ajouter un alternant. Ces procédures sont une adaptation à la polystructure, des opérations de base sur les arbres syntaxiques décrites par C. PAIR [1970].

3.4.1.3.1. juxtaposer (ou concaténer)

Dans le cas d'un arbre syntaxique cette procédure a pour paramètres deux listes de noeuds fils1 et fils2. Elle concatène ces deux listes et le résultat est dans fils1. Si nous voulons généraliser cette procédure au cas d'une polystructure, il faut tout d'abord tenir compte du fait qu'un même noeud N1 peut être utilisé dans plus d'une structure syntaxique, et donc que N1 peut, dans une structure, avoir pour frère un noeud N2 et, dans une autre, un noeud N3. Pour parer à cette éventualité soit nous créons sur un noeud autant de liens "frère" qu'il en est besoin (on ne peut le savoir a priori), soit nous créons une copie de fils1 et concaténons la copie de fils1 à fils2 ; c'est le choix que nous avons fait.

procedure juxtaposer (fils1, fils2 : pnoeud ; var aine : pnoeud) ;

```
debut
   copier fils1 dans aine ;
   si aine = nil alors aine := fils2
   sinon
      affecter au dernier frère de aine, fils2 comme frère
fin
```

3.4.1.3.2. enraciner

La procédure enraciner a pour paramètres les noeuds aîné et père. Elle est conforme à celle décrite par C. PAIR sur les arbres, puisqu'une racine n'a ni frère ni alternant.

procedure enraciner (aine : pnoeud ; var pere : pnoeud) ;

```
debut
  creer un noeud pere ;
  pere^.aine := aine ;
  pere^.frere := nil ;
  pere^.alter := nil ;
fin
```

3.4.1.3.3. ajouter un alternant

Cette procédure consiste à ajouter un alternant à une liste composée sur la relation d'alternance, de noeuds déjà construits. Ces noeuds déjà construits peuvent avoir contribué à la construction de la polystructure à un niveau supérieur. Or des noeuds en relation d'alternance sont les aînés et frères des mêmes noeuds. En conséquence, ajouter un alternant à une liste d'alternants c'est le fondre au sein de la polystructure déjà construite au niveau supérieur. On prendra donc le soin d'insérer un nouvel alternant en fin de la liste des alternants, ainsi il héritera des noeuds "père" et "frère" des alternants déjà construits.

procédure ajouter_alternant (racine, alter : pnoeud) ;

```
insérer alter en queue de la liste racine suivant la relation
d'alternance.
```

3.4.2. Le parcours d'une polystructure

A l'issue de l'analyse, la polystructure est construite. Pour éditer le résultat de l'analyse, il faut imprimer la polystructure. Le parcours d'une polystructure s'opère à partir de la racine, donc pour nous à partir de la racine P associée à l'état <0,2,0,$> de l'accepteur. Il doit satisfaire aux exigences suivantes :

1. si la polystructure est réduite à un arbre, le parcours se réduit à un parcours d'arbre.

2. si la polystructure contient plusieurs arbres, tous les arbres inclus doivent être parcourus.

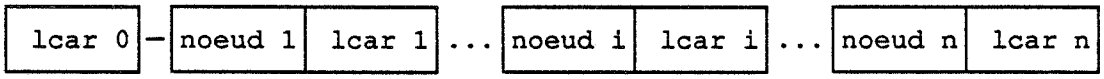
3. le résultat du parcours est l'impression sur un fichier de chaque arbre sous forme parenthésée.

4. le parcours choisi est le parcours en profondeur.

La procédure centrale du parcours d'une polystructure est donc une procédure classique de parcours d'arbre au sein de laquelle un traitement particulier s'opère sur les noeuds ayant des alternants. En effet, lorsque lors du parcours d'arbre, on rencontre un tel noeud, plusieurs alternatives pour poursuivre le parcours apparaissent ; il y en a autant que d'éléments dans la liste des alternants de ce noeud. Dans ce cas on arrête sur ce noeud le processus de descente, on ne passe donc pas à l'aîné, et l'on conserve dans une phrase le parcours déjà effectué suivi de l'adresse du noeud ayant des alternants. Puis on poursuit le parcours à partir des frères de ce noeud jusqu'à rencontrer un autre noeud avec alternants, noeud qui est traité comme précédemment.

Si l'on soumet la racine de la polystructure à une telle procédure on obtient en sortie une phrase composée :

- soit d'une liste de caractères, s'il n'y a qu'un arbre - soit d'une séquence d'éléments de la forme :



L'écriture rigoureuse de ce parcours nécessite une structure de données pour représenter une phrase :

```
pcar = ^tcar ;
tcar =        record
              car : char ;
              suivant : pcar ;
      end ;
tliste_car =         record
              tete : pcar ;
              queue : pcar
      end ;
pphrase = ^tphrase ;
tphrase =     record
              noeud : pnoeud ;
              lcar : tliste_car ;
              suivant : pphrase ;
      end ;
```

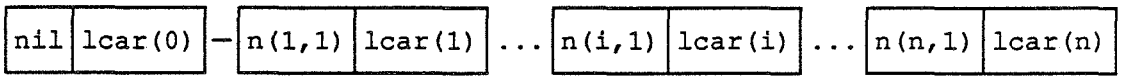
Avant de parcourir la polystructure, on crée une phrase qui contient un seul élément dont tous les champs sont initialisés à nil. La procédure parcours s'écrit récursivement de la façon suivante :

```
parcours (noeud : pnoeud) ;
debut
      si noeud^.alter <> nil alors
            créer un nouvel élément de phrase
            dont le champ noeud contient noeud et
            dont le champ lcar est vide ;
            insérer cet élément en fin de phrase ;
      fin
      sinon
            éditer (noeud^.symbole) ;
            si noeud^.aine <> nil alors
                  éditer ('[') ;
                  parcours (noeud^.aine, phrase) ;
                  éditer (']') ;
            fin
      fin

      si noeud^.frere <> nil alors
            éditer (',') ;
            parcours (noeud^.frere, phrase) ;
      fin
fin
```

Remarque

la procédure "éditer" a pour paramètre une chaîne de caractères et pour fonction d'ajouter cette chaîne de caractères en fin de la liste de caractères du dernier élément de la phrase.

A l'issue du parcours de la racine, on obtient donc dans le cas le plus général une phrase de la forme :



Chaque noeud n[i,1] pour i = 1 à n est tête d'une liste d'alternants. Soient donc n[i,j] pour j = 1 à p_i les alternants de cette liste. Il est clair qu'à chaque alternant n[i,j] correspond une polystructure de racine n[i,j] distincte des polystructures de racine n[i,j'] pour j'≠j. Donc pour obtenir l'ensemble des polystructures différentes issues de cette phrase, il faut énumérer tous les multiplets à n éléments construits à partir des listes :

(n[1,1]...n[1,j_1]...n[1,p_1]) ... (n[i,1]...n[i,j_i]...n[i,p_i]) ... (n[n,1]...n[n,j_n]...n[n,p_n])

en prenant un élément dans chaque liste. Les multiplets sont au nombre de $p_1 \times p_2 \times ... \times p_n$. Il nous faut les construire afin d'éclater la phrase initiale en autant de phrases différentes, chacune représentant une polystructure différente. Cette opération sera conduite par la procédure éclatement dont le paramètre d'entrée est la phrase initiale et le paramètre de sortie, un texte considéré comme une liste de phrases.

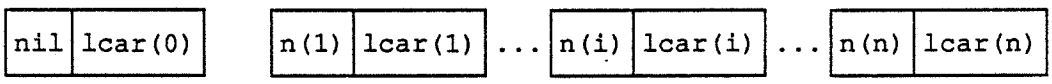
soit la structure de données :

```
ptexte = ^ttexte ;
ttexte =     record
             phrase : pphrase ;
             suivant : ptexte ;
      end ;
```

procedure éclater (phrase : pphrase ; var texte : ptexte) ;

```
var
  laux : ptexte ; naux : pnoeud ; paux : pphrase ;

debut
   initialiser texte : texte^.phrase <-- phrase
                       texte^.suivant <-- nil

   pour i = 1 à n faire
        laux <-- nil ;
        pour j = 1 à p_i faire
             pour chaque phrase ph de texte faire

                  créer une racine naux construite à partir de n[i,j]
                  avec naux^.alter = nil et naux^.frère = nil

                  créer une phrase paux égale à ph
                   mais dont le champ noeud du ième élément
                   vaut naux

                  insérer paux en queue de laux
             fin
         fin
         texte <-- laux
   fin
```

L'éclatement de la phrase en un texte nous permet de considérer maintenant chaque phrase du texte. Ces phrases sont de la forme :



où chaque noeud n[i] a été coupé de ses alternants. Donc chaque n[i] est racine d'une polystructure qu'il faut maintenant parcourir. Pour chacun, on initialise une phrase ph[i] avec l'élément :

noeud = nil et lcar = vide ;

et on appelle la procédure parcours (n[i]) sur ph[i]) ; à l'issue de ce traitement, on range le résultat dans une phrase P obtenue par la concaténation des phrases ph[i] pour i = 1 à n. Cette phrase P est à son tour soumise à la procédure éclatement.

Remarque

La procédure parcours ne progresse que sur les noeuds qui n'ont pas d'alternants. Donc la progression dans le parcours de la polystructure est assurée par le fait qu'une racine de polystructure n'a pas d'alternants. C'est pourquoi, la procédure éclatement sépare le noeud de ses alternants pour que ce noeud puisse devenir racine de polystructure.

La procédure impression a pour paramètre la phrase à traiter, phrase dont tous les champs "noeud" sont des racines de polystructure. Sa fonction est d'enchaîner les procédures précédentes afin d'obtenir l'impression de tous les arbres inclus.

procedure impression (phrase : pphrase) ;

```
initialisations :
          texte^.phrase <-- phrase
          texte^.suivant <-- nil

pour chaque phrase du texte faire

   si phrase^.suivant = nil alors
      imprimer (phrase^.lcar)

   sinon
      P <-- vide
      pour chaque élément de phrase (à partir du 2ème) faire
         ph <-- nil vide
         dernier <-- ph
         parcours (élément^.noeud) ;
         dernier <-- nil
         P <-- concaténation (P,ph)
      fin
      eclater (P, texte) ;

      pour chaque phrase de texte faire impression (phrase) ;
   fin
fin
```

Cet ensemble de procédures est amorcé par l'appel de la procédure impression avec une phrase qui contient deux éléments. Tous les champs du premier sont à nil sauf le champ suivant qui pointe sur le deuxième élément. Tous les champs du deuxième élément sont à nil excepté le champ noeud qui contient l'adresse du noeud racine de la polystructure à imprimer.

Cet algorithme permet de contruire la polystructure pendant l'opération de reconnaissance, puis d'imprimer dans une deuxième phase, tous les arbres inclus dans cette polystructure.

3.4.3. Deuxième méthode.

3.4.3.1. Fonctionnement de l'analyseur

Cette méthode exploite les résultats de l'accepteur. Elle est proche de celle exposée dans [AHO et ULLMAN, 1972, tome 1]. On utilisera donc les listes d'états V[0]...V[n+1]. La chaîne d'entrée x est élément de L(G) si l'état <0,2,0,$> est dans V[n+1]. Partant de cet état équivalent à [PHI --> S $ *, 0, $] on cherche à reconstruire la polystructure engendrée par S. Le résultat de l'analyseur sera une séquence de numéros de règles de R, avec alternants, donnant une analyse droite de x.

D'une façon générale, l'algorithme fonctionne ainsi : soit un état final <r,l(r),j,a> de V[i], noté :

[A --> B[1]...B[k]...B[l(r)] *, j, a].

La structure engendrée par A résulte de l'application de la règle r, puis, pour chacun des éléments B[k], k décroissant de l(r) à 1, de l'application de règles définies ainsi [Figure 2.9] :

B[l(r)] est soit un terminal, auquel cas la valeur de ce terminal est donnée par la règle r, soit un non terminal ; alors B[l(r)] engendre une chaîne terminale se terminant en x[i]. Il existe donc dans V[i] au moins un état final <r',l(r'),j[l(r)],a'> tel que

1 B[l(r)] soit partie gauche de r'

2 a' = a puisque le symbole attendu après B[l(r)], est identique à celui attendu après A.

3 j[l(r)] < i puisque la chaîne engendrée par B[l(r)], x[j(l(r))+1]...x[i] ne peut être vide.

Une fois trouvée dans V[i] la règle à partir de laquelle B[l(r)] engendre la sous-chaîne x[j(l(r))+1]...x[i], il faut trouver la règle à partir de laquelle B[l(r)-1] engendre une sous-chaîne se terminant en x[l(r)]. On la trouvera parmi les états finals de V[l(r)], en considérant ceux dont la règle a pour partie gauche B[l(r)-1], le symbole attendu est élément de Prem(B[l(r)]), le point de départ dans la chaîne d'entrée est compris entre j et j(l(r))-1. On opère de façon identique pour chaque B[k] jusqu'à ce que B[1] ait été traité.

A chaque B[k] correspond un indice j(k), la sous-chaîne engendrée par B[k] débutant à l'indice j(k)+1 et se terminant à l'indice j(k+1). La concaténation de ces sous chaînes pour k variant de 1 à l(r), doit fournir la sous-chaîne x[j+1]...x[i]. Pour cela, les indices j(k) doivent réaliser une partition de x[j+1]...x[i], et donc respecter les contraintes suivantes :

1 j(k) < j(k+1)
2 j(1) = j et j(l(r)) < i
3 (j(2) - j) +...+ (j(k+1) - j(k)) +...+ (i - j(l(r)) = (i - j)

Pour chaque B[k], il peut exister plusieurs états répondant aux conditions demandées. B[k] est alors racine de plusieurs structures, structures que l'on séparera par le caractère +. De ce fait, à l'état <r, l(r), j, a> de V[i] peuvent correspondre différentes partitions de la chaîne x[j+1]...x[i]. [Figure 2.9].

Remarque

Parmi tous les états engendrés par l'accepteur, seuls les états finals sont utiles à la construction de la structure. On peut donc envisager, une fois la reconnaissance de la chaîne d'entrée réalisée, de nettoyer ces listes pour ne garder que les états finals.

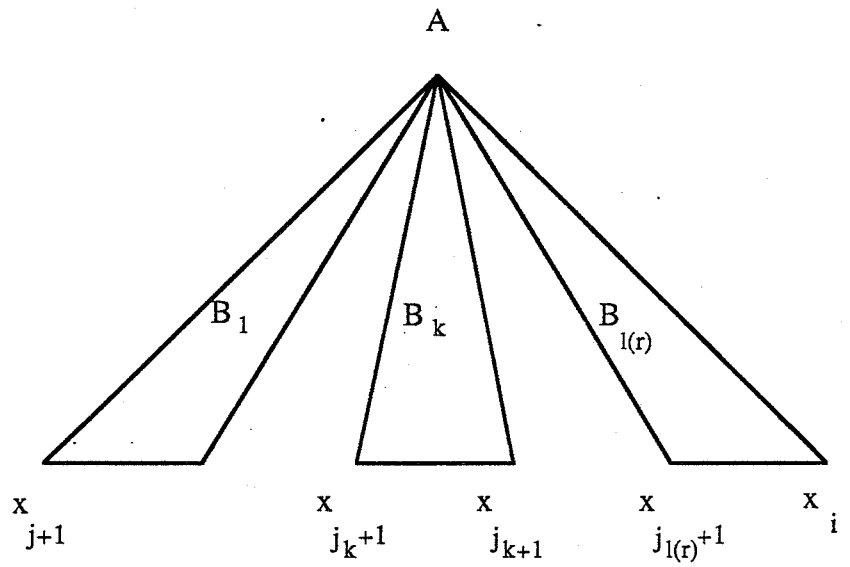


FIGURE 2.9

3.4.3.2. Algorithme

procédure IMPRESSION

```
début
      RES = NIL
      parcours (<0,2,0,$>, n+1)
fin
```

procédure PARCOURS (<r,l(r),j,a>, i)

```
début
      RES = r||RES
      k = l(r)
      l = i

      répéter
                  si B[k] est terminal alors début
                  k =k-1 ;
                  l = l-1 ;
            finsi
            si B[k] est non terminal alors chercher dans V[l] tous
            les états <r', l(r'), j', b> tels que
            - B[k] soit partie gauche de r'
            - j' pris entre j et l-1,
            j' respectant la partition de la sous-chaîne x[j+1]...x[i]
            - b élément de Prem(B[k+1]) si k<l(r) ou b=a si k=l(r)

            pour chacun de ces états faire
                  PARCOURS (<r',l(r'),j',b>, l)
                  RES = +||RES
            fin

            k = k-1
            l = j'
      jusqu'à k=0 ;
```

3.4.3.3. Exemple

A partir de l'exemple donné pour l'accepteur d'Earley et donc des listes d'états, l'algorithme précédent fournit le résultat suivant :

[6 7 7 2 1 5 4 3 + 6 7 3 7 2 1 5 4] 2 1 0 par appels récursifs de la procédure parcours [Figure 2.10]. Ce résultat correspond à une polystructure [Figure 2.11].

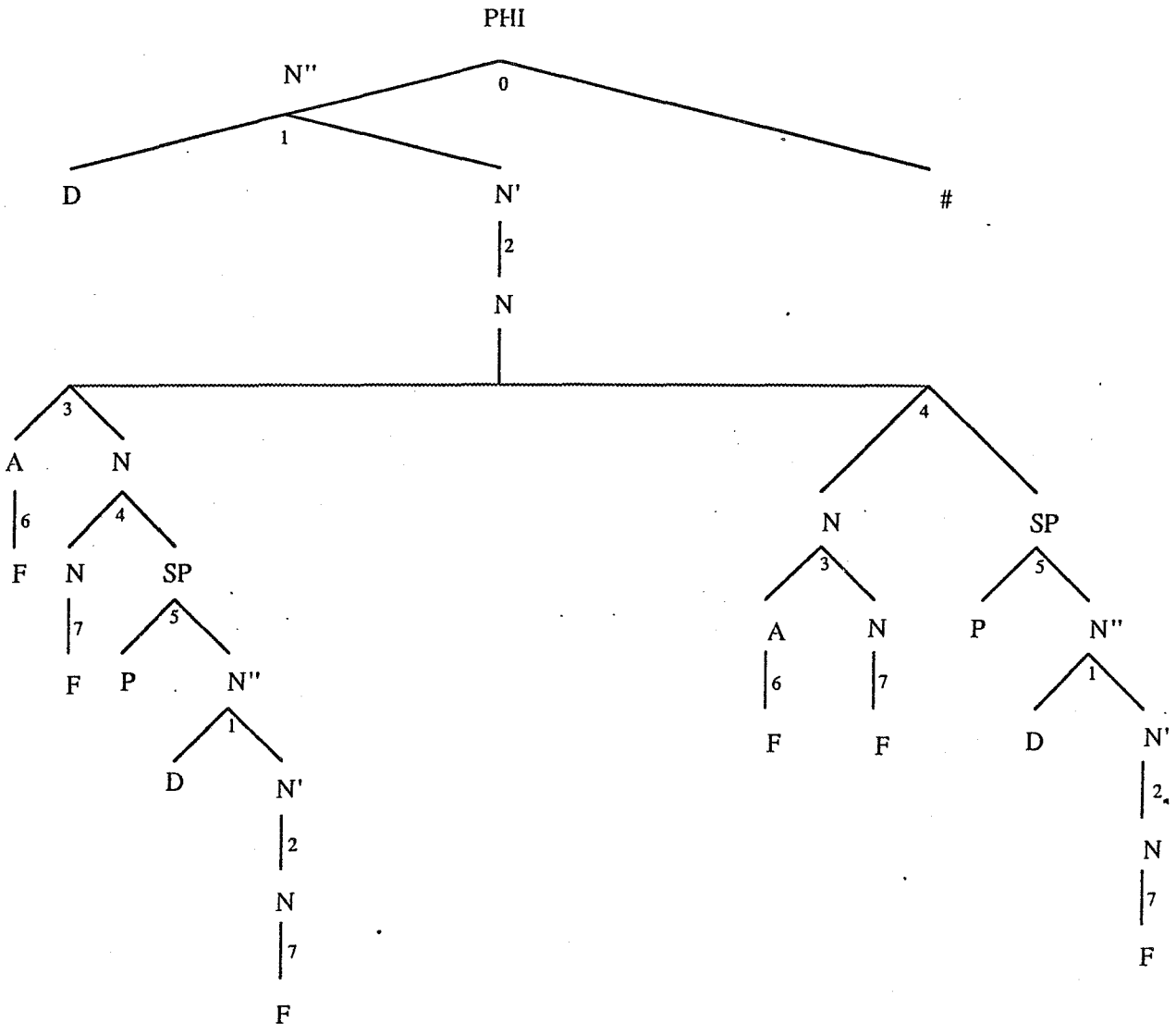

PHI
N''
0
1
D
N'
#
2
N
3
4
A
N
N
SP
6
4
3
5
F
N
SP
A
N
P
N''
7
5
6
7
1
F
P
N''
F
F
D
N'
1
2
D
N'
N
2
N
7
7
F
F

FIGURE 2.10

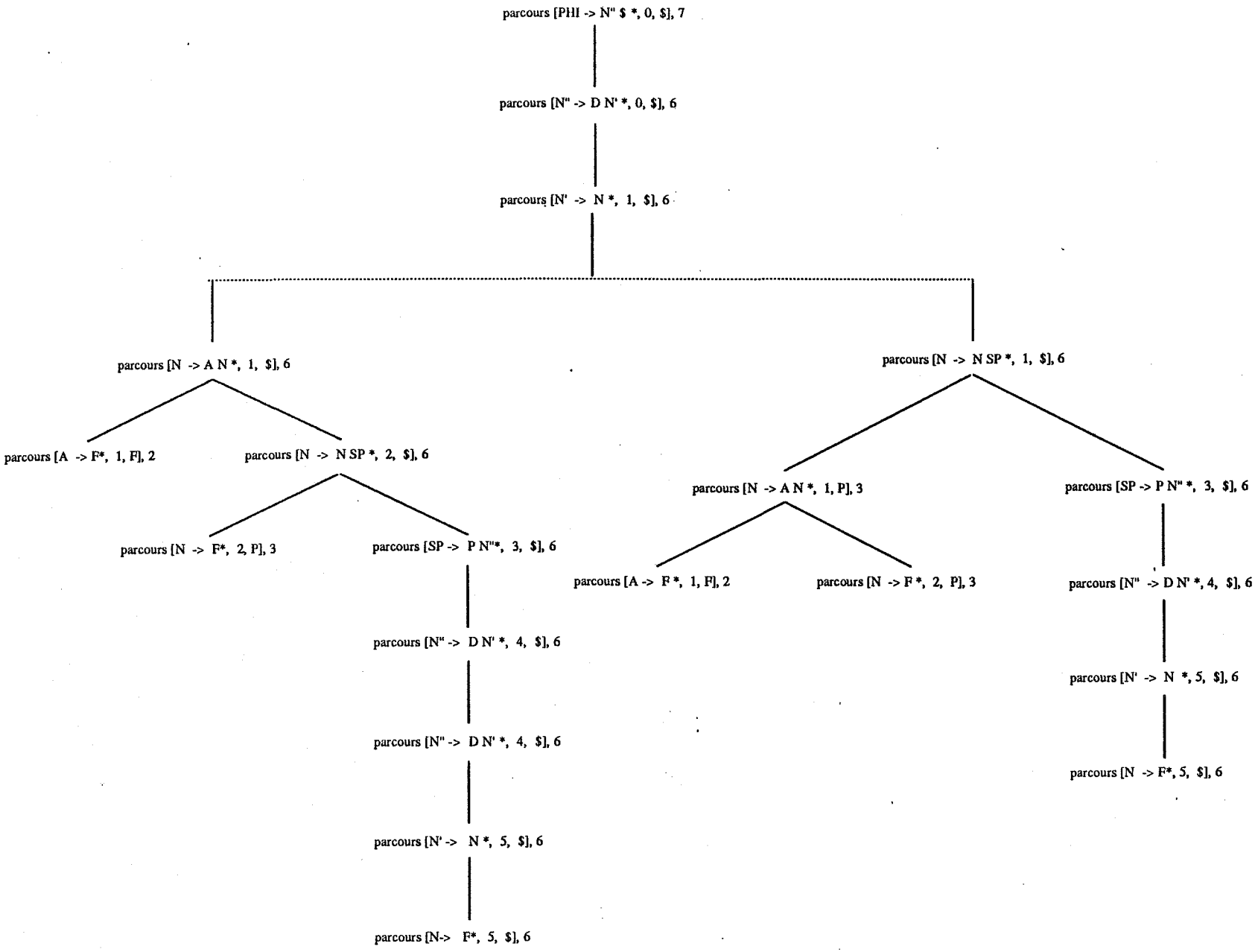

parcours [PHI -> N" $ *, 0, $], 7
parcours [N" -> D N' *, 0, $], 6
parcours [N' -> N *, 1, $], 6
parcours [N -> A N *, 1, $], 6
parcours [N -> N SP *, 1, $], 6
parcours [A -> F*, 1, F], 2
parcours [N -> N SP *, 2, $], 6
parcours [N -> A N *, 1, P], 3
parcours [SP -> P N" *, 3, $], 6
parcours [N -> F*, 2, P], 3
parcours [SP -> P N"*, 3, $], 6
parcours [A -> F *, 1, F], 2
parcours [N -> F *, 2, P], 3
parcours [N" -> D N' *, 4, $], 6
parcours [N" -> D N' *, 4, $], 6
parcours [N' -> N *, 5, $], 6
parcours [N" -> D N' *, 4, $], 6
parcours [N' -> N *, 5, $], 6
parcours [N -> F*, 5, $], 6
parcours [N-> F*, 5, $], 6

FIGURE 2.11

4. COMPLEXITE DES ALGORITHMES DE RECONNAISSANCE

4.1. AUTOMATE A PILE

Comme tout langage hors contexte peut-être reconnu par un automate à pile non déterministe, la programmation directe de cet automate est un accepteur pour le langage. Ses performances, très faibles, constituent, pour cette raison, une référence pour les autres algorithmes. Or, on sait que pour reconnaître une chaîne de longueur n, cet accepteur nécessite une place mémoire proportionnelle à n et un nombre d'opérations élémentaires de la forme k^n, où k est une constante (dépendant de la grammaire G). [voir AHO, ULLMAN 1972, pp 298-299].

4.2. ALGORITHMES DE COCKE ET YOUNGER

Les algorithmes de Cocke et Younger ont des performances meilleures ; ils nécessitent un espace en n^2, taille de la table d'analyse. Le nombre d'opérations élémentaires est en n^2 pour chaque ligne, donc globalement en n^3, pour l'accepteur. Ce nombre est en n^2 pour l'analyseur seul ; il est donc de n^3 pour l'ensemble accepteur et analyseur.

Remarque : Variante de Valiant.
VALIANT [1975] a proposé une amélioration de l'algorithme de Younger. Elle consiste à calculer la fermeture transitive d'une matrice booléenne par l'algorithme de Strassen. Le nombre d'opérations élémentaires (multiplications) est ainsi réduit à n puissance 2,81.

4.3. ALGORITHME D'EARLEY

L'algorithme d'Earley a des caractéristiques particulières dans la mesure où ses performances varient avec les caractéristiques de la grammaire (alors que celles de l'algorithme de Cocke sont constantes). L'espace requis est en $O(n^2)$. Dans le cas le plus général, la complexité en temps est en $O(n^3)$. Cependant sur la classe des grammaires à degré d'ambiguïté borné - classe qui contient les grammaires non ambiguës - il est en $O(n^2)$. Enfin, sur la classe des grammaires qui engendrent des ensembles d'états de taille bornée - classe à laquelle s'apparentent les grammaires LR(k) - il est en O(n). Les indications de calcul de complexité données ci-après, sont exposées dans [Earley 68] et [Earley 70].

4.3.1. Notations

Soient |R| le nombre de règles de R, |T|, le nombre de symboles terminaux, L la longueur maximale des règles de R, A le nombre maximal de règles ayant même partie gauche.

Soient encore, les deux listes chaînées, sous-listes de V[i], définies ainsi :

$0 \leq j \leq i$ $\quad v[i,j] = \{<r,k,j,a> / <r,k,j,a> \varepsilon V[i]\}$

$X \varepsilon N$ $\quad w[i,X]=\{<r,k,j,a> / <r,k,j,a> \varepsilon V[i]$ et $B[k+1]=X\}$

4.3.2. Espace requis par les listes d'états

Le vecteur V, a pour composantes V[0],...,V[i],...V[n+1] ; V[i] est un ensemble d'états de la forme <r,k,j,a> ; V[i] ne peut contenir deux états identiques. Donc le nombre maximum d'états de V[i] correspond à toutes les valeurs possibles du quadruplet <r,k,j,a> quand r parcourt R, k varie de 0 à L, j de 0 à i, et a parcourt T, soit :

$$|V[i]| \leq |R|*(L+1)*(i+1)*|T|$$

comme $|V| = |V[0]|+...+|V[i]|+...|V[n+1]|$

$$|V| \leq |R|*(L+1)*|T|*(n+1)*(n+2) / 2 \approx K\ n^2$$

où K est une constante propre à la grammaire G. L'espace requis est donc en $O(n^2)$.

4.3.3. Temps requis par l'accepteur

4.3.3.1. Cas général

Une opération élémentaire sera ici la comparaison de deux états. Soit, alors, un état <r,k,j,a> de V[i] ; il est soumis à l'une des opérations suivantes :

dérivation si B[k+1] est non terminal.

Cette opération engendre au plus A*|T| états <r',0,i,b> puisqu'il y a au plus A règles qui ont B[k+1] en partie gauche et puisque a parcourt T. Chacun de ces états est comparé à la sous-liste v[i,i] de V[i], qui contient les états <r,k,i,a> où r parcourt R, k varie de 0 à L, i est fixé, et a parcourt T. v[i,i] contient au plus |R|*(L+1)*|T| états. La prédiction s'effectue donc en un nombre d'opérations élémentaires maximum de :

$$A*|T^2|*|R|*(L+1) \quad (1)$$

balayage si B[k+1] est terminal.

Cette opération n'engendre qu'un état qui doit être comparé aux états de v[i+1,j], soit à au plus |R|*(L+1)*|T| états. D'où un nombre d'opérations élémentaires de

$$|R|*(L+1)*|T| \quad (2)$$

complétion si k=l(r) et si a=x[i+1].

Soit X le symbole gauche de r, on prend tous les états de la sous-liste w[j,X] qui en contient au plus |R|*(L+1)*(j+1)*|T| d'après les calculs de l'espace requis pour V[j]. Chacun de ces états est comparé aux états de v[i,j] qui en contient |R|*(L+1)*|T| au plus. D'où un nombre d'opérations élémentaires de :

$$|R|^2 * (L+1)^2 * (j+1) * |T|^2$$

or $j \leq i$ donc cette quantité est majorée par :

$$|R|^2 * (L+1)^2 * (i+1) * |T|^2 \quad (3)$$

Ces calculs ont été faits pour un état. Or V[i] contient au plus |R|*(L+1)*(i+1)*|T| états, donc le nombre maximal d'opérations élémentaires requis pour l'analyse d'une chaîne de longueur n est la somme pour i variant de 0 à (n+1) de :

$$|R| * (L+1) * (i+1) * |T| * [(1) + (2) + (3)]$$

Le résultat est en $O(n^3)$.

4.3.3.2. Cas d'ambiguïté bornée

4.3.3.2.1. Définitions

Le degré d'ambiguïté directe d'un non terminal A par rapport à une chaîne x[1]...x[n] est le nombre de façons différentes dont A peut engendrer la chaîne. Plus précisément, pour chacune des règles de R ayant A en partie gauche, A --> B[1]...B[k]...B[l(r)], A engendre x[1]..x[n], signifie qu'il existe au moins une suite j[1], j[2],..., j[k], j[k+1],...,j[l(r)], j[l(r)+1], où j[1]=0 et j[l(r)+1]=n, réalisant une partition de la chaîne. Le degré d'ambiguïté directe de A par rapport à la chaîne x[1]...x[n] est alors la somme, pour chacune des règles de R ayant A en partie gauche, du nombre de partitions.

Une grammaire est de degré d'ambiguïté borné si tous les non terminaux ont par rapport à toutes les chaînes de T^*, un degré d'ambiguïté directe borné.

4.3.3.2.2. Complexité en temps

Soit b la borne du degré d'ambiguïté de la grammaire. L'opération de complétion sur un état de V[i], engendre au plus b états dans V[i], nombre qui n'est plus dépendant de i. De ce fait, l'algorithme se déroule alors en un temps de l'ordre de $O(n^2)$. Ce résultat est a fortiori valable si la grammaire est non ambiguë.

4.3.3.3. Cas où le nombre d'états est borné

4.3.3.3.1. Définition

Une grammaire est à nombre d'états borné, pour une prédiction portant sur m symboles, s'il existe une limite e, borne du nombre d'états engendrés dans V[i] par toute chaîne x[1]...x[i+m].

4.3.3.3.2. Complexité en temps

Soit G une grammaire à nombre d'états borné par e ; alors V[i] contient au plus e états, e étant indépendant de i. L'algorithme a dans ce cas une complexité en espace et en temps en O(n).

De ce chapitre, nous pouvons retenir que, parmi les algorithmes existants, l'algorithme d'Earley est le plus général, car il ne nécessite aucune normalisation de la grammaire. De plus, sa complexité en temps varie en fonction du type de langage traité, et dans chacun des cas, il est aussi performant que les algorithmes particuliers. C'est donc cet algorithme qui nous servira de point de départ dans l'analyse syntaxique des syntagmes nominaux.

Chapitre 3

STRATEGIE D'ANALYSE

1. CHOIX DU MODELE HORS CONTEXTE

Pour mener à bien l'analyse syntaxique d'un langage L, il faut au préalable lui associer une grammaire G, telle que toute phrase du langage puisse être engendrée par la grammaire et que toute phrase engendrée par la grammaire soit élément du langage. Chomsky a proposé une classification des langages en fonction de la grammaire qui leur est associée.

1.1. LA CLASSIFICATION DE CHOMSKY

Soit L un langage engendré par une grammaire G= (T, N, S, R) [cf. notations du chapitre 2].

Une grammaire est de type 0, si les règles de R sont de la forme A --> u, où A et u sont éléments de V^*.

Une grammaire est de type 1, ou contextuelle, si les règles de R vérifient la condition : X A Y --> X u Y où u, X, Y sont éléments de V^* et A élément de N ;

Une grammaire est de type 2, ou hors contexte, si les règles de R vérifient la condition : A --> u où A est élément de N et u de V^*.

Une grammaire est de type 3, ou régulière, si les règles sont toutes de la forme :

A --> a ou A --> a B

ou toutes de la forme :

A --> a ou A --> B a

avec A et B éléments de N et a de T.

Un langage sera dit régulier, hors contexte ou contextuel selon qu'il existe une grammaire de type 3, de type 2 ou de type 1 qui l'engendre.

1.2. OU SE SITUE LA LANGUE NATURELLE DANS LA CLASSIFICATION DE CHOMSKY ?

La réponse à cette question nécessite l'existence préalable d'une grammaire qui engendre toutes les phrases de la langue naturelle et aucune autre. Cette grammaire n'existe pas à l'heure actuelle, et quand bien même existerait-elle, serait-elle alors utilisable par l'informaticien ? Puisqu'il n'existe pas de grammaire de la langue, le problème se transforme et se pose dans les termes suivants : trouver la meilleure grammaire pour une application donnée. De ce fait, le choix d'une grammaire s'effectue en fonction de critères d'utilisation . Dans notre cas, la reconnaissance automatique de textes écrits en langue naturelle en vue d'une indexation automatique nous suggère d'opter pour une grammaire ayant les caractéristiques suivantes :

grammaire de reconnaissance

elle doit donc reconnaître les phrases de la langue, mais n'est pas contrainte à n'engendrer que celles-ci.

grammaire conçue pour un traitement automatique

Parmi les grammaires, celles que les informaticiens savent le mieux traiter sont les grammaires de type régulier ou de classe 3. Les analyseurs associés sont des automates d'états finis. Les structures syntaxiques obtenues ont la forme d'un peigne puisque l'application d'une règle entraîne la reconnaissance d'un terminal. Ce type de structures bien que souvent attesté dans la langue naturelle ne rend compte que très partiellement de son fonctionnement.

Exemple : La structure syntaxique du syntagme :
"le niveau de vie moyen"
représentée sur la figure 3.1. n'est pas régulière.

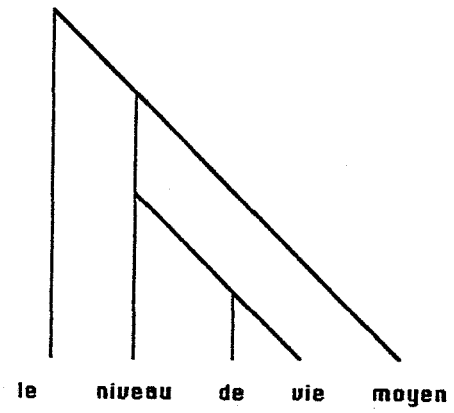


FIGURE 3.1.

La classe suivante dans la classification de Chomsky est celle des grammaires hors contexte. Ces grammaires engendrent des langages plus riches que les grammaires de classe 3. De plus, on sait traiter automatiquement de telles grammaires. Les analyseurs associés sont de type automates à pile. C'est ce que l'on a vu dans le chapitre précédent. L'on sait encore que les structures syntaxiques obtenues sont des arborescences ; ceci présente deux avantages : ces structures sont bien connues des informaticiens et de ce fait, leur construction et leur parcours relèvent de techniques éprouvées. Le deuxième avantage réside dans leur facilité d'interprétation par l'utilisateur.

Les grammaires hors contexte sont, en résumé, des grammaires qui se prêtent au traitement automatique. Cette conclusion n'a d'intérêt que si les linguistes peuvent concevoir de telles grammaires pour décrire le fonctionnement des langues naturelles. Cette question est encore ouverte dans la communauté des linguistes [M. KING, 1983], mais comme nous l'avons déjà dit, il ne s'agit pas d'écrire une grammaire qui rende compte de tous les phénomènes de la langue mais d'une grammaire adaptée aux objectifs visés. C'est la raison pour laquelle, en accord avec A. Berrendonner, nous opterons pour une grammaire hors contexte. Ce choix justifie a posteriori les régularisations opérées lors du traitement préalable. (cf. chapitre 1).

