



HAL
open science

Syntaxe et sémantique dans un système de dialogue oral homme-machine finalisé en langage naturel

Pierre Mousel

► **To cite this version:**

Pierre Mousel. Syntaxe et sémantique dans un système de dialogue oral homme-machine finalisé en langage naturel. Informatique [cs]. Université Henri Poincaré - Nancy 1, 1990. Français. NNT : 1990NAN10038 . tel-01754424

HAL Id: tel-01754424

<https://hal.univ-lorraine.fr/tel-01754424v1>

Submitted on 30 Mar 2018

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

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



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Syntaxe et sémantique dans un système de dialogue oral homme-machine finalisé en langage naturel



Thèse

présentée et soutenue publiquement le 6 avril 1990

pour l'obtention du

Doctorat de l'Université de Nancy I

(Mention Informatique)

par

Pierre MOUSEL

Composition du Jury

Président :

Jean-Paul HATON

Rapporteurs :

Daniel COULON

Gérard SABAH

Directeur de thèse :

Jean-Marie PIERREL

Examineurs :

Jean-Claude ASSELBORN

Jacques BERLEUR

Remerciements



Le travail de recherche décrit dans cette thèse aurait certainement été impossible sans l'aide, les conseils et les encouragements de bon nombre de personnes que je tiens à remercier vivement.

Jacques Berleur, Recteur des Facultés Universitaires Notre-Dame de la Paix de Namur, a assuré le cours de « Méta-Informatique » à l'Institut d'Informatique des Facultés, cours qui a suscité mon intérêt pour le traitement automatique de la parole. Il a accepté de diriger mon travail de fin d'études de Licencié et Maître en Informatique et m'a permis de nouer les premiers contacts avec le Centre de Recherche en Informatique de Nancy.

Jean-Paul Haton, Professeur à l'Université de Nancy I et Directeur de Recherche à L'INRIA, m'a accueilli dans son équipe, dans le cadre de mon travail de fin d'études de Licencié et Maître en Informatique d'abord, dans le cadre du DEA et de mon travail de thèse ensuite. Il m'a fait l'honneur de présider le jury chargé d'évaluer mon travail.

Jean-Marie Pierrel, Professeur à l'Université de Nancy I, a dirigé mes travaux dès mon arrivée au Centre de Recherche en Informatique de Nancy. C'est lui qui m'a appris le métier de chercheur. Les qualités du travail présenté ici sont dues à sa compétence.

Jean-Claude Asselborn, Professeur au Centre Universitaire de Luxembourg, a relu mon manuscrit avec beaucoup d'attention. Grâce à ses remarques toujours pertinentes, j'ai pu éliminer un grand nombre d'imperfections du document écrit.

Daniel Coulon, Professeur à l'Ecole des Mines de Nancy, ainsi que Gérard Sabah, Directeur de Recherche au LIMSI, ont accepté de rédiger le rapport sur ce travail de thèse.

Fernand Reinig, Administrateur du Centre de Recherche Public du Centre Universitaire de Luxembourg, m'a permis de terminer la rédaction du manuscrit dans des conditions matérielles idéales.

Il va sans dire que je n'aurais pas pu réaliser ce travail sans les discussions fructueuses que nous avons eues dans le cadre du groupe « dialogue oral homme-machine » au CRIN. Je voudrais particulièrement remercier mes amis Guy Deville et Azim Roussanaly qui ont été tout au long de ces an-

nées des co-équipiers de tout premier choix. Je leur dois en grande partie l'aboutissement de ce travail.

Je tiens également à remercier des amis qui n'ont pas été directement impliqués dans le projet mais dont les remarques, les encouragements ou tout simplement la présence m'ont permis de mener ce travail à terme, il s'agit de Béchir El Ayeb, Victor Henx, Gérard Masini, Hans Paulussen, Paul Retter et Nino Silverio.

Enfin je ne puis trouver de mots pour exprimer ma gratitude envers mon épouse Josiane qui a fait preuve d'une patience et d'une compréhension extrêmes durant ces dernières années, envers ma fille Sonia qui n'a cessé de me réjouir par sa fraîcheur, ainsi qu'envers mes parents qui ont su me convaincre de l'utilité d'une solide éducation et qui m'ont prêté un appui constant.

Pierre Mousel
Nancy, avril 1990

TABLE DES MATIERES

TABLE DES MATIERES	i
LANGAGE, PAROLE ET INFORMATIQUE	1
1 A propos de Frankenstein, de savants fous et d'ingénieurs indignes de confiance	1
2 Vers des systèmes performants	3
3 Vers des systèmes qui comprennent	4
4 Vers des systèmes qui parlent	6
MOTIVATIONS DES RECHERCHES EN PAROLE	9
1 Y a-t-il des applications pour la parole ?	9
2 Y a-t-il un marché pour la parole ?	11
LES RECHERCHES EN PAROLE, HIER ET AUJOURD'HUI	15
1 Une histoire riche en rebondissements	15
1.1 Les premiers résultats prometteurs	15
1.2 Désillusion et déception	16
1.3 La reprise des travaux	17
1.4 Les travaux sur la parole en France	17
2 Etat de l'art et tendances	18
3 Techniques et approches	20
3.1 Reconnaître la parole ou comment maîtriser une quantité énorme d'informations	20
3.1.1 Différents points de vues pour aborder la reconnaissance de la parole	20
3.1.2 La parole, un phénomène continu	21
3.1.3 Digitalisation de la parole	21
3.1.4 Compression du signal de parole	23
3.1.5 Extraction de paramètres	24
3.1.6 Reconnaissance globale ou analytique	25
3.2 Limiter le problème	26
3.2.1 Langages artificiels ou naturels	27
3.2.2 Univers de blocs ou réalité perçue	28
3.2.3 Menus ou initiative partagée	29
3.2.4 Reconnaître ou comprendre	29
3.2.5 Mots isolés ou parole continue	30
3.2.6 Mono-locuteur ou multi-locuteurs	31

COMMUNICATION ORALE HOMME-MACHINE FINALISEE EN LANGAGE NATUREL	33
1 Communication	33
2 Communication homme-machine	36
3 Communication orale	37
4 Communication en langage naturel	38
5 Communication finalisée	38
 QUELQUES REPERES LINGUISTIQUES	41
1 Des modèles linguistiques, points d'appui à notre étude	41
1.1 Les modèles syntaxiques	42
1.1.1 Les grammaires formelles	44
1.1.2 Les automates	48
1.2 Les modèles sémantiques	55
1.2.1 Les grammaires de cas	56
1.2.2 Les grammaires fonctionnelles	59
1.3 Les modèles cognitifs	64
1.4 Des modèles linguistiques adaptés au traitement de la parole	67
2 La notion de sous-langage, justification méthodologique de notre approche	67
2.1 Les sous-langages	67
2.2 Particularités des sous-langages	68
2.3 De l'intérêt des sous-langages	71
 QUELQUES REALISATIONS INFORMATIQUES	73
1 Un échantillon de systèmes de compréhension de parole	73
2 Le système KEAL	73
3 Le système MYRTILLE I	75
4 Les systèmes HWIM ET MYRTILLE II	77
5 Le système HEARSAY II	80
6 Une architecture de système adaptée au dialogue oral homme-machine	81
 LA MISE EN ŒUVRE DE CONNAISSANCES DANS LE SYSTEME DIAL	83
1 Le système DIAL	83
1.1 La compréhension de phrases dans les systèmes MYRTILLE I et MYRTILLE II	83
1.2 La compréhension de dialogues dans le système DIAL	84
1.3 Les objectifs du système DIAL	85
1.4 Le choix d'une application test	86
2 Le choix d'une architecture	88
2.1 Les caractéristiques particulières d'un dialogue oral finalisé en langage naturel	88
2.2 Une typologie opératoire des connaissances linguistiques	89
2.3 Proposition pour une architecture répartie	91
3 La composante syntaxico-sémantique	95
3.1 Communication avec les autres composantes	95
3.2 Structure de la composante syntaxico-sémantique	95
4 La composante pragmatique	97
4.1 Communication avec les autres composantes	97
4.2 Structure de la composante pragmatique	97