Ce choix a encore pour avantage de placer le linguiste dans les meilleures conditions pour concevoir une grammaire. En effet, les contraintes qui pèsent sur les règles d'une grammaire hors contexte sont moindres que dans le type régulier. Elles permettent de rendre compte de phénomènes linguistiques réels. De plus, le caractère non contextuel donne à chaque règle une portée globale et évite ainsi de multiplier les cas à envisager et par la même, le nombre de règles de la grammaire. En effet, une grammaire de grande taille devient très vite difficile à maîtriser. C'est une des raisons, d'ailleurs, pour lesquelles nous n'envisagerons pas d'adopter une grammaire contextuelle pour la syntaxe de la langue naturelle. Les autres raisons qui nous poussent à abandonner le modèle contextuel sont d'ordre informatique, car c'est un domaine où les algorithmes sont peu performants, car coûteux en temps et en espace. De plus, les résultats obtenus sont difficiles à interpréter et à manipuler car les structures syntaxiques ne sont plus des arborescences.

En conclusion, puisque nous ne pouvons trouver de réponse à la question posée, nous adopterons **le modèle hors-contexte** car il répond à la fois aux exigences des linguistes et des informaticiens, et car il nous semble le mieux adapté à l'objectif poursuivi : l'analyse syntaxique de textes écrits en langue naturelle.

2. NECESSITE D'UNE STRATEGIE D'ANALYSE

Un analyseur de type hors contexte comme ceux que nous avons étudiés au chapitre précédent, fonctionne de façon purement combinatoire. Il fournit donc en résultat des solutions souvent nombreuses parmi lesquelles on distinguera les solutions grammaticales des solutions acceptables.

2.1. SOLUTIONS GRAMMATICALES ET SOLUTIONS ACCEPTABLES

Soit G une grammaire de la langue naturelle (ou supposée telle à l'issue du paragraphe précédent). Une telle grammaire est nécessairement ambiguë, dans le sens où l'on trouvera toujours des phrases de la langue naturelle sur lesquelles la grammaire engendrera plus d'une structure syntaxique. De ce fait, pour une chaîne d'entrée, l'analyseur associé à la grammaire donne l'un des deux résultats suivants :

1. soit la chaîne n'est pas engendrée par la grammaire et elle est agrammaticale.

2. soit la chaîne est engendrée par la grammaire et elle est grammaticale. Dans ce cas, la grammaire peut générer soit une structure syntaxique, solution simple, soit plus d'une structure syntaxique, solution multiple.

Si le résultat obtenu est une solution simple, l'analyse de la chaîne est résolue. Par contre, il nous faut étudier plus précisément le cas d'une solution multiple.

L'analyseur peut construire plusieurs structures syntaxiques parce que la grammaire est ambiguë ; la grammaire hérite cette caractéristique du langage qu'elle reconnaît. De ce fait, les ambiguïtés peuvent provenir de deux origines différentes. Soit elles sont inhérentes à la langue, et dans ce cas il est heureux que la grammaire en rende compte. Soit elles sont produites par la grammaire sans être attestées par la langue. Ce sont alors des solutions parasites.

Exemple :Soit la grammaire :

```
N' --> D N
N' --> N
N --> F
N --> N SP
N --> A N
N --> N A
SP --> P N'
```

et la chaîne :

```
les principales caractéristiques de ces résistances
 D      A              F         P   D       F
```

Les deux structures obtenues apparaissent sur la figure 3.2. Elles traduisent une ambiguïté inhérente à la langue naturelle : les deux structures sont également admises par la langue. Seules des données sur le contexte nous permettraient de choisir l'une où l'autre. Cependant au niveau purement syntaxique, il y a ambiguïté et les deux structures sont solutions, elles doivent donc être conservées.

Exemple : Soit avec la même grammaire, la chaîne d'entrée :

```
les maladies à virus de la pomme de terre
 D     F     P   F   P  D    F   P    F
```

La figure 3.3 nous montre que la grammaire engendre sur cette chaîne cinq structures différentes alors que l'usage courant de la langue n'en admet qu'une, la solution 2. Les autres sont des solutions parasites engendrées par la grammaire.

Face à une telle situation, deux attitudes sont possibles :

1 conserver toutes les solutions en espérant que les analyses de niveau supérieur (sémantique, pragmatique) effectueront un tri. Cette stratégie est peu réaliste, car les étapes suivantes devront gérer en plus de leurs propres ambiguïtés celles des niveaux précédents : d'où un risque important de créer des solutions parasites en grand nombre et de perdre la maîtrise de l'analyse.

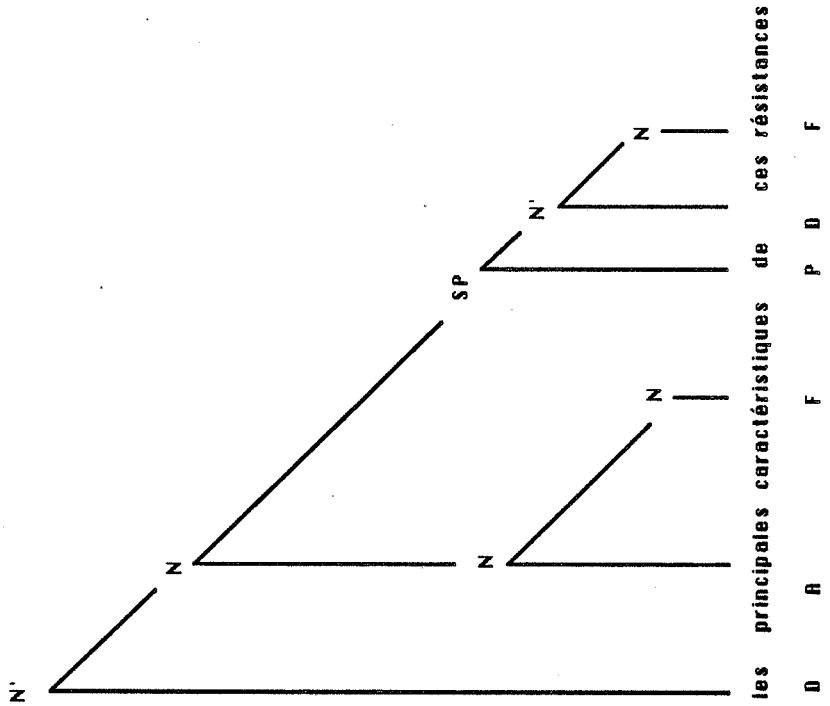
2 élaborer pour l'analyse syntaxique, une stratégie qui, intervenant soit avant l'analyse soit au cours de l'analyse, permette de de ne pas construire les solutions parasites. C'est, par cohérence avec la méthodologie du projet dans son ensemble (cf. chapitre 1), que nous préférerons cette voie.

2.2. STRATEGIE D'ANALYSE

Un analyseur syntaxique guidé par la seule grammaire donne pour une chaîne toutes les structures grammaticales. Parmi ces structures, on ne peut distinguer les structures parasites des structures attestées par la langue. Un tel analyseur ne suffit donc pas pour atteindre l'objectif visé : obtenir les seules structures attestées par la langue.

L'analyseur ne suffit plus, il faut lui associer une **stratégie d'analyse,** stratégie qui le guidera au cours de la construction des structures vers celles qui sont attendues. Une stratégie d'analyse peut être considérée comme un superviseur qui guide le travail de l'analyseur. Elle est définie d'après des hypothèses qui traduisent le fonctionnement de la langue naturelle. La mise en oeuvre d'une stratégie d'analyse est coûteuse et elle ne réduit pas pour autant la complexité de l'algorithme d'analyse. En effet, la stratégie ne modifie pas le déroulement de l'analyse mais elle intervient tout au long de l'analyse pour la contraindre. Elle ralentit donc le processus d'analyse et de ce fait ne réduit pas sa complexité. Donc une stratégie n'est intéressante que si elle conduit à des résultats plus satisfaisants que le seul analyseur. D'où la nécessité de choisir la "bonne" stratégie.

Les critères de choix d'une bonne stratégie varient suivant l'objectif visé. Ces critères peuvent privilégier la rapidité de l'analyse ou sa justesse. Pour illustrer notre propos nous avons choisi de présenter deux stratégies différentes : la stratégie déterministe de Marcus et la stratégie de Kimball. Puis nous terminerons par celle que nous proposons.


N'
N
N
SP
N'
N
N
les principales caractéristiques de ces résistances
D A F P D F
SOLUTION 2

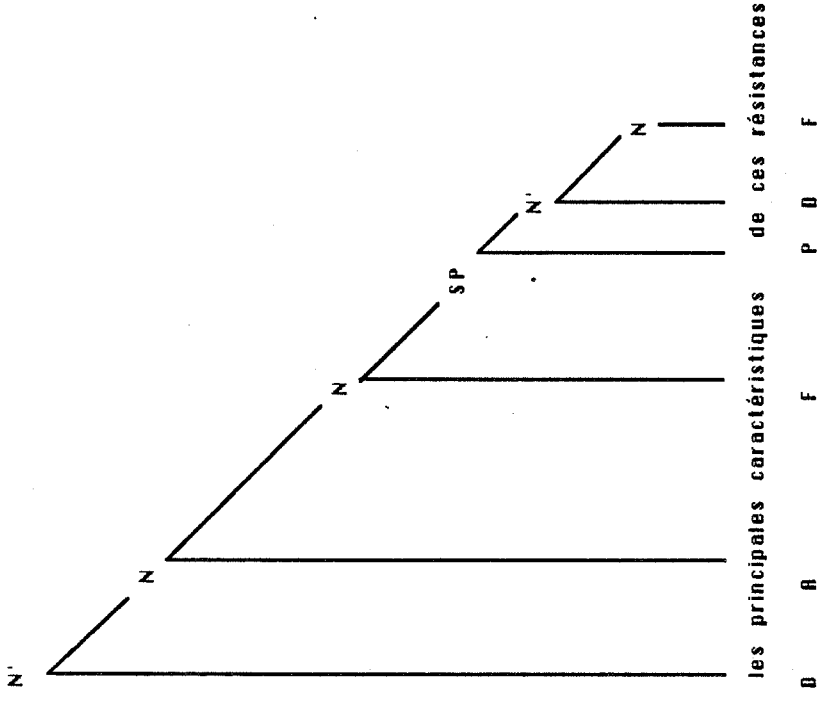

N'
N
N
SP
N'
N
N
les principales caractéristiques de ces résistances
D A F P D F
SOLUTION 1

FIGURE 3.2.

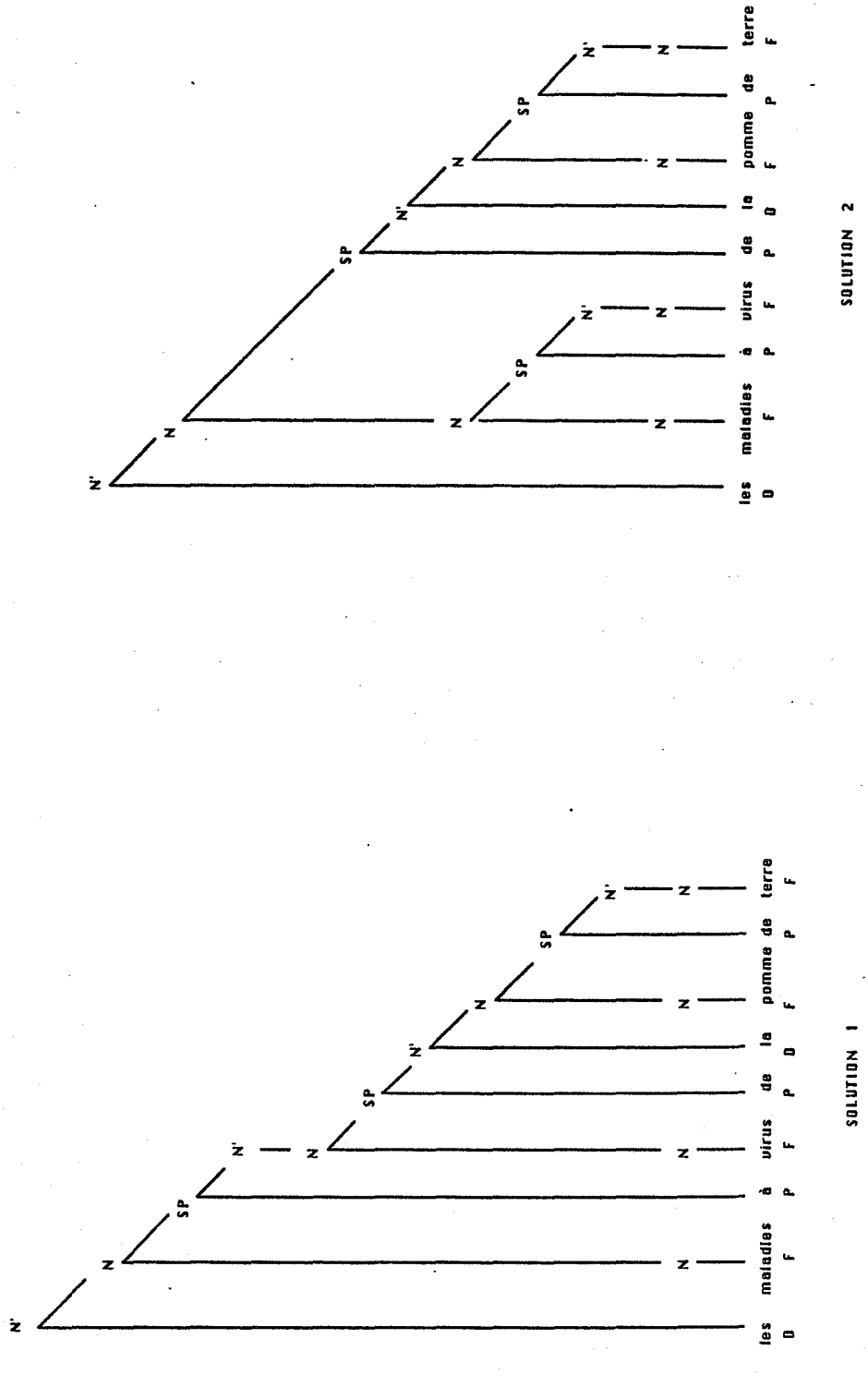
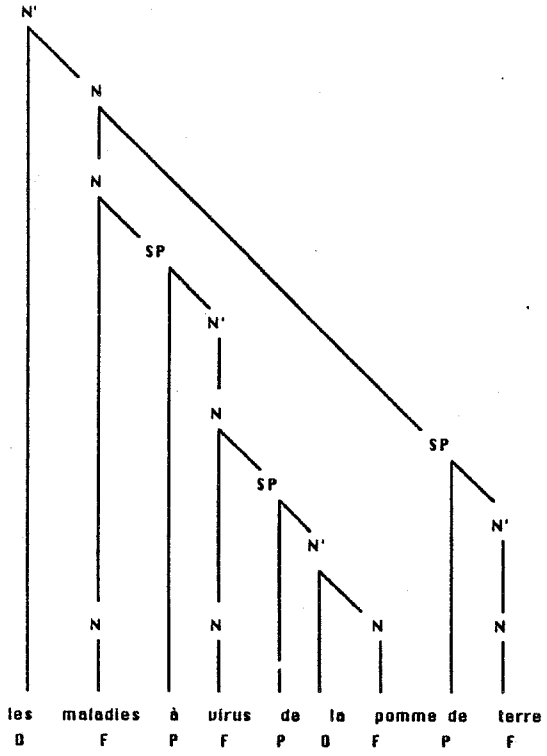
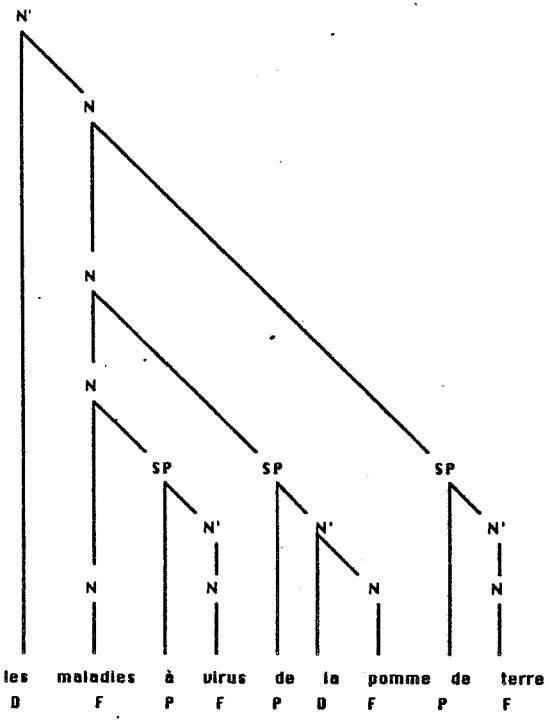

N'
N
SP
N'
N
SP
N'
N
les maladies à virus de la pomme de terre
D F P F P D F P F
SOLUTION 2
N'
N
SP
N'
N
SP
N'
N
les maladies à virus de la pomme de terre
D F P F P D F P F
SOLUTION 1

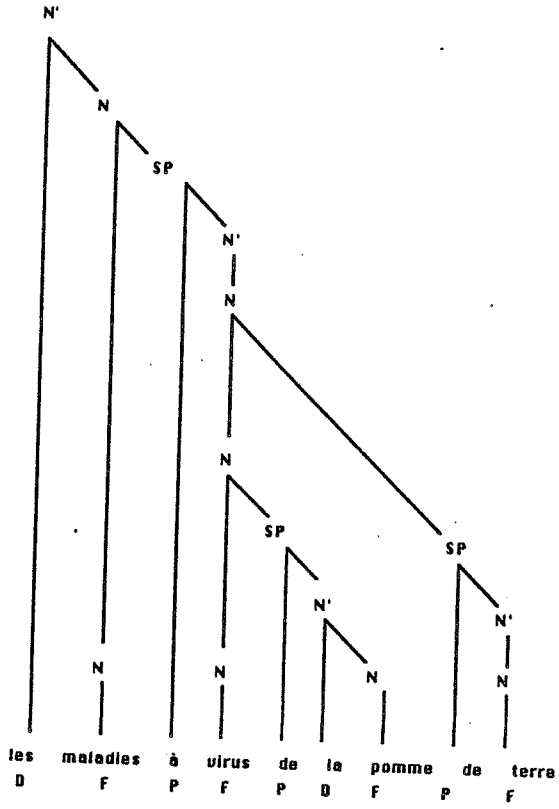
FIGURE 3.3



SOLUTION 3



SOLUTION 4



SOLUTION 5

FIGURE 3.3

3. LA STRATEGIE DETERMINISTE DE MARCUS.

La stratégie de Marcus [Marcus, 1979] se fonde sur des hypothèses psycho-linguistiques. Elle part du principe que l'esprit humain construit la structure des phrases qu'il perçoit au fur et à mesure de leur perception. Il ne fonctionne donc ni par retour arrière, ni en construisant plusieurs structures simultanément ; retour arrière ou pseudo-parallélisme étant les deux outils qui permettent de simuler des mécanismes non déterministes sur des ordinateurs qui eux fonctionnent toujours de façon déterministe. D'où l'idée d'une stratégie déterministe dans le sens où l'analyseur ne remet jamais en cause une structure déjà construite. Le déterminisme de cette stratégie s'appuie sur un outil puissant de prédiction. En effet, l'analyseur gère à la fois une pile de noeuds dont le seul sommet est actif. A chaque noeud est attaché un paquet de règles activé lorsque le noeud est au sommet de la pile. La règle qui sera alors appliquée doit appartenir au paquet de règles et sera choisie parmi les autres en fonction des noeuds en attente dans une mémoire tampon à trois cases. Cette mémoire tampon est l'outil de prédiction. Elle contient soit des terminaux de la chaîne d'entrée, soit des sous-structures qui, ayant été en sommet de pile n'ont pu permettre à l'analyseur de progresser. Celles-ci sont alors déposées dans la mémoire tampon en attendant d'être appelées par une règle d'un noeud situé plus bas dans la pile.

Cette stratégie d'analyse traduit bien le fait que l'esprit humain rejette a priori les structures indéfiniment imbriquées puisque la taille de la mémoire tampon est limitée. Elle contraint l'analyseur dans le choix des règles à appliquer : une règle ne sera appliquée que si elle appartient au paquet du noeud actif dans la pile et si elle est compatible avec les noeuds en attente. Pour satisfaire l'hypothèse de déterminisme, il faut que dans toute situation, les contraintes soient telles qu'elles ne sélectionnent qu'une seule règle, c'est-à-dire qu'il n'y ait qu'une issue possible. Or c'est à notre avis, ici que l'analyseur montre sa faiblesse, car il nie une évidence à savoir que la langue naturelle est ambiguë et que, comme nous l'avons vu dans le paragraphe 2.1., il peut exister plus d'une structure syntaxique correcte pour une phrase donnée. Dans un tel cas, l'analyseur ne fournit qu'une structure ; Marcus propose alors d'indiquer sur cette structure l'endroit où le choix de la règle a été arbitraire et d'analyser à nouveau les mêmes données en modifiant la règle choisie. Cette démarche revient à simuler un mécanisme non déterministe, elle est donc contraire aux hypothèses.

Un autre cas d'école, plus courant dans la langue anglaise que dans la nôtre, met cet analyseur en échec. Ce sont les phrases "labyrinthe" ou "garden-path". Elles ont la particularité suivante : leur structure est déterminée par un mot éloigné du début de la phrase. Donc tant que ce mot n'a pas été lu par l'analyseur il y a ambiguïté. Aussi, si l'analyseur de Marcus privilégie arbitrairement une structure non compatible avec le mot présent, l'analyse de la phrase échoue.

En résumé, la stratégie de Marcus est guidée par l'hypothèse de déterminisme. Cette hypothèse est intéressante car elle conduit à une analyse efficace donnant souvent de bons résultats. Sa faiblesse principale réside dans le fait qu'elle nie l'ambiguïté de la langue naturelle. C'est pour cette raison que nous ne la retiendrons pas car nous considérons que les ambiguïtés sont une des particularités de langue avec laquelle il faut oeuvrer.

Face à la démarche de Marcus qui est : le principe de déterminisme est posé d'emblée et la langue naturelle doit s'y soumettre, une démarche différente consiste à partir de données de la langue naturelle à analyser, à construire une stratégie d'analyse. C'est la démarche de Kimball.

4. LA STRATEGIE DE KIMBALL

D'après Kimball [Kimball, 1973], un analyseur syntaxique de la langue naturelle doit accepter la nature ambiguë de la langue et donc pouvoir construire plusieurs structures sur une même chaîne. Mais pour éviter de construire des solutions parasites, l'analyseur doit être guidé par une stratégie. Cette stratégie vise à éliminer les structures trop complexes, c'est-à-dire les structures qui ne sont pas naturelles à ceux qui pratiquent cette langue. Kimball note que l'homme ne procède pas comme un ordinateur pour construire la structure des phrases qu'il perçoit. Sa remarque se fonde sur le fait que l'homme ne mémorise qu'une quantité d'informations limitée dans l'activité d'analyse alors que l'ordinateur peut utiliser une mémoire de taille beaucoup plus importante. De ce fait, l'ordinateur crée des solutions parasites que l'homme élimine faute de place mémoire. La stratégie de Kimball repose en grande partie sur cette hypothèse à savoir que l'homme utilise une mémoire restreinte, et donc que les algorithmes d'analyse doivent eux-aussi limiter le nombre d'informations à mémoriser afin de tendre vers une analyse correcte. Kimball se place dans le cadre d'une grammaire hors contexte. Sa stratégie s'énonce à travers les principes suivants :

1 La langue naturelle se prête bien à une analyse descendante, qui associée à la lecture de quelques caractères en avant permet de limiter le nombre de structures projetées. Quelques phénomènes linguistiques comme la coordination et l'apposition, rompent la régularité de l'analyse descendante car, dans ce cas il faut insérer un noeud entre deux noeuds déjà existants.

2 Un symbole terminal sera rattaché de préférence au noeud non terminal de niveau le plus bas. En effet, les phrases de la langue naturelle sont le plus souvent construites sur des structures branchées à droite.

Exemple : Les structures engendrées par la même grammaire sur la chaîne :

"une zone de la toundra arbustive"
D F P D F F

sont données par la figure 3.4. Dans ce cas, on préférera en accord avec le principe énoncé la solution 1 à la solution 2. Le choix fait est ici contraire à celui proposé dans le premier exemple. Nous le justifierons au chapitre suivant car il est fondé sur des données linguistiques.

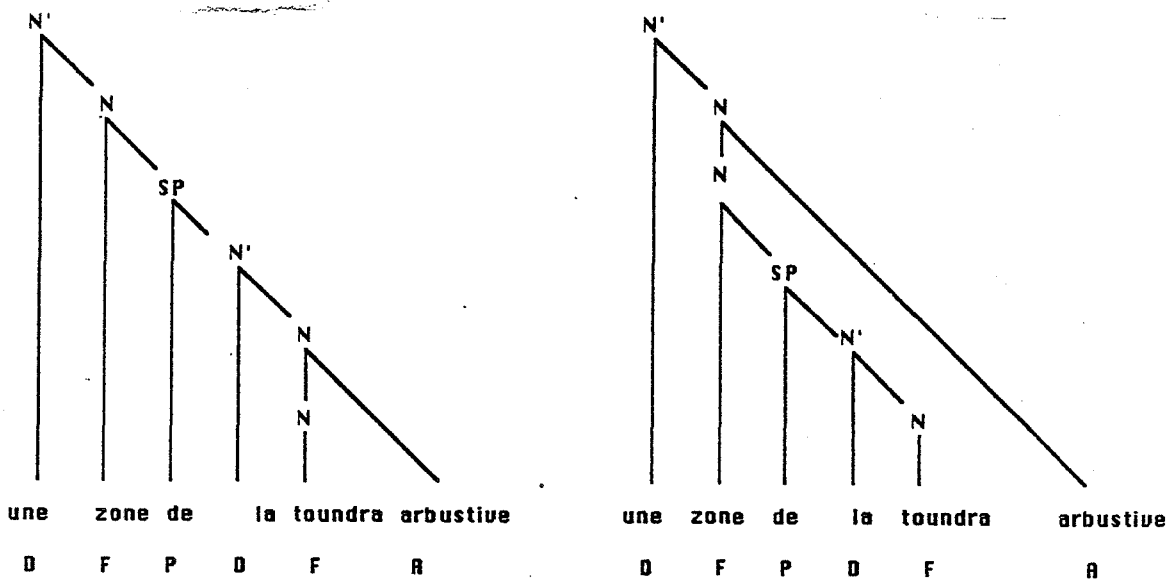


FIGURE 3.4.

3 La construction d'un nouveau noeud est annoncé par un mot-outil de la langue, comme les déterminants, les prépositions, les bornes propositionnelles. En effet, la lecture d'un mot-outil, par exemple, une préposition indique qu'un noeud étiqueté SP (syntagme prépositionnel) doit être construit.

4 L'analyseur ne peut opérer simultanément sur plus de deux propositions. Cette restriction est fondée sur le fonctionnement de la langue, et non sur la capacité des algorithmes. En effet, une phrase constituée de propositions fortement imbriquées est très difficile à comprendre donc très rarement utilisée.

5 L'analyse d'une proposition sera considérée achevée si le noeud suivant n'est pas constituant obligé de cette proposition. C'est le principe de fermeture. Lorsqu'il n'est pas respecté, la phrase n'est pas aisément compréhensible.

6 Dès que le dernier constituant d'une proposition a été formé, et la proposition fermée, suivant le principe 5, il est coûteux en terme de compréhension, de revenir en arrière pour réorganiser les constituants de cette proposition. On retrouve ici le cas des phrases "labyrinthe". La capacité d'un analyseur à éviter ce piège est liée au nombre de symboles lus en avant dans la chaîne d'entrée. Donc d'après ce principe, il est moins coûteux de lire la chaîne en avant, que de remettre en cause une structure déjà construite. Ce principe n'intervient que dans le cas où l'analyseur construit les structures les unes après les autres, car s'il les construit simultanément, il n'y aura pas retour arrière mais abandon d'une structure non attestée par la chaîne.

7 Lorsqu'une proposition est fermée, elle peut être soumise aux traitements ultérieurs, et effacée de la mémoire restreinte.

Les principes énoncés par Kimball sont illustrés par des exemples issus de la langue anglaise. Cependant, il pense qu'ils ont une portée universelle. Nous ne le contredirons pas sur ce point, car d'une part, la stratégie que nous avons adoptée pour le français a de nombreux points communs avec la sienne, et d'autre part cela nous entraînerait bien loin du sujet qui nous préoccupe et bien au-delà de nos compétences.

5. LA STRATEGIE DU GROUPE SYDO

La stratégie que nous allons exposer maintenant n'a pas la prétention d'être universelle. Son domaine d'application se limite à l'analyse de textes rédigés en français. Il n'existe pas à notre connaissance de travaux sur ce domaine. Ceci s'explique peut être d'une part parce que peu de linguistes s'intéressent à la reconnaissance de la langue écrite, encore moins à la langue française écrite, et d'autre part, parce que les informaticiens, sans l'aide de linguistes, ont tendance à formaliser la langue naturelle pour pouvoir la traiter avec des outils développés pour les langages de programmation plutôt qu'à considérer la langue naturelle comme étant un domaine d'application différent de celui des langages de programmation. La structure du groupe de recherche SYDO qui réunit linguistes et informaticiens se prête bien à une telle recherche.

La stratégie adoptée s'énonce dans son principe en ces termes : étant données une grammaire hors contexte et une chaîne d'entrée, avant de les confier à un analyseur syntaxique, il faut exploiter les indications syntaxiques portées par les mots de la chaîne d'entrée.

En effet, certains mots de la langue sont porteurs d'informations syntaxiques. Nous distinguerons parmi eux les indicateurs de structure grammaticaux, des indicateurs de structures lexicaux (ISL). Les premiers sont repérables par la catégorie morphologique qui leur est assignée. Ce sont, par exemple, les prépositions, les bornes propositionnelles. Ils introduisent une structure déterminée par la grammaire : un syntagme prépositionnel, une proposition. Ainsi leur présence engendrera l'application des seules règles compatibles.

Les indicateurs de structure lexicaux sont soit des verbes soit des noms ou adjectifs qui régissent des compléments, compléments précédés éventuellement d'une préposition. Leur caractère lexical s'explique par le fait que la seule analyse morphologique ne permet pas de prédire de façon univoque la structure qu'ils engendrent. C'est l'adéquation entre les prépositions présentes dans la chaîne d'entrée et leur comportement rectionnel, enregistré dans un lexique, qui permet de prédire la structure qu'ils annoncent.

La présence d'ISL dans la chaîne permet d'en prédire la structure. L'absence d'ISL n'est pas dénuée d'informations. En effet, dans le cas le plus général, la langue naturelle a tendance à admettre un fonctionnement régulier. Nous étayerons cette affirmation au cours du chapitre suivant.

Ainsi la lecture de la chaîne permet de choisir, en s'appuyant sur certains mots les règles de la grammaire qui doivent être appliquées. Les critères à partir desquels le choix s'effectue sont fondés sur le fonctionnement de la langue. Nous les verrons plus en détail au chapitre 4. L'analyseur syntaxique qui entrera en action à l'issue de cette phase de prédiction, devra à tout instant restreindre l'exploration systématique de la grammaire aux seules règles prédites. La mise en oeuvre d'un tel analyseur à partir de l'algorithme d'Earley sera l'objet du chapitre 5.

Chapitre 4

LES FONDEMENTS LINGUISTIQUES DE L'ANALYSE SYNTAXIQUE

"Maudit soit le père de l'épouse du forgeron qui forgea le fer de la cognée avec laquelle le bûcheron abattit le chêne dans lequel on sculpta le lit où fut engendré l'arrière-grand-père de l'homme qui conduisit la voiture dans laquelle ma mère rencontra mon père."

Robert DESNOS, "La colombe de l'arche"

Le chapitre précédent a montré que l'analyse syntaxique automatique de la langue naturelle était à la fois un problème informatique et linguistique et qu'un analyseur de type combinatoire ne pouvait opérer efficacement sans prendre en compte le fonctionnement de la langue naturelle. Il nous faut donc maintenant construire les données linguistiques nécessaires au processus d'analyse. Ce sera l'objet de ce chapitre. Nous fixerons d'abord le cadre du modèle linguistique, puis étudierons les éléments de la stratégie d'analyse définie au chapitre précédent : les indicateurs de structure. Enfin nous associerons ces données, modèle et indicateurs de structure, pour mettre en oeuvre la stratégie.

Notre étude se limite aux syntagmes nominaux en raison de l'application visée : l'indexation automatique. Cette restriction aux seuls syntagmes nominaux pourrait être perçue comme un cloisonnement de la langue qui viserait à isoler l'étude des syntagmes nominaux de celle du reste de la langue. Il n'en est rien : l'analyse syntaxique du syntagme nominal passe par la prise en compte de phénomènes linguistiques qui affectent l'ensemble de la langue : propriétés rectionnelles, coordination... De ce fait, si l'on progresse dans l'analyse des syntagmes nominaux, l'on progresse parallèlement dans l'analyse de la langue.

1. LE MODELE LINGUISTIQUE HORS CONTEXTE.

Le modèle linguistique a pour fonction de rendre compte du fonctionnement de la langue en respectant les contraintes du modèle formel choisi. Dans notre cas, nous avons privilégié le modèle hors-contexte pour les raisons vues au chapitre 3, donc le modèle linguistique doit lui-aussi être hors-contexte.

Le pivot du modèle linguistique en est la grammaire, bien évidemment hors-contexte. Cependant, une telle grammaire ne peut prendre en compte tous les aspects linguistiques, car certains ne relèvent pas de ce modèle, comme la coordination.

1.1. LA GRAMMAIRE DU SYNTAGME NOMINAL

La grammaire s'exprime de façon classique au moyen de symboles et de règles. Les symboles terminaux sont des catégories morphologiques. Les règles peuvent faire intervenir, outre les catégories morphologiques, les variables associées à ces catégories, par le biais de conditions d'application.

1.1.1. Les symboles

La grammaire du syntagme nominal est une grammaire expérimentale élaborée avec l'aide de A. Berrendonner. C'est une grammaire hors contexte dont l'axiome est N'' et où les symboles sont :

V_T = { F, W, D, P }
V_N = { N'', N', N, A'', A', A, SP, D', K }

Les symboles terminaux correspondent aux catégories morphologiques que l'on rencontre au sein d'un syntagme nominal. Comme l'analyse morphologique associe à chaque forme du texte initial une catégorie et des valeurs de variables (cf. chapitre 1), les règles de la grammaire pourront en tenir compte. Les valeurs de variables ici mentionnées ont la même signification que dans le chapitre 1.

La catégorie F recouvre l'ensemble des noms et des adjectifs, la catégorie W, l'ensemble des adverbes, la catégorie D, l'ensemble des déterminants, la catégorie P, l'ensemble des prépositions.

Parmi les symboles non terminaux, il faut tout d'abord distinguer N'', qui représente l'axiome de la grammaire, et donc un syntagme nominal. Les autres symboles correspondent à des noeuds intermédiaires qui permettent de passer des symboles terminaux à l'axiome. Les notations choisies traduisent, le cas échéant, une relation d'inclusion sur les noeuds. Ainsi, on notera :

X' un syntagme contenant X et de niveau supérieur à X
X'' un syntagme contenant X' et de niveau supérieur à X'

En accord, avec cette notation, on notera A un syntagme adjectival de niveau 0, A' un syntagme adjectival de niveau 1, et enfin A'' un syntagme adjectival de niveau 2. De même,

les symboles N, N' et N'' représenteront respectivement les syntagmes nominaux de niveau 0, 1 et 2.

Remarque
Lorsque le niveau d'un syntagme nominal ou adjectival n'est pas précisé, il s'agit alors d'un syntagme de niveau 2, noté N'' ou A''.

1.1.2. Les règles de la grammaire

SP désigne un syntagme prépositionnel et se réécrit :

SP --> P N'' [1]

"de la maison", "à café"...

A est un syntagme adjectival de niveau 0, ouvert à droite et à gauche, il se réécrit :

A --> F (ADJ) [2]
A --> F (NAN) [3]

A' est un syntagme adjectival de niveau 1 complété à gauche et se réécrit :

A' --> W (AAJ) A [4]
A' --> A [5]

"très mauvais"

A'' est un syntagme adjectival de niveau 2, complété à droite et à gauche. Si la variable DQ n'est pas associé à A' alors :

A'' --> A' [6]

mais dans le cas où la variable DQ a une valeur on appliquera l'une ou l'autre de ces règles suivant le nombre de SP régis :

A'' --> A' SP [7]
A'' --> A' SP SP [8]

"protégé par un vaccin"
"protégé par un vaccin contre les maladies"

D' désigne un déterminant de niveau 1. Il s'obtient par :

D' --> D [9]
D' --> D (DEF) D (NUM) [10]
D' --> de /le, la ou les/ [11]

"le", "un", "les deux", "ces quelques"...

N désigne un syntagme nominal, de niveau 0, qui peut être complété à droite et à gauche. Les règles de réécriture de N sont :

N --> F (NOM) [12]
N --> F (NAN) [13]

N --> N SP [14]
N --> A' N [15]
N --> N A'' [16]
N --> A' [17]
N --> A'' [18]

"Nestor", "maison", "artiste", "machine à café",
"écosystème de la prairie", "si beau spectacle",
"grand cheval protégé par un vaccin contre les maladies",
"le plus petit", "(le seul) capable de brutalité"

Remarque

Les règles 17 et 18 n'appartiennent pas à la grammaire de base. En effet, de telles règles conduisent, dans le cas général, à de nombreuses solutions parasites. Elles ne seront appliquées que dans le cas où la grammaire de base est mise en échec.

N' désigne un syntagme nominal de niveau 1, fermé à droite. Il s'obtient soit à partir d'un N si celui-ci ne régit pas de complément :

N' --> N [19]

soit, comme pour les adjectifs, à partir de l'une des règles suivantes en cas de rection :

N' --> N SP [20] *
N' --> N SP SP [21]
N' --> N SP SP SP [22]

ou encore :

N' --> N -ci [23]
N' --> N -là [24]

puisque ces particules ferment , à droite, le syntagme nominal de niveau 0.