LES CONNAISSANCES LINGUISTIQUES DU SYSTEME DIAL	99
1 Les Réseaux à Nœuds Procéduraux	99
1.1 Objectifs des Réseaux à Nœuds Procéduraux	99
1.2 Définition des RNP	100
1.3 Puissance de description des RNP et comparaison avec d'autres types de représentation	103
1.4 La modélisation syntaxique du sous-langage de notre application test	104
2 La grammaire de cas	106
2.1 Objectifs de la grammaire de cas	106
2.2 Définition de la grammaire de cas	107
2.3 Comparaison avec d'autres modèles sémantiques	110
2.4 La modélisation sémantique du sous-langage de notre application test	112
3 Les règles de correspondance	114
3.1 Objectifs des règles de correspondance	114
3.2 Définition des règles de correspondance	115
3.3 Les règles de correspondance du sous-langage de notre application test	115
 LA COMPOSANTE SYNTAXICO-SEMANTIQUE DU SYSTEME DIAL	 117
1 La composante SYNSEM	117
1.1 Différents types d'analyse	117
1.2 Quelques concepts de base	119
1.2.1 Espace des théories	119
1.2.2 Le type d'une théorie	120
1.2.3 Etat d'une théorie	122
1.2.4 Espace des hypothèses	124
1.3 L'analyse d'un énoncé dans SYNSEM	125
2 Autour de SYNSEM	128
2.1 Outils d'aide à l'étude de corpus linguistiques	128
2.2 Outils de création de bases de connaissances linguistiques	130
2.3 Outil de consultation de l'espace des théories	133
3 Réalisation d'un prototype d'évaluation	134
3.1 Simulation des interfaces	134
3.2 Paramétrisation par les connaissances	135
3.3 Paramétrisation par les stratégies	136
3.3.1 Stratégies de sélection	136
3.3.2 Stratégies de propagation	138
3.4 Facilités	141
3.4.1 Abandon et reprise d'une analyse	141
3.4.2 Statistiques sur l'évolution de l'espace des théories	142
3.4.3 Trace chronologique des événements	143
3.4.4 Sauvegarde automatique de l'espace des théories	143
3.4.5 Nettoyage automatique de l'espace des théories	144
4 Premiers résultats	144
 ORIENTATIONS FUTURES	 147
1 Ce qui a été fait	147
2 Ce qu'il reste à faire	148
3 Quelques remarques en guise de conclusion	149
 BIBLIOGRAPHIE	 151

LANGAGE, PAROLE ET INFORMATIQUE



Le premier chapitre se propose de situer notre travail dans le domaine de l'intelligence artificielle. Il met en évidence les relations qu'entretient l'informatique depuis ses débuts avec les sciences de la langue écrite ou parlée et présente les spécificités de la parole par rapport à l'écrit.

1 A propos de Frankenstein, de savants fous et d'ingénieurs indignes de confiance

« Ce fut en novembre, pendant une nuit affreuse, que je vis l'accomplissement de mes travaux. Dans une inquiétude voisine de l'agonie, je rassemblai autour de moi les instruments propres à donner la vie, pour introduire une étincelle d'existence dans cette matière inanimée qui était à mes pieds. L'airain avait déjà sonné la première heure après minuit ; la pluie battait, avec un sifflement horrible, contre mes fenêtres ; ma lumière était près de s'éteindre, lorsqu'à cette lueur vacillante, je vis s'ouvrir l'œil jaune et stupide de la créature : elle respira avec force, et ses membres furent agités d'un mouvement convulsif. »

C'est dans une ambiance cauchemardesque que Victor Frankenstein réalise un vieux rêve de l'humanité : insuffler de la vie à ce qui est mort (Shelley, 1818). Mais sa déception est immédiate, il voit l'œil jaune et stupide de sa créature et son cœur ira même jusqu'à se remplir d'une horreur et d'un dégoût affreux.