"Nestor", "résistance aux virus",
"étude du docteur X sur le comportement",
"expédition d'un satellite dans l'espace par la NASA"...,
"(cette) maison-ci"

K représente une expression quantitative qui peut être centre de syntagme, et s'obtient :

K --> W (QUA) [25]

---

* Les règles [14] et [20] ont même partie droite ; cependant, elles ne s'appliquent pas dans les mêmes conditions. Pour nous, la règle [20] s'applique lorsque le SP est régi par le N qui le précède (cas marqué par la présence de la variable DQ), alors que la règle [14] s'applique dans tous les autres cas. Nous ne suivons pas en celà la grammaire de A. Berrendonner à la lettre, qui applique la règle [14] aux mots composés et la règle [20] dans tous les autres cas. Mais alors, puisque l'on ne sait distinguer un mot composé sur des critères syntaxiques, les deux règles sont toujours en concurrence, d'où la création d'ambiguïtés.

K --> D (NUM) N (MES) [26]
K --> D (DEF) A' [27]

où la variable QUA définit sur les adverbes une sous-classe de ceux qui fonctionnent comme des expressions quantitatives (beaucoup), et où MES marque les noms de quantité (unité de mesure ou "ensemble", "tas", "masse"...)

```
"beaucoup", "deux kilos", "chaque ensemble",
"le premier"
```

N'' est un syntagme nominal de niveau 2. Il s'obtient par les règles suivantes :

N'' --> N' [28]
N'' --> D' N' [29]
N'' --> N'' N'' [30]
N'' --> K de N'' [31]
N'' --> K [32]

```
"Nestor", "l'expédition d'un satellite...",
"l'empereur Jules I", "trois douzaines d'huîtres"
```

1.1.3. Conditions d'application des règles

Les règles de la grammaire exposée ci-dessus font intervenir outre les symboles de V, des valeurs de variables morphologiques. La mention explicite de ces valeurs indique que l'application de la règle est restreinte aux symboles de V affectés de cette valeur. C'est le cas par exemple des quatre règles :

N --> F(NOM) [12]
N --> F(NAN) [13]
A --> F(ADJ) [2]
A --> F(NAN) [3]

qui, sans restriction, s'exprimeraient :

N --> F
A --> F

On peut donc interpréter ces restrictions, comme des conditions d'application des règles, conditions qui permettent d'exploiter l'information contenue par les variables pour limiter les solutions parasites.

Mais il existe aussi des règles soumises à des conditions d'application implicites. Ce sont celles qui sous-entendent l'accord préalable en genre et en nombre des symboles de la partie droite. En effet, des règles comme :

N'' --> D' N' [29]
N --> N A'' [16]

ne s'appliquent que si les valeurs de genre et de nombre des symboles de la partie droite sont compatibles. Considérons, par exemple, la règle [17]. Elle doit s'entendre comme :

N --> N(gr1, nb1) A''(gr2, nb2)

où les valeurs de genre gr1 et gr2 sont compatibles, ainsi que les valeurs de nombre nb1 et nb2. Comme gr1 et gr2 prennent leurs valeurs dans {MAS, FEM, GRN} et comme nb1 et

nb2 les prennent dans {SNG, PLU, NBN}, il existe 81 (9^2) combinaisons différentes de (gr1,nb1) avec (gr2,nb2), où gr1, gr2, nb1 et nb2 décrivent leur ensemble de valeurs respectif. Parmi ces combinaisons, certaines ne respectent pas l'accord en genre et en nombre. Ce sont toutes celles où l'on a simultanément :

gr1 = MAS et gr2 = FEM
ou gr1 = FEM et gr2 = MAS
ou nb1 = SNG et nb2 = PLU
ou nb1 = PLU et nb2 = SNG

Il reste donc 49 (7^2) combinaisons licites, chacune correspondant à une condition d'application de la règle [16]. Cette règle recouvre donc implicitement 49 règles qui mettent en jeu les valeurs de genre et de nombre.

Ainsi, parmi les règles de la grammaire certaines s'appliquent sans réserve comme N --> N SP, d'autres, sous conditions qu'elles soient implicites ou explicites. Soumettre une règle hors-contexte à restriction, comme N --> F ou N -->N A'' revient à la remplacer par un certain nombre de règles mettant en jeu les valeurs des variables morphologiques. Le nombre de ces règles est fini. En effet, une variable morphologique ne peut prendre qu'un nombre fini de valeurs [voir chapitre 1]. La partie droite d'une règle hors contexte contient un nombre fini de symboles de V. Chaque symbole est marqué par un nombre fini de variables. Donc, soumettre les règles d'une grammaire hors contexte à des condition d'application revient à leur substituer un nombre fini de règles hors contexte. De ce fait, la grammaire reste de nature hors contexte.

1.1.4. Les règles de transfert

Puisque l'application d'une règle peut être fonction des valeurs des variables morphologiques associées aux symboles de sa partie droite (éléments de V), et que, au départ, ces valeurs sont portées par les symboles terminaux de la chaîne à analyser, il faut lors de l'application d'une règle, transférer certaines valeurs sur la partie gauche. Plusieurs cas peuvent se présenter :

Soit une valeur est un critère de choix parmi l'ensemble des règles. A l'issue de la sélection, l'information ne réside plus dans cette valeur mais dans la règle choisie. Il serait donc redondant de transférer cette valeur sur le symbole gauche.

```
Exemple :
Soit F(ADJ, MAS, SNG) le symbole à analyser. La valeur ADJ de la
variable NA permet de choisir parmi les règles de réécriture de F, la
règle A --> F(ADJ). Le noeud A ainsi construit est par définition un
syntagme qui contient un adjectif, il est donc inutile de transférer
sur ce noeud la valeur ADJ. Ainsi :
        A(MAS, SNG) --> F(ADJ, MAS, SNG)
```

Soit une valeur n'est pas un critère de choix ; n'étant pas exploitée, elle sera transférée sur le symbole gauche. C'est le cas des variables de genre et de nombre dans l'exemple précédent.

Soit les valeurs transférées sur le symbole gauche sont calculées à partir de celles affectées aux symboles droits. C'est le cas de l'accord en genre et en nombre.

Exemple :Soit la chaîne :
les [D (DEF, GRN, PLU)] animaux [F (NOM, MAS, PLU)] nuisibles [F (ADJ, GRN, PLU)]
Le transfert des variables au cours de l'analyse de cette chaîne est illustré par la figure 4.1. On observera que :

1 les valeurs NOM et ADJ ne sont pas transférées parce que redondantes sur les noeuds construits, respectivement N et A.

2 lors de l'application de la règle N --> N A'', le symbole gauche se verra affecter les valeurs MAS et PLU calculées à partir des valeurs de genre et de nombre de la partie droite.

3 le transfert des variables, lors de l'application de la règle N'' --> D' N' s'opère de manière identique.

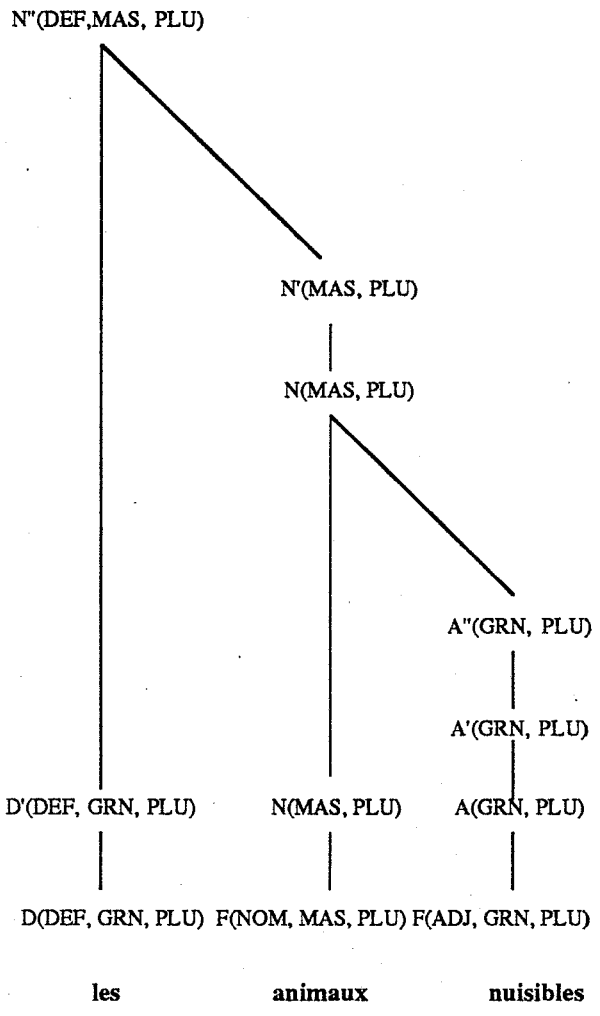


FIGURE 4.1.

Les règles de transfert doivent donc définir d'abord quelles sont les variables soumises au transfert, puis auprès de quels symboles de la partie droite se situent les valeurs d'origine et enfin comment calculer les valeurs transférées.

En conclusion, le modèle linguistique n'impose pas mais propose que les règles soient soumises à des conditions d'application. En effet, ces conditions ne modifient pas la nature de la grammaire et sont souvent utiles pour limiter le nombre de solutions parasites engendrées par l'analyse. Aussi, avons-nous retenu cette solution. Une règle de la grammaire se décompose alors en quatre champs :

1. le symbole gauche, élément de V_N
2. la liste des symboles de la partie droite, éléments de V
3. la condition d'application
4. la règle de transfert

1.2. LES EXCEPTIONS AU MODELE HORS CONTEXTE

Le modèle hors contexte est un modèle formel qui ne peut rendre compte de tous les phénomènes linguistiques. Les linguistes, entre eux, n'ont pas pris de position unanime sur les rapports entre langage hors-contexte et langue naturelle [GAZDAR, 1983]. Nous prenons le parti, ici, de considérer que l'accord en genre et en nombre relève du modèle hors-contexte. Il n'en reste pas moins quelques exceptions à ce modèle. Nous avons vu au chapitre 1, le traitement de régularisation effectué sur les morphèmes discontinus afin de ne pas sortir de ce modèle. Cependant deux exceptions demeurent : la coordination et l'apposition. Ces deux phénomènes sont de même nature, la coordination étant plus courante, nous nous y attarderons.

La coordination se traduit par la présence de conjonctions ("et", "ou"), et/ou de virgules. Elle peut affecter la plupart des symboles de la grammaire et de plus, sa portée n'est pas limitée. On ne peut pas en rendre compte par un nombre fini de règles du type :

X --> X et X et ... et X

Elle ne peut donc se traiter dans le cadre hors contexte. Aussi, pour résoudre le problème de la coordination, il n'est pas envisagé d'intégrer de telles règles dans la grammaire, car alors le modèle linguistique ne serait plus hors-contexte.

Une solution proposée par A. Berrendonner, consisterait à confier à une méta-grammaire le soin de traiter la coordination. Une méta-grammaire est une grammaire, donc un ensemble de règles. Les règles de la méta-grammaire opèrent sur les règles de la grammaire hors-contexte. La méta-grammaire est donc une grammaire qui gère l'application des règles de la grammaire hors-contexte.

Illustrons sur un exemple, ce que devrait être le traitement de la coordination. Soit la chaîne :

```
les variétés de  Melon , Piment et Tomate résistantes à les virus
 D  F(NOM)    P  F(NOM) T F(NOM) C  F(NOM) F(ADJ, DVB) P  D F(NOM)
```

L'analyse de cette chaîne s'arrête sur la lecture de la virgule. La structure alors construite est indiquée sur la figure 4.2.

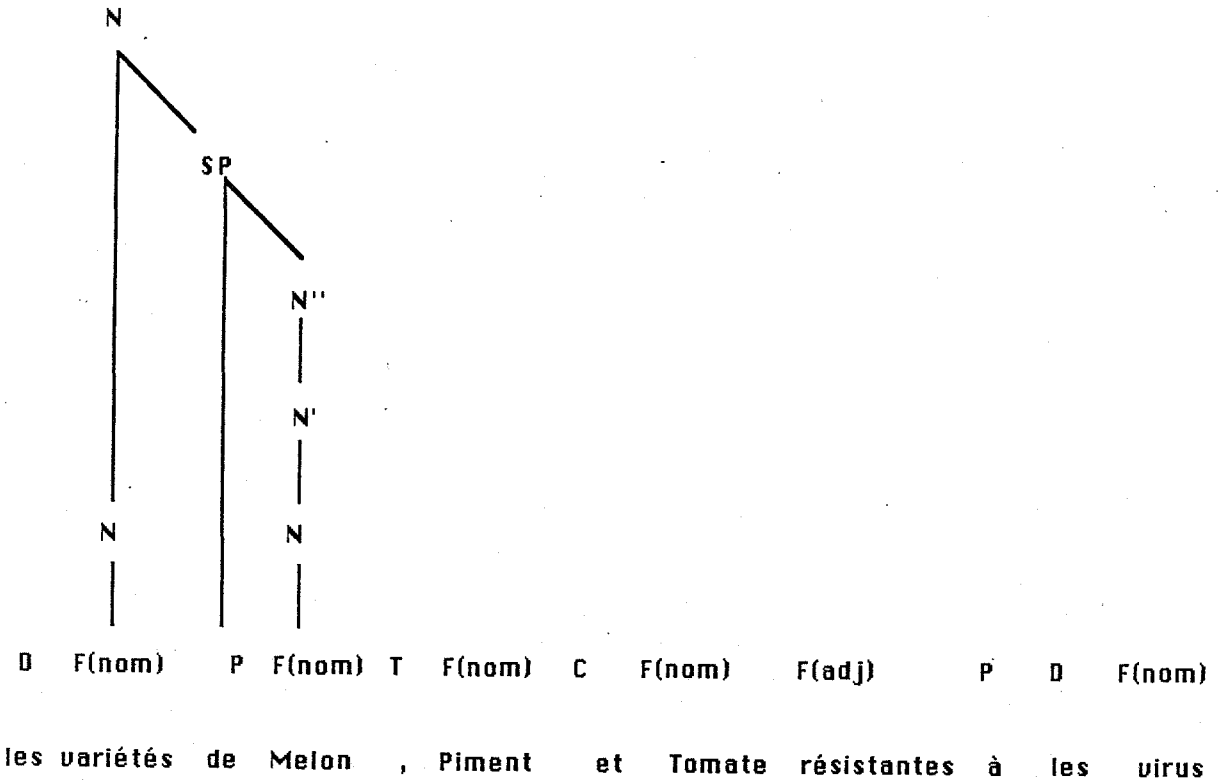


FIGURE 4.2.

La méta-grammaire intervient ; elle suspend l'analyse en cours, lance celle de la sous-chaîne qui débute après la virgule. Cette dernière analyse s'interrompt à la lecture de la conjonction "et". La structure construite sur cette sous-chaîne correspond à celle de la figure 4.3.

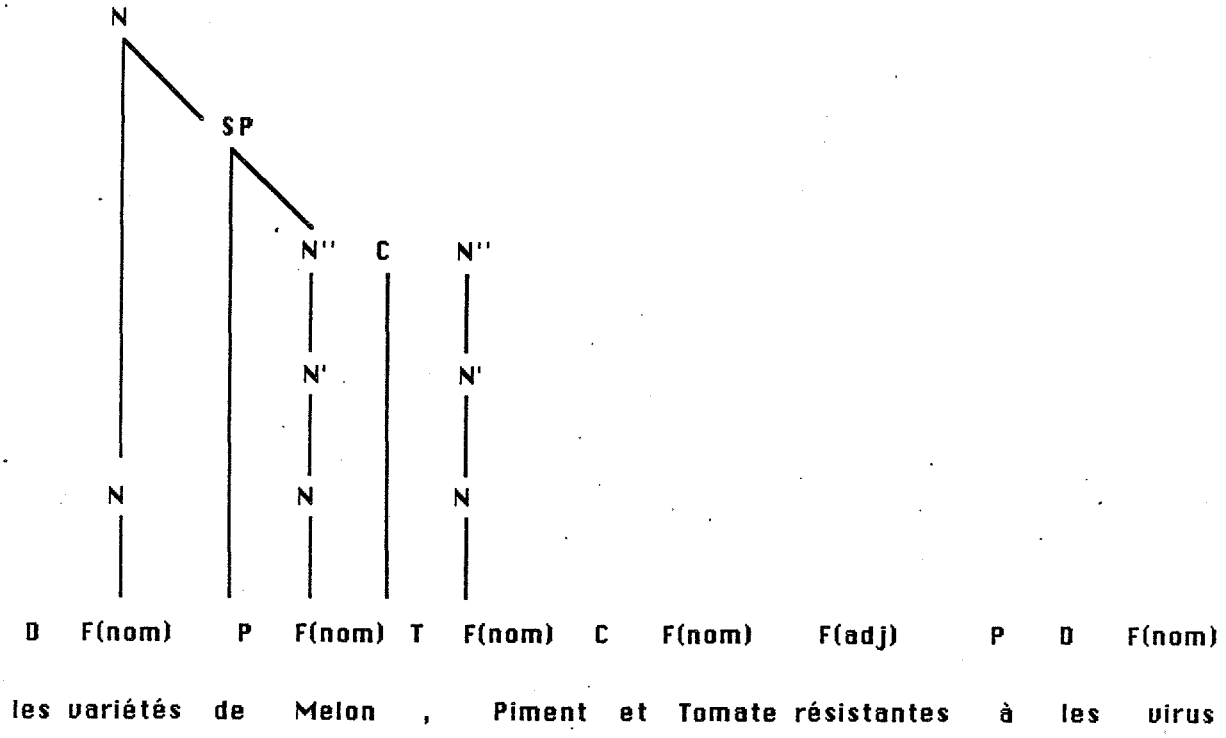


FIGURE 4.3.

La méta-grammaire substitue à ces deux structures, la structure coordonnée. Elle opère ainsi : en cassant la première structure contruite pour en isoler le noeud présentant le plus de symétrie avec la sous-chaîne qui suit la virgule, en rattachant les noeuds symétriques autour du coordonnant et enfin en substituant dans la structure globale, un seul noeud N'' résultant de la séquence N'' C N''. D'où la nouvelle structure figurant en 4.4.

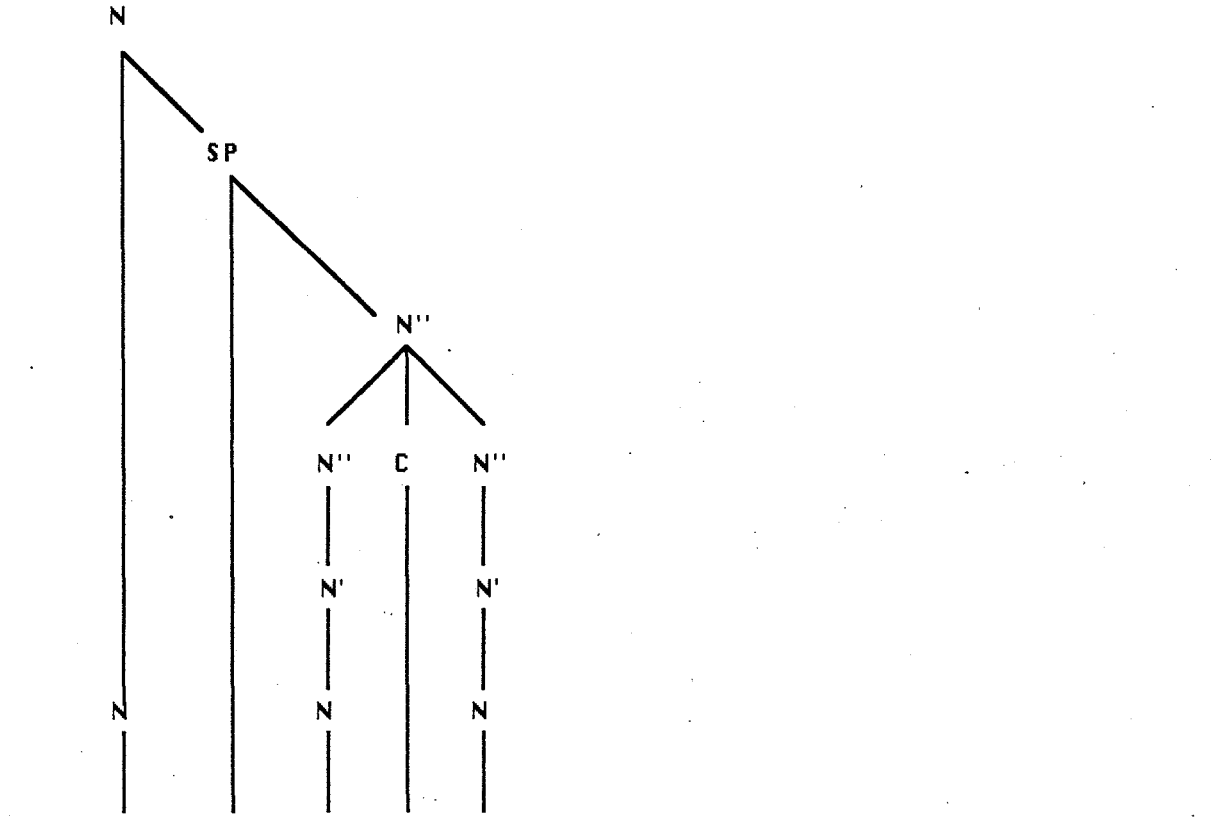


FIGURE 4.4.

L'analyse de "Tomate" s'opère de même et la méta-grammaire engendre la structure de la figure 4.5. Le mot suivant "résistantes" n'est pas un candidat coordonnant. C'est un adjectif qui ne peut s'accorder à "Tomate". La méta-grammaire a donc terminé sa tâche et remet à l'analyseur la structure coordonnée; l'analyse se poursuit normalement.

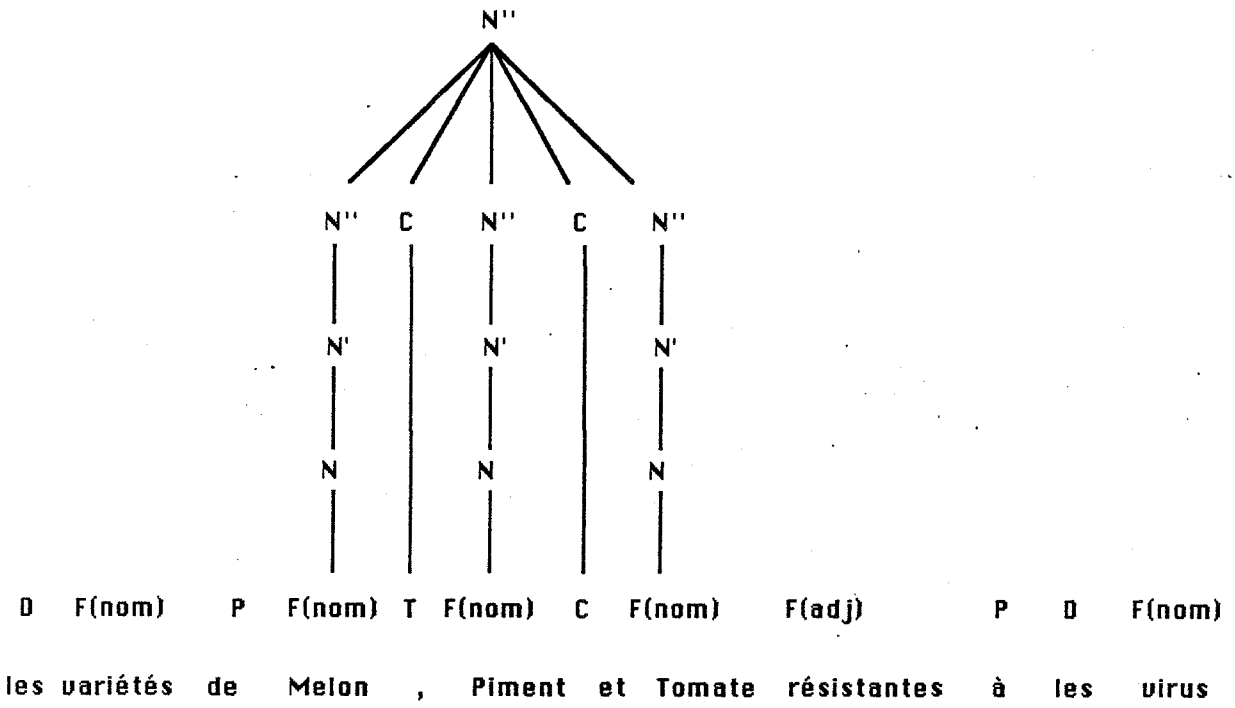


FIGURE 4.5

A travers cet exemple, nous avons illustré les rôles respectifs de la grammaire et de la méta-grammaire. La mise en oeuvre de la méta-grammaire est déclenchée par la détection d'un candidat coordonnant. Cette méta-grammaire a pour fonction de délimiter la portée du coordonnant à partir de critères de symétrie, et enfin de substituer pour la grammaire hors-contexte, les noeuds coordonnés par un seul noeud de même nature.

C'est donc ainsi que l'on envisage de traiter le problème de la coordination. Ce traitement s'avère très délicat pour deux raisons. La première réside dans le fait qu'il n'est pas simple de déterminer en surface parmi les candidats coordonnants, ceux qui sont réellement des opérateurs de coordination. Ni la virgule, ni la conjonction "et" ne sont univoques. La seconde provient de la difficulté de repérer les éléments de la symétrie. En effet, la démarche est la suivante : la détection d'un coordonnant permet de prédire qu'il existe une symétrie de part et d'autre. Mais alors, il faut déterminer ces éléments de symétrie, sans en connaître a priori ni la nature, ni la portée. C'est pourquoi, nous nous limitons à évoquer le problème sans y apporter de solution.

2. LES OUTILS DE LA STRATEGIE : LES INDICATEURS DE STRUCTURE

Un indicateur de structure est une forme dont l'occurrence renseigne sur la structure syntaxique d'un segment de texte auquel elle appartient. On distinguera les indicateurs de structure grammaticaux (ISG) dont les propriétés syntaxiques sont issues de la catégorie morphologique à laquelle ils appartiennent, des indicateurs de structure lexicaux (ISL) dont les propriétés syntaxiques sont issues du lexique.

2.1. LES INDICATEURS DE STRUCTURE GRAMMATICAUX

Sont ISG, toutes les formes qui appartiennent, sans ambiguïté, à certaines catégories ou sous-catégories morphologiques. Elles permettent une ébauche de la structure syntaxique. Sont considérées comme ISG des formes des catégories Q, T, Y, V, P, D.

2.1.1. Q : les bornes propositionnelles

Une borne propositionnelle indique le début d'une proposition en Q. Elle est donc frontière gauche de Q-phrase et prédit une proposition.

2.1.2. T : les ponctuations

Parmi les ponctuations, les ponctuations fortes étiquetées VP1, comme ".", "!", "?", signalent :
(1) la frontière droite de tous les syntagmes amorcés
(2) la frontière gauche d'une phrase, si cette ponctuation n'est pas la dernière forme d'un texte.

Les ponctuations faibles étiquetées VP2, comme ";" ou ":" qui séparent des membres de phrase indiquent :
(1) la frontière droite de tous les syntagmes de niveau inférieur à celui de la phrase
(2) la frontière gauche d'un membre de phrase

Remarque L'interprétation de la virgule est beaucoup plus délicate, car très dépendante du contexte. Nous ne la traiterons pas ici.

2.1.3. Y et V : les pronoms préverbaux et les verbes

Le premier symbole Y rencontré à partir de la gauche signale :
(1) la frontière gauche d'un syntagme verbal
(2) la frontière droite de tout autre syntagme de niveau inférieur à celui de la proposition

Si l'on rencontre un symbole V sans avoir rencontré au préalable un symbole Y, alors V porte les mêmes informations de frontière que Y.

```
"je la lui ai donnée", " (les arbres) perdent souvent (leurs feuilles)"
```

2.1.4. P : les prépositions

Une préposition indique la frontière gauche d'un syntagme prépositionnel et annonce un syntagme nominal qui débute à la forme suivante

```
"sur le petit banc", "(tasse) à café"
```

2.1.5. D : les déterminants

Un déterminant débute un syntagme nominal. Il est suivi soit d'un autre déterminant, soit d'un syntagme nominal de niveau 1.

```
"les trois pommes", " le plus grand animal"
```

2.2. LES INDICATEURS DE STRUCTURE LEXICAUX

La présentation des indicateurs de structure lexicaux, faite ici, doit beaucoup aux travaux des linguistes, ainsi M. GROSS[1975] et A. CULIOLI[1970], qui ont entrepris une étude approfondie du fonctionnement syntaxique des verbes, en vue d'un traitement automatique.

Un ISL est un mot qui possède des propriétés rectionnelles. L'information sur le comportement syntaxique de ce mot n'est pas contenue dans sa catégorie morphologique - tous les verbes sont catégorisés V et n'ont pas les mêmes propriétés rectionnelles. Cette information est d'ordre lexical. Parmi les ISL, on distinguera ceux qui sont porteurs directs de cette information, de ceux, indirects, dont le comportement provient d'un mot dont ils dérivent. Aux premiers correspond dans le lexique l'indication de leur comportement syntaxique ; aux seconds ne correspondra qu'un renvoi au mot dont ils héritent leurs propriétés rectionnelles. Sont indicateurs directs, tous les verbes, quelques adjectifs et quelques noms. Sont indicateurs indirects tous les noms déverbaux ou déadjectivaux et tous les adjectifs déverbaux.

2.2.1. Les indicateurs directs.

2.2.1.1. Les verbes

Le comportement syntaxique d'un verbe est défini par la nature des compléments qu'il régit. Un complément est un syntagme nominal ou prépositionnel (nous n'aborderons pas, ici, les compléments sous forme de Q-phrase). La formalisation des propriétés rectionnelles des verbes passe par celle des compléments.

2.2.1.1.1. Les compléments

On distinguera six types de compléments, notés de C_0 à C_5 en fonction des critères suivants :

C_0 : syntagme nominal, non précédé d'une préposition et anaphorisable en "il"
`Dans "le chat dort", le SN "le chat" est de type` C_0.

C_1 : syntagme nominal, non précédé d'une préposition et anaphorisable en "le".
`Dans "il manque son train", le SN "son train" est de type` C_1

C_2 : syntagme nominal précédé de la préposition "à" et anaphorisable en "lui" ou "leur".
`Dans "il le donne à son chien", le SN "son chien" est de type` C_2

C_3 : syntagme nominal précédé d'une préposition locative ("à", "dans", "sur"...) et anaphorisable en "y".
`Dans "je monte sur le toit", le SN "le toit" est de type` C_3

C_4 : syntagme nominal précédé d'une préposition "de" et anaphorisable en "en".
`Dans "je viens du jardin", le SN "le jardin" est de type` C_4

C_5 : syntagme nominal précédé d'une préposition et non anaphorisable.
`Dans "il lutte contre le feu", le SN "le feu" est de type` C_5

Remarque

1- Pour la préposition "à", on distinguera le cas où le syntagme régi est anaphorisable en "lui" de celui où il est anaphorisable en "y", en les notant respectivement "à1" et "à2".

2- Les compléments circonstanciels n'entrent dans aucune de ces catégories.

3- Le complément noté C_0 n'est pas en toute rigueur un complément du verbe, mais le sujet. Nous l'assimilons cependant à un complément puisque le sujet d'un verbe devient par nominalisation un complément introduit par les prépositions "de" ou "par".

4- l'exposé ne tient pas compte des phénomènes de supplétisme :
`"je lui joins" mais "je l'y joins"`

2.2.1.1.2. Propriétés rectionnelles des verbes

Les propriétés rectionnelles des verbes seront indiquées dans le lexique. Chaque construction verbale se verra affecter 6 cases numérotées de 0 à 5 La case i contiendra une préposition "p" si le verbe régit un complément C_i précédé par la préposition "p".

Exemple :
le verbe "donner" qui régit les compléments C_0, C_1 et C_2 se verra affecter dans le dictionnaire les données suivantes :

donner [Ø] [Ø] [à1] [] [] []

Remarque
le symbole Ø signifie que le syntagme régi n'est pas précédé d'une préposition, c'est donc un syntagme nominal. Ce cas est à distinguer du symbole [] qui lui signifie la non existence de syntagme régi.

Dans le cas où un même complément peut être régi par plusieurs prépositions, la case correspondant à ce complément contiendra la liste des ces prépositions. Alors, l'occurrence de l'une des prépositions exclut celle des autres prépositions occupant la même position.

Exemple :
le verbe "monter" admet un complément de type C_3 régi par "à", "dans" ou "sur"... Dans le dictionnaire on aura donc :

monter [Ø] [Ø] [] [à2, dans, sur] [] []

L'occurrence de la préposition "sur" exclut la présence des prépositions "à2" et "dans" pour le complément de type C_3.

* je monte sur le toit dans la cheminée

Si un verbe admet plusieurs comportements syntaxiques, on créera une entrée pour chacun des comportements.

Exemple :
le verbe "manquer" admet cinq constructions syntaxiques différentes, l'une excluant les autres. Ainsi :

manquer1	[Ø] [] [à1] [] [] []	il lui manque
manquer2	[Ø] [] [] [] [de] []	il en manque
manquer3	[Ø] [Ø] [] [] [] []	il le manque
manquer4	[Ø] [] [] [] [] []	il manque
manquer5	[Ø] [] [] [à2] [] []	il y manque

2.2.1.2. Les adjectifs

Certains adjectifs se comportent comme des ISL directs. Ce sont, par exemple : "sûr de", "susceptible de", "apte à"... Le nombre de ces compléments est le plus souvent limité à un seul. Cependant, par souci d'homogénéité, nous utiliserons la même formalisation que pour les verbes. Ainsi, un adjectif régira un complément C_i, si dans une expression du type :

"il est ADJ PREP SN"
le syntagme nominal répond aux conditions d'anaphorisation définies ci-dessus.

"il est apte au travail".
Le SN "le travail" est anaphorisable en "y", donc "apte" régit un complément de type C_3. Le dictionnaire indiquera :
apte [] [] [] [à2] [] []

2.2.1.3. Les noms

Il existe aussi quelques noms qui régissent des compléments, comme "père de", "fonction de". On procèdera alors, comme pour les verbes et les adjectifs, en partant de l'expression "il est le NOM Prep SN".

"il est le père de Jules".
Jules est anaphorisable en "en" et donc le dictionnaire contiendra :
père [] [] [] [] [de] []

2.2.2. Les indicateurs indirects

2.2.2.1. Les noms et adjectifs déverbaux

2.2.2.1.1. Définition

Une forme de la catégorie F sera dite déverbale si :
- elle est dérivée d'un verbe
- son comportement syntaxique est issu de celui du verbe et attesté en surface

Pour marquer qu'une forme de catégorie F dérive d'une autre forme (ici un verbe) on utilisera la variable DQ qui aura pour valeur :

- DVB pour un nom d'action déverbal
"déménagement" sera analysé F (NOM, DVB)

- AGE pour un nom d'agent déverbal
"déménageur" sera analysé F (NAN, AGE)

- DVB pour un adjectif déverbal
"transmissible" sera analysé F (ADJ, DVB)

Quant aux participes présents et passés, ils sont marqués lors de l'analyse morphologique par la variable PA, avec les valeurs respectives PPR et PPA. La variable DQ n'apporte pas dans ce cas d'informations supplémentaires. Ainsi, seront considérés comme adjectifs déverbaux :

- les participes présents au sens strict

Exemple :
"résistant" dans une expression comme "les variétés résistant aux virus" est analysé F (ADJ, PPR). Par contre "résistantes" dans l'expression "les variétés résistantes aux virus" (expression non académique mais attestée par certains corpus) sera analysé comme F (ADJ, DVB), ce qui nous ramène au cas (3).

- les participes passés dans la mesure où ceux-ci ne sont pas éléments d'une forme verbale composée.

Exemple :
"transmis" est analysé F (ADJ, PPA) dans une expression comme "les virus transmis par les animaux", alors qu'il est partie de verbe dans "il a transmis la nouvelle".

2.2.2.1.2. Nature des informations du lexique

A l'heure actuelle, seule la composante flexionnelle de l'analyseur morphologique est réalisée. La composante dérivationnelle qui permettrait de regrouper sous une même entrée lexicale tous les mots ayant même radical, n'existe pas encore. Avec cet outil, il serait possible, à partir d'un mot de la catégorie F d'identifier, le cas échéant, le verbe ou l'adjectif dont il est issu, son mode de dérivation (agent, déverbal). Pour combler cette lacune temporaire, il devient nécessaire de constituer un lexique qui indique pour chaque forme déverbale :

- le verbe dont elle dérive
- le mode de dérivation

Ainsi, le lexique contiendrait ces informations, informations qui doivent permettre de retrouver le comportement syntaxique. A titre d'exemple, donnons le contenu du lexique pour les cas traités ci-dessus :

"déménagement"	: "déménager"	DVB
"déménageur"	: "déménager"	AGE
"transmissible"	: "transmettre"	DVB
"résistant"	: "résister"	PPR
"résistant"	: "résister"	DVB
"transmis"	: "transmettre"	PPA

2.2.2.1.3. Le passage du lexique au comportement syntaxique

Les données présentes dans le lexique doivent suffire à identifier le comportement syntaxique d'une forme donnée. En effet, connaissant le verbe dont elle dérive, le fonctionnement syntaxique de ce verbe et le mode de passage de la forme au verbe, on peut par un jeu de règles simples retrouver le comportement syntaxique de la forme.

Les participes présents (PPR) se comportent comme les verbes dont ils sont issus. Cependant, le syntagme qui joue le rôle de complément de type C_0 n'apparaît pas comme tel, puisque la grammaire le considérera comme le symbole "N" qui suivi du symbole "A''" se réécrira "N" [règle 17].

Les déverbaux à caractère d'agent (AGE), se comportent comme le verbe, à une réserve près : ils ne gouvernent pas de complément de type C_0.

Les participes passés (PPA) se comportent comme les verbes pour les compléments autres que C_0 et C_1. En effet, le complément C_0 est régi par la préposition "par" et le complément C_1 n'apparaît pas, puisque la grammaire le traitera comme un symbole "N" qui suivi de"A''" se réécrira "N" [règle 17].

Les autres déverbaux (DVB) se comportent comme les verbes pour les compléments autres que C_0 et C_1. En effet, le complément C_1 est alors régi par une préposition "de", et C_0 par une préposition "de" ou "par".

"l'ordinateur traite les données"

devient

"le traitement des données par l'ordinateur"

Ces quelques règles permettent de transposer le comportement syntaxique des verbes aux formes qui en sont issues.

2.2.2.2. Les noms déadjectivaux

Une forme est déadjectivale si :
- elle dérive d'un adjectif non déverbal
- son comportement syntaxique est issu de celui de l'adjectif et attesté en surface.

"aptitude" dérive de "apte" et régit un complément en "à" comme "apte".

Ces noms seront marqués par la valeur DAJ de la variable DQ. Dans le lexique, on trouvera donc :

"aptitude" : "apte" DAJ

Le comportement syntaxique des formes déadjectivales se déduit directement de celui de l'adjectif.

"apte au service" devient "l'aptitude au service"

3. MISE EN OEUVRE DE LA STRATEGIE A PARTIR DES DONNEES LINGUISTIQUES

La stratégie d'analyse du syntagme nominal repose sur la détection des indicateurs de structure lexicaux contenus dans la chaîne à analyser. De ce fait, deux cas se présentent suivant qu'il y a présence ou non d'ISL dans le syntagme nominal. Tous deux débutent par la lecture de la chaîne afin d'y détecter l'existence d'ISL.

3.1. LA CHAINE CONTIENT DES ISL

3.1.1. La stratégie adoptée

Le fonctionnement du français nous permet d'édicter quelques principes respectés le plus souvent :

Principe 1 :
Les compléments régis par un ISL figurent à droite de celui-ci.

```
la production de matière sèche
* de matière sèche la production
```

Principe 2 :
Si un ISL admet d'après le lexique, la même préposition pour introduire deux compléments d'ordre différent, ces deux compléments n'apparaissent jamais simultanément, régis par une préposition identique.

```
Exemple:
  Le lexique donne :
        production [de, par] [de] [] [] [] []
Les expressions suivantes sont admises ou rejetées :
        la production de matière sèche
        la production du couvert végétal
        la production de matière sèche par le couvert végétal
        * la production de matière sèche du couvert végétal
```

Ainsi, si un ISL est suivi de deux prépositions identiques, seule la plus proche pourra introduire un SP gouverné par cet ISL.

Principe 3 :
Si d'après le lexique, un complément peut être régi par plus d'une préposition, dans la chaîne à analyser ce complément n'apparaîtra qu'une fois, introduit par l'une de ces prépositions. Donc l'occurrence d'une préposition exclut la présence des autres.

```
Exemple :
  Le lexique donne :
        monter [Ø] [Ø] [] [sur, dans, à2] [] []
Les expressions suivantes sont admises ou rejetées :
```

```
il monte dans le train
* il monte dans le train sur le quai
```

Principe 4:
Si un SP peut être régi par deux ISL d'une même chaîne, il sera rattaché au plus proche.

```
"l'étude sur le comportement du docteur X"
```

"étude" et "comportement" régissent tous deux un SP en "de" ; le plus proche du SP "du docteur X" est "comportement" ; donc ce SP sera régi par "comportement".

Ces principes ne sont pas posés de façon absolue. Il existe en effet quelques contre-exemples :

```
"l'explusion de l'Irak de la ligue arabe"
```

Ils sont cependant le plus souvent respectés pour des raisons de clarté. Leur application permet de ne construire qu'une seule structure syntaxique, la structure attestée, dans de nombreux cas. Nous présenterons sur un exemple les solutions obtenues par la seule analyse combinatoire, puis le résultat obtenu en associant prédiction et analyse combinatoire.

3.1.2. Un exemple

Si l'on confie à l'analyseur combinatoire la grammaire du paragraphe 1 (à l'exception des règles 12, 18, 19, et 31) et la chaîne ci-après :

```
"l'" [D (DEF, GRN, SNG)]
"étude" [F(NOM, DVB, FEM, SNG)]
"de"  [P]
"le" [D (DEF, MAS, SNG)]
"fonctionnement" [F( NOM, DVB, MAS, SNG)]
"de" [P]
"un" [D (NUM, MAS, SNG)]
"écosystème" [F (NOM, MAS, SNG)]
"prairial" [F (ADJ, MAS, SNG)]
```

il construit 9 structures différentes, alors que la langue n'en admet qu'une ! [annexe 1]. Les origines de cette multiplicité de solutions sont :

1 la grammaire conduit l'analyseur à construire deux structures concurrentes sur chaque F (DVB), l'une provenant de l'application de la règle N' --> N SP, l'autre de la séquence de règles N' --> N puis N --> N SP.

2 sur le mot "étude", on peut aussi appliquer la règle N' --> N SP SP puisqu'il est suivi de deux prépositions "de".

3 dans les cas où la chaîne "fonctionnement d'un écosystème prairial" se réécrit "N", alors les règles N --> N A'' et N --> N SP engendrent des structures différentes suivant l'ordre dans lequel elles sont appliquées.

La mise en oeuvre de la stratégie sur cette même chaîne s'opère ainsi :

1 "fonctionnement" est le déverbal le plus à droite dans la chaîne. Il est suivi de la préposition "de". Les principes 1 et 4 impliquent que si "fonctionnement" régit un syntagme prépositionnel en "de", ce ne peut être que celui-ci. Or, d'après le lexique, le mot "fonctionnement" est doté du comportement suivant :

fonctionnement [de] [] [] [] [] []

On peut donc prédire l'application de la règle N' --> N SP(de) sur la sous-chaîne qui débute en "fonctionnement".

2 "étude" est un déverbal suivi de deux prépositions en "de". La plus à droite est régie par "fonctionnement" (Principe 4) ; de plus un même déverbal ne peut régir deux syntagmes prépositionnels débutant par la même préposition (Principe 2). Donc "étude" ne peut régir qu'un seul syntagme prépositionnel en "de", le premier. Le lexique donne :

étude [de, par] [de] [] [] [] []

Ainsi, la règle N' -> N SP(de) s'applique sur la sous-chaîne qui débute en "étude".

Les prédictions opérées à partir des ISL de la chaîne conduisent à la construction de la seule structure syntaxique attestée et élimine donc les huit solutions parasites.

Remarque

Le modèle choisi ne nous permet pas de déterminer dans ce cas la nature du complément de "fonctionnement", C_0 ou C_1, saturé par ce syntagme en "de".

3.1.3. Les limites d'une telle stratégie

Bien que le plus souvent efficace, la prédiction à partir des ISL soulève trois problèmes ; le premier vient du fait qu'un ISL ne se comporte pas toujours comme tel, le second vient de la présence de compléments circonstanciels, compléments qui n'entrent pas dans la catégorie des compléments régis, et enfin, le troisième émane de la portée de la prédiction sur la chaîne et de son intégration dans le processus d'analyse.

3.1.3.1. La non-permanence du trait ISL

Considérons, parmi les ISL, les noms déverbaux. Le trait caractéristique de ces noms est qu'ils sont marqués par la valeur DVB de la variable DQ. Ce trait est dans notre cas une donnée lexicale (il pourrait aussi résulter d'une analyse morphologique dérivationnelle). Or, un même nom ne se comporte pas toujours comme un déverbal : c'est le cas en français de tous les termes qui désignent à la fois un processus et un produit, "analyse", "étude", "classement", "construction"... Ceux-ci sont ISL lorsqu'ils désignent le processus, ne le sont plus lorsqu'ils en désignent le produit. Pour rendre compte de ce double fonctionnement, il est possible de créer deux entrées lexicales par nom de ce type, traduisant chacune un

fonctionnement différent. En contrepartie, l'analyseur morphologique fournira une solution multiple pour chacune de ces formes, multiplicité délicate à réduire sans renseignement de nature syntaxique. Dans le cas particulier du corpus traité pour cette étude, le fonctionnement déverbal est de loin le plus couramment attesté. C'est pourquoi nous avons choisi de privilégier, pour chaque nom de ce type, son fonctionnement déverbal.

3.1.3.2. Les compléments circonstanciels

Nous poserons ici le problème de ces compléments parce qu'il est réel, mais ne visons pas à le résoudre. Ces compléments ne sont pas intégrés dans la typologie des compléments régis. Ils ne sauraient y trouver leur place étant donnée leur nature : ils ne sont pas appelés par un verbe ou par toute autre forme déverbale. Dans une phrase, ils jouent le rôle de modifieur de verbe au même titre que les adverbes.

Exemple :
"Ce processus conduit dans toutes les circonstances à une protection suffisante de l'hôte"
Le SP "dans toutes les circonstances" est un circonstanciel et se distingue du SP "à une protection suffisante de l'hôte" qui lui est régi par le verbe "conduit".

De façon similaire, les circonstanciels peuvent affecter un SN dont le centre est un déverbal.

1. "l'identification de résistances aux virus chez les plantes maraîchères"
2. "l'observation en France d'une race contournant ce gène"

Les circonstanciels sont ici "chez les plantes maraîchères" et "en France". La grammaire engendre dans le premier cas une solution parasite, en rattachant le SP circonstanciel au symbole N qui le précède, ici "virus" par l'application de la règle N --> N SP. Dans le deuxième cas, la même règle s'applique et rattache le circonstanciel "en France" à "observation". La solution construite est alors acceptable : c'est en effet, la plus juste que la grammaire puisse construire, mais elle a l'inconvénient d'assimiler la fonction du circonstanciel à celle d'un SP banal comme dans "la vie de la plante". Le traitement des circonstanciels nécessiterait l'ajout d'une règle spécifique. Cependant quelle que soit la règle adoptée, il est nécessaire de régulariser la chaîne à analyser avant d'entamer le processus d'analyse. Cette régularisation consiste à translater les circonstanciels auprès du déverbal qu'ils affectent. Pour ce faire, il faut être capable de détecter les circonstanciels. C'est pourquoi, nous laisserons ce problème ouvert.

3.1.3.3. La portée de la prédiction

La prédiction à partir des ISL consiste, après lecture de la chaîne et consultation du lexique, à préciser :

1. la règle qui sera appliquée
2. le symbole dans la chaîne à partir duquel cette règle s'applique.

La règle prédite a toujours pour symbole gauche N', symbole auquel la grammaire ne nous permet pas de rattacher des syntagmes adjectivaux antéposés. Dans une telle configuration, la prédiction conduit à un rejet du syntagme nominal par l'analyseur.

```
l' excellente      étude     de  Paul
D    F (ADJ)   F(NOM, DVB)   P  F(NOM)
```

Si l'on considère, d'après le paragraphe précédent, que "étude" est un déverbal, la prédiction impose l'application de la règle N' -> N SP à partir de ce mot. On construira donc :

N' [N [F, étude] SP [[P, de] [F,Paul]]]

structure qui ne peut permettre l'intégration de l'adjectif "excellente", puisque la règle N -> A' N s'applique au niveau N et non au niveau N'. C'est la raison pour laquelle l'analyse échoue.

Face à ce problème, on peut proposer la solution suivante. Il faut alors considérer que l'analyse d'un syntagme nominal du type "D F[ADJ] F[NOM, DVB] SP" ne peut s'opérer en une seule passe. En effet, la lecture de la chaîne ne permet pas de déterminer sur quel symbole débute la construction du noeud N'. Par contre, l'application préalable des règles :

```
A       -> F[ADJ]
A'      -> A
N[DVB]  -> F[NOM, DVB]
N[DVB]  -> A' N[DVB]
```

conduit à intégrer l'adjectif antéposé au noeud N et donc à prédire que l'application de la règle N' -> N SP débute sur le symbole F[ADJ].

Ainsi, nous retrouvons au niveau de l'analyse syntaxique, ce trait de dualité déjà présent dans les étapes antérieures : l'analyse syntaxique s'opère efficacement après prédiction, mais une prédiction juste peut requérir au préalable une analyse syntaxique partielle.

3.2. LA CHAINE DEPOURVUE D'ISL

Supposons maintenant que la chaîne à analyser ne contienne aucun ISL. Les symboles terminaux, éléments de syntagme nominal sont : F, D, W, P et parmi les règles qui peuvent s'appliquer pour en construire la structure, on trouve :

```
N -> N SP
N -> A' N
N -> N A''
```

Elles ont en commun d'être récursives à droite ou à gauche et de ce fait d'engendrer des structures multiples. Après étude du corpus, nous avons dégagé trois chaînes types génératrices de solutions multiples. Ces chaînes illustrent les sources principales d'ambiguïtés, sources qui peuvent se combiner au sein d'un même syntagme. Les chaînes types sont de

l'une des deux formes suivantes où X représente soit un syntagme adjectival (A'') soit un syntagme prépositionnel (SP).

type 1 : D F[ADJ] F[NOM] X

type 2 : D F[NOM] P D F[NOM] X

type 3 : F[NAN] F[NAN]

Remarque

1- La présence dans chacune des chaînes des déterminants (D) est facultative

2- Le respect de l'accord en genre et en nombre peut éliminer des solutions parasites, pour le type 2. Nous nous placerons dans le cas où le respect de cet accord n'apporte aucun supplément d'information quant à la structure.

3.2.1. Les chaînes de type 1 : D F[ADJ] F[NOM] X

A priori elles admettent deux analyses concurrentes [figure 4.6]. Le schéma 1.A est dit régulier à droite (structure de rateau) et s'oppose sur ce critère au schéma 1.B.

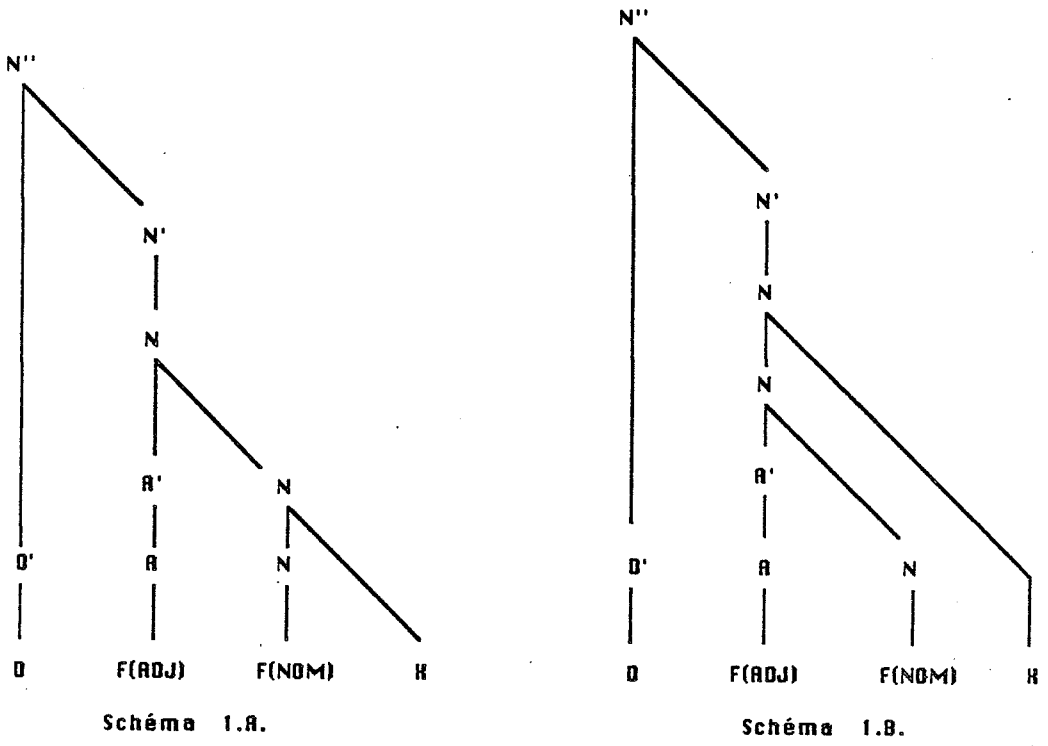


FIGURE 4.6

Etudions sur des syntagmes divers, l'impact de ces structures. Soient, par exemple, les syntagmes :

1.a. les différentes sources de pollen
1.b. une meilleure qualité d'écoute
1.c. les principales caractéristiques de ces résistances
1.d. une grande variété de substrats
1.e. les différents apports distaux
1.f. les nombreuses espèces annuelles
1.g. une jeune fille élégante

Rappelons que l'objectif visé par l'analyse syntaxique, est de détecter les syntagmes nominaux inclus dans un syntagme nominal. Les syntagmes nominaux, par ordre décroissant de complétude, sont ceux étiquetés par les noeuds N'', N' et N. Pour un syntagme nominal donné, la bonne structure est donc celle qui est attestée par la langue ; mais en cas d'ambiguïté on pourra se permettre de choisir celle qui donne le meilleur découpage en vue d'une indexation automatique.

En vertu de ces critères, analysons les syntagmes ci-dessus. En dehors de tout contexte, les syntagmes 1.a., 1.c., 1.d., 1.e. et 1.f. peuvent accepter les structures 1.A. et 1.B. ; il est linguistiquement impossible de choisir une solution plutôt qu'une autre. Ensuite, étudions la pertinence du découpage induit par chaque structure.

Les syntagmes nominaux de niveau 1 et 2 sont identiques pour l'une et l'autre structure. Les syntagmes de niveau 0 diffèrent :

Sur le syntagme 1.a., le schéma 1.A. conduit à des noeuds N qui recouvrent "pollen", "sources de pollen", "différentes sources de pollen". Le schéma 1.B. conduit au découpage "pollen", différentes sources" et "différentes sources de pollen". Dans un but d'indexation, il est plus pertinent d'obtenir les noeuds N donné par le schéma 1.A. puisque "sources de pollen" est plus riche en information que "différentes sources". Les syntagmes 1.c. et 1.d. conduisent à la même conclusion.

Sur le syntagme 1.e., le schéma 1.A. conduit à des noeuds N qui recouvrent "apports", "apports distaux" et "différents apports distaux". Le schéma 1.B. conduit au découpage, "apports", "différents apports" et "différents apports distaux". Là encore, le schéma 1.A. se révèle le plus pertinent. Le syntagme 1.f. conduit à la même conclusion. On préférera donc dans ce cas, le schéma 1.A.

Considérons maintenant le syntagme 1.b. ; il diffère des précédents car il n'admet qu'une analyse correcte, celle correspondant au schéma 1.A. En effet, "qualité d'écoute" est un mot composé. On rejette donc le découpage "meilleure qualité" et "écoute".

Reste enfin le syntagme 1.g. ; celui-ci ne supporte que la structure 1.B.. La cause de ce fonctionnement particulier est la présence du mot composé "jeune fille". Nous reviendrons ultérieurement sur le problème des mots composés car il se pose aussi pour les chaînes de type 2.

En conclusion, dans tous les cas, sauf 1.g., le schéma 1.A., dit régulier, est le meilleur pour ce type de chaîne.

3.2.2. Les chaînes de type 2 : D F[NOM] P D F[NOM] X

Elles admettent, elles aussi, deux structures différentes [figure 4.7].

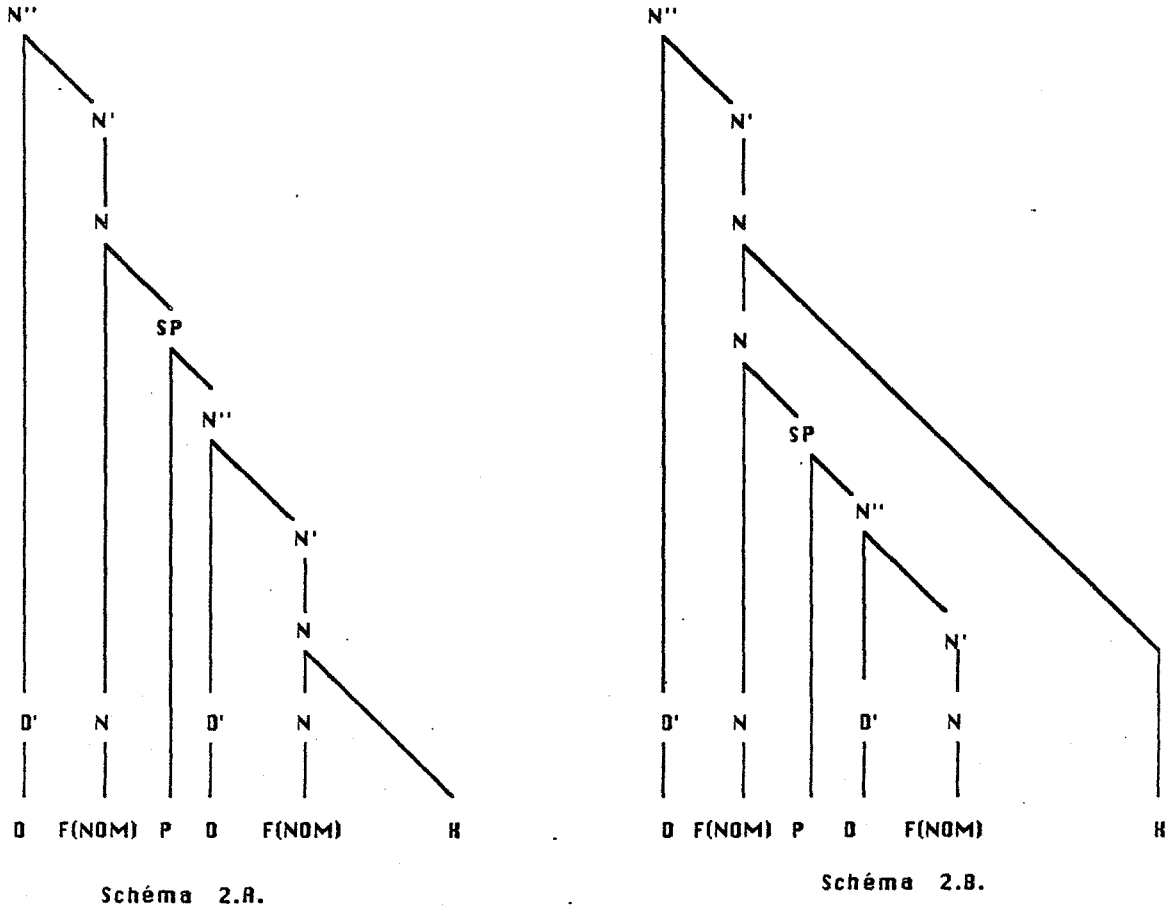


FIGURE 4.7.

Soient, à titre d'illustration les syntagmes :

2.a. un exemple de la pourriture noble
2.b. une résistance à contrôle monogénique
2.c. une sapinière à épinette rouge
2.d. une pomme de terre cultivée
2.e. l'agence de presse soviétique
2.f. l'épaisseur des parois des cellules
2.g. des segments de plantules de blé
2.h. les maladies à virus des hêtres

En suivant la même démarche que pour les chaînes de type 1, on constate que les syntagmes 2.a., 2.b., 2.c., 2.f. et 2.g. n'acceptent que la structure 2.A. Par contre, 2.e. accepte l'une et l'autre, et enfin 2.d. et 2.h. n'acceptent que la structure 2.B.. Cette constatation nous conduit à poser à nouveau le problème, crucial en indexation automatique, des mots composés. En effet, ici encore la structure attestée dépend de la présence de mots composés.

En résumé, l'analyse syntaxique combinatoire sur les chaînes de type 1 et 2, attribue :

1 soit une solution unique dans le cas où l'accord en genre et en nombre élimine une solution parasite.

```
"le niveau de vie moyen"
"les forêts de bouleaux blancs"
```

2 soit deux solutions ; dans ce cas, la structure régulière sera préférée sauf s'il y a présence de mots composés. Ceci nous amène à poser explicitement le problème des mots composés.

3.2.3. Le problème des mots composés

Un mot composé se définit classiquement sur des critères sémantiques comme étant un syntagme nominal complexe correspondant à une unité sémantique. Une telle définition n'est pas très rigoureuse. En effet, si tout le monde s'accorde à reconnaître "pomme de terre" comme un mot composé, il n'en est pas de même pour "cancer du sein". Dans ce cas, suivant la discipline considérée, et donc le champ sémantique, cette expression recouvrira soit une unité sémantique, soit deux : "cancer" et "sein". De plus, une telle définition n'a pas de réalité dans notre étude, puisque fondée sur des critères sémantiques alors que le problème des mots composés se pose ici sous des aspects syntaxiques. Les formes les plus courantes des mots composés sont N A, A N, N SP ou N N. Parmi celles-ci, seules les formes A N et N SP posent problème pour la grammaire, car elles rompent la régularité de la structure syntaxique comme :

```
"la belle mère de mon frère"
 D    A      N  P  D     N

"le champion de le monde de ski"
 D      N    P  D    N   P   N
```

Afin d'éviter la production de structures parasites, structures qui seraient validées si l'on privilégiait la structure la plus régulière, il est nécessaire de détecter la présence de mots composés avant analyse.

Ne disposant pas à l'heure actuelle de moyens linguistiques pour détecter les mots composés avant l'analyse syntaxique, la seule solution envisageable est de les stocker dans un lexique, et de substituer à chacune de leur occurrence dans les chaînes à analyser, chacun des symboles terminaux qui les composent par un seul.

```
[pomme, F] [de, D] [terre, F] devient [pomme de terre, F].
```

Cette solution a deux avantages :
- les mots composés, qui ont un fonctionnement syntaxique particulier, sont remplacés avant l'analyse par un seul symbole terminal. D'où une régularisation de la chaîne à analyser.
- après régularisation des mots composés, les chaînes dépourvues d'ISL se comportent le plus souvent de façon régulière. Ce résultat confirme le quatrième principe de KIMBALL, à savoir qu'un terminal est le plus souvent rattaché au noeud le plus bas dans la structure.

3.2.4. Les chaînes de type 3 : F[NAN] F[NAN]

La séquence F[NAN] F[NAN] est toujours source d'ambiguïtés, car chacun des F peut être soit N soit A. Ce cas est illustré par des syntagmes comme "artiste peintre", "couvert végétal", "un grand noir". On ne peut dans ce cas prédire quel est le symbole F qui joue le rôle de nom. Une solution proposée par A. Berrendonner consiste à substituer préalablement à l'analyse, toute séquence de ce type par un seul symbole F[NAN]. Ainsi :

```
"artiste" "peintre" est remplacé par le bloc "artiste peintre"
 F[NAN]     F[NAN]                                    F[NAN]
```

et de ce fait, l'amalgame se verrait attribuer lors de l'analyse le rôle nom ou adjectif requis par l'analyse.

Exemple :
Considérons le syntagme nominal "un artiste peintre", tout d'abord comme une séquence de deux F[NAN], puis comme un bloc F[NAN]. Essayons sur chacun d'eux, la grammaire abrégée donc sans tenir compte des règles 18 et 19.

Analyse de la séquence : l'[D] artiste[F(NAN)] peintre[F(NAN)]

Solution 1:

```
N'' {GRN SNG }
      D' {DEF GRN SNG }
            D   l'{DEF GRN SNG }
      N' {GRN SNG }
            N   {GRN SNG }
                  A' {GRN SNG }
                        A   {GRN SNG }
                              F   artiste {NAN GRN SNG }
                  N   {GRN SNG }
                        F   peintre {NAN GRN SNG }
```

Solution 2:

```
N'' {GRN SNG }
      D' {DEF GRN SNG }
            D   l'{DEF GRN SNG }
      N' {GRN SNG }
            N   {GRN SNG }
                  N   {GRN SNG }
                        F   artiste {NAN GRN SNG }
                  A' {GRN SNG }
                        A' {GRN SNG }
                              A   {GRN SNG }
                                    F   peintre {NAN GRN SNG }
```

Solution 3:

```
N'' {GRN SNG }
      N' {GRN SNG }
            D' {DEF GRN SNG }
                  D   l'{DEF GRN SNG }
            N' {GRN SNG }
                  N   {GRN SNG }
                        F   artiste {NAN GRN SNG }
      N' {GRN SNG }
            N' {GRN SNG }
                  N   {GRN SNG }
                        F   peintre {NAN GRN SNG }
```

Analyse de la séquence : l'[D] artiste peintre[F(NAN)]

```
N'' {GRN SNG }
      D' {DEF GRN SNG }
            D   l'{DEF GRN SNG }
      N' {GRN SNG }
            N   {GRN SNG }
                  F   artiste peintre {NAN GRN SNG }
```

Les chaînes de chacun de ces trois types ne représentent pas de façon exhaustive toutes les sources d'ambiguïtés, mais la plupart de celles rencontrées dans les corpus étudiés, engendrées par la grammaire sur les syntagmes nominaux. C'est pourquoi nous nous fonderons sur ces résultats pour affiner la stratégie d'analyse.

4. CONCLUSION : STRATEGIE D'ANALYSE ET ISL

La démarche adoptée pour étudier les structures syntaxiques des syntagmes nominaux pourrait laisser penser qu'il n'existe que deux types de SN : ceux contenant des ISL et les autres. En fait, les SN peuvent soit appartenir à l'un ou l'autre type, soit encore être mixtes. En effet, lorsqu'un SN contient des ISL, la prédiction porte soit sur une partie de la chaîne, et alors le reste de la chaîne devient une chaîne sans ISL ; soit encore un ISL peut prédire des noeuds qui peuvent être engendrés de plusieurs manières par une sous-chaîne.

"Les principaux critères de sélection"

Le mot "sélection" est, ici, un exemple de déverbal qui n'est pas en réalité un indicateur de structure. En effet, son comportement syntaxique n'est pas attesté dans la chaîne puisqu'il n'est suivi d'aucun SP. Dans ce cas, la prédiction est inopérante.

"Une réflexion sur les différentes stratégies d'emploi de les gènes"

"réflexion" et "emploi" sont des ISL ; ils régissent respectivement le SP en "sur" et le SP en "de" ; cependant la sous-chaîne "les différentes stratégies d'emploi" est un cas particulier d'une chaîne sans ISL de type 1, et de ce fait engendre deux structures concurrentes. On est donc ramené au cas type vu précédemment et la structure régulière sur cette sous-chaîne devra être préférée.

Ainsi, pour analyser un SN de type quelconque, il faut allier les deux aspects : prédiction à partir des ISL et régularité de la structure. Le problème qui se pose alors est de déterminer les rôles respectifs de la prédiction et de la régularité dans l'analyse d'une chaîne quelconque. Deux démarches s'affrontent.

La première consiste à formuler l'hypothèse que le fonctionnement de la langue est régulier sauf s'il y a présence d'ISL ou de mots composés. Cette démarche a pour avantages de limiter le nombre de solutions. En effet, le modèle régulier est contraignant et donc élimine un grand nombre d'ambiguïtés. De plus, l'analyse est peu coûteuse en temps et en espace. Par contre, elle présente des inconvénients. Ceux-ci résident dans le fait qu'elle rejette a priori toute chaîne non conforme au modèle régulier ; par conséquent, pour éviter le rejet d'une séquence correcte, on multipliera les exceptions à ce modèle, en créant artificiellement des ISL et des mots composés.

Exemple :
Sur la chaîne "niveau de vie moyen", la grammaire n'engendre qu'une structure ; cette structure est non régulière. Donc un analyseur conforme à l'hypothèse de régularité la rejettera, sauf si l'on déclare "niveau de vie" comme étant un mot composé.

Finalement, cette démarche guidée par l'hypothèse de régularité, s'apparente à celle de MARCUS dans la mesure où son efficacité est incontestable et où la solution obtenue, même unique n'est pas forcément celle attestée par la langue mais la plus conforme au modèle régulier.

Une deuxième démarche, partant des mêmes données, consiste à analyser la chaîne en tenant compte seulement de la prédiction des ISL. Puis on considère le résultat de l'analyseur. Trois cas de figures se présentent alors :

Soit la chaîne est rejetée : on en déduit que la chaîne est agrammaticale ou encore que les informations portées par les ISL, le cas échéant, sont inadaptées à la chaîne.

Soit la chaîne est acceptée et n'engendre qu'une seule structure. On est en droit de supposer que cette structure est attestée par la langue et donc que l'analyse est achevée.

Soit la chaîne est acceptée et engendre plus d'une structure. On considérera alors les seules sous-chaînes génératrices d'ambiguïtés. La plupart de ces sous-chaînes sont composées sur la base des types de chaînes sans ISL étudiés précédemment. Ces sous-chaînes seront analysées à nouveau, cette fois-ci sous l'hypothèse de régularité en tenant compte des mots composés. On élimine ainsi un grand nombre de solutions parasites.

Une variante consiste à opérer l'analyse en une seule passe, et à affecter en cas d'ambiguïté, le plus fort poids à la solution la plus proche du modèle régulier.

Cette seconde démarche est évidemment très coûteuse puisqu'elle construit toutes les solutions, même celles qui ne seront pas retenues. Mais elle a l'avantage d'être dirigée par les données contenues dans la chaîne, l'hypothèse de régularité n'intervenant que lorsque ces données sont épuisées.

Conscients de ses inconvénients, c'est cette deuxième démarche que nous adopterons car elle apparaît comme la plus riche d'un point de vue expérimental. C'est elle qui, en effet, nous obligera à appréhender le fonctionnement de la langue de façon progressive alors que la mise en oeuvre de la première démarche nous conduirait à tester la justesse de l'hypothèse de régularité.

Chapitre 5

MISE EN OEUVRE INFORMATIQUE DE LA STRATEGIE

Le problème posé maintenant est celui de la construction d'un algorithme d'analyse syntaxique pour le syntagme nominal. Au cours des chapitres précédents, nous avons été amenés à préciser de nombreux points utiles à l'élaboration de cet algorithme.

Ce fut d'abord le choix du modèle hors-contexte. Puis, parmi les analyseurs hors-contexte existants, l'algorithme d'Earley s'est avéré le mieux adapté, puisqu'à la fois le plus général et le plus performant [chapitre 2]. Malgré ses qualités, ni l'algorithme d'Earley, ni aucun de ses concurrents, ne peut à lui seul apporter une solution satisfaisante à l'analyse de la langue naturelle, puisque le fonctionnement de celle-ci n'est pas purement combinatoire, mais contraint. D'où la nécessité de soumettre l'analyseur à une stratégie d'analyse, la stratégie adaptée étant guidée par des informations linguistiques portées par la chaîne d'entrée [chapitre 3]. L'interprétation de ces informations oriente l'analyseur vers la construction d'un nombre de structures bien moindre que la seule analyse combinatoire, parmi lesquelles figurent les solutions attestées [chapitre 4].

L'analyseur issu d'une telle démarche est une adaptation de l'algorithme d'Earley. De façon plus précise, il s'agit de construire un algorithme d'analyse dont le moteur est l'automate défini par Earley, le fonctionnement de ce moteur étant soumis à la stratégie d'analyse.

Au cours de ce chapitre nous étudierons tout d'abord les données, puis l'adaptation de l'algorithme d'Earley. Les résultats feront l'objet du chapitre suivant.

1. LES DONNEES

Les données requises par un analyseur classique sont une grammaire et une chaîne à analyser. L'application visée nécessite des informations plus nombreuses. Parmi les données linguistiques, outre la grammaire, l'analyseur requiert l'identification des variables morphologiques et des valeurs qu'elles peuvent prendre, un lexique des ISL indiquant leur comportement syntaxique et un lexique de mots composés. Enfin, la chaîne à analyser doit être conforme à certaines normes que nous donnerons.

1.1. LES DONNEES LINGUISTIQUES

1.1.1. Le fichier GRAMMAIRE

La grammaire se présente comme un fichier de texte où chaque ligne contient une règle suivie de deux entiers : le premier traduisant la condition d'application de la règle, le deuxième la règle de transfert des valeurs de variables.

La première ligne contient obligatoirement la règle :

PHI --> S $ 0 0

où S est l'axiome de la grammaire, PHI un nouveau symbole non terminal et $ un nouveau symbole terminal marqueur de fin de chaîne.

1.1.1.1. Les conditions d'application

Rappelons que les conditions d'application limitent le domaine d'application d'une règle à certaines valeurs des variables morphologiques de la partie droite d'une règle. Pour chaque condition d'application, nous en donnerons l'énoncé suivi d'un exemple.

1. la variable NA du symbole droit de la règle doit avoir pour valeur NOM.
   N --> F(NOM)

2. la variable NA du symbole droit de la règle doit avoir pour valeur NAN.
   N --> F(NAN)

3. la variable NA du symbole droit de la règle doit avoir pour valeur ADJ.
   A --> F(ADJ)

4. le premier symbole de la partie droite de la règle doit être accompagné d'une valeur de la variable DQ : AGE, DVB ou DAJ.
   N' --> N (DQ ∈ {AGE, DVB, DAJ}) SP SP

5. les variables de genre GR et de nombre NB associées à chacun des deux symboles de la partie droite de la règle doivent avoir des valeurs compatibles avec l'accord en genre et en nombre.
   N --> N A''

6. la variable NU du premier symbole de la partie droite doit avoir pour valeur DEF, celle du deuxième symbole NUM.

D' --> D(DEF) D(NUM)

7. la variable VW du premier symbole de la partie droite doit avoir la valeur AAJ.

A' --> W(AAJ) A

8. la variable VW du symbole de la partie droite doit avoir pour valeur QUA.

K --> W(QUA)

9. la variable NU du premier symbole de la partie droite doit avoir pour valeur NUM ou NNU. De plus, les valeurs des variables GR et NB attachées à ces symboles doivent respecter l'accord en genre et en nombre.

K --> D(NUM ou NNU) N

10. le symbole P, premier de la partie droite de la règle, doit correspondre à la préposition "de"; le symbole D, deuxième de la partie droite, à l'un des déterminants : "le", "la" ou "les".

D' --> "de" / "le", "la" ou "les"/

11. la variable NU du premier symbole de la partie droite doit avoir pour valeur DEF. De plus, les valeurs des variables GR et NB attachées à chacun des deux symboles de la partie droite doivent respecter l'accord en genre et en nombre.

K --> D(DEF) A'

12. le symbole P, deuxième de la partie droite de la règle, doit correspondre à la préposition "de".

N'' --> K "de" N''

1.1.1.2. Les règles de transfert

Les règles de transfert régissent le passage des valeurs des variables morphologiques de la partie droite sur la partie gauche. Comme les conditions d'application, elles sont représentées formellement par des entiers de la manière suivante :

1. transfert de toutes les variables du symbole de la partie droite de la règle, sur le symbole de la partie gauche, sauf NA.

N --> F(NOM)

2. transfert de toutes les variables du symbole de la partie droite de la règle, sur le symbole de la partie gauche, sauf DQ, devenue inutile.