Les chercheurs en intelligence artificielle, et plus particulièrement ceux qui travaillent dans le domaine de la reconnaissance de la parole, se

voient parfois comparés au célèbre savant genevois (Mosanto, 1988)¹ ou pire, traités de savants fous ou d'ingénieurs indignes de confiance (Pierce, 1969)². Pour certains contemporains, il est évident qu'après une première période de grandes promesses où tout semblait possible en intelligence artificielle, nous vivons actuellement à l'heure du désenchantement, voire même de la déception (Dreyfus et Dreyfus, 1986)³. Pour d'autres, les chercheurs en intelligence artificielle ont toujours été des modèles publics d'humilité et de modestie (Bobrow, 1985)⁴ et l'intelligence artificielle est désormais entrée dans sa phase de maturité (Winston, 1984)⁵.

Quels sont donc les buts poursuivis par ces personnes qui essaient d'apprendre aux machines à voir, à entendre, à comprendre et à parler, bref, de les rendre plus intelligentes ? Jean-Marie Pierrel pense que depuis toujours, l'homme a voulu créer des machines qui imitent les mouvements, les fonctions ou les actes des corps animés et qu'on peut voir l'intelligence artificielle comme le prolongement naturel de cette évolution (Pierrel, 1987)⁶. Plus concrètement, il part du constat qu'il existe actuellement une masse d'informations sur support informatique. Dès lors, il faut tout faire pour que cette information gérée par des machines soit facile à utiliser et à

¹ P. MOSANTO, dans *La compréhension des langages naturels : libre tribune*, pense que : « (...) l'utilisation des indéniables acquis de la reconnaissance de la parole par les spécialistes de l'I.A. ressemble furieusement à celle de l'électricité par Frankenstein. ».

² J.R. PIERCE les appelait des « (...) mad scientists and untrustworthy engineers (...) » dans sa lettre à l'éditeur du *Journal of the Acoustical Society of America* intitulée *Whither Speech Recognition ?*

³ H.L. et S.E. DREYFUS, dans *Künstliche Intelligenz*, pages 127-128, voient l'avenir de l'intelligence artificielle avec pessimisme : « Die Zukunft der KI sieht in der Tat düster aus. Wie wir schon früher sahen, haben die als logische Maschinen programmierten Computer auf Bilder oder sonstige bildähnliche Repräsentationen nur dann Zugriff, wenn sie sie in Beschreibungen übersetzen. Ebenso können logische Maschinen mit Ähnlichkeiten nur etwas anfangen, wenn sie sie in Listen von identischen Merkmalen zerlegen. Und es gibt keinen Grund für die Hoffnung, daß Computer, die aus eindeutigen Fakten logische Schlüsse ziehen und genau definierte Merkmale mit anderen genau definierten Merkmalen vergleichen können, Bilder in einer Weise ganzheitlich miteinander zu vergleichen vermögen, die Urteile über Ähnlichkeiten ermöglicht. ».

⁴ P. McCORDUCK, dans *Artificial Intelligence - Where Are We? (D.G. Bobrow, P.J. Hayes Eds.)*, écrit : « I don't believe AI need worry about industrial disillusionment. That isn't to say there won't be disillusioned industrialists: there will be, and they will complain to the media, and AI will be accused once more of making promises it can't deliver on. This too shall pass. I observe that nobody ever investigates *who* made the promises: aside from Simon's prehistorical predictions, and a few off-the-wall remarks by Minsky, the promises have been mainly made by outsiders extrapolating wildly. For better or worse, AI researchers themselves are public models of humility, modesty and decorum, dreary dull compared to, oh say, paleoanthropologists or genetic engineers. ».

⁵ P.H. WINSTON, dans la préface de *Artificial Intelligence*, écrit : « In the early days, most researchers in the field were content with *illustrations* that showed ideas at work on simple, blocks-world problems, without showing the limits of those ideas. As the field matured, there came to be a gradual shift toward *demonstrations*, showing that there are ideas that are powerful enough to handle problems of practical interest. ».

⁶ A ce sujet on pourra également consulter l'article de P. MORRISON, *The Mind of the Machine*. L'auteur y discute un problème lié depuis toujours à ce désir de l'homme : « With each stage of seemingly magical machines that simulate aspects of life, the question why life is different from non-life returns on a new level. The ancient's simulations of purposeful motion were magic, until man came to understand that life meant more than the ability to move. In an age of computers, what shall we say if machines come to think? ». Les photos accompagnant ce texte sont consacrées aux androïdes Jaquet-Droz, des automates construits au dix-huitième siècle capables de mouvements assez étonnants.

mettre à jour. Il est alors nécessaire de permettre à ces machines d'acquérir des moyens de raisonnement, les systèmes experts, et des moyens de communication : parole, vision ou langage naturel les plus proches possibles de ceux de l'homme (Gaines et Shaw, 1986a ; Haton, 1984 ; Pierrel, 1987)¹. Un rôle privilégié revient alors au traitement de la langue, écrite ou parlée, car elle va servir à la fois de support et de véhicule pour l'information et la connaissance, et de langage d'interrogation ou d'accès à ces vastes bases de connaissances (Angeletti et Beysson, 1984 ; Arnold, 1986 ; Biermann et al. 1983 ; Bolc et al., 1985 ; Carbonell et al., 1984 ; Cozannet, 1985 ; Danlos, 1985 ; Finin et al., 1986 ; Hayes et al., 1981 ; Hayes et Reddy, 1983 ; Hendrix et al., 1978 ; Ledgard et al., 1980 ; Maguire, 1982 ; Nagao et al., 1986 ; Sabah, 1988 ; Sabah, 1989 ; Sammet, 1966 ; Schwarcz et al., 1970 ; Shapiro and Kwasny, 1975 ; Simmons, 1970 ; Slocum, 1986 ; Waltz, 1978 ; Weischedel, 1986 ; Young et al., 1989).

2 Vers des systèmes performants

En effet, depuis ses débuts, l'informatique entretient des rapports très étroits avec le langage. On peut distinguer essentiellement deux approches dans le traitement automatique de la langue (Coulon et Kayser, 1986 ; Pierrel, 1987) :

- L'approche ergonomique. L'être humain qui communique avec une machine a besoin d'un langage. Dans certains cas (pas dans tous), le langage naturel est celui qui convient le mieux à l'utilisateur. Il faut donc examiner les procédés par lesquels le dialogue homme-machine peut se rapprocher, lorsque cela est souhaitable, d'un dialogue normal. Certains parlent d'une approche pragmatique. L'objectif est de réaliser des programmes capables de faire au moins aussi bien qu'un individu moyen dans un domaine de compétence et pour lesquels il ne nous est pas possible de décrire des algorithmes déterministes.
- L'approche philosophique. L'être humain, depuis la plus haute antiquité, consacre des réflexions à son propre langage ; cette réflexion s'inscrit dans des cadres qui changent avec les époques, sous des appellations qui évoluent : grammaire, philologie, linguistique, ... ; l'informatique apporte avec elle une façon de poser les questions et d'exprimer les réponses qui est susceptible de renouveler profondément la pensée sur le langage. Il faut, par conséquent, examiner avec tout le recul possible les voies qui se sont ainsi ouvertes. Encore qualifiée de psychologique, matérialiste même, le but de cette approche est d'expliquer l'homme et de construire des machines qui soient de véritables modèles du fonctionnement du cerveau humain.