N' --> N (DVB) SP SP

3. transfert des valeurs des variables GR et NB après accord et des autres variables du deuxième symbole de la partie droite.

N --> A' N

4. transfert des valeurs des variables GR et NB après accord et des autres variables du premier symbole de la partie droite.

N --> N A''

5. aucun transfert de variables.

SP --> P N''

6. transfert de toutes les variables du symbole de la partie droite

N' --> N

7. transfert de toutes les variables du deuxième symbole de la partie droite.

A' --> W(AAJ) A

8. transfert de toutes les variables du premier symbole de la partie droite.

A'' --> A' SP SP

9. transfert des valeurs des variables GR et NB après accord et des autres variables du premier symbole.

D' --> D(DEF) D(NUM)

10. transfert des valeurs des variables GR et NB après accord, de la valeur de la variable NU du premier symbole et des autres variables du deuxième symbole.

N'' --> D' N'

1.1.1.3. Exemple de fichier GRAMMAIRE

```
PHI -> N'' FIN 0 0
N -> F 1 1
N -> F 2 1
A -> F 3 1
A -> F 2 1
N -> N SP 0 8
N -> A' N 5 3
N -> N A'' 5 4
A'' -> A' SP 4 2
A'' -> A' 0 2
SP -> P N'' 0 5
A' -> A 0 6
A' -> W A 0 7
N' -> N SP 4 2
N' -> N 0 6
N'' -> N' 0 6
N'' -> D' N' 5 10
D' -> D 0 2
N' -> N SP SP 4 2
A'' -> A' SP SP 4 2
```

```
N'' -> N'' N'' 5 3
```

1.1.2. Le fichier VARIABLE

Le fichier VARIABLE est un fichier texte dont chaque ligne est constituée de l'identificateur d'une variable (2 caractères), suivi des identificateurs des valeurs que peut prendre cette variable. Chaque identificateur de valeurs est représenté par 3 caractères. Une même valeur ne peut être attribuée à deux variables différentes.

Exemple de fichier VARIABLE

```
NA NOM NAN ADJ
GR MAS FEM GRN
NB SNG PLU NBN
DQ DVB DAJ AGE
PA PPR PPA
VB INF FIN
VX AUX ORD
NN PRP COM PRO
NU DEF NUM NNU
VW AVJ TAM ANA QUA
```

1.1.3. Le lexique des ISL

Chacun des mots de la catégorie F muni d'une valeur de la variable DQ peuvent être des ISL. A ce titre, ils sont enregistrés dans un fichier texte, LEXIQUE_ISL, dont chaque ligne est composée de la base, c'est-à-dire de l'entrée lexicale, suivie de six paires de crochets, chacune contenant la liste des prépositions acceptées par le complément régi d'ordre i-1.

Exemple de LEXIQUE_ISL

```
fonctionnement [] [de] [] [] [] []
étude [de , par] [de] [] [] [] []
production [de, par] [de] [] [] [] []
résistance [de] [] [à] [] [] []
infection [de, par] [de] [] [] [] []
```

1.1.4. Le lexique des mots composés

Le lexique des mots composés est un fichier texte dont chaque ligne contient les bases et catégories morphologiques des composants, suivies d'un séparateur "$" et de l'amalgame avec sa catégorie morphologique.

Exemple :

```
"pomme" [F] "de"[P] "terre"[F] $   "pomme de terre" [F]
```

Remarque
L'information contenue dans ces deux lexiques est spécifique à l'analyse syntaxique; d'où la création au moment opportun de ces lexiques. Cependant, on doit envisager d'intégrer ces renseignements dans un dictionnaire qui contiendrait toutes les informations nécessaires à l'ensemble des opérations, analyse syntaxique incluse.

1.2. LE FICHIER GENERATEUR

Ce fichier texte contient les chaînes à analyser. La fin d'une chaîne est matérialisée par le caractère "$", marqueur de fin de chaîne. Une chaîne est composées d'unités lexicales. Chaque unité correspond à une ligne du fichier. Une unité lexicale se compose de :

- la forme : directement issue du texte, elle est surtout utile à l'édition des résultats.
- la base : c'est elle qui nous donnera accès aux différents lexiques.
- la catégorie morphologique : c'est la donnée essentielle, puisqu'elle joue le rôle de symbole terminal.
- éventuellement, la liste des valeurs des variables, ces valeurs étant utiles à la vérification des conditions d'application.

Le format sous lequel doivent apparaître ces données est le suivant :
FORME "BASE" [CAT (VAL1,...,VALN)]

Exemple de fichier GENERATEUR

```
chaque "chaque" [D(DEF, MAS, SNG)]
laboratoire "laboratoire" [F(NOM, MAS, SNG)]
travaillant "travaillant" [F(DVB, ADJ, GRN, NBN, PPR)]
sur "sur" [P]
un "un" [D(NNU, MAS, SNG)]
meçanisme "meçanisme" [F(NOM, MAS, SNG)]
s'inte'grant "s'inte'grant" [F(DVB, ADJ, GRN, NBN, PPR)]
dans "dans" [P]
le "le" [D(DEF, MAS, SNG)]
projet "projet" [F(DVB, NOM, MAS, SNG)]
de "de" [P]
mode`le "mode`le" [F(DVB, NOM, MAS, SNG)]$
les "le" [D(DEF, GRN, PLU)]
proble`mes "proble`me" [F(NOM, MAS, PLU)]
pose's "pose¨ [F(ADJ, MAS, PLU, DVB, PPA)]
par "par" [P]
le "le" [D(DEF, MAS, SNG)]
fonctionnement "fonctionnement" [F(NOM, DVB, MAS, SNG)]
de "de" [P]
une "un" [D(NNU, FEM, SNG)]
e'quipe "e'quipe" [F(NOM, FEM, SNG)]
```

```
plurilocaliseé "plurilocalise¨ [F(ADJ, FEM, SNG)]$
```

L'ensemble de ces fichiers constitue les données de l'analyseur. Un analyseur classique de type combinatoire n'utilise que les fichiers de base, GRAMMAIRE et GENERATEUR. La mise en oeuvre de la stratégie nécessite des données supplémentaires contenues dans les fichiers VARIABLE, LEXIQUE_ISL et MOT_COMPOSE.

2. L'ANALYSEUR

L'algorithme d'analyse qui répond aux spécifications linguistiques définies au chapitre précédent, résulte d'une adaptation de l'algorithme d'Earley.

L'algorithme d'Earley, exposé en détail au chapitre 2, a été retenu car il est parmi les analyseurs hors-contexte le mieux adapté à notre problème. Rappelons qu'il construit en une seule passe toutes les structures syntaxiques engendrées sur une chaîne par une grammaire hors-contexte quelconque. Cet algorithme est complété par un module de parcours de la polystructure construite, pour en donner toutes les structures incluses.

Les adaptations apportées à cet algorithme interviennent soit avant la phase d'analyse proprement dite, soit au cours de l'analyse. En effet, dans une phase préalable, il est nécessaire de prendre en compte toutes les données linguistiques pour les transformer en informations directement utilisables par l'analyseur. Puis, il faut modifier l'analyseur pour qu'il prenne en compte ces informations.

2.1. LES TRAITEMENTS PREALABLES A L'ANALYSE.

Ces traitements se décomposent en modules qui ont en commun de lire les données sur chacun des fichiers et de les organiser dans des structures de données en fonction des besoins de l'analyseur.

2.1.1. GRAM : traitement du fichier GRAMMAIRE

Données : le fichier GRAMMAIRE

Résultats :

1. une table des symboles TABLE_SYMBOLE qui contient tous les symboles de la grammaire et la mention de leur caractère terminal ou non terminal. Un caractère est terminal s'il n'apparaît jamais en partie gauche de règle, et non terminal s'il apparaît au moins une fois en partie gauche de règle.

2. un tableau ENSEMBLE_REGLE indicé par les symboles, qui contient la liste des règles qui ont ce symbole en partie gauche.

3. un tableau TABLE_REGLE indicé par les numéros de règle; le numéro 0 est affecté à la première règle du fichier GRAMMAIRE, le numéro 1 à la seconde... Ce tableau contient le symbole gauche de la règle et l'adresse mémoire où réside la règle.

4. un tableau PREM indicé par les symboles de la grammaire qui contient pour chacun d'eux l'ensemble des terminaux qui peuvent figurer en tête de la réécriture de ce symbole.

2.1.2. VABVAL : traitement du fichier VARIABLE

Données : le fichier VARIABLE

Résultats :

1. le tableau TABVAB, table des identificateurs de variables

2. le tableau TABVAL, table des identificateurs de valeurs.

3. la matrice booléenne TABVABVAL, dont les lignes sont indicées par les variables, les colonnes par les valeurs, et définie par :

TABVABVAL [I, J] = 1 si la variable I peut prendre la valeur J.
TABVABVAL [I, J] = 0 sinon.

2.1.3. LECTURE_CHAINE : lecture de la chaîne

Données : le fichier GENERATEUR

Résultats : construit pour chaque chaîne à analyser, un tableau CHAINE indicé par un entier qui contient chacune des unités lexicales à analyser, et ajoute en fin de chaîne deux unités lexicales contenant le symbole terminal marqueur de fin de chaîne.

2.1.4. PREDICTION_ISL : prédiction à partir des ISL

Données : le tableau CHAINE et le fichier LEXIQUE_ISL

Traitements :

1. recherche des ISL et des prépositions dans CHAINE

2. On considère chacun des ISL de la chaîne en commençant par la droite de la chaîne. Pour chacun d'eux : repérer les prépositions candidates à la rection, en les considérant de gauche à droite.

Une préposition est candidate :
- si elle située à la droite de l'ISL considéré.
- si elle n'est pas régie par un ISL situé plus à droite, donc déjà traité.
- si elle n'est pas identique à une préposition candidate pour cet ISL, donc à une préposition candidate située entre l'ISL et la préposition considérée.

Après sélection, pour un ISL, des prépositions candidates dans la chaîne, on consulte le lexique ISL. Une préposition candidate est effectivement régie si elle est requise par cet ISL pour régir un complément non encore saturé.

On cherche dans la grammaire la règle qui rend compte de la rection par l'ISL considéré, des syntagmes prépositionnels introduits par les prépositions régies, et l'on prédit que cette règle s'applique à partir de cet ISL.

Résultats : un tableau ENS_PRED indicé par un entier i qui contient la liste des numéros de règles prédites à partir des ISL, sur la sous-chaîne qui débute à la ième unité lexicale de la chaîne.

2.1.5. MOT_COMPO : substitution des mots composés

Données : CHAINE et le fichier MOT_COMPOSE

Résultat: un nouveau tableau CHAINE construit à partir du tableau CHAINE initial en substituant toutes les unités lexicales constituant un mot composé du lexique par une seule.

2.2. MODIFICATIONS DE L'ANALYSEUR

On dispose maintenant de toutes les informations nécessaires pour adapter l'algorithme d'Earley à la stratégie d'analyse. Comme l'algorithme se décompose en trois opérations qui sont la dérivation, le balayage et la complétion, ces modifications s'intégreront donc au sein de ces opérations.

2.2.1. Conditions d'application et règles de transfert.

Nous avons vu que la condition d'application d'une règle restreint l'application de cette règle à certaines valeurs des variables morphologiques attachées aux symboles de la partie droite. Pour pouvoir décider si une condition d'application est respectée, il faut connaître les valeurs de ces variables. Or ces valeurs proviennent initialement de la chaîne et donc on ne peut en avoir connaissance qu'après reconnaissance de tous les symboles de la partie droite d'une règle. Pour cette raison, la condition d'application d'une règle doit être vérifiée sur un état final. Or, l'opération qui traite les états finals est l'opération de complétion. De ce fait, quand l'analyseur arrive sur un état final, <r,k,j,a>, il vérifie que les variables de la partie droite respectent les conditions d'application de la règle r, avant d'entreprendre l'opération de complétion proprement dite. Dans l'affirmative, on transfère, sur le symbole, les valeurs de variables requises par la règle de transfert de r et l'opération de complétion peut se poursuivre sur cet état. Sinon, cet état n'est pas soumis à la complétion, mais tout simplement abandonné.

Algorithme

La procédure complétion est appelée pour traiter un état final <r,k,j,a,P> de V[i]. En tête de la procédure complétion on effectue le traitement suivant :

1 connaissant r le numéro de la règle, on cherche les valeurs de ca, la condition d'application et de rt, la règle de transfert, attachées à cette règle.

2 connaissant P, racine de la polystructure associée à cet état final, on a accès aux valeurs des variables associées à ses fils.

3 on vérifie que les valeurs respectent la condition d'application ca :
- SI OUI :
  transférer sur le symbole gauche, donc sur la racine P, les valeurs requises par rt.
  poursuivre la procédure complétion
- SI NON :
  sortir de la procédure complétion

2.2.2. prédiction à partir des ISG.

Au cours du chapitre précédent, nous avons présenté deux types d'indicateurs de structure : grammaticaux (ISG) et lexicaux (ISL). Les ISG ne sont pas donnés par un lexique, mais par la grammaire. En effet, ce sont des symboles terminaux qui permettent de prédire le noeud dominant, et les constituants de ce noeud. Parmi les ISG définis, ceux qui entrent dans la composition des syntagmes nominaux sont les prépositions (P) et les déterminants (D). Si, de façon précise, l'on se réfère à la grammaire, il s'agit lorsque l'on rencontre un symbole P, en position i, dans la chaîne d'entrée, de construire d'après la règle (la seule ayant P en partie droite) SP --> P N'', le noeud dominant, SP, qui débute en x[i] et de prédire un noeud N'' qui débutera en x[i+1]. Cela revient à modifier l'algorithme d'Earley de la façon suivante : au cours de l'étape i, si x[i] = P, les seules règles sur lesquelles pourra porter l'opération de dérivation sont celles de la forme A --> C[1]...C[l(r)] telles que P soit élément de Prem(C[1]), car le caractère à reconnaître à l'étape i est P.

On pourrait faire le même raisonnement pour D, autre indicateur de structure du syntagme nominal. Dans ce cas, si x[i] = D, la règle N'' --> D' N' permet de construire un noeud dominant, N'', et de prédire ses deux noeuds constituants : D' qui débute en x[i] et N' qui débutera dans la chaîne à l'issue de D'. Cette règle engendrera donc des états dans V[i]. Puis, les règles D' --> D D ou D' --> D engendreront des états de V[i], états qui conduiront à la reconnaissance du symbole x[i] = D. Une fois que la sous-chaîne engendrée par D' sera reconnue, alors le processus de reconnaissance de la sous-chaîne engendrée par N' pourra débuter. On voit ici, que les seuls états générés par la lecture de x[i] dans V[i] le sont par les règles de R telles que D soit préfixe de la sous-chaîne engendrée par la partie droite. Donc, l'opération de dérivation ne devrait s'appliquer qu'aux seules règles qui peuvent engendrer une sous-chaîne débutant par x[i], après lecture de ce caractère. Ce procédé limite les états engendrés à l'étape i aux seuls compatibles avec le caractère x[i] si celui-ci est un indicateur de structure grammatical.

Le fait de limiter l'application de ce procédé aux seuls indicateurs de structure, rend l'algorithme dépendant du langage traité. Pour cette raison, il nous a semblé plus cohérent de l'étendre à tous les symboles terminaux d'une grammaire, car tout symbole terminal lu en avance dans la chaîne apporte de l'information sur la structure à construire. Cependant, certains en apportent plus que d'autres : c'est le cas des symboles P et D, car ils limitent le nombre de règles qui peuvent s'appliquer à une ou deux. Ceci revient à transformer l'algorithme d'Earley pour tenir compte dans les états engendrés à une étape i, du ième symbole de la chaîne. L'on est conduit ainsi, par le biais des ISG, à construire un analyseur semblable à celui proposé par BOUCKAERT, PIROTTE et SNELLING [1975]. Ceux-ci montrent que, en modifiant l'algorithme d'Earley, de façon à prendre en compte le caractère à analyser dans la chaîne, on en améliorait les performances. En effet, le nombre d'états créés à chaque étape est bien moindre, et de ce fait l'espace nécessaire à leur stockage et le temps requis pour examiner l'ensemble des états bien inférieurs. Cependant, la complexité globale de l'algorithme reste dans le pire des cas de l'ordre de n^3.

Algorithme :

Dérivation à partir de l'état <r,k,j,a> de V[i], c'est-à-dire de l'état [A --> B[1]...B[k]*B[k+1] B[k+2]...B[l(r)], j, a] de V[i].

Pour toutes les règles r' de R, de la forme B[k+1] --> C[1]...C[l(r)] telles que x[i] soit élément de Prem(C[1]), et pour b parcourant Prem(B[k+2]) ajouter <r',0,i,b> à V[i].

Cette modification répond aux exigences du premier principe de la stratégie. Elle met en jeu un déterminisme local, qui,à la lecture du caractère suivant dans la chaîne d'entrée, n'engendre que les états compatibles avec ce caractère.

modifications annexes

La lecture en avance d'un caractère dans la chaîne d'entrée, effectuée dans le cadre de l'opération de dérivation, peut être généralisée aux deux autres opérations de base balayage et complétion. On obtient alors exactement l'automate d'analyse proposé par BOUCKAERT, PIROTTE et SNELLING [1975], et qu'ils nomment M(1,1).

Balayage à partir d'un état <r,k,j,a> de V[i], soit [A --> B[1]...B[k]*B[k+1]B[k+2]...B[l(r)], j, a]

Si x[i] = B[k+1] et si, suivant la valeur de k , x[i+1] répond à l'une des conditions suivantes :

(1) si k+1 = l(r), a = x[i+1]

(2) si k+1 < l(r), a est élément de Prem(B[k+2])

alors ajouter <r,k+1,j,a> à V[i+1]

Complétion à partir de l'état <r,k,j,a> de V[i] où k=l(r), soit [A --> B[1]...B[l(r)]*, j, a] de V[i].

Soit <q,k',j',a'> un état de V[j] candidat à la complétion. Alors, si le (k'+1)ème symbole de la règle q est identique à la partie gauche de r et si, suivant la valeur de k', x[i+1] répond à l'une des conditions suivantes :

(1) k'+1 = l(q), x[i+1] = a

(2) k'+1 < l(q), x[i+1] est préfixe du (k'+2)ème symbole de q

alors ajouter l'état <q,k'+1,j',a'> à V[i].

2.2.3. Prise en compte des ISL.

Dans la phase initiale de traitement, le module PREDICTION_ISL construit un vecteur ENS_PRED indicé par un entier correspondant au numéro d'ordre dans la chaîne de l'ISL concerné. Chaque élément de ce vecteur contient la liste des numéros de règles qui doivent être appliquées à cette étape. Or, dans l'algorithme d'Earley, les règles qui peuvent être appliquées à une étape i sont choisies par l'opération de dérivation. Donc les informations contenues dans ENS_PRED seront utilisées lors de la dérivation. Cette opération consiste à partir d'un non terminal à construire tous les états dictés par les règles de la grammaire dont ce non terminal est partie gauche. Or l'information apportée par un ISL permet de ne pas prendre en compte toutes les règles de réécriture de ce non-terminal mais seulement celles de ENS_PRED[i]. D'où la modification de l'opération de dérivation :

Algorithme

Dérivation à partir de l'état <r,k,j,a> de V[i], c'est-à-dire de l'état [A --> B[1]...B[k]*B[k+1] B[k+2]...B[l(r)], j, a] de V[i].

Si le symbole B[k+1] est partie gauche d'une règle r' de ENS_PRED[i], alors les seuls états engendrés seront les états <r',0,i,b> où b parcourt Prem(B[k+2]).

Sinon, l'opération de dérivation s'opère normalement.

2.2.4. La régularité de la structure

2.2.4.1. Notion intuitive de régularité

Rappelons tout d'abord ce que nous entendons par structure régulière, car nous restons ici dans le cadre du modèle hors-contexte. Il ne s'agit donc pas d'une structure régulière, au sens de la théorie des langages, c'est-à-dire engendrée par une grammaire régulière. Notre grammaire est bien hors-contexte, mais comme elle est de plus ambiguë, l'analyseur, bien qu'il soit soumis à la prédiction des ISL, construit dans certains cas plus d'une structure. Ces cas sont illustrés par les chaînes de type 1 et de type 2 vues au chapitre 4. Or, nous avons montré que dans le cas où plusieurs analyses étaient possibles, celle dont la structure était la plus régulière à droite, était le plus souvent attestée par la langue.

Intuitivement, la structure la plus régulière résulte du fait que, lorsqu'un noeud nouvellement construit peut être attaché à plusieurs noeuds de la structure déjà construite, ce rattachement s'opère sur le noeud de niveau inférieur.

Exemple :

Soit la chaîne :

"les différentes variétés de résistance"

D F(ADJ) F(NOM) P F(NOM) L'analyseur construit d'abord la sous-structure de la figure 5.1, donc deux noeuds étiquetés N.

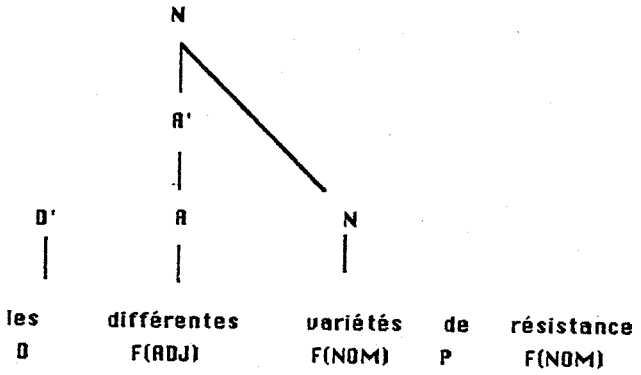


figure 5.1

Le noeud de niveau inférieur résulte de l'application de la règle N --> F(NOM), le noeud de niveau supérieur est construit sur le noeud N précédent par la règle N --> A' N. L'analyse se poursuit par la construction d'un noeud SP. Celui-ci, en vertu de la règle N --> N SP peut être rattaché à chacun des noeuds N déjà construits [figure 5.2].

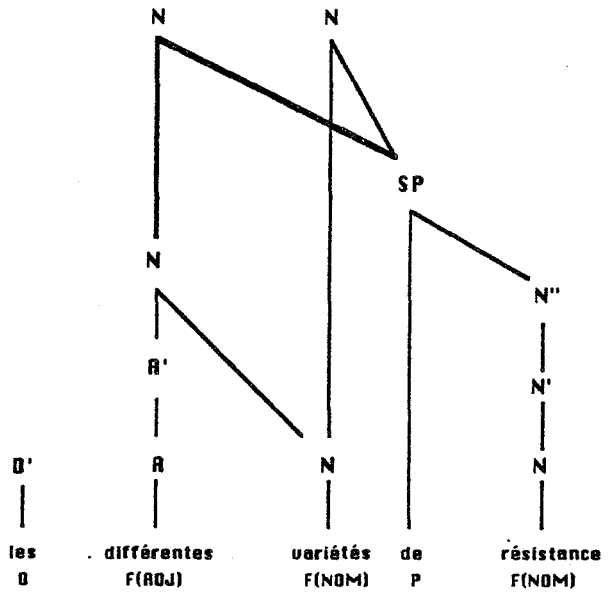


figure 5.2

D'où la construction de deux structures concurrentes et équivalentes aux yeux de la grammaire. On est alors face à une ambiguïté de nature syntaxique. Si par contre, on privilégie, en cours d'analyse, avec des moyens que l'on verra ultérieurement, la construction de la structure la plus régulière, les structures construites ne sont plus équivalentes, car elles se distinguent sur le critère de la régularité.

Rappelons encore que la contrainte de régularité peut s'exercer de deux façons différentes sur l'analyseur. La première consiste à ne construire qu'une seule structure, la plus régulière, et s'oppose à la seconde qui construit toutes les structures et qui, au cours de leur construction, enregistre leur degré de régularité. C'est cette seconde démarche que nous tenterons de réaliser.

2.2.4.2. Formalisation

Nous cherchons maintenant à quantifier la régularité des structures engendrées sur les chaînes de type 1 et de type 2 par la grammaire. Analysons tout d'abord plus précisément l'origine de cette double solution. Au cours de l'analyse de ces chaînes, le noeud X (A'' ou SP) peut être rattaché à deux noeuds N de niveau différent, par le biais des règles :

N --> N SP

N --> N A''

Ces deux règles ont en commun d'être récursives à gauche; elles autorisent donc le rattachement du noeud X à tout noeud N préalablement construit. C'est donc leur application qui perturbe la régularité à droite de la structure et non pas l'application de la règle N --> A' N. Celle-ci est en effet récursive à droite et engendre des structures naturellement régulières à droite. La contrainte de régularité devra donc s'exercer lors de l'application des règles récursivesà gauche.

Puisque le caractère de régularité est lié à la structure et que cette structure est construite, au sein de l'algorithme d'Earley, au cours de la complétion, c'est donc cette opération qu'il nous faut d'abord modifier.

Le rattachement d'un noeud X, nouvellement construit, à la structure existante, s'opère lorsque l'état courant <r,k,j,a> de V[i] est final et de la forme [X --> B[1]...B[l(r)]*, j, a]. On cherche donc dans V[j] les états candidats <q,k',j',a'>. Ils sont de la forme [N --> N*X, j', a']. Leur nombre est égal au nombre d'entiers j' définis par :

N engendre x[j'+1]...x[j] avec 0 < j' <j.

S'il existe plusieurs j' répondant à cette condition, cela signifie que le noeud X peut être attaché à plus d'un noeud N, chacun étant associé à une valeur de j'.

Rapport entre niveau du noeud N et valeur de j' :

Soient j'_1 et j'_2 tels que $0 < j'_1 < j'_2 < j$; cela signifie qu'il existe un noeud N_1 qui engendre la sous-chaîne $x[j'_1+1]...x[j]$ et qu'il existe un noeud N_2 qui engendre $x[j'_2+1]...x[j]$. Comme $j'_1 < j'_2$, la sous-chaîne $x[j'_2+1]...x[j]$ est incluse dans la sous-chaîne $x[j'_1+1]...x[j]$. De plus, l'on sait que N_2 engendre la plus petite sous-chaîne et N_1 la plus grande, donc le noeud N_1 engendre le noeud N_2. Le noeud N_1 est donc de niveau supérieur au noeud N_2.

De ce qui précède, on déduit que rattacher un noeud X engendrant la sous-chaîne x[j+1]...x[i] au noeud N de niveau inférieur, revient à choisir parmi les noeuds N celui qui engendre la chaîne x[j']...x[j] avec le j' le plus grand.

Puisque l'on ne veut pas éliminer mais classer les solutions en fonction de leur degré de régularité, il faut les pondérer. Nous choisirons un poids p, compris entre 0 et 1, calculé de la manière suivante :

$$p = (j - j') / n$$

où n est la longueur de la chaîne, et où j et j' sont des points dans la chaîne, vérifiant $0 < j' < j < n$. Ce poids p calculé lors de l'opération de complétion est affecté à la racine de la polystructure associée à l'état construit à partir de l'état courant et de l'état candidat.

modification de l'opération de complétion

Complétion à partir de l'état <r,k,j,a> de V[i] où k=l(r), soit [X --> B[1]...B[l(r)]*, j, a] de V[i].

Soient <q,k',j',a'> les états candidats de V[j].

Si parmi eux, tous les numéros de règle q sont différents, affecter un poids nul aux états construits.

Si, par contre, il existe plusieurs états candidats, ayant :
même numéro de règle q,
même point dans la règle k',
même symbole attendu a' mais dont les points dans la chaîne j' diffèrent, alors affecter à chacun des états construits un poids p = (j - j') / n.

D'après son mode de calcul, le poids le plus faible affecte le rattachement du noeud courant au noeud candidat le plus bas dans la structure, alors que le poids le plus fort traduit un rattachement au noeud candidat le plus élevé, et donc la structure affectée du plus faible poids est la plus régulière à droite. A ce niveau, cette information n'est pas exploitable; elle marque en effet des états non liés entre eux. Ce n'est qu'au moment où ces états concurrents donneront naissance à une ambiguïté syntaxique, c'est-à-dire, lorsqu'un même symbole engendrera, par le biais des états concurrents, plus d'une structure différente sur la même sous-chaîne, que l'on pourra comparer la régularité des structures obtenues. D'où la nécessité d'intégrer ces poids au cours de la construction de la structure.

La construction de la polystructure se décompose en trois opérations [cf. chapitre 2] : juxtaposition, enracinement et ajout d'un alternant. Il s'agit d'étudier les actions opérées sur les poids par chacune de ces procédures.

juxtaposition : Cette opération consiste à concaténer au sens du lien frère, deux polystructures, chacune pouvant être affectée d'un poids . Dans ce cas, on calcule la somme de ces poids, elle-même inférieure à 1. Cette somme sera affectée au noeud père.

enracinement : On affecte au noeud père le poids égal à la somme de ses fils, poids obtenu lors de l'opération de juxtaposition.

ajout d'un alternant : Les noeuds, en relation d'alternance, issus d'états concurrents sont affectés d'un poids. Puisque la structure recherchée est celle de poids moindre, on affectera au noeud fictif représentant de l'ensemble des noeuds en relation d'alternance, le poids inférieur.

Lors de la construction de la structure, on pondère chacun des états de l'analyseur et donc la polystructure associée. Tout état non affecté d'un poids par calcul ou par transfert se verra attribuer par défaut un poids nul. A l'issue de cette étape, il reste à parcourir la polystructure en exploitant la pondération. La polystructure se parcourt de la racine vers les feuilles [cf. chapitre 2], et chaque fois que l'on rencontre un noeud avec alternants, l'on constitue une liste ordonnée de ces alternants, liste dont chaque élément sera considéré

comme racine de polystructure, et fera l'objet de la procédure parcours. L'ordre qui régit cette liste est jusqu'alors, l'ordre de création des alternants (chaque nouvel alternant est inséré en queue de liste). Il suffit de modifier l'ordre de cette liste en tenant compte des poids, et donc de parcourir les alternants en commençant par celui de poids inférieur. Ce choix s'opérera récursivement et donc la première structure imprimée sera celle dont tous les sous-arbres sont de poids inférieur, la suivante celle dont tous les sous-arbres, sauf le plus bas, sont de poids inférieur... On obtient ainsi un ordre partiel sur les structures syntaxiques incluses dans une polystructure.

Grâce à ce système de pondération, on peut dégager parmi l'ensemble des structures attestées par la grammaire, la structure la plus régulière à droite. Il nous reste à prouver que cette structure est aussi linguistiquement correcte, dans la plupart des cas, et ensuite, que parmi les structures attestées par la grammaire et par la langue, elle est la plus pertinente pour l'objectif visé : l'indexation automatique.

Conclusion

L'analyseur que nous avons décrit tout au long de ce chapitre, a été réalisé, dans sa presque totalité. Il se présente sous la forme d'un programme rédigé en langage Pascal standard. D'abord conçu sur le système d'exploitation MULTICS du HB-68 de Grenoble, il a pu être transféré très aisément, à l'aide de Kermit, sur une SM-90, fonctionnant sous UNIX. Ce programme peut être utilisé pour toute grammaire hors-contexte : ainsi a-t-il servi à tester une grammaire du langage APL. Le chapitre suivant sera consacré à une application particulière de l'analyseur : l'indexation automatique.

Chapitre 6

VERS L'INDEXATION AUTOMATIQUE

L'analyse syntaxique du syntagme nominal que nous avons étudiée jusqu'ici peut être utilisée à des fins diverses ; cependant, celle que nous visons est l'indexation automatique. Au cours de ce chapitre, nous tenterons d'exploiter les résultats de l'analyse syntaxique en vue de l'indexation automatique.

Auparavant, il nous faut rappeler les hypothèses qui régissent notre approche de l'indexation automatique. Classiquement, l'opération d'indexation manuelle d'un document peut se décomposer en trois phases. La première consiste à appréhender le contenu du document ; c'est la phase de compréhension. La seconde phase consiste à extraire du contenu du document, les notions jugées les plus importantes. Le choix de ces notions s'effectue toujours, de manière plus ou moins consciente, en fonction des utilisateurs finals et en fonction du fonds documentaire dans lequel ce document sera immergé. La troisième est une traduction de ces notions en terme de descripteurs.

L'opération d'indexation se révèle donc être une opération intellectuelle complexe et il serait bien utopique et ambitieux d'envisager l'indexation automatique comme étant la simulation par ordinateur de chacune des phases, compréhension, résumé, traduction, ainsi que de leur enchaînement. Il faut donc trouver une autre voie pour atteindre cet objectif.

C'est pourquoi nous proposerons une alternative cohérente avec l'hypothèse formulée dans l'introduction, selon laquelle le contenu informatif d'un texte est fortement lié à la forme de surface de ce texte. Donc, l'indexation pour le groupe SYDO se pose en ces termes : il s'agit de repérer dans un texte, les éléments informatifs de ce texte, à partir de traits de surface. Cela nécessite d'une part que l'on sache analyser la surface d'un texte, objet des chapitres précédents, et d'autre part que l'on dispose de critères de surface pour

distinguer un élément informatif. C'est ici qu'intervient une autre hypothèse de travail, formulée par M. LE GUERN [1983]. Elle définit un descripteur comme étant un syntagme nominal extrait du texte. Un descripteur est donc un élément du discours et non pas, comme dans l'acception classique, un élément lexical. C'est en celà qu'il est vecteur d'information. De plus, sa nature le rend aisément repérable à la surface d'un texte, au moyen d'une analyse morpho-syntaxique. Ainsi l'indexation automatique passe pour le groupe SYDO par l'analyse morpho-syntaxique des textes.

Cependant, il est nécessaire de nuancer davantage la définition d'un descripteur en fonction de la représentation des textes traités. En effet, les corpus sur lesquels nous avons travaillé, sont constitués à partir d'articles scientifiques rédigés en français par des auteurs francophones. Pour chacun des articles, seuls les titres et résumés d'auteurs ont été saisis et traités, ceci pour des raisons de coût. L'indexation de ces corpus ne se pose pas dans les mêmes termes que l'indexation de corpus de textes pleins, puisque dans notre cas, l'extraction des idées principales est déjà effectuée. Donc si tous les syntagmes nominaux des titres et des résumés peuvent être considérés comme des descripteurs potentiels, il n'en est pas de même pour les syntagmes nominaux issus des textes pleins. D'ailleurs, M. Le Guern précise que dans un texte plein, les descripteurs sont les syntagmes nominaux qui jouent le rôle de thèmes. La détection des thèmes dans un texte plein est encore à l'étude actuellement et nous ne l'aborderons pas plus avant.

Ce chapitre sera donc consacré à l'étude des syntagmes nominaux dans un corpus de titres et de résumés. Ce corpus est soumis à deux segmentations successives : en texte (titre + résumé) puis, pour chacun des textes, en syntagmes nominaux maximaux. Aussi, notre étude comportera trois phases : nous considérerons, dans un premier temps, chacun des syntagmes nominaux maximaux isolément et tenterons d'interpréter leurs structures syntaxiques ; puis, tenant compte de la segmentation du corpus en documents, nous analyserons l'ensemble des syntagmes nominaux d'un résumé, pour n'en retenir que les descripteurs. Enfin, nous considérerons les descripteurs au niveau d'un corpus.

1. EXPLOITATION DE LA STRUCTURE SYNTAXIQUE DU SN

Parmi l'ensemble des syntagmes nominaux, on distinguera les syntagmes nominaux maximaux simples des syntagmes nominaux maximaux complexes. En effet, **un syntagme nominal maximal** est un syntagme nominal qui n'est pas contenu dans un syntagme nominal plus large, mais celui-ci peut contenir ou ne pas contenir d'autres syntagmes nominaux plus simples. **Un syntagme nominal maximal simple** est donc un syntagme nominal qui n'est pas contenu dans un autre syntagme nominal et qui n'en contient pas lui-même. Un **syntagme nominal maximal complexe** est un syntagme nominal non contenu dans un syntagme nominal plus large, mais contenant des syntagmes nominaux plus simples. Ainsi la structure syntaxique d'un syntagme nominal complexe traduit une relation entre syntagmes nominaux, information que l'on ne trouve pas dans la structure syntaxique d'un syntagme nominal simple. C'est pourquoi nous distinguerons ces deux cas .

1.1. LES SYNTAGMES NOMINAUX MAXIMAUX SIMPLES

1.1.1. Leur structure

Par définition les syntagmes nominaux simples ne contiennent pas d'autres syntagmes nominaux, donc leur structure ne peut découler de l'application des règles suivantes :

SP --> P N''

N'' --> N'' N''

règles où N'' apparaît en partie droite. L'exclusion de ces règles entraînent l'exclusion des règles où le symbole SP apparaît en partie droite, donc :

N --> N SP

N' --> N SP (SP...SP)

A'' --> A' SP (SP...SP)

Les structures des syntagmes nominaux simples résultent donc de l'application d'un sous-ensemble de la grammaire de départ. D'autre part, dans le cas de structures multiples, nous avons choisi de privilégier la structure la plus régulière à droite. Il en résulte que, dans le cas général, on peut distinguer en remontant le long de cette structure, du centre de syntagme vers l'axiome, plusieurs strates successives [figure 6.1].

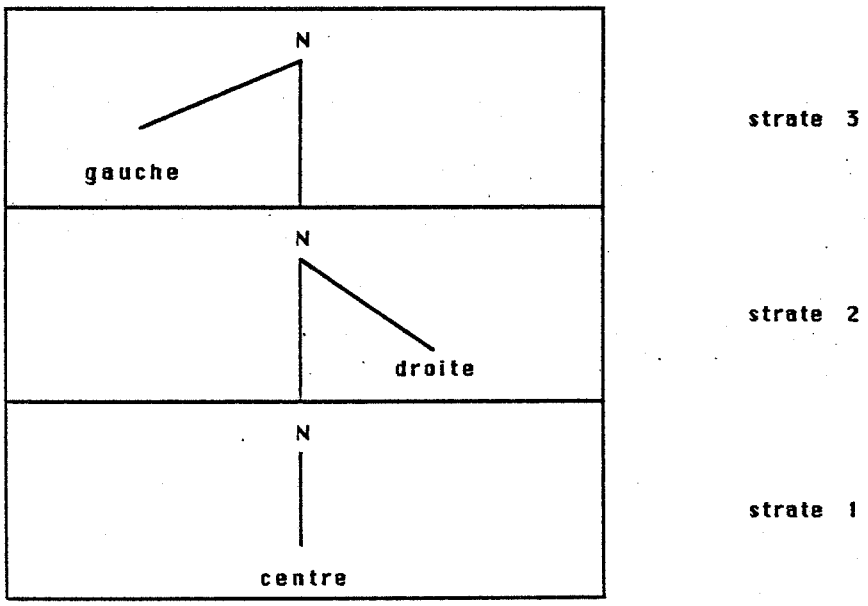


FIGURE 6.1

La première strate est celle qui est obtenue par la seule réécriture du centre de syntagme.