¹ Il est intéressant de confronter cette vue des choses à celle exprimée par T. WINOGRAD dans *Artificial Intelligence - Where Are We?* (D.G. Bobrow, P.J. Hayes Eds.) : « My own work underwent a major change, as I moved away from the assumption that the way to make better and more useful computers (and interfaces) was to get them to be intelligent and use natural language. I recognized the depth of the difficulties in getting a machine to understand language in any but a superficial and misleading way, and am convinced that people will be much better served by machines that do well-defined and understandable things than those that appear to be like a person until something goes wrong (which won't take long), at which point there is only confusion. ».

Dans nos recherches, ce sont essentiellement les travaux de la première catégorie qui ont retenu notre attention. Nous voulons réaliser un système informatique capable de mener un dialogue qui se rapproche le plus possible d'un dialogue normal¹.

3 Vers des systèmes qui comprennent

La construction d'un tel système pose deux problèmes fortement liés, mais néanmoins dissociables² :

- d'une part, ce système doit accepter en entrée des énoncés proches de la langue naturelle. Il doit pouvoir les transformer en une représentation interne qu'il pourra manipuler par la suite. En d'autres mots il doit avoir, dans une certaine mesure, la faculté de comprendre du langage naturel ;
- d'autre part, il doit souvent être capable de fournir une réponse à l'utilisateur en langue naturelle. Problème inverse de la compréhension, il s'agit de générer, à partir d'une représentation interne au système, un texte aussi proche que possible de la langue naturelle.

Nous avons essentiellement étudié le problème de la compréhension. Bien sûr, nous sommes tout à fait conscient que dans un système automatique de dialogue, le problème de la génération de réponses à l'utilisateur est incontournable mais, étant donné que ce problème peut très bien être traité séparément, que d'autres équipes de recherche y travaillent de toutes façons (Danlos, 1985), nous avons consacré, pour l'instant, nos efforts à la compréhension.

¹ Cette façon d'envisager l'utilisation des ordinateurs est considérée comme caractéristique de l'approche IA par J. KAMMERSGAARD dans *Four Different Perspectives on Human-Computer Interaction* : « A perspective on the use of computers which play a role especially within the AI community is the dialogue partner perspective. When applying this perspective, humans and computers are regarded as partners in a dialogue. This interaction process is regarded as a communication process in which user and computer application act as both sender and receiver, and the computer application is seen as being able to show communicative behaviour similar to that of its human partner. ». Approche d'ailleurs mise en question dans sa conclusion où il écrit : « In the last four sections I have described four different perspectives on the use of computers. It will be clear from the description that I find the dialogue partner perspective problematic. Although it has gained widespread acceptance, I find its relevance very limited. ».

² Cette distinction est également faite par certains linguistes. C.J. FILLMORE, dans *Subjects, Speakers, and Roles*, écrit page 23 : « (...) I so frequently find myself speaking without any understanding of what I am saying that I quite naturally think of the ability to produce a sentence as involving essentially different principles from those that are employed in figuring out what if anything its utterer intended. », tandis que S. DIK, dans *Basic Principles of Functional Grammar*, affirme page 6 : « (...) Generative grammars have often been said to be completely neutral with respect to this division. But a functional grammar, in seeking to achieve pragmatic and psychological adequacy, cannot simply dissociate itself from production and comprehension.: why should a grammar be neutral as between producing and comprehending linguistic expressions, when producing and comprehending such expressions is just what the grammar is there for? It seems that a better fit with psychological reality can be achieved if we conceptualize the grammar itself as consisting of (a) a production model, (b) a comprehension model, (c) a store of elements and principles which are used in both (a) and (b) (...) ».

Avec la notion de compréhension, nous rencontrons une première difficulté : comment définir ce que nous entendons par « comprendre ». Une définition formelle et applicable semble difficile voire impossible (Kayser, 1985 ; Weizenbaum, 1967)¹. Néanmoins certains critères qualitatifs (Bonnet, 1984) nous permettent de préciser ce qu'on attend, en partie au moins, d'un tel système dit de compréhension du langage naturel :

- la capacité de répondre de façon appropriée à des questions ;
- la capacité de paraphraser ;
- la capacité de faire des inférences ;
- la capacité de traduire d'une langue à une autre ;
- la capacité de résoudre des problèmes de références ;
- et finalement réussir le test de Turing.