La seconde strate poursuit la première : elle correspond à la structure engendrée sur la sous-chaîne droite du syntagme. Cette seconde strate contient éventuellement des structures imbri-

quées.

La dernière strate poursuit la troisième et correspond à la structure engendrée par l'adjonction de la sous-chaîne gauche.

A cette succession de strates dans la structure correspond une segmentation de la surface du syntagme nominal. En surface, on distinguera le centre de syntagme, objet de la première strate, mot de la catégorie F, ou plus rarement , de la catégorie W (QUA), toujours présent. Les strates 2 et 3 facultatives, correspondent respectivement aux constituants suivant et précédant le centre de syntagme.

1.1.2. Interprétation

Une étude sur un corpus de botanique, de la correspondance entre ces strates et la forme de surface des syntagmes nominaux, illustrée par le tableau suivant :

éléments antéposés	centre de syntagme	éléments postposés
	[sélection]	
[le]	[virus]	
[sa]	[synthèse]	
[une]	[dormance]	
	[lamelles]	[moyennes]
[l']	[organe]	[contaminé]
[les deux]	[populations]	
[les]	[surfaces]	[isolées]
[des]	[milieux]	[trop variés]
[le]	[revêtement]	[piléique celluleux]
[des]	[surfaces]	[planes bien draînées]
[les nombreuses]	[espèces]	[tubérifères]

montre que :
1. le centre de syntagme est, dans la plupart des cas, l'élément central du sens du syntagme. Ainsi : "virus", "surfaces" ; le cas du mot "espèces" est plus délicat.
2. les éléments postposés précisent beaucoup le sens du centre de syntagme, et ce, de façon progressive. On a affaire ici à une relation hyperonyme / hyponyme. Ainsi "revêtement" est un hyperonyme de "revêtement piléique" qui est à son tour un hyperonyme de "revêtement piléique celluleux".
3. les éléments antéposés jouent bien davantage le rôle de lien référentiel entre le texte qui précède et le syntagme nominal qu'ils introduisent, et apporte donc plus d'informations à la compréhension globale du texte qu'à la détermination du sens du syntagme considéré isolément. Ainsi, "les nombreuses', "sa", "une"...

Ces considérations, très empiriques et de plus fragmentaires, nous conduiraient cependant à ne pas considérer un syntagme nominal comme une succession de constituants, mais à interpréter les différents niveaux de sa structure syntaxique, à partir du centre de syntagme. Il s'en dégage une relation d'hyperonymie / hyponymie entre éléments d'un même syntagme nominal, relation qui s'établit entre des êtres syntaxiques étiquetés "N", et non pas des syntagmes nominaux.

1.2. LES SYNTAGMES NOMINAUX COMPLEXES

Les syntagmes nominaux complexes résultent de l'application des règles :

N'' --> N'' N''
SP --> P N''

et donc des règles :

N --> N SP
N' --> N SP (SP...SP)
A'' --> A' SP (SP...SP)

Nous distinguerons trois situations suivant la séquence des règles appliquées. La première est engendrée par l'application de la règle N'' --> N'' N''. La seconde par la séquence SP --> P N'', puis N --> N SP. La troisième, enfin, par la règle SP --> P N'' puis l'une des deux règles N' --> N SP (SP...SP) ou A'' --> A' SP (SP...SP)

1.2.1. Cas 1 : règle N'' --> N'' N''

La règle N'' --> N'' N'' concatène deux syntagmes nominaux en un troisième. Les syntagmes nominaux de la partie droite ne sont pas indépendants l'un de l'autre puisqu'ils peuvent se juxtaposer. Au sein du corpus étudié, cette règle est attestée par des séquences où le deuxième N'' est un nom propre (présence d'une majuscule).

"la lignée Vpm"
"le champignon Botrytis cinerea"

La relation qui intervient entre les syntagmes nominaux "la lignée" et "le champignon" d'une part, et "Vpm" et "Botrytis cinerea" d'autre part, sont dans ce corpus de nature générique / spécifique dans la mesure où "Botrytis cinerea" est une espèce de "champignon". Cette relation, toujours attestée dans le corpus, ne peut être généralisée sans prudence à d'autres corpus. En effet, les naturalistes identifient toujours les végétaux par un nom de genre suivi d'un nom d'espèce ; ce procédé leur est donc familier.

En conséquence, pour un tel corpus il y a deux attitudes face à cette situation. La première consiste à retenir les trois syntagmes nominaux et donc, par exemple :

"le champignon"
"Botrytis cinerea"
"le champignon Botrytis cinerea"

La seconde se limiterait à ne retenir que les deux syntagmes nominaux de la partie droite et la relation générique / spécifique qui les relie.

genre
"le champignon" <==========> Botrytis cinerea"
espèce

1.2.2. Cas 2 : SP --> P N" puis N --> N SP

Ce cas se distingue du suivant dans la mesure où le syntagme prépositionnel n'est pas annoncé par le N auquel il s'associe. Alors la plus grande partie des SP en question est introduite par l'une des deux prépositions "de" ou "à".

1.2.2.1. Etude de N --> N SP(à)

1.2.2.1.1. Domaine d'application de cette règle

Puisque nous nous plaçons dans le cas des SP non prédits, la première question à se poser est : peut-on distinguer un SP(à) requis par le N qui le précéde d'un SP(à) non requis. En d'autres termes : lorsqu'un mot de la catégorie F régit, d'après le lexique, un SP(à) et est suivi d'un tel SP(à), a-t-on le moyen de prédire laquelle des règles

N --> N SP
N' --> N SP

s'applique. Il semble que l'on puisse proposer une réponse en ces termes : lorsqu'un F régit un SP(à), ce SP est toujours déterminé (le problème demeure pour les noms propres). Ainsi :

"résistance à les virus"

s'analyse à l'aide de la règle N' --> N SP. Par contre, si le SP(à) qui suit est non déterminé :

"résistance à contrôle monogénique"

alors la règle N --> N SP doit s'appliquer.

Dans les autres cas, c'est-à-dire, lorsque le SP(à) n'est pas requis, la présence ou l'absence du déterminant n'est pas significative. Ainsi :

"veau aux hormones"
"vergers à haute densité"
"maladies à virus"

s'analysent avec la règle N --> N SP.

1.2.2.1.2. Interprétation du lien induit par la règle N --> N SP(à)

Toujours au sein de notre corpus, nous avons relevé les syntagmes nominaux sur lesquels s'appliquent cette règle. Ainsi :

"[les maladies à [virus]]"
"[Les vergers à [haute densité]]"
"[les résistances à [contrôle polygénique]]"

La décomposition en syntagmes nominaux est figurée par les parenthèses. On obtient dans ces cas, deux syntagmes nominaux , par exemple :

"les résistances à contrôle polygénique"
"contrôle polygénique"

Si l'on s'en tient aux seuls syntagmes nominaux, on ne peut aller plus loin ; par contre si l'on s'autorise à revenir aux constituants de la règle N --> N SP, alors l'on peut remarquer une relation entre le N de la partie gauche et le N de la partie droite, qui s'énoncerait ainsi : le N de la partie gauche est un hyponyme de celui de la partie droite. Par exemple, le N "maladies à virus" représente une notion plus précise que le N "maladies".

Comme dans le cas des syntagmes nominaux simples, la relation hyperonyme / hyponyme induite par la règle N --> N SP ne porte pas non plus sur des syntagmes nominaux, mais sur des N, ce qui rend son exploitation délicate.

Afin de compléter cette étude des SP(à), nous mentionnerons les occurrences de locutions prépositionnelles débutant avec la préposition "à", comme "à la suite de", "à le cours de". Ces locutions introduisent le plus souvent un complément circonstanciel, et présentent de ce fait un double obstacle : le premier déjà mentionné au chapitre 4, dû à la non prise en compte de tels compléments, le second causé par la non-reconnaissance de ces locutions qui seront donc analysées comme de simple SP(à) incluant un SP(de).

1.2.2.2. La règle N --> N SP(de)

La préposition "de" est de loin la plus couramment utilisée, mais elle est aussi la plus difficile à interpréter. Aussi nous limiterons nous à quelques traits saillants, car il serait bien imprudent de vouloir aborder en profondeur tous les aspects du fonctionnement de cette préposition sur un corpus particulier et de taille réduite.

Tout d'abord, nous n'avons pas le moyen, contrairement à la préposition "à", de distinguer sur les syntagmes nominaux suivants laquelle des règles N --> N SP ou N' --> N SP s'applique :

"la construction de bois"
"la construction de maisons"

Remarque Le problème ne se pose que si le substantif n'est précédé d'un déterminant, c'est-à-dire lorsqu'il est au pluriel ou précédé d'un adjectif ou encore lorsque c'est un nom de matière.

Nous avons fait le choix de privilégier la règle N' --> N SP et avons justifié ce choix au chapitre 4. Nous ne considérons donc ici que les syntagmes nominaux engendrés par l'application de la règle N --> N SP, limitée au cas où le N de la partie droite ne régit pas, d'après le lexique, de SP(de). De tels syntagmes nominaux sont illustrés par :

"[plusieurs formes de [résistances partielles]]"
"[différents stades de [l'infection virale]]"
"[un taux de [plantes infectées] plus faible]"
"[certaines parties de [la plante]]"
"[le niveau de [la protection]]"
"[les principales caractéristiques de [ces résistances]]"
"[l'épaisseur de [les parois de [les cellules épidermiques]]]"

Les syntagmes nominaux inclus sont indiqués par les parenthèses. Ainsi, dans le syntagme nominal "l'épaisseur de les parois de les cellules épidermiques", on distingue, outre ce syntagme nominal, le syntagme nominal "les parois des cellules épidermiques" et le syntagme nominal "les parois épidermiques".

Une interprétation de la relation induite par la règle N --> N SP(de) pourrait s'exprimer ainsi : le syntagme N de la partie gauche de règle représente une partie (au sens ensembliste) du syntagme N'' contenu dans le SP(de). On a alors une relation partie-tout entre SN. Ainsi : "Les cellules épidermiques" représentent un surensemble du syntagme nominal "les parois des cellules épidermiques" qui représentent à leur tour un surensemble du syntagme nominal "l'épaisseur des cellules des parois épidermiques".

Cette relation a l'avantage de porter sur des syntagmes nominaux ce qui n'était pas le cas des relations induites par les règles précédentes. D'autre part, elle confirme les résultats annoncés par M. PRADILLA [1982] et R. ALVAREZ[1982] qui stipulaient qu'au sein d'un syntagme nominal complexe, le syntagme nominal le plus à droite jouait un rôle majeur dans le sens du syntagme nominal global.

1.2.2.3. Les autres prépositions.

Outre les prépositions "de" et "à" introduisant des SP non régis, nous avons rencontré au sein des syntagmes nominaux les prépositions :
- "chez" dans "exploitation de résistances aux virus chez les plantes maraîchères"
- "dans" dans "la réduction du nombre de traitements dans l'avenir"
- "pour" dans "quelques unités pour cent"...

Leurs occurrences sont trop faibles pour que l'on puisse distinguer leurs différents emplois. Cependant, lorsqu'elles introduisent un circonstanciel, alors elles conduisent à une analyse erronée.

1.2.3. Cas 3 : les structures engendrées par les ISL

Nous retrouvons ici les structures engendrées par des indicateurs de structure lexicaux (ISL). Le chapitre 4 portait sur la construction de ces structures ; il s'agit maintenant d'exploiter l'information qu'elles contiennent.

Les structures engendrées par les ISL mettent en évidence un ou plusieurs SP qui jouent le rôle de compléments par rapport au nom ou à l'adjectif qui les régit. Une telle structure fait apparaître autant de syntagmes nominaux que de compléments régis, plus un syntagme nominal qui englobe le gouverneur et l'ensemble de ses compléments. Les syntagmes nominaux compléments se situent tous au même niveau dans la structure, et chacun sature un C_i différent. De ce fait, il existe deux relations distinctes entre les constituants :
- la première s'établit entre le gouverneur et chacun des syntagmes nominaux compléments.
- la deuxième est une relation entre les syntagmes nominaux compléments d'un même gouverneur.

Schématiquement, ces relations peuvent se représenter sous une forme étoilée [figure 6.2].

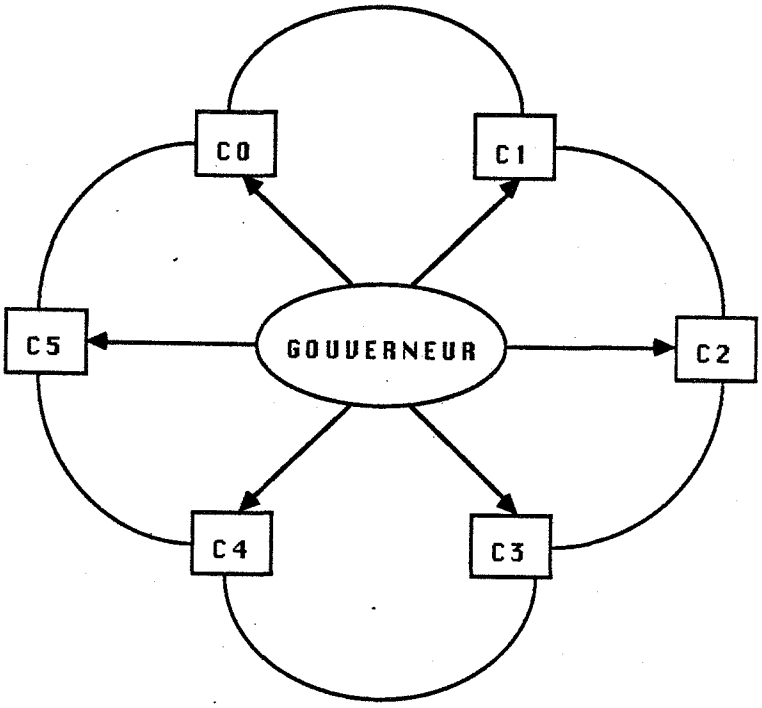


FIGURE 6.2

Exemple :

Soit le syntagme nominal "la résistance de le pommier à la tavelure". La structure syntaxique de ce syntagme nominal [figure 6.3], exhibe deux syntagmes nominaux "le pommier" et "la tavelure", et l'analyseur permet d'affirmer en outre que "le pommier" est un complément C_0 et "la tavelure" un complément C_2. Le schéma étoilé est plus riche [figure 6.4] ; il rend compte de toutes les relations qui s'exercent d'une part entre le gouverneur et chacun des C_i et d'autre part entre les C_i :

"résistance" a pour C_0 "le pommier"

"résistance" a pour C_2 " la tavelure"

"le pommier" et "la tavelure" sont simultanément compléments de "résistance".

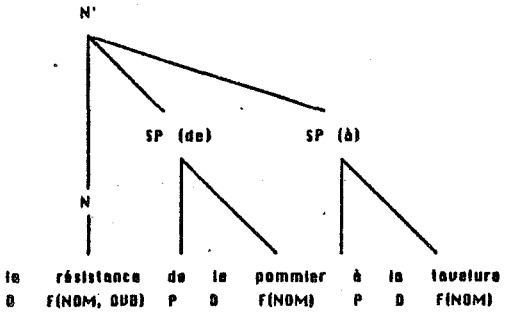


figure 6.3

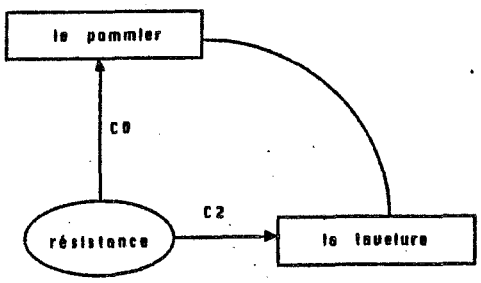


figure 6.4

On exprime ainsi toutes les relations induites par la règle N' --> N SP (SP...SP), mais l'expression de ces relations passent par une normalisation de la surface du texte, et l'on s'éloigne alors de l'hypothèse de départ : un descripteur est un élément du discours et donc un syntagme nominal. Cette normalisation de l'énoncé pourrait cependant être exploitée dans le cadre de travaux ultérieurs à une double fin : structurer le vocabulaire d'indexation et détecter les paraphrases.

La décomposition en syntagmes nominaux déduite de la structure syntaxique est en accord avec l'hypothèse de départ, mais l'information qu'elle apporte est beaucoup plus faible que celle dont on dispose et qui est rendue par le schéma étoilé. Il faut donc chercher, tout en restant dans le domaine des syntagmes nominaux, à mieux rendre compte de cette information. Une solution empirique consiste, à partir d'un ISL régissant n compléments, à engendrer outre les n+1 syntagmes nominaux effectivement présents, tous les syntagmes nominaux obtenus tour à tour par l'effacement :

nominaux obtenus tour à tour par l'effacement :

- de chacun des compléments
- de chaque paire de compléments
- de chaque (n-1)-uplet de compléments

On obtient ainsi tous les syntagmes nominaux sous-jacents à une telle structure.

Exemple :

Du syntagme nominal "la résistance de le pommier à la tavelure", avec une telle méthode on extrait, en outre, les syntagmes nominaux :

"résistance de le pommier"
"résistance à la tavelure"

Du syntagme nominal "L'expédition d'un satellite dans l'espace par la NASA" on extrait

"un satellite"
"l'espace"
"la NASA"
"L'expédition d'un satellite"
"L'expédition dans l'espace"
"L'expédition par la NASA"
"L'expédition d'un satellite dans l'espace"
"L'expédition d'un satellite par la NASA"
"L'expédition dans l'espace par la NASA"
"L'expédition d'un satellite dans l'espace par la NASA"

La représentation sous forme étoilée, qui reste la plus complète, est à rapprocher des travaux sur les mots signaux effectués par Florence VEILEX [1985]. En effet, un syntagme nominal engendré par la règle N' --> N SP (SP...SP) est une forme nominalisée. Par exemple, "la résistance de le pommier à la tavelure" est une forme nominalisée de la phrase "le pommier résiste à la tavelure". Dans cette phrase, le verbe "résiste" est un mot signal qui relie les syntagmes nominaux "le pommier" et "la tavelure". Par transposition à la forme nominalisée, on peut interpréter le mot "résistance" comme un mot signal reliant les mêmes syntagmes nominaux. Cette approche qui mériterait d'être approfondie a l'avantage de permettre l'utilisation des travaux antérieurs de l'équipe, tout en justifiant le fait que la relation induite par les ISL ne peut s'exprimer totalement au moyen de syntagmes nominaux.

2. INDEXATION AU MOYEN DE SYNTAGMES NOMINAUX

Pour aborder ce paragraphe, nous donnerons une autre définition de l'indexation :

"Indexer un document c'est modifier la représentation de ce document afin de faciliter l'accès à l'information qu'il contient"

Cette définition fait apparaître la notion de représentation d'un document ainsi que la dualité entre indexation et interrogation.

2.1. LA REPRESENTATION D'UN DOCUMENT

Dans le cadre de notre application, un document est caractérisé par ses références bibliographiques et par des données textuelles, en l'occurrence, le titre et le résumé d'auteurs. Voici sur un exemple la représentation initiale d'un document :

REPRESENTATION 1

Références bibliographiques
LECOQ (H.), POCHARD (E.), PITRAT (M.), LATERROT (H.), MARCHOUX (G.). Identification et exploitation de résistances aux virus chez les plantes maraîchères. *Cryptogamie, Mycologie,* 1982, tome 3, pp. 333-365.

Titre
Identification et exploitation de résistances aux virus chez les plantes maraîchères.

Résumé
La création de variétés résistantes aux virus apparaît comme le moyen le plus efficace et le plus économique de lutter contre ces agents pathogènes.
Plusieurs formes de résistances partielles sont utilisées pour la création de variétés de Melon, Piment et Tomate résistantes aux virus. Nous présentons quelques mécanismes intervenant à différents stades de l'infection virale. La résistance peut s'exprimer lors de l'inoculation du virus ce qui se traduit par un taux de plantes infectées plus faible (tendance à échapper à l'infection). Le virus est parfois séquestré dans l'organe contaminé ou demeure localisé à certaines parties de la plante sans devenir systémique (résistance à la migration) ou sa synthèse est réduite (résistance à la multiplication). Enfin le virus peut être difficilement accessible aux vecteurs, ce qui peut réduire le développement ultérieur des épidémies (résistance à l'acquisition du virus). Pour un certain nombre de ces cas des tests simples, appliqués en sélection, ont été mis au point.
L'association chez une même plante de mécanismes de résistances agissant à des stades différents du développement de l'infection virale a pour but d'augmenter le niveau de la protection. Cependant cette addition de résistances partielles ne conduit pas dans toutes les circonstances à une protection suffisante de l'hôte. L'association de pratiques culturales retardant le développement des épidémies virales et de variétés partiellement résistantes peut alors conduire à une protection efficace de la culture.

Les données textuelles sont soumises à l'analyseur morpho-syntaxique : on obtient alors dans un premier temps, la liste des générateurs maximaux formés à partir des catégories F, P, D, W. Voici donc la nouvelle représentation obtenue :

REPRESENTATION 2

Références bibliographiques

LECOQ (H.), POCHARD (E.), PITRAT (M.), LATERROT (H.), MARCHOUX (G.). Identification et exploitation de résistances aux virus chez les plantes maraîchères. *Cryptogamie, Mycologie,* 1982, tome 3, pp. 333-365.

les générateurs maximaux

identification
exploitation de résistances à les virus chez les plantes maraîchères
la création de variétés résistantes à les virus
ainsi le moyen le plus efficace
contre ces agents pathogènes
plusieurs formes de résistances partielles
utilisées pour la création de variétés de melon
piment
tomate résistantes à les virus
quelques mécanismes intervenant à différents stades de l'infection virale
la résistance
lors de l'inoculation de le virus
par un taux de plantes infectées plus faible
tendance
à l'infection
le virus
parfois séquestré dans l'organe contaminé
localisé à certaines parties de la plante sans
résistance à la migration
sa synthèse
résistance à la multiplication
enfin le virus
difficilement accessible à les vecteurs
le développement ultérieur de les épidémies
résistance à l'acquisition de le virus
pour un certain nombre de ces cas de les tests simples
appliqués en sélection
mis à le point
l'association chez une même plante de mécanismes de résistances agissant à de les stades différents de le développement de l'infection virale
pour but
le niveau de la protection
cependant cette addition de résistances partielles
dans toutes les circonstances à une protection suffisante de l'hôte
l'association de pratiques culturales retardant le développement de les épidémies virales
de variétés partiellement résistantes
à une protection efficace de la culture

La représentation effectivement obtenue est beaucoup plus riche en information dans la mesure où chaque forme du texte est accompagnée du résultat de l'analyse morphologique. Ainsi, "le niveau de la protection" recouvre : "le [le, D (DEF, MAS, SNG)] niveau [niveau, F (NOM, MAS, SNG)] de [de, P] la [la, D (DEF, FEM, SNG)] protection [protection, F [NOM, DVB, FEM, SNG)]. Nous avons choisi une présentation simplifiée pour ne pas alourdir notre propos.

Ces générateurs sont à leur tour soumis à l'analyse syntaxique. De ce traitement, sont issues deux nouvelles représentations : une représentation intermédiaire constituée des syntagmes nominaux maximaux et une représentation finale obtenue après décomposition des syntagmes nominaux complexes. Voici donc ces deux représentations :

REPRESENTATION 3

Références bibliographiques

LECOQ (H.), POCHARD (E.), PITRAT (M.), LATERROT (H.), MARCHOUX (G.). Identification et exploitation de résistances aux virus chez les plantes maraîchères. *Cryptogamie, Mycologie*, 1982, tome 3, pp. 333-365.

les syntagmes nominaux maximaux

identification
exploitation de résistances à les virus chez les plantes maraîchères
la création de variétés résistantes à les virus
le moyen le plus efficace
ces agents pathogènes
plusieurs formes de résistances partielles
la création de variétés de melon
piment
tomate
les virus
quelques mécanismes intervenant à différents stades de l'infection virale
la résistance
l'inoculation de le virus
un taux de plantes infectées plus faible
tendance
l'infection
le virus
l'organe contaminé
certaines parties de la plante
résistance à la migration
sa synthèse
résistance à la multiplication
le virus
les vecteurs
le développement ultérieur de les épidémies

résistance à l'acquisition de le virus
un certain nombre de ces cas de les tests simples
sélection
le point
l'association chez une même plante de mécanismes de résistances agissant à de les stades différents de le développement de l'infection virale
but
le niveau de la protection
cette addition de résistances partielles
toutes les circonstances à une protection suffisante de l'hôte
l'association de pratiques culturales retardant le développement de les épidémies virales
variétés partiellement résistantes
une protection efficace de la culture

REPRESENTATION 4

Références bibliographiques

LECOQ (H.), POCHARD (E.), PITRAT (M.), LATERROT (H.), MARCHOUX (G.). Identification et exploitation de résistances aux virus chez les plantes maraîchères. *Cryptogamie, Mycologie*, 1982, tome 3, pp. 333-365.

les syntagmes nominaux maximaux et les syntagmes nominaux inclus

identification
exploitation de résistances à les virus chez les plantes maraîchères
    les plantes maraîchères
    les virus chez les plantes maraîchères
    résistances à les virus chez les plantes maraîchères
la création de variétés résistantes à les virus
    les virus
    variétés résistantes à les virus
le moyen le plus efficace
    le plus efficace
    le moyen
ces agents pathogènes
plusieurs formes de résistances partielles
    résistances partielles
la création de variétés de melon
    melon
    variétés de melon
piment
tomate
les virus
quelques mécanismes intervenant à différents stades de l'infection virale
    l'infection virale
    différents stades de l'infection virale

la résistance
l'inoculation de le virus
le virus
un taux de plantes infectées plus faible
plantes infectées
tendance
l'infection
le virus
l'organe contaminé
certaines parties de la plante
la plante
résistance à la migration
la migration
sa synthèse
résistance à la multiplication
la multiplication
le virus
les vecteurs
le développement ultérieur de les épidémies
les épidémies
résistance à l'acquisition de le virus
le virus
l'acquisition de le virus
un certain nombre de ces cas de les tests simples
les tests simples
ces cas de les tests simples
sélection
le point
l'association chez une même plante de mécanismes de résistances agissant à de les stades différents de le développement de l'infection virale
l'infection virale
le développement de l'infection virale
les stades différents de le développement de l'infection virale
résistances agissant à de les stades différents de le développement de l'infection virale
mécanismes de résistances agissant à de les stades différents de le développement de l'infection virale
une même plante de mécanismes de résistances agissant à de les stades différents de le développement de l'infection virale
but
le niveau de la protection
la protection
cette addition de résistances partielles
résistances partielles
toutes les circonstances à une protection suffisante de l'hôte

l'hôte
une protection suffisante de l'hôte
l'association de pratiques culturales retardant le développement de les épidémies virales
les épidémies virales
le développement de les épidémies virales
pratiques culturales retardant le développement de les épidémies virales
variétés partiellement résistantes
une protection efficace de la culture
la culture

Cette dernière représentation devrait elle aussi être complétée en prenant en compte les structures syntaxiques. Ainsi "le niveau de la protection" recouvre effectivement les données suivantes :

```
N'' {MAS SNG }
      D' {DEF MAS SNG }
           D   le {DEF MAS SNG }

      N' {MAS SNG }
           N   {MAS SNG }
                 N   {MAS SNG }
                       F   niveau {NOM MAS SNG }

                 SP
                       P   de
                       N'' {DVB FEM SNG }
                             D' {DEF FEM SNG }
                                   D   la {DEF FEM SNG }

                             N' {DVB FEM SNG }
                                   N   {DVB FEM SNG }
                                         F   protection {NOM DVB FEM SNG
```

Ainsi, un document est représenté par les structures syntaxiques des syntagmes nominaux qu'il contient. Le résultat auquel on parvient n'est pas dénué d'inexactitudes. En effet, quelques imperfections demeurent tant dans la représentation 2 que dans la représentation 3.

Dans la représentation 2, on distingue deux types d'erreurs :

1 les erreurs provoquées par l'élimination des ponctuations et des conjonctions de coordination : ainsi "la création de variétés de melon, piment et tomate résistantes aux virus" conduit aux générateurs "la création de variétés de melon", "piment" et "tomate résistantes aux virus". Ce dernier générateur est syntaxiquement incorrect ; son analyse donne donc naissance à deux syntagmes nominaux maximaux "tomate" et "les virus".

2 les erreurs provoquées par l'extraction des générateurs. Ainsi : "toutes les circonstances à une protection suffisante de l'hôte" "certain nombre de ces cas des tests simples" sont considérés comme un seul syntagme nominal maximal alors qu'ils en contiennent deux, le premier étant un complément circonstanciel de la phrase.

La représentation finale hérite donc des imperfections de la représentation précédente auxquelles viennent s'ajouter les quelques inexactitudes dues à l'analyseur syntaxique. En effet, lorsqu'un syntagme nominal complexe admet plusieurs structures syntaxiques, sa décomposition en syntagmes nominaux plus simples n'est pas toujours unique. Puisque nous avons fait le choix de ne retenir que la structure la plus régulière, nous obtenons parfois une décomposition erronée. Ce cas est illustré par la décomposition du syntagme nominal très complexe "l'association chez une même plante de mécanismes de résistances agissant à des stades différents de l'infection virale". Ce syntagme nominal pose deux problèmes : celui des circonstanciels "chez une même plante" que la grammaire ne sait traiter, et celui des participes présents. En effet un participe présent est invariable, il n'est donc marqué ni en genre ni en nombre, aussi peut-il s'associer à tous les noms qui le précèdent. Si, dans les faits, il doit être rattaché au nom le précédant immédiatement, alors la structure régulière est la bonne structure. Si par contre, comme c'est le cas ici, il doit être rattaché à un nom plus avant, alors l'analyse obtenue est erronée. Une solution consisterait alors à déclarer "mécanismes de résistances" comme mot composé, ce qui n'est pas sans poser d'autres problèmes.

Le fait de ne retenir que la décomposition en syntagmes nominaux issue de la structure la plus régulière réduit considérablement la dimension de la représentation. En effet, le syntagme nominal précédent admet plus de 36 structures toutes inexactes à cause du circonstanciel, et donne donc naissance à un grand nombre des syntagmes nominaux plus simples. Ce choix conduit le plus souvent à la bonne solution, de plus il a l'avantage de réduire le nombre de syntagmes nominaux à envisager, cependant il n'en est pas moins arbitraire et peut donc être remis en cause par la suite.

La représentation 3 est, malgré ses imperfections déjà signalées, une représentation très riche en renseignements linguistiques.

2.2. ANALYSE FONCTIONNELLE DE LA REPRESENTATION D'UN DOCUMENT.

Au sein d'un système documentaire, la représentation d'un document sera stockée de manière à permettre une recherche efficace. La représentation d'un document peut donc être évaluée à travers deux fonctions : le stockage et l'interrogation.

2.2.1. Le stockage

Quelle que soit la structure des fichiers sur lesquels sera stockée la représentation des documents, il est évident que l'espace requis est au moins proportionnel à la taille de la représentation. Or la représentation 4 est d'une taille importante : il faut donc envisager de la réduire sans pour autant nuire à l'interrogation. Or l'on remarque que certains syntagmes nominaux, le plus souvent simples, apparaissent plus d'une fois : "le virus", "les virus", "l'infection virale". Pour satisfaire ces critères, on pourrait ne retenir qu'une seule occurrence

de chaque syntagme nominal.

L'on peut de plus s'interroger sur la nécessité de conserver des syntagmes nominaux sans intérêt ("tendance", "le point") ou très complexes ("l'association chez une même plante...l'infection virale") ; en effet, il est fort peu probable que de tels syntagmes soient l'objet de requêtes ; cependant, l'élimination automatique de tels syntagmes devrait, par cohérence, s'effectuer sur des critères morpho-syntaxiques. Or de tels critères ne permettent pas de distinguer "le point", syntagme à rejeter, de "le virus", syntagme à conserver. L'élimination de ces syntagmes parasites pourrait s'opérer soit à l'aide de modèle probabiliste (comme celui de HARTER), soit à l'aide d'un lexique de "syntagmes nominaux parasites", soit encore manuellement.

2.2.2. l'interrogation

L'interrogation d'un fonds documentaire où chaque document est représenté par une liste de syntagmes nominaux se doit de partir d'une question formulée en langue naturelle. De cette question, en sont extraits les syntagmes nominaux qui seront comparés à ceux présents dans la base. Sans entrer dans les détails d'une stratégie d'interrogation, stratégie qui est encore à l'état d'étude, l'on peut cependant avancer que, la représentation choisie est très pertinente pour une recherche documentaire, malgré quelques handicaps.

2.2.2.1. Avantages d'une telle représentation

Le fait de représenter un document par l'ensemble des syntagmes nominaux qu'il contient, présente de nombreux avantages par rapport aux systèmes documentaires classiques. En effet, cette représentation est très riche en informations linguistiques. De ce fait, il est possible de distinguer certaines homographies :

lorsque le sens dépend du genre : le syntagme nominal "le voile universel" est de genre masculin, et donc le mot "voile" représente "le voile" et non "la voile".

lorsque le sens peut être déterminé par le contexte du syntagme nominal : le syntagme nominal "la maturité des baies" exclut le sens "découpage côtier".

Cette représentation permet aussi de rendre compte du lien syntaxique entre deux termes : ainsi, "la résistance aux virus" et "la résistance des virus" recouvrent deux notions différentes que les opérateurs booléens ne peuvent distinguer.

Enfin, cette représentation permet de retrouver tous les syntagmes nominaux contenant un syntagme nominal donné et donc d'affiner la formulation de la question. Compte tenu de ces facilités, l'indexation obtenue est d'une grande finesse, qualité de base d'un système documentaire "peu bruyant"(*).

(*) Dans un système documentaire, le bruit est représenté par les documents sélectionnés non pertinents.

2.2.2.2. Inconvénients d'une telle représentation

Malheureusement un système documentaire bâti sur une indexation très fine, est souvent "trop silencieux"(**). Et la représentation choisie favorise effectivement le silence dans la mesure où elle ne rend pas compte des synonymies. Nous avons rencontré quelques cas de synonymie auxquels l'analyse morpho-syntaxique peut apporter une solution.

Il s'agit tout d'abord des variantes en nombre d'un même syntagme. Ainsi, "le virus" et "les virus" dénotent des classes d'objets équivalents pour une application documentaire. Cette transformation est aisée, mais n'est pas toujours valide ("théorie des langages" et "théorie du langage" ne recouvrent pas les mêmes notions).

Nous avons trouvé aussi des expressions équivalentes comme "l'écosystème prairial" et "l'écosystème de la prairie", qui proviennent du rôle syntaxique similaire que jouent dans ce cas l'adjectif et le syntagme prépositionnel. Une analyse morphologique dérivationnelle donnerait des résultats.

Enfin, une illustration plus dense et plus complexe de la synonymie est traduite par les expressions :

"quelques mécanismes intervenant à différents stades de l'infection virale"

"mécanismes de résistances agissant à des stades différents de l'infection virale".

La résolution de ces synonymies ne fait pas intervenir d'autres informations que celles issues de l'analyse morpho-syntaxique. Elle est un préalable indispensable à la construction d'un système documentaire efficace. Il existe d'autres synonymies plus difficiles à résoudre comme celle existant entre "virose" et "maladies à virus", car celles-ci font appel à des informations de nature sémantique.

La non-détection de la synonymie entre syntagmes nominaux n'est pas le seul facteur de silence. La non-prise en compte des relations hyperonymie / hyponymie va dans le même sens. Ainsi, si le syntagme nominal "résistance à la tavelure" figure dans la représentation d'un document, ce document ne sera pas identifié par le syntagme nominal "résistance aux maladies". Il est donc nécessaire de disposer de l'information supplémentaire "la tavelure est une maladie".

En conclusion, le mode d'indexation choisi conduirait vers un système documentaire "peu bruyant" mais "trop silencieux", où la représentation des documents est de taille importante. Pour lutter contre le silence, il est nécessaire de rendre compte des relations de synonymie et d'hyperonymie / hyponymie entre syntagmes. Or dans certains cas, ces relations sont perceptibles à travers l'analyse morpho-syntaxique, comme nous venons de le voir. Aussi pour terminer, nous montrerons sur un exemple qu'il est possible, en exploitant les résultats de l'analyse morpho-syntaxique, d'obtenir des informations sémantiques.

(**) Le silence est représenté par les documents pertinents non sélectionnés.

3. ESSAI DE STRUCTURATION DU VOCABULAIRE D'INDEXATION

A travers quelques résumés, nous avons relevé tous les syntagmes nominaux complexes dont le centre était le nom "résistance". En voici la liste :

résistances à contrôle polygénique
résistances à contrôle monogénique
résistance à les viroses
résistance à les maladies
résistances à les virus
la résistance à l'infection
la résistance à l'infection de le virus \y
résistance à la migration
résistance à la multiplication
résistance à l'acquisition de le virus
la résistance variétale de les plantes
résistance partielle chez les plantes
résistance de le pommier à la tavelure
la résistance de le hêtre à le chancre
résistances à les virus chez les plantes maraîchères
la résistance à les virus chez la pomme de terre
les résistances extrêmes à le virus \x.

On distingue parmi ces syntagmes ceux dans lequel "résistance" n'est pas un ISL comme :

résistances à contrôle polygénique
résistances à contrôle monogénique,

qui sont des types de "résistance" de ceux où il est ISL et donc régit des compléments. On observera aussi les circonstanciels qui bien que non régis se révèlent intéressants.

Voici la liste des syntagmes compléments C_0 :

les plantes
le pommier
le hêtre.

Ce sont des noms de végétaux.

Puis la liste des circonstanciels :

les plantes
les plantes maraîchères
la pomme de terre.