Des réalisations ont montré qu'il est possible de répondre à certains de ces critères, notamment réussir le test de Turing, sans mettre en œuvre d'autres traitements que des manipulations assez simplistes de structures d'informations (Colby et al., 1966 ; Weizenbaum, 1966). Certains chercheurs, comme Terry Winograd par exemple, pensent que de toutes façons, les ordinateurs ne feront jamais rien d'autre que de tels traitements aveugles (Winograd et Flores, 1986). Nous pensons que cela n'est pas particulièrement gênant pourvu qu'un tel système réponde à certaines exigences, par exemple qu'il réponde de façon appropriée à des questions dans un nombre assez important de cas².

¹ Dans son article *Des machines qui comprennent notre langue*, D. KAYSER situe le problème : « Parmi les perspectives ouvertes par l'informatique, il en est une particulièrement fascinante : la conception de machines qui comprendraient notre propre langue. En fait, un tel objectif est très mal défini. On peut le considérer comme déjà atteint, car il existe aujourd'hui des machines qui répondent à des questions exprimées en français ou dans une autre langue, pourvu que ce soit sur des domaines étroitement limités ; mais on peut aussi le déclarer inaccessible : comprendre un être ne signifie-t-il pas partager avec lui tant de références, y compris non verbalisables, qu'il est bien rare que deux êtres humains eux-mêmes se « comprennent » véritablement. ». Ce constat n'est pas récent, en effet J. WEIZENBAUM a tenté d'en fournir une explication dans *Contextual Understanding by Computers* « There can be no total understanding and no absolutely reliable test of understanding. To know with certainty that a person understood what has been said to him is to perceive his entire belief structure and that is equivalent to sharing his entire life experience. ».

² Ce point de vue est également partagé par B.R. GAINES et M.L.G. SHAW dans *Foundations of Dialog Engineering: The Development of Human-Computer Interaction. Part II* : « If the sender has to build a model of the receiver then a new level of complexity emerges. The complexity of a system able to create a model of another is necessarily much greater than that of the system being modeled. The information necessary to model a system is very much greater than that necessary to pass messages between them. If two dissimilar systems are attempting to establish communication based on mutual understanding, then we have a paradox. On the above argument, each must be more complex than the other in order to achieve this. Hence complete understanding is impossible between dissimilar systems. This is a reasonable result. In practice, an adequate level of mutual understanding to achieve certain goals may involve only a very small part of the two systems' structures. They may restrict their complexity to that which is necessary to cope with the goals. ».



fig. 1.1 : « Répondre de façon appropriée à des questions »
 extrait de (Chapman, 1985)

4 Vers des systèmes qui parlent

Un système de compréhension du langage naturel peut se baser sur la parole ou sur l'écrit. Nous voulons construire un système capable de mener un dialogue oral, ce qui complique légèrement les choses par rapport à l'écrit. La parole a en effet certaines spécificités qui la distinguent de la langue écrite (Bonnet, 1984 ; Lea, 1983 ; Mariani, 1985) et qui constituent autant de problèmes supplémentaires à résoudre :