Ils appartiennent au même champ sémantique que les C_0. Ce n'est pas un hasard. En effet, les circonstanciels introduits ici par la préposition "chez" ne sont pas à rattacher à "résistance" mais à un ISL situé plus avant :

"exploitation des résistances aux virus chez les plantes maraîchères"
"exploitation des caractères de résistance partielle chez les plantes"
"amélioration de la résistance aux virus chez la pomme de terre"

qui sont des paraphrases plus élégantes que :

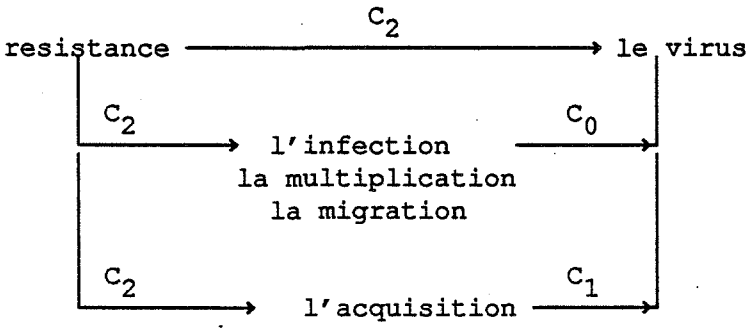
"exploitation des résistances des plantes maraîchères aux virus"
"exploitation des caractères de résistance partielle des plantes"
"amélioration de la résistance de la pomme de terre aux virus".

Dans ce corpus, l'usage de la préposition "chez", habituellement réservée aux êtres animés, est courant chez les botanistes qui assimilent les végétaux à des êtres animés : leur nom débute par une majuscule (Melon, Piment, Tomate, Pomme de terre...).

Voici enfin la liste des compléments C_2 :

les viroses
les maladies
la tavelure
le chancre
les virus
les virus
le virus \x
les virus
l'infection
l'infection de le virus \y
la migration
la multiplication
l'acquisition de le virus.

Parmi eux, on distingue les compléments formés sur un déverbal (les cinq derniers) des précédents. On constate que les non déverbaux sont des noms de vecteurs de maladie (virus), ou les maladies elles-mêmes (viroses, chancre, tavelure...). Les déverbaux recouvrent différents modes de transfert des maladies. D'où le schéma :



Cet exemple a pour objet de montrer qu'il est possible d'approcher le sens des syntagmes nominaux à partir de leur rôle syntaxique (compléments régis ou non, type de compléments), et d'informations lexicales (déverbal). Si l'on ajoute à ces renseignements de nature sémantique, ceux qui sont issus des relations entre syntagmes étudiées en tête de ce chapitre, l'on peut espérer structurer le vocabulaire d'indexation.

En conclusion, l'indexation de documents au moyen des syntagmes nominaux contenus dans le titre et dans le résumé conduit à un résultat d'une grande finesse. Cependant, il faut envisager de réduire la taille de cette représentation sans trop en altérer les qualités : par l'élimination des syntagmes redondants, des syntagmes trop complexes et des syntagmes sans intérêt par rapport au sujet traité.

L'utilisation d'un tel mode d'indexation à des bases documentaires très spécialisées permettrait de plus de structurer "sémantiquement" le vocabulaire d'indexation à partir d'informations morpho-syntaxiques.

Enfin, ce chapitre ne prétend rien affirmer, car les données traitées sont trop restreintes. Son objet est plus simplement de proposer à ceux qui poursuivront ce travail des idées pour exploiter les résultats de l'analyse syntaxique à des fins d'indexation.

CONCLUSION

L'analyseur syntaxique réalisé a été conçu avant tout comme un outil pour tester des données linguistiques. En effet, la grammaire, la liste des variables morphologiques, le lexique des indicateurs de structure lexicaux sont des données de l'analyseur. L'analyseur est donc de type déclaratif, de ce fait, ses performances en temps et espace ne sont pas optimales. De plus, l'analyseur construit toutes les structures compatibles avec les données linguistiques. Comme ces structures sont souvent nombreuses, l'espace et le temps requis pour les construire ne sont pas négligeables. Il est cependant rassurant de constater que le critère informatique de performance et le critère linguistique de justesse ne se contrarient pas, mais bien au contraire vont dans le même sens. En effet, des données linguistiques plus précises limitent le nombre de structures possibles et donc rendent l'analyseur plus performant. A titre d'exemples, le fait que l'analyseur tienne compte de l'ordre d'apparition des compléments régis limite le nombre de solutions parasites construites.

Les améliorations qui devront être apportées à ce programme devront donc d'abord être de nature linguistique. C'est seulement dans une phase ultérieure, lorsque la grammaire sera figée et les informations linguistiques stabilisées, que l'on pourra alors réellement s'attacher à améliorer les performances de l'algorithme.

Malheureusement, les règles qui régissent le fonctionnement de la langue ne s'énoncent pas toujours sous une forme calculable ; en d'autres termes, il n'existe pas toujours un algorithme traduisant l'application de ces règles. C'est le cas, par exemple, de l'interprétation des mots-composés (une même suite de mots peut suivant les cas être ou non un mot-composé), de la ponctuation "virgule" et des conjonctions de coordination, et de la détection des circonstanciels. Il n'est d'ailleurs pas surprenant que l'analyse syntaxique bute sur ces problèmes. En effet, ils ne peuvent être résolus que partiellement au niveau syntaxique, et nécessitent donc une interprétation à un niveau linguistique supérieur, sémantique

ou pragmatique. Dans le pire des cas, lorsque l'ambiguïté est totale, il n'existe, à aucun niveau, de solutions uniques.

Une solution envisageable serait de construire un système déductif à bases de connaissances qui, à partir d'une base de connaissances, et de la chaîne à analyser orienterait l'analyseur syntaxique vers la solution la plus probable. Les connaissances, à définir avec des linguistes, pourraient être :

pour la détection des circonstanciels, la liste des prépositions avec leur probabilité d'introduire un circonstanciel, la liste de certaines locutions comme "de bon matin", "dans toutes les circonstances"...

pour la détermination du rôle de la virgule, l'observation de son contexte d'occurrence ; existe-t-il un coordonnant à droite et non accolé à la virgule ? une proposition à droite ? ...

On disposerait alors d'un système d'analyse syntaxique plus performant, au sens où le nombre d'ambiguïtés serait limité, d'un système souvent plus juste, car il pourrait prendre en compte des phénomènes linguistiques non résolus par la syntaxe, mais d'un système capable de construire parfois des solutions fausses.

L'indexation automatique de documents n'est qu'une application parmi d'autres de l'analyse syntaxique de la langue naturelle. A titre d'exemples, nous pourrions citer dans le même ordre d'idées, le dépouillement des questions ouvertes d'une enquête, ou encore dans un autre domaine, la détection de fautes d'orthographe dans un texte.

Dans le cadre de l'indexation automatique, les résultats obtenus sont d'un intérêt certain bien que très parcellaires. Il est indispensable de poursuivre ce travail suivant deux axes complémentaires.

Le premier consiste à condenser davantage encore la représentation d'un document. Or pour condenser cette représentation, il semble, de prime abord, nécessaire de faire appel à des critères sémantiques : éliminer les syntagmes vides de sens, regrouper les syntagmes de sens voisins. La prise en compte de tels critères serait contraire à notre hypothèse de départ. Il faut donc mettre en oeuvre d'autres outils comme l'évaluation de la distance entre deux structures syntaxiques, outils qui, testés sur un corpus de taille importante devrait permettre de détecter des synonymies entre syntagmes.

La deuxième démarche consiste à considérer la question d'un utilisateur, et à partir d'elle, à générer par paraphrasage des énoncés équivalents que l'on comparerait à la représentation des documents.

Ces deux démarches se complètent car l'essentiel dans un système documentaire est que la représentation des documents d'une part, et la question d'autre par se rejoignent. Quant à définir à quel endroit se situe le point de jonction, celà est du ressort de la stratégie du système documentaire.

Le chemin qu'emprunteront ceux qui poursuivront ce travail, est encore long et semé d'embûches, mais j'ai acquis la conviction, tout au long de ce travail, qu'il est le seul qui conduise à ce que l'on appelle "la compréhension de la langue naturelle".

BIBLIOGRAPHIE

AHO (Alfred V.), CORASICK (Margaret J.). Efficient string matching : an aid to bibliographic search. *Communications of the A.C.M., 18, 6, pp. 333-343, 1975.*

AHO (Alfred V.), HOPCROFT (John E.), ULLMAN (Jeffrey D.). Data structures and algorithms. Reading, Addison-Wesley, 1983.

AHO (Alfred V.), HOPCROFT (John E.), ULLMAN (Jeffrey D.). The design and analysis of computer algorithms. Reading, Addison-Wesley, 1974.

AHO (Alfred V.), ULLMAN (Jeffrey D.). The theory of parsing, translation and compiling . Volume I : parsing. Englewood Cliffs, Prentice Hall, 1972.

ALVAREZ (Ramon). Aspects lexicographiques en documentation. Université des Sciences Sociales de Grenoble, Rapport de DEA IMSS, 1982.

ANTONIADIS (Georges). Elaboration d'un système d'analyse morpho-syntaxique d'une langue naturelle . Application en informatique documentaire. Thèse 3ème cycle, Université des Sciences Sociales de Grenoble, 20 juin 1984.

BERRENDONNER (Alain). [1983 a]. Grammaires formelles et analyse morpho-syntaxique automatique. *LE COADIC (Y.), ROUAULT (J.) ed., Ecole d'Eté des Sciences de l'Information, org. DBMIST, Vignieu, septembre 1983.*

BERRENDONNER (Alain). [1983 b]. Grammaire pour un analyseur : aspects morphologiques. *LE COADIC (Y.), ROUAULT (J.) ed., Ecole d'Eté des Sciences de l'Information, org. DBMIST, Vignieu, septembre 1983.*

BOUCKAERT (M.), PIROTTE (A.), SNELLING (M.). Efficient parsing algorithms for general context-free parsers. *Information Sciences, 1975, 8, pp. 1-26.*

BRUANDET (M.F.). Modèle partiel de connaissances pour un système de recherche d'informations. *Actes RIAO 85, Recherche d'informations assistée par ordinateur, Grenoble, mars 1985, pp. 101-114.*

BRUANDET (M.F.), CHIARAMELLA (Y.), KERKOUBA (D.). Interrogation des NEF. Documentation du Projet CONCERTO, no 7267, 1985.

CHIARAMELLA (Y.), KERKOUBA (D.). Indexation automatique des NEF. Documentation du projet CONCERTO, no 7236 INT, 1984.

COURTIN (Jacques). Algorithmes pour le traitement interactif des langues naturelles. Thèse Etat, Sciences, Université Scientifique et Médicale de Grenoble, 28/10/1977.

CULIOLI (A.), FUCHS (C.), PECHEUX (M.). Considérations théoriques à propos du traitement formel du langage. Documents de linguistique quantitative no 7. Paris, Dunod, 1970.

DREYFUS (Hubert L.). Intelligence artificielle : mythes et limites. Paris, Flammarion, 1984.

EARLEY (Jay Clark). An efficient context-free parsing algorithm. Carnegie-Mellon University, Ph.D., 1968.

EARLEY (Jay Clark). An efficient context-free parsing algorithm. *Communications of the ACM, vol. 13, no 2, feb. 1970, pp. 94-102.*

FLUHR (C.). Analyse de certaines fonctions que doit remplir un système linguistique dans une utilisation documentaire. *L'informatique documentaire, Bulletin du Centre de Hautes Etudes Internationales d'Informatique Documentaire, 1982, no 5, pp. 27-36.*

FLUHR (C.). Problèmes d'optimisation de l'accès à l'information dans les bases de données textuelles. *Actes du Colloque Traitement automatique du langage naturel, Nantes, 1984, vol.2, pp. 144-161.*

GALIOTOU (H.). construction d'un analyseur morphologique du français. Université des Sciences Sociales de Grenoble, I.M.S.S., Rapport de DEA, 1983.

GRANDJEAN (E.), VEILLON (G.). Utilisation d'une composante linguistique dans les logiciels de recherche d'informations : "logiciel prototype PIAFDOC". *Journée d'étude AFCET, Possibilités de traitement de textes en documentation, Paris, 24 avril 1980.*

GROSS (M.). Méthodes en syntaxe. Paris, Hermann, 1975.

HOROWITZ (E.), SAHNI (S.). Fundamentals of data structures. London, Pitman, 1976.

KALLAS (Ghassan). titre à préciser. Thèse de lüniversité des Sciences Sociales de Grenoble, 1986.

KIMBALL (John). Seven principles of surface structure parsing in natural language. *Cognition, 1973, vol.2, no 1, pp. 15-47.*

KING (Margaret) ed. Parsing natural languages . London, Academic Press, 1983.

LE GUERN (M.). Sémantique et syntaxe des descripteurs. *LE COADIC (Y.), ROUAULT (J.) ed., Ecole d'Eté des Sciences de l'Information, org. DBMIST, Vignieu, septembre 1983.*

MAEGAARD, SPANG HANSSEN (E.). Le programme segmentation. *Congrès ICCL, Pise 1973.*

MARCUS (Mitchell P.). A theory of syntactic recognition for natural language. *WINSTON (P.H.), BROWN (R.H.) ed. Artificial intelligence : an MIT perspective, volume 1, pp. 191-229.*

MERLE (Alain). Un analyseur présyntaxique pour la levée des ambiguïtés dans des documents écrits en langue naturelle : applications à l'indexation automatique. Thèse Docteur-Ingénieur, Génie Informatique, Institut National Polytechnique de Grenoble, 22/09/1982.

PAIR (Claude). Sur des notions algébriques liées à l'analyse syntaxique. *R.I.R.O., R-3, 1970, pp. 3-29.*

PRADILLA (Magdalena). Recherche de descripteurs en indexation automatique de documents. Université des Sciences Sociales de Grenoble, Thèse 3ème cycle, 1982.

ROUAULT (Jacques). Linguistique automatique et informatique documentaire. *Colloque Franco-Anglais, org. DBMIST, Paris, Décembre 1983.*

ROUAULT (Jacques). Linguistique automatique : applications documentaires. Berne, Lang, 1987.

SPARCK JONES (Karen), WILKS (Yorick), ed. Automatic natural language parsing. Chichester, Ellis Horwood Ltd., 1983.

TOMITA (Masaru). An efficient context-free parsing algorithm for natural languages and its applications. Ph. D. Thesis, Computer Science Department, Carnegie-Mellon University, May 1985.

VALIANT (Leslie G.). General context-free recognition in less than cubic time. *Journal of computer and system sciences 10, pp. 308-315, 1975.*

VAN RIJSBERGEN (C.J.). Information retrieval. Second edition. London, Butterworths, 1979.

ANNEXES

1 - Analyse combinatoire d'un syntagme nominal

2 - Analyse prédictive d'un syntagme nominal

3 - Analyse prédictive sur un résumé

ANNEXE 1

Analyse combinatoire d'un syntagme nominal

```
************   ANALYSE COMBINATOIRE - GRAMMAIRE ABREGEE   ***************

1. GRAMMAIRE

PHI -> N'' FIN      0     0
N -> F              1     1
A -> F              3     1
N -> N SP           0     8
N -> N A            5     4
SP -> P N''         0     5
N' -> N SP          4     2
N' -> N             0     6
N'' -> N'           0     6
N'' -> D N'         5     3
N' -> N SP SP       4     2

2. GENERATEUR

l' "l'" [D (DEF, GRN, SNG)]
e'tude "e'tude" [F (NOM, DVB, FEM, SNG)]
de "de" [P]
le "le" [D (DEF, MAS, SNG)]
fonctionnement "fonctionnement" [F(NOM, DVB, MAS, SNG)]
de "de" [P]
un "un" [D (NUM, MAS, SNG)]
e'cosyste`me "e'cosyste`me" [F(NOM, MAS, SNG)]
prairial "prairial" [F(ADJ, MAS, SNG)]$

3. RESULTATS


PHI
      N'' {FEM SNG }
            D   l' {DEF GRN SNG }
            N'  {FEM SNG }
                  N   {DVB FEM SNG }
                        F   e'tude {NOM DVB FEM SNG }

                  SP
                        P   de
                        N'' {DVB MAS SNG }
                              D   le {DEF MAS SNG }
                              N'  {DVB MAS SNG }
                                    N   {DVB MAS SNG }
                                          F   fonctionnement {NOM DVB MAS SNG }



                  SP
                        P   de
                        N'' {MAS SNG }
                              D   un {NUM MAS SNG }
                              N'  {MAS SNG }
                                    N   {MAS SNG }
                                          N   {MAS SNG }
                                                F   e'cosyste`me {NOM MAS SNG }

                                          A   {MAS SNG }
```

```
                                F   prairial {ADJ MAS SNG }

FIN

PHI
N'' {FEM SNG }
      D   l' {DEF GRN SNG }
      N'  {FEM SNG }
            N   {DVB FEM SNG }
                  N   {DVB FEM SNG }
                        F   e'tude {NOM DVB FEM SNG }

                  SP
                        P   de
                        N'' {DVB MAS SNG }
                              D   le {DEF MAS SNG }
                              N'  {DVB MAS SNG }
                                    N   {DVB MAS SNG }
                                          F   fonctionnement {NOM DVB MAS SNG }



            SP
                  P   de
                  N'' {MAS SNG }
                        D   un {NUM MAS SNG }
                        N'  {MAS SNG }
                              N   {MAS SNG }
                                    N   {MAS SNG }
                                          F   e'cosyste`me {NOM MAS SNG }

                                    A   {MAS SNG }
                                          F   prairial {ADJ MAS SNG }



FIN

PHI
N'' {FEM SNG }
      D   l' {DEF GRN SNG }
      N'  {FEM SNG }
            N   {DVB FEM SNG }
                  F   e'tude {NOM DVB FEM SNG }

            SP
                  P   de
                  N'' {DVB MAS SNG }
```

```
                    D   le {DEF MAS SNG }
                    N'  {DVB MAS SNG }
                      N   {DVB MAS SNG }
                        N   {DVB MAS SNG }
                          N   {DVB MAS SNG }
                            F   fonctionnement {NOM DVB MAS SNG }

                          SP
                            P   de
                            N'' {MAS SNG }
                              D   un {NUM MAS SNG }
                              N'  {MAS SNG }
                                N   {MAS SNG }
                                  F   e'cosyste`me {NOM MAS SNG }



                        A   {MAS SNG }
                          F   prairial {ADJ MAS SNG }



FIN

PHI
    N'' {FEM SNG }
      D   l' {DEF GRN SNG }
      N'  {FEM SNG }
        N   {DVB FEM SNG }
          F   e'tude {NOM DVB FEM SNG }

        SP
          P   de
          N'' {DVB MAS SNG }
            D   le {DEF MAS SNG }
            N'  {DVB MAS SNG }
              N   {DVB MAS SNG }
                N   {DVB MAS SNG }
                  F   fonctionnement {NOM DVB MAS SNG }

                SP
                  P   de
                  N'' {MAS SNG }
                    D   un {NUM MAS SNG }
                    N'  {MAS SNG }
                      N   {MAS SNG }
                        N   {MAS SNG }
                          F   e'cosyste`me {NOM MAS SNG }

                        A   {MAS SNG }
                          F   prairial {ADJ MAS SNG }
```

```
FIN

PHI
N'' {FEM SNG }
      D   l' {DEF GRN SNG }
      N'  {FEM SNG }
            N   {DVB FEM SNG }
                  F   e'tude {NOM DVB FEM SNG }

            SP
                  P   de
                  N'' {DVB MAS SNG }
                        D   le {DEF MAS SNG }
                        N'  {MAS SNG }
                              N   {DVB MAS SNG }
                                    F   fonctionnement {NOM DVB MAS SNG }

                              SP
                                    P   de
                                    N'' {MAS SNG }
                                          D   un {NUM MAS SNG }
                                          N'  {MAS SNG }
                                                N   {MAS SNG }
                                                      N . {MAS SNG }
                                                            F   e'cosyste`me {NOM MAS SNG }

                                                      A   {MAS SNG }
                                                            F   prairial {ADJ MAS SNG }

FIN

PHI
N'' {FEM SNG }
      D   l' {DEF GRN SNG }
      N'  {DVB FEM SNG }
            N   {DVB FEM SNG }
                  N   {DVB FEM SNG }
                        N   {DVB FEM SNG }
                              F   e'tude {NOM DVB FEM SNG }

                        SP
                              P   de
                              N'' {DVB MAS SNG }
```

```
                                    D    le {DEF MAS SNG }
                                    N'   {DVB MAS SNG }
                                      N    {DVB MAS SNG }
                                        F    fonctionnement {NOM DVB MAS SNG }



                  SP
                        P    de
                        N'' {MAS SNG }
                          D    un {NUM MAS SNG }
                          N'   {MAS SNG }
                            N    {MAS SNG }
                              N    {MAS SNG }
                                F    e'cosyste`me {NOM MAS SNG }

                              A    {MAS SNG }
                                F    prairial {ADJ MAS SNG }



FIN

PHI
      N'' {FEM SNG }
        D    l' {DEF GRN SNG }
        N'   {DVB FEM SNG }
          N    {DVB FEM SNG }
            N    {DVB FEM SNG }
              F    e'tude {NOM DVB FEM SNG }

            SP
                  P    de
                  N'' {DVB MAS SNG }
                    D    le {DEF MAS SNG }
                    N'   {DVB MAS SNG }
                      N    {DVB MAS SNG }
                        N    {DVB MAS SNG }
                          N    {DVB MAS SNG }
                            F    fonctionnement {NOM DVB MAS SNG }

                          SP
                                P    de
                                N'' {MAS SNG }
                                  D    un {NUM MAS SNG }
                                  N'   {MAS SNG }
                                    N    {MAS SNG }
                                      F    e'cosyste`me {NOM MAS SNG }
```

```
                                        A    (MAS SNG )
                                          F   prairial (ADJ MAS SNG )
```

```
FIN

PHI
N'' (FEM SNG )
      D   l' (DEF GRN SNG )
      N'  (DVB FEM SNG )
            N   (DVB FEM SNG )
                  N   (DVB FEM SNG )
                        F   e'tude (NOM DVB FEM SNG )

                  SP
                        P   de
                        N'' (DVB MAS SNG )
                              D   le (DEF MAS SNG )
                              N'  (DVB MAS SNG )
                                    N   (DVB MAS SNG )
                                          N   (DVB MAS SNG )
                                                F   fonctionnement (NOM DVB MAS SNG )

                                          SP
                                                P   de
                                                N'' (MAS SNG )
                                                      D   un (NUM MAS SNG )
                                                      N'  (MAS SNG )
                                                            N   (MAS SNG )
                                                                  N   (MAS SNG )
                                                                        F   e'cosyste`me (NOM MAS SNG )

                                                                  A   (MAS SNG )
                                                                        F   prairial (ADJ MAS SNG )
```

```
FIN

PHI
N'' (FEM SNG )
      D   l' (DEF GRN SNG )
      N'  (DVB FEM SNG )
            N   (DVB FEM SNG )
                  N   (DVB FEM SNG )
```

```
        F   e'tude {NOM DVB FEM SNG }

SP
        P   de
        N'' {DVB MAS SNG }
              D   le {DEF MAS SNG }
              N'  {MAS SNG }
                    N   {DVB MAS SNG }
                          F   fonctionnement {NOM DVB MAS SNG }

                    SP
                          P   de
                          N'' {MAS SNG }
                                D   un {NUM MAS SNG }
                                N'  {MAS SNG }
                                      N   {MAS SNG }
                                            N   {MAS SNG }
                                                  F   e'cosyste`me {NOM MAS SNG }

                                            A   {MAS SNG }
                                                  F   prairial {ADJ MAS SNG }

FIN
```

ANNEXE 2

Analyse prédictive d'un syntagme nominal

```
***************   ANALYSE PREDICTIVE - GRAMMAIRE ABREGEE  ****************

1. GRAMMAIRE

PHI -> N'' FIN      0     0
N -> F              1     1
A -> F              3     1
N -> N SP           0     8
N -> N A            5     4
SP -> P N''         0     5
N' -> N SP          4     2
N' -> N             0     6
N'' -> N'           0     6
N'' -> D N'         5     3
N' -> N SP SP       4     2

3. LEXIQUE


fonctionnement  [] [de] [] [] [] []
e'tude           [de , par] [de] [] [] [] []


2. GENERATEUR

l' "l'" [D (DEF, GRN, SNG)]
e'tude "e'tude" [F (NOM, DVB, FEM, SNG)]
de "de" [P]
le "le" [D (DEF, MAS, SNG)]
fonctionnement "fonctionnement" [F(NOM, DVB, MAS, SNG)]
de "de" [P]
un "un" [D (NUM, MAS, SNG)]
e'cosyste`me "e'cosyste`me" [F(NOM, MAS, SNG)]
prairial "prairial" [F(ADJ, MAS, SNG)]$

3.RESULTAT


PHI
      N'' (FEM SNG )
            D   l' (DEF GRN SNG )
            N'  (FEM SNG )
                  N   (DVB FEM SNG )
                        F   e'tude (NOM DVB FEM SNG )

                  SP
                        P   de
                        N'' (MAS SNG )
                              D   le (DEF MAS SNG )
                              N'  (MAS SNG )
                                    N   (DVB MAS SNG )
                                          F   fonctionnement (NOM DVB MAS SNG )

                                    SP
                                          P   de
                                          N'' (MAS SNG )
                                                D   un (NUM MAS SNG )
```

```
N'   (MAS SNG )
       N    (MAS SNG )
              N    (MAS SNG )
                     F    e'cosyste`me (NOM MAS SNG )

              A    (MAS SNG )
                     F    prairial (ADJ MAS SNG )
```

FIN

ANNEXE 3

Analyse prédictive sur un résumé

****

ANALYSE PREDICTIVE SUR UN RESUME

****

1. GRAMMAIRE

PHI -> N'' FIN	0	0
N -> F	1	1
N -> F	2	1
A -> F	3	1
A -> F	2	1
N -> N SP	0	8
N -> A' N	5	3
N -> N A''	5	4
A'' -> A' SP	4	2
A'' -> A'	0	2
SP -> P N''	0	5
A' -> A	0	6
A' -> W A	0	7
N' -> N SP	4	2
N' -> N	0	6
N'' -> N'	0	6
N'' -> D' N'	5	3
D' -> D	0	2
D' -> D D	6	9
N' -> N SP SP	4	2
N' -> N SP SP SP	4	2
A'' -> A' SP SP	4	2
A'' -> A' SP SP SP	4	2
N'' -> N'' N''	5	3
D' -> P D	10	7
K -> W	8	0
K -> D N	9	0
K -> D A'	11	0
N'' -> K P N''	12	0

2. LEXIQUE

fonctionnement [] [de] [] [] [] []
association [de, par] [de] [] [] [] []
e'tude [de , par] [de] [] [] [] []
production [de, par] [de] [] [] [] []
re'sistance [de] [] [a`] [] [] []
infection [de, par] [de] [] [] [] []
agissant [] [] [] [a`, dans, lors] [] []
e'volution [de] [] [] [] [] []
confronte' [par] [] [] [a`] [] []
re'flexion [de] [] [] [sur] [] []
emploi [de, par] [de] [] [] [] []
propre [] [] [] [a`] [] []
rendu [par] [] [] [] [] []
ta^tonnement [par, de] [de] [] [] [] []
recherche [] [sur] [] [] [] []
projet [] [de] [] [] [] []
mode`le [] [de] [] [] [] []

```
travaillant [] [] [] [sur] [] []
s'inte'grant [] [] [] [dans] [] []
construction [de, par] [de] [] [] [] []
de'finition [de, par] [de] [] [] [] []
comportement [de] [] [] [] [] []
mesure [de, par] [de] [] [] [] []
pose' [par] [] [] [] [] []
pre'occupation [de] [] [] [] [] []
```

3. GENERATEURS

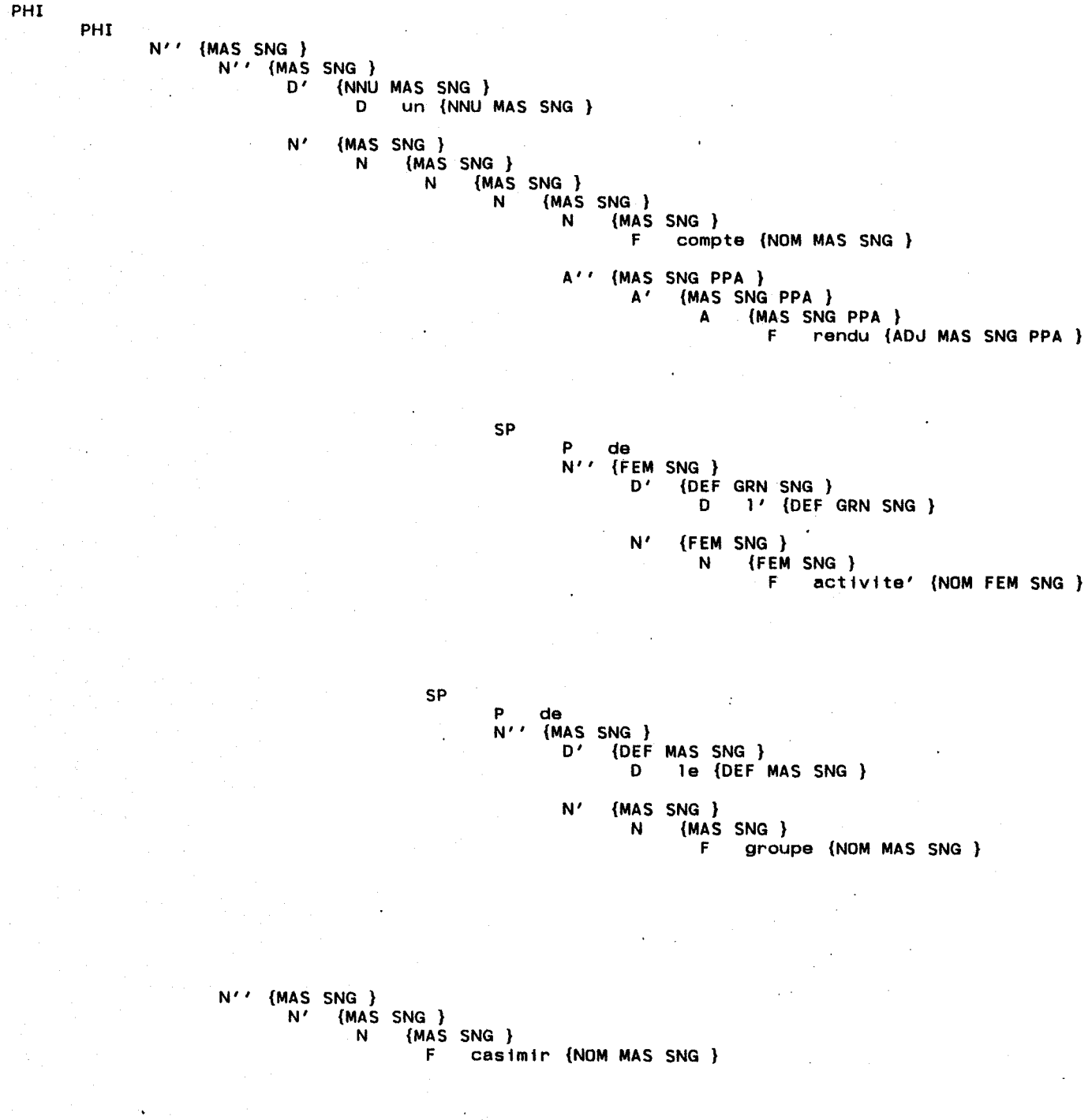
```
l' "l'" [D (DEF, GRN, SNG)]
article "article" [F (NOM, MAS, SNG)]$
un  "un" [D(NNU, MAS, SNG)]
compte "compte" [F(NOM, MAS, SNG)]
rendu "rendu" [F(ADJ, MAS, SNG, DVB, PPA)]
de "de" [P]
l' "l'" [D(DEF, GRN, SNG)]
activite' "activite'" [F(NOM, FEM, SNG)]
de "de" [P]
le "le" [D(DEF, MAS, SNG)]
groupe "groupe" [F(NOM, MAS, SNG)]
casimir "casimir" [F(NOM, MAS, SNG)]$
l' "l'" [D(DEF, GRN, SNG)]
e'gide "e'gide" [F(NOM, FEM, SNG)]
de "de" [P]
la "la" [D(DEF, FEM, SNG)]
DGRST "DGRST" [F(NOM, FEM, SNG)]$
le "le" [D(DEF, MAS, SNG)]
CNRS "CNRS" [F(NOM, MAS, SNG)]$
l' "l'" [D(DEF, GRN, SNG)]
e'tude "e'tude" [F(NOM, DVB, FEM, SNG)]
de "de" [P]
le "le" [D(DEF, MAS, SNG)]
fonctionnement "fonctionnement" [F(NOM, DVB, MAS, SNG)]
de "de" [P]
l' "l'" [D(DEF, GRN, SNG)]
e'cosyste`me "e'cosyste`me" [F(NOM, MAS, SNG)]
prairial "prairial" [F(ADJ, MAS, SNG)]$
une "une" [D(NNU, FEM, SNG)]
phase "phase" [F(NOM, FEM, SNG)]
initiale "initial" [F(ADJ, FEM, SNG)]
de "de" [P]
ta^tonnements "ta^tonnement" [F(DVB, NOM, MAS, PLU)]$
la "la" [D(DEF, FEM, SNG)]
recherche "recherche" [F(DVB, NOM, FEM, SNG)]$
un "un" [D(NNU, MAS, SNG)]
projet "projet" [F(DVB, NOM, MAS, SNG)]
de "de" [P]
mode`le "mode`le" [F(DVB, NOM, MAS, SNG)]
de "de" [P]
e'cosyste`me "e'cosyste`me" [F(NOM, MAS, SNG)]
prairial "prairial" [F(ADJ, MAS, SNG)]
fonctionnel "fonctionnel" [F(ADJ, MAS, SNG)]
de "de" [P]
vocation "vocation" [F(NOM, FEM, SNG)]
ge'ne'rale "ge'ne'ral" [F(ADJ, FEM, SNG)]$
un "un" [D(NNU, MAS, SNG)]
```

```
mode`le "mode`le" [F(DVB, NOM, MAS, SNG)]
pour "pour" [P]
une "un" [D (NNU, FEM, SNG)]
localite' "localite'" [F(NOM, FEM, SNG)]
de'finie "de'fini" [F(ADJ, FEM, SNG)]$
l' "l'" [D( GRN, SNG)]
e'quipe "e'quipe" [F(NOM, FEM, SNG)]$
terrain "terrain" [F(NOM, MAS, SNG]
commun "commun" [F(ADJ, MAS, SNG)]
de "de" [P]
recherche "recherche" [F(NOM, DVB, FEM, SNG)]$
chaque "chaque" [D(DEF, MAS, SNG)]
laboratoire "laboratoire" [F(NOM, MAS, SNG)]
travaillant "travaillant" [F(DVB, ADJ, GRN, NBN, PPR)]
sur "sur" [P]
un "un" [D(NNU, MAS, SNG)]
me'canisme "me'canisme" [F(NOM, MAS, SNG)]
s'inte'grant "s'inte'grant" [F(DVB, ADJ, GRN, NBN, PPR)]
dans "dans" [P]
le "le" [D(DEF, MAS, SNG)]
projet "projet" [F(DVB, NOM, MAS, SNG)]
de "de" [P]
mode`le "mode`le" [F(DVB, NOM, MAS, SNG)]$
moindre "moindre" [F(ADJ, GRN, SNG)]
cou^t "cou^t" [F(NOM, MAS, SNG)]$
l' "l'" [D( GRN, SNG)]
activite' "activite'" [F(NOM, FEM, SNG)]
de "de" [P]
le "le" [D(DEF, MAS, SNG)]
groupe "groupe" [F(NOM, MAS, SNG)]$
la "la" [D(DEF, FEM, SNG)]
construction "construction" [F(NOM, DVB, FEM, SNG)]
de "de" [P]
mode`les "mode`le" [F(DVB, NOM, MAS, PLU)]
partiels "partiel" [F(ADJ, MAS, PLU)]$
la "la" [D(DEF, FEM, SNG)]
de'finition "de'finition" [F(NOM, DVB, FEM, SNG)]
ope'rationnelle "ope'rationnel" [F(ADJ, FEM, SNG)]
de "de" [P]
les "le" [D(DEF, GRN, PLU)]
comportements "comportement" [F(DVB, NOM, MAS, PLU)]$
les "le" [D(DEF, GRN, PLU)]
flux "flux" [F(NOM, MAS, NBN)]$
le "le" [D(DEF, MAS, SNG)]
cas "cas" [F(NOM, MAS, SNG)]
de "de" [P]
les "le" [D(DEF, GRN, PLU)]
sols "sol" [F(NOM, MAS, PLU)]$
proble`mes "proble`me" [F(NOM, MAS, SNG)]
difficiles "difficile" [F(ADJ, GRN, SNG)]$
ces "ce" [D(DEF, MAS, PLU)]
derniers "dernier" [F(NAN, MAS, PLU)]$
des "des" [D(DEF, GRN, PLU)]
recherches "recherche" [F(DVB, NOM, FEM, PLU)]
sur "sur" [P]
la "la" [D(DEF, FEM, SNG)]
mesure "mesure" [F(DVB, NOM, FEM, SNG)]
de "de" [P]
les "les" [D(DEF, GRN, PLU)]
flux "flux" [F(NOM, MAS, NBN)]
```

```
de "de" [P]
carbone "carbone" [F(NOM, MAS, SNG)]$
azote "azote" [F(NOM, MAS, SNG)]$
des "des" [D(DEF, GRN, PLU)]
donne'es "donne'e" [F(NOM, FEM, PLU)]
expe'rimentales "expe'rimental" [F(ADJ, FEM, PLU)]
valables "valable" [F(ADJ, GRN, PLU)]$
les "le" [D(DEF, GRN, PLU)]
dernie`res "dernier" [F(NAN, FEM, PLU)]
anne'es "anne'e" [F(NOM, FEM, PLU)]$
les "le" [D(DEF, GRN, PLU)]
proble`mes "proble`me" [F(NOM, MAS, PLU)]
pose's "pose'" [F(ADJ, MAS, PLU, DVB, PPA)]
par "par" [P]
le "le" [D(DEF, MAS, SNG)]
fonctionnement "fonctionnement" [F(NOM, DVB, MAS, SNG)]
de "de" [P]
une "un"  [D(NNU, FEM, SNG)]
e'quipe "e'quipe" [F(NOM, FEM, SNG)]
plurilocalise'e "plurilocalise'" [F(ADJ, FEM, SNG)]$
le "le" [D(DEF, MAS, SNG)]
centre "centre" [F(NOM, MAS, SNG)]
de "de" [P]
les "le" [D(DEF, GRN, PLU)]
pre'occupations "pre'occupation" [F(DVB, NOM, FEM, PLU)]$
les "le" [D(DEF, GRN, PLU)]
proble`mes "proble`me" [F(NOM, MAS, PLU)]
pose's "pose'" [F(ADJ, MAS, PLU, DVB, PPA)]
par "par" [P]
le "le" [D(DEF, MAS, SNG)]
cycle "cycle" [F(NOM, MAS, SNG)]
de "de" [P]
l' "l'" [D(DEF, GRN, SNG)]
azote "azote" [F(NOM, MAS, SNG)]$
les "le" [D(DEF, GRN, PLU)]
avantages "avantage" [F(NOM, FEM, PLU)]
de "de" [P]
le "le" [D(DEF, MAS, SNG)]
syste`me "syste`me" [F(NOM, MAS, SNG)]$
```

4. RESULTATS

```
PHI
      PHI
            N'' {MAS SNG }
                  D'  {DEF GRN SNG }
                        D   l' {DEF GRN SNG }

                  N'  {MAS SNG }
                        N   {MAS SNG }
                              F   article {NOM MAS SNG }


            FIN
```


PHI
PHI
N'' {MAS SNG }
N'' {MAS SNG }
D' {NNU MAS SNG }
D un {NNU MAS SNG }
N' {MAS SNG }
N {MAS SNG }
N {MAS SNG }
N {MAS SNG }
N {MAS SNG }
F compte {NOM MAS SNG }
A'' {MAS SNG PPA }
A' {MAS SNG PPA }
A {MAS SNG PPA }
F rendu {ADJ MAS SNG PPA }
SP
P de
N'' {FEM SNG }
D' {DEF GRN SNG }
D l' {DEF GRN SNG }
N' {FEM SNG }
N {FEM SNG }
F activite' {NOM FEM SNG }
SP
P de
N'' {MAS SNG }
D' {DEF MAS SNG }
D le {DEF MAS SNG }
N' {MAS SNG }
N {MAS SNG }
F groupe {NOM MAS SNG }
N'' {MAS SNG }
N' {MAS SNG }
N {MAS SNG }
F casimir {NOM MAS SNG }

```
FIN

PHI
PHI
N'' {MAS SNG }
  N'' {MAS SNG }
    D'  {NNU MAS SNG }
      D   un {NNU MAS SNG }

    N'  {MAS SNG }
      N   {MAS SNG }
        N   {MAS SNG }
          N   {MAS SNG }
            F   compte {NOM MAS SNG }

          A'' {MAS SNG PPA }
            A'  {MAS SNG PPA }
              A   {MAS SNG PPA }
                F   rendu {ADJ MAS SNG PPA }



        SP
          P   de
          N'' {FEM SNG }
            D'  {DEF GRN SNG }
              D   l' {DEF GRN SNG }

            N'  {FEM SNG }
              N   {FEM SNG }
                N   {FEM SNG }
                  F   activite' {NOM FEM SNG }

                SP
                  P   de
                  N'' {MAS SNG }
                    D'  {DEF MAS SNG }
                      D   le {DEF MAS SNG }

                    N'  {MAS SNG }
                      N   {MAS SNG }
                        F   groupe {NOM MAS SNG }

  N'' {MAS SNG }
    N'  {MAS SNG }
      N   {MAS SNG }
```

```
                                F   casimir {NOM MAS SNG }


        FIN


PHI
      PHI
        N'' {FEM SNG }
          D'  {DEF GRN SNG }
                D   l' {DEF GRN SNG }

          N'  {FEM SNG }
                N   {FEM SNG }
                      N   {FEM SNG }
                            F   e'gide {NOM FEM SNG }

                      SP
                            P   de
                            N'' {FEM SNG }
                              D'  {DEF FEM SNG }
                                    D   la {DEF FEM SNG } .

                              N'  {FEM SNG }
                                    N   {FEM SNG }
                                          F   DGRST {NOM FEM SNG }




        FIN


PHI
      PHI
        N'' {MAS SNG }
          D'  {DEF MAS SNG }
                D   le {DEF MAS SNG }

          N'  {MAS SNG }
                N   {MAS SNG }
                      F   CNRS {NOM MAS SNG }


        FIN


PHI
      PHI
        N'' {FEM SNG }
          D'  {DEF GRN SNG }
```

```
                D   l' {DEF GRN SNG }

          N'  {FEM SNG }
                N   {DVB FEM SNG }
                      F   e'tude {NOM DVB FEM SNG }

                SP
                      P   de
                      N'' {MAS SNG }
                            D'  {DEF MAS SNG }
                                  D   le {DEF MAS SNG }

                            N'  {MAS SNG }
                                  N   {DVB MAS SNG }
                                        F   fonctionnement {NOM DVB MAS SNG }

                                  SP
                                        P   de
                                        N'' {MAS SNG }
                                              D'  {DEF GRN SNG }
                                                    D   l' {DEF GRN SNG }

                                              N'  {MAS SNG }
                                                    N   {MAS SNG }
                                                          N   {MAS SNG }
                                                                F   e'cosyste`me {NOM MAS SNG }

                                                          A'' {MAS SNG }
                                                                A'  {MAS SNG }
                                                                      A   {MAS SNG }
                                                                            F   prairial {ADJ MAS SNG }
```

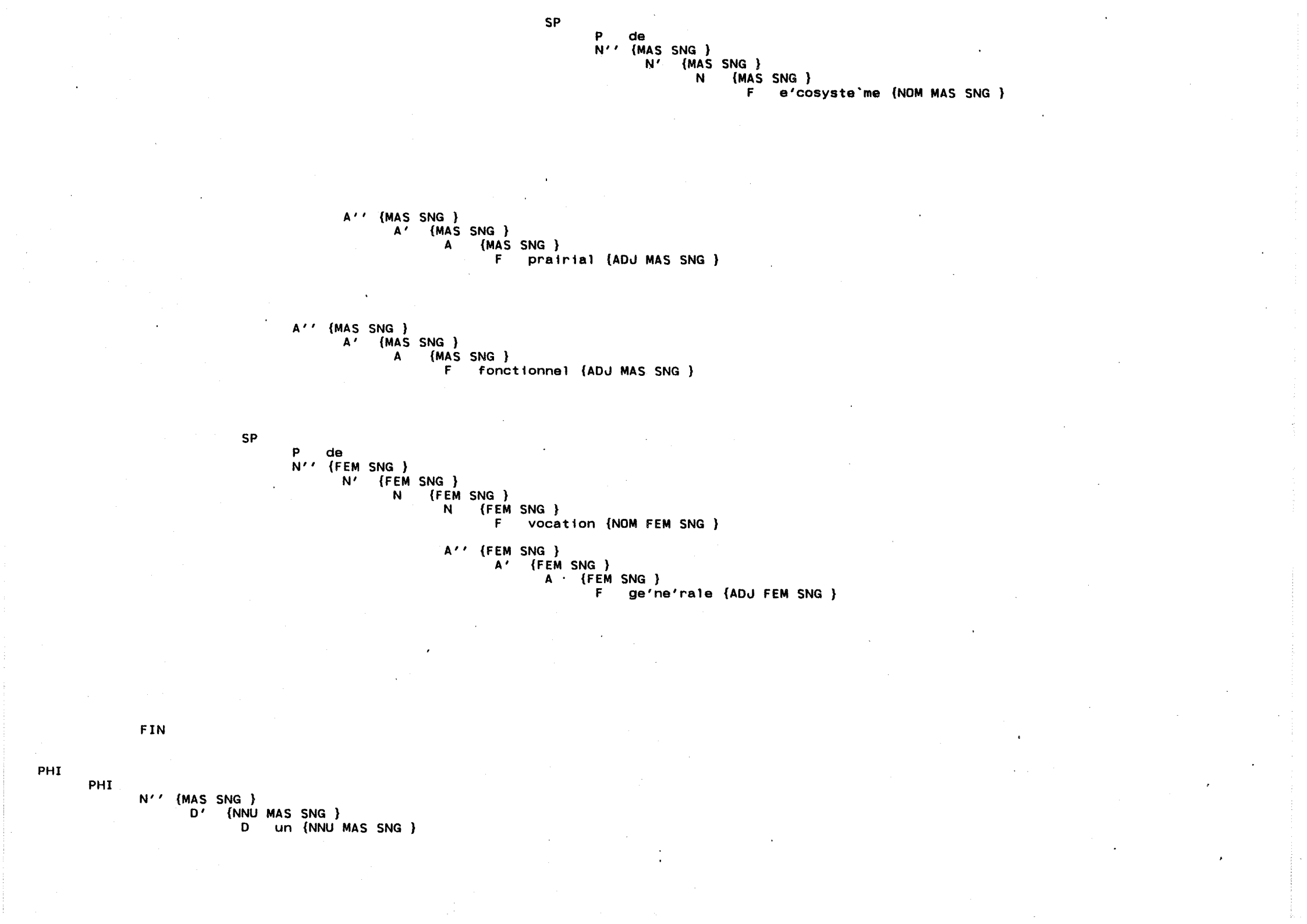
FIN

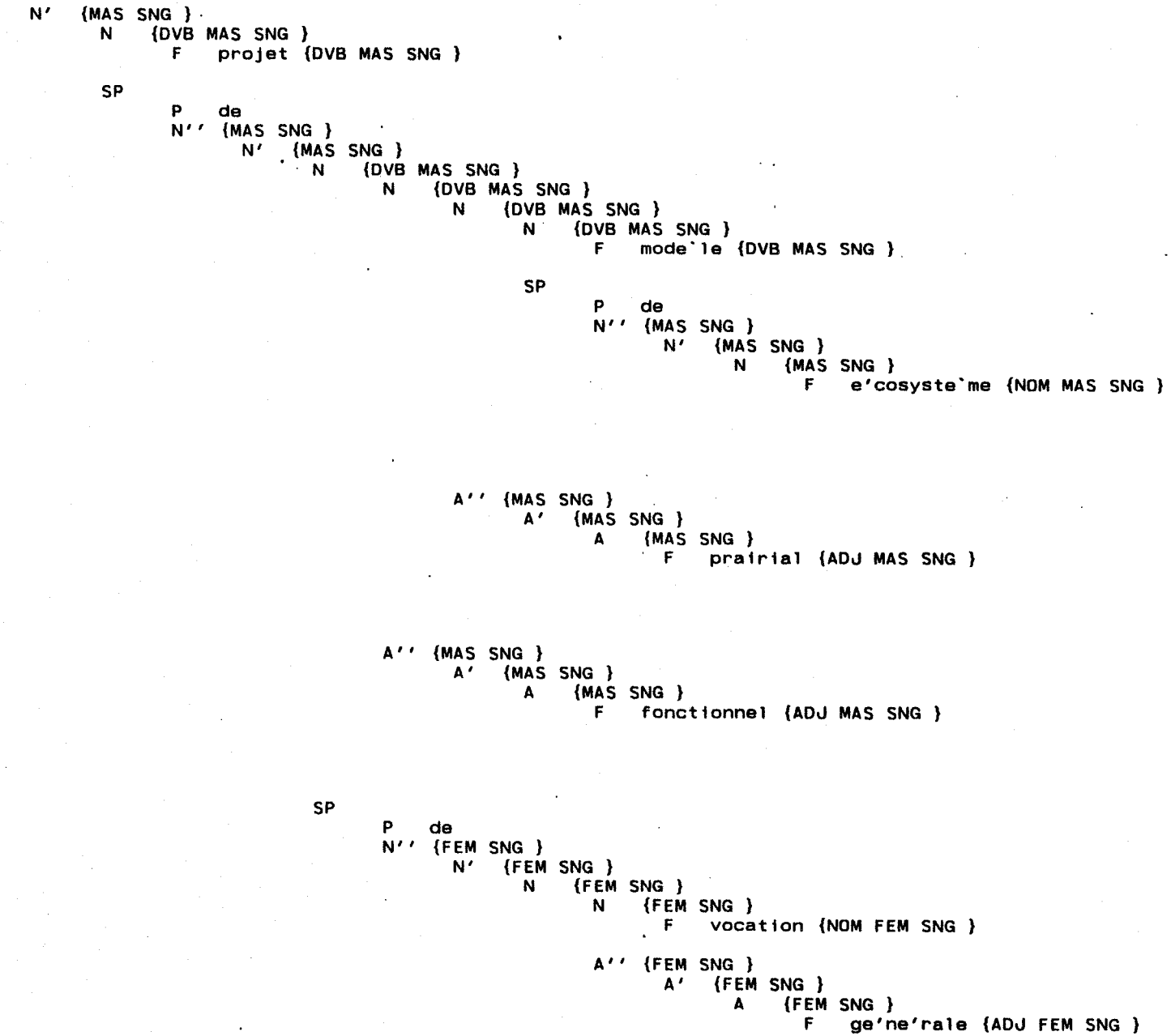
PHI

PHI

```
N'' {FEM SNG }
      D'  {NNU FEM SNG }
            D   une {NNU FEM SNG }

      N'  {FEM SNG }
            N   {FEM SNG }
                  N   {FEM SNG }
                        N   {FEM SNG }
                              F   phase {NOM FEM SNG }

                  A'' {FEM SNG }
```

```
                                    A'  {FEM SNG }
                                      A   {FEM SNG }
                                        F   initiale {ADJ FEM SNG }


                              SP
                                    P   de
                                    N'' {DVB MAS PLU }
                                      N'  {DVB MAS PLU }
                                        N   {DVB MAS PLU }
                                          F   ta^tonnements {DVB MAS PLU }


        FIN


PHI
      PHI
        N'' {DVB FEM SNG }
              D'  {DEF FEM SNG }
                D   la {DEF FEM SNG }

              N'  {DVB FEM SNG }
                N   {DVB FEM SNG }
                  F   recherche {DVB FEM SNG }


        FIN


PHI
      PHI
        N'' {MAS SNG }
              D'  {NNU MAS SNG }
                D   un {NNU MAS SNG }

              N'  {MAS SNG }
                N   {DVB MAS SNG }
                  N   {DVB MAS SNG }
                    N   {DVB MAS SNG }
                      N   {DVB MAS SNG }
                        F   projet {DVB MAS SNG }

                    SP
                      P   de
                      N'' {MAS SNG }
                        N'  {MAS SNG }
                          N   {DVB MAS SNG }
                            F   mode`le {DVB MAS SNG }
```


SP
P de
N'' (MAS SNG)
N' (MAS SNG)
N (MAS SNG)
F e'cosyste`me (NOM MAS SNG)
A'' (MAS SNG)
A' (MAS SNG)
A (MAS SNG)
F prairial (ADJ MAS SNG)
A'' (MAS SNG)
A' (MAS SNG)
A (MAS SNG)
F fonctionnel (ADJ MAS SNG)
SP
P de
N'' (FEM SNG)
N' (FEM SNG)
N (FEM SNG)
N (FEM SNG)
F vocation (NOM FEM SNG)
A'' (FEM SNG)
A' (FEM SNG)
A (FEM SNG)
F ge'ne'rale (ADJ FEM SNG)
FIN
PHI
PHI
N'' (MAS SNG)
D' (NNU MAS SNG)
D un (NNU MAS SNG)


N' {MAS SNG }
N {DVB MAS SNG }
F projet {DVB MAS SNG }
SP
P de
N'' {MAS SNG }
N' {MAS SNG }
N {DVB MAS SNG }
N {DVB MAS SNG }
N {DVB MAS SNG }
N {DVB MAS SNG }
F mode`le {DVB MAS SNG }
SP
P de
N'' {MAS SNG }
N' {MAS SNG }
N {MAS SNG }
F e'cosyste`me {NOM MAS SNG }
A'' {MAS SNG }
A' {MAS SNG }
A {MAS SNG }
F prairial {ADJ MAS SNG }
A'' {MAS SNG }
A' {MAS SNG }
A {MAS SNG }
F fonctionnel {ADJ MAS SNG }
SP
P de
N'' {FEM SNG }
N' {FEM SNG }
N {FEM SNG }
N {FEM SNG }
F vocation {NOM FEM SNG }
A'' {FEM SNG }
A' {FEM SNG }
A {FEM SNG }
F ge'ne'rale {ADJ FEM SNG }

```
        FIN

PHI
    PHI
        N'' {MAS SNG }
          D'  {NNU MAS SNG }
                D   un {NNU MAS SNG }

          N'  {MAS SNG }
                N   {DVB MAS SNG }
                  F   projet {DVB MAS SNG }

                SP
                      P   de
                      N'' {MAS SNG }
                            N'  {MAS SNG }
                                  N   {DVB MAS SNG }
                                    F   mode`le {DVB MAS SNG }

                                  SP
                                        P   de
                                        N'' {MAS SNG }
                                              N'  {MAS SNG }
                                                    N   {MAS SNG }
                                                          N   {MAS SNG }
                                                                N   {MAS SNG }
                                                                      N   {MAS SNG }
                                                                            N   {MAS SNG }
                                                                              F   e'cosyste`me {NOM MAS SNG }

                                                                      A'' {MAS SNG }
                                                                            A'  {MAS SNG }
                                                                                  A   {MAS SNG }
                                                                                    F   prairial {ADJ MAS SNG }



                                                          A'' {MAS SNG }
                                                                A'  {MAS SNG }
                                                                      A   {MAS SNG }
                                                                        F   fonctionnel {ADJ MAS SNG }



                                                    SP
                                                          P   de
                                                          N'' {FEM SNG }
                                                                N'  {FEM SNG }
                                                                      N   {FEM SNG }
                                                                            N   {FEM SNG }
                                                                              F   vocation {NOM FEM SNG }

                                                                      A'' {FEM SNG }
```

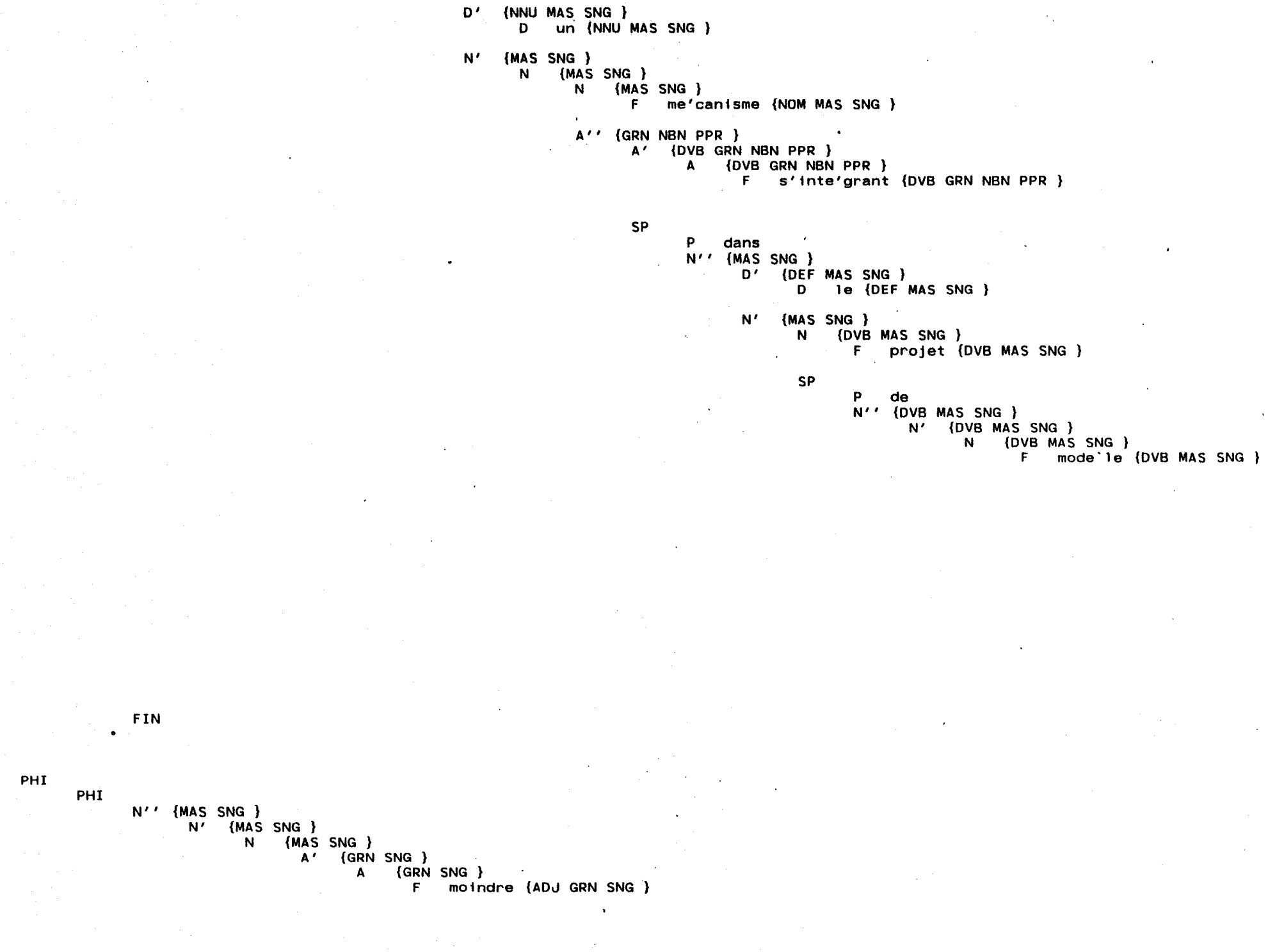
```
                                                                A'  {FEM SNG }
                                                                  A   {FEM SNG }
                                                                    F   ge'ne'rale {ADJ FEM SNG
}

          FIN


PHI
      PHI
          N'' {MAS SNG }
                D'  {NNU MAS SNG }
                      D   un {NNU MAS SNG }

                N'  {MAS SNG }
                      N   {DVB MAS SNG }
                            F   mode`le {DVB MAS SNG }

                      SP
                            P   pour
                            N'' {FEM SNG }
                                  D'  {NNU FEM SNG }
                                        D   une {NNU FEM SNG }

                                  N'  {FEM SNG }
                                        N   {FEM SNG }
                                              N   {FEM SNG }
                                                    F   localite' {NOM FEM SNG }

                                        A'' {FEM SNG }
                                              A'  {FEM SNG }
                                                    A   {FEM SNG }
                                                          F   de'finie {ADJ FEM SNG }




          FIN
```

```
PHI
    PHI
        N'' {MAS SNG }
            D'  {NNU MAS SNG }
                D   un {NNU MAS SNG }

            N'  {DVB MAS SNG }
                N   {DVB MAS SNG }
                    N   {DVB MAS SNG }
                        F   mode`le {DVB MAS SNG }

                    SP
                        P   pour
                        N'' {FEM SNG }
                            D'  {NNU FEM SNG }
                                D   une {NNU FEM SNG }

                            N'  {FEM SNG }
                                N   {FEM SNG }
                                    N   {FEM SNG }
                                        F   localite' {NOM FEM SNG }

                                    A'' {FEM SNG }
                                        A'  {FEM SNG }
                                            A   {FEM SNG }
                                                F   de'finie {ADJ FEM SNG }

        FIN


PHI
    PHI
        N'' {FEM SNG }
            D'  {GRN SNG }
                D   l' {GRN SNG }

            N'  {FEM SNG }
                N   {FEM SNG }
                    F   e'quipe {NOM FEM SNG }


        FIN


PHI
    PHI
        N'' {MAS SNG }
```

```
                N'  {MAS SNG }
                  N   {MAS SNG }
                        N   {MAS SNG }
                              N   {MAS SNG }
                                    F   terrain {NOM MAS SNG }

                              A'' {MAS SNG }
                                    A'  {MAS SNG }
                                          A   {MAS SNG }
                                                F   commun {ADJ MAS SNG }



                              SP
                                    P   de
                                    N'' {DVB FEM SNG }
                                          N'  {DVB FEM SNG }
                                                N   {DVB FEM SNG }
                                                      F   recherche {NOM DVB FEM SNG }

        FIN


PHI
      PHI
        N'' {MAS SNG }
              D'  {DEF MAS SNG }
                    D   chaque {DEF MAS SNG }

              N'  {MAS SNG }
                    N   {MAS SNG }
                          N   {MAS SNG }
                                N   {MAS SNG }
                                      F   laboratoire {NOM MAS SNG }

                          A'' {GRN NBN PPR }
                                A'  {DVB GRN NBN PPR }
                                      A   {DVB GRN NBN PPR }
                                            F   travaillant {DVB GRN NBN PPR }


                                SP
                                      P   sur
                                      N'' {MAS SNG }
                                            D'  {NNU MAS SNG }
                                                  D   un {NNU MAS SNG }

                                            N'  {MAS SNG }
                                                  N   {MAS SNG }
                                                        F   me'canisme {NOM MAS SNG }
```

```
                    A''  {GRN NBN PPR }
                      A'   {DVB GRN NBN PPR }
                        A    {DVB GRN NBN PPR }
                          F    s'inte'grant {DVB GRN NBN PPR }


                      SP
                          P    dans
                          N''  {MAS SNG }
                            D'   {DEF MAS SNG }
                              D    le {DEF MAS SNG }

                            N'   {MAS SNG }
                              N    {DVB MAS SNG }
                                F    projet {DVB MAS SNG }

                              SP
                                  P    de
                                  N''  {DVB MAS SNG }
                                    N'   {DVB MAS SNG }
                                      N    {DVB MAS SNG }
                                        F    mode`le {DVB MAS SNG }
```

```
    FIN


PHI
  PHI
    N''  {MAS SNG }
      D'   {DEF MAS SNG }
        D    chaque {DEF MAS SNG }

      N'   {MAS SNG }
        N    {MAS SNG }
          N    {MAS SNG }
            F    laboratoire {NOM MAS SNG }

          A''  {GRN NBN PPR }
            A'   {DVB GRN NBN PPR }
              A    {DVB GRN NBN PPR }
                F    travaillant {DVB GRN NBN PPR }


            SP
                P    sur
                N''  {MAS SNG }
```


D' {NNU MAS SNG }
D un {NNU MAS SNG }
N' {MAS SNG }
N {MAS SNG }
N {MAS SNG }
F me'canisme {NOM MAS SNG }
A'' {GRN NBN PPR }
A' {DVB GRN NBN PPR }
A {DVB GRN NBN PPR }
F s'inte'grant {DVB GRN NBN PPR }
SP
P dans
N'' {MAS SNG }
D' {DEF MAS SNG }
D le {DEF MAS SNG }
N' {MAS SNG }
N {DVB MAS SNG }
F projet {DVB MAS SNG }
SP
P de
N'' {DVB MAS SNG }
N' {DVB MAS SNG }
N {DVB MAS SNG }
F mode`le {DVB MAS SNG }
FIN
PHI
PHI
N'' {MAS SNG }
N' {MAS SNG }
N {MAS SNG }
A' {GRN SNG }
A {GRN SNG }
F moindre {ADJ GRN SNG }

```
                    N    {MAS SNG }
                      F    cou^t {NOM MAS SNG }



      FIN


PHI
    PHI
      N'' {FEM SNG }
        D'  {GRN SNG }
          D   1' {GRN SNG }

        N'  {FEM SNG }
          N   {FEM SNG }
            N   {FEM SNG }
              F   activite' {NOM FEM SNG }

            SP
              P   de
              N'' {MAS SNG }
                D'  {DEF MAS SNG }
                  D   le {DEF MAS SNG }

                N'  {MAS SNG }
                  N   {MAS SNG }
                    F   groupe {NOM MAS SNG }




      FIN


PHI
    PHI
      N'' {FEM SNG }
        D'  {DEF FEM SNG }
          D   la {DEF FEM SNG }

        N'  {FEM SNG }
          N   {DVB FEM SNG }
            F   construction {NOM DVB FEM SNG }

          SP
            P   de
            N'' {DVB MAS PLU }
              N'  {DVB MAS PLU }
                N   {DVB MAS PLU }
                  N   {DVB MAS PLU }
                    F   mode`les {DVB MAS PLU }

                  A'' {MAS PLU }
```

```
                                        A'  (MAS PLU )
                                          A   (MAS PLU )
                                            F   partiels (ADJ MAS PLU )


        FIN


PHI
      PHI
        N'' (FEM SNG )
              D'  (DEF FEM SNG )
                    D   la (DEF FEM SNG )

              N'  (FEM SNG )
                    N   (DVB FEM SNG )
                          N   (DVB FEM SNG )
                                F   de'finition (NOM DVB FEM SNG )

                          A'' (FEM SNG )
                                A'  (FEM SNG )
                                      A   (FEM SNG )
                                        F   ope'rationnelle (ADJ FEM SNG )


                    SP
                          P   de
                          N'' (DVB MAS PLU )
                                D'  (DEF GRN PLU )
                                      D   les (DEF GRN PLU )

                                N'  (DVB MAS PLU )
                                      N   (DVB MAS PLU )
                                        F   comportements (DVB MAS PLU )


        FIN


PHI
      PHI
        N'' (MAS PLU )
              D'  (DEF GRN PLU )
                    D   les (DEF GRN PLU )
```

```
                N'  {MAS PLU }
                  N   {MAS PLU }
                        F   flux {NOM MAS PLU }


        FIN


PHI
    PHI
        N'' {MAS SNG }
              D'  {DEF MAS SNG }
                    D   le {DEF MAS SNG }

              N'  {MAS SNG }
                    N   {MAS SNG }
                          N   {MAS SNG }
                                F   cas {NOM MAS SNG }

                          SP
                                P   de
                                N'' {MAS PLU }
                                      D'  {DEF GRN PLU }
                                            D   les {DEF GRN PLU }

                                      N'  {MAS PLU }
                                            N   {MAS PLU }
                                                  F   sols {NOM MAS PLU }




        FIN


PHI
    PHI
        N'' {MAS SNG }
              N'  {MAS SNG }
                    N   {MAS SNG }
                          N   {MAS SNG }
                                F   proble`mes {NOM MAS SNG }

                          A'' {GRN SNG }
                                A'  {GRN SNG }
                                      A   {GRN SNG }
                                            F   difficiles {ADJ GRN SNG }



        FIN
```

```
PHI
      PHI
            N'' {MAS PLU }
                  D'  {DEF MAS PLU }
                        D   ces {DEF MAS PLU }

                  N'  {MAS PLU }
                        N   {MAS PLU }
                              F   derniers {NAN MAS PLU }



            FIN


PHI
      PHI
            N'' {FEM PLU }
                  D'  {DEF GRN PLU }
                        D   des {DEF GRN PLU }

                  N'  {FEM PLU }
                        N   {DVB FEM PLU }
                              N   {DVB FEM PLU }
                                    F   recherches {DVB FEM PLU }

                              SP
                                    P   sur
                                    N'' {FEM SNG }
                                          D'  {DEF FEM SNG }
                                                D   la {DEF FEM SNG }

                                          N'  {FEM SNG }
                                                N   {DVB FEM SNG }
                                                      F   mesure {DVB FEM SNG }

                                                SP
                                                      P   de
                                                      N'' {MAS PLU }
                                                            D'  {DEF GRN PLU }
                                                                  D   les {DEF GRN PLU }

                                                            N'  {MAS PLU }
                                                                  N   {MAS PLU }
                                                                        F   flux {NOM MAS PLU }




                        SP
                              P   de
                              N'' {MAS SNG }
```

```
                                N'  {MAS SNG }
                                  N   {MAS SNG }
                                    F   carbone {NOM MAS SNG }




        FIN


PHI
    PHI
        N'' {FEM PLU }
            D'  {DEF GRN PLU }
                D   des {DEF GRN PLU }

            N'  {FEM PLU }
                N   {DVB FEM PLU }
                    F   recherches {DVB FEM PLU }

                SP
                    P   sur
                    N'' {FEM SNG }
                        D'  {DEF FEM SNG }
                            D   la {DEF FEM SNG }

                        N'  {FEM SNG }
                            N   {DVB FEM SNG }
                                N   {DVB FEM SNG }
                                    F   mesure {DVB FEM SNG }

                                SP
                                    P   de
                                    N'' {MAS PLU }
                                        D'  {DEF GRN PLU }
                                            D   les {DEF GRN PLU }

                                        N'  {MAS PLU }
                                            N   {MAS PLU }
                                                F   flux {NOM MAS PLU }



                            SP
                                P   de
                                N'' {MAS SNG }
                                    N'  {MAS SNG }
                                        N   {MAS SNG }
                                            F   carbone {NOM MAS SNG }
```

```
FIN

PHI
PHI
N'' {FEM PLU }
  D'  {DEF GRN PLU }
    D   des {DEF GRN PLU }

  N'  {FEM PLU }
    N   {DVB FEM PLU }
      F   recherches {DVB FEM PLU }

    SP
      P   sur
      N'' {FEM SNG }
        D'  {DEF FEM SNG }
          D   la {DEF FEM SNG }

        N'  {FEM SNG }
          N   {DVB FEM SNG }
            F   mesure {DVB FEM SNG }

          SP
            P   de
            N'' {MAS PLU }
              D'  {DEF GRN PLU }
                D   les {DEF GRN PLU }

              N'  {MAS PLU }
                N   {MAS PLU }
                  N   {MAS PLU }
                    F   flux {NOM MAS PLU }

                SP
                  P   de
                  N'' {MAS SNG }
                    N'  {MAS SNG }
                      N   {MAS SNG }
                        F   carbone {NOM MAS SNG }

FIN

PHI
```

```
PHI
      N'' {MAS SNG }
            N'  {MAS SNG }
                  N   {MAS SNG }
                        F   azote {NOM MAS SNG }


      FIN


PHI
      PHI
      N'' {FEM PLU }
            D'  {DEF GRN PLU }
                  D   des {DEF GRN PLU }

            N'  {FEM PLU }
                  N   {FEM PLU }
                        N   {FEM PLU }
                              N   {FEM PLU }
                                    F   donne'es {NOM FEM PLU }

                              A'' {FEM PLU }
                                    A'  {FEM PLU }
                                          A   {FEM PLU }
                                                F   expe'rimentales {ADJ FEM PLU }



                        A'' {GRN PLU }
                              A'  {GRN PLU }
                                    A   {GRN PLU }
                                          F   valables {ADJ GRN PLU }



      FIN


PHI
      PHI
      N'' {FEM PLU }
            D'  {DEF GRN PLU }
                  D   les {DEF GRN PLU }

            N'  {FEM PLU }
                  N   {FEM PLU }
                        A'  {FEM PLU }
                              A   {FEM PLU }
                                    F   dernie`res {NAN FEM PLU }


                        N   {FEM PLU }
```

```
                        F   anne'es {NOM FEM PLU }


        FIN


PHI
      PHI
        N'' {MAS SNG }
              D'  {DEF MAS SNG }
                    D   le {DEF MAS SNG }

              N'  {MAS SNG }
                    N   {MAS SNG }
                          N   {MAS SNG }
                                F   centre {NOM MAS SNG }

                          SP
                                P   de
                                N'' {DVB FEM PLU }
                                      D'  {DEF GRN PLU }
                                            D   les {DEF GRN PLU }

                                      N'  {DVB FEM PLU }
                                            N   {DVB FEM PLU }
                                                  F   pre'occupations {DVB FEM PLU }




        FIN


PHI
      PHI
        N'' {MAS PLU }
              D'  {DEF GRN PLU }
                    D   les {DEF GRN PLU }

              N'  {MAS PLU }
                    N   {MAS PLU }
                          N   {MAS PLU }
                                N   {MAS PLU }
                                      F   proble`mes {NOM MAS PLU }

                                A'' {MAS PLU PPA }
                                      A'  {MAS PLU PPA }
                                            A   {MAS PLU PPA }
                                                  F   pose's {ADJ MAS PLU PPA }


                                      SP
                                            P   par
```

```
                                    N''  {MAS SNG }
                                          D'   {DEF MAS SNG }
                                                D    le {DEF MAS SNG }

                                          N'   {MAS SNG }
                                                N    {MAS SNG }
                                                      F    cycle {NOM MAS SNG }



                              SP
                                    P    de
                                    N''  {MAS SNG }
                                          D'   {DEF GRN SNG }
                                                D    l' {DEF GRN SNG }

                                          N'   {MAS SNG }
                                                N    {MAS SNG }
                                                      F    azote {NOM MAS SNG }



FIN


PHI
      PHI
            N''  {FEM PLU }
                  D'   {DEF GRN PLU }
                        D    les {DEF GRN PLU }

                  N'   {FEM PLU }
                        N    {FEM PLU }
                              N    {FEM PLU }
                                    F    avantages {NOM FEM PLU }

                              SP
                                    P    de
                                    N''  {MAS SNG }
                                          D'   {DEF MAS SNG }
                                                D    le {DEF MAS SNG }

                                          N'   {MAS SNG }
                                                N    {MAS SNG }
                                                      F    syste`me {NOM MAS SNG }
```

ANALYSE SYNTAXIQUE AUTOMATIQUE DU FRANCAIS ECRIT : APPLICATIONS A L'INDEXATION AUTOMATIQUE

Geneviève LALLICH-BOIDIN

Résumé

L'analyse syntaxique d'une langue naturelle consiste à définir une grammaire de cette langue, grammaire nécessairement ambiguë, à choisir un algorithme d'analyse non déterministe et à élaborer une stratégie d'analyse afin d'éviter la construction de structures syntaxiques parasites.

Dans le cadre de ce travail, nous définissons une grammaire du syntagme nominal du français écrit. Nous étudions les analyseurs hors-contexte et non-déterministes de Cocke-Younger-Kasami et d'Earley, et retenons ce dernier. Puis, au dessus de cet analyseur, nous élaborons une stratégie d'analyse qui, à partir des données linguistiques portées par le texte à analyser, permet de prédire localement la structure juste et qui limite de ce fait le nombre de solutions parasites.

L'indexation automatique de documents à partir des syntagmes nominaux contenus dans leur résumé est une application de l'analyseur construit.

Mots-clés :

Analyse syntaxique
Langue naturelle
Algorithme d'Earley
Stratégie d'analyse
Ambiguïté
Indexation automatique
Grammaire hors contexte
Linguistique