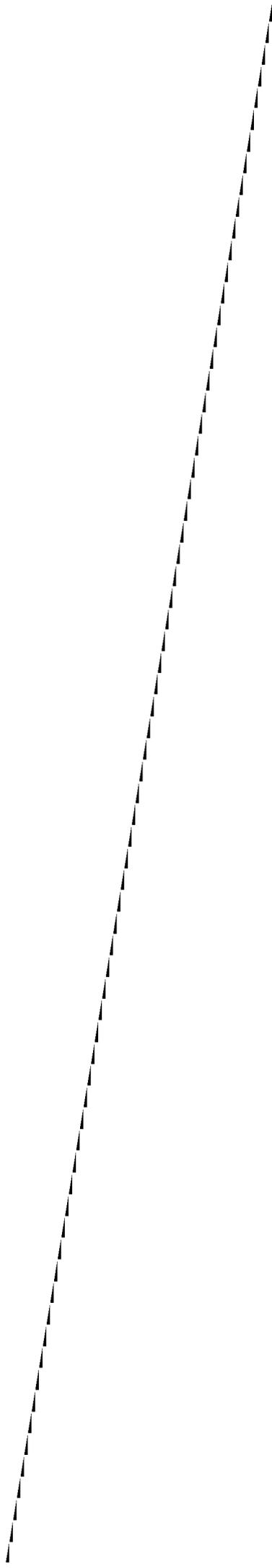
- le signal résultant de l'émission de parole comporte des « bruits » et des « réverbérations » c'est-à-dire des non-informations à éliminer dues à l'environnement ;
- certaines perturbations sont dues à la prise de son, selon le type du microphone, la distance et l'orientation par rapport au locuteur etc. ;
- la prononciation des locuteurs n'est pas parfaite et varie d'un individu à l'autre suivant le sexe, l'âge, l'accent etc. ;
- la prononciation d'un locuteur donné est variable suivant son état (psychologique, physiologique) ;
- la prononciation d'un phonème voire d'un mot n'est pas la même suivant que ces derniers sont isolés ou en contexte ;

- la frontière entre les mots successifs d'une phrase n'apparaît pas sur le signal vocal ; il peut très bien y avoir des intervalles de silence au milieu d'un mot et aucun intervalle entre deux mots successifs ;
- des mots totalement différents peuvent avoir la même prononciation : c'est le problème des homophones.

Un système de dialogue oral, contrairement à un système basé sur l'écrit doit par conséquent être capable de (Young et al., 1989) :

- Gérer l'indéterminisme. Lors du traitement, le système est confronté à des incertitudes et à chaque niveau du traitement, ces incertitudes sont combinées. Le résultat de chaque étape est en général exprimé en termes de probabilité ou de vraisemblance car plusieurs choix différents doivent pouvoir être évalués.
- Identifier des mots. Les frontières entre les mots n'apparaissant pas sur le signal, la localisation des mots est tout à fait incertaine. A un endroit donné, des mots peuvent commencer, se terminer, se chevaucher etc. De plus, sur une plage de signal donnée, il y a incertitude sur le mot qui a été réellement prononcé et plusieurs mots sont en général candidats avec des plausibilités différentes. Enfin, très souvent, les mots sont coarticulés de telle façon que les frontières entre les mots s'estompent.
- Gérer l'homophonie. Certains mots, ou suites de mots, se prononçant de la même façon, ce n'est que la prise en compte d'un contexte plus large qui permet de les distinguer.
- Résister aux phénomènes d'altérations phonologiques. Le langage parlé ayant tendance à être concis, les locuteurs humains omettent ou transforment inconsciemment des syllabes entières.
- Intégrer des mécanismes d'analyse robustes. Un mot prononcé par un locuteur peut ne pas être reconnu. Le système doit donc pouvoir travailler avec des informations incomplètes. De plus, le langage parlé est caractérisé par une grammaire beaucoup plus souple que celle de l'écrit et des phénomènes de reprise et d'hésitation qui ne font que compliquer les choses.

Une grande partie de ces problèmes n'a pas encore trouvé de solution. Dès lors, il semble évident que le recours à la parole pour une application informatique ne soit, à l'heure actuelle, envisageable que dans des cas très particuliers où l'usage de ce moyen de communication est absolument indispensable.



MOTIVATIONS DES RECHERCHES EN PAROLE

Le deuxième chapitre montre dans quelles conditions, l'ajout de la parole comme moyen de communication avec une machine peut être utile. Les applications potentielles de la parole sont présentées ainsi que l'importance économique qu'elles pourront représenter.

1 Y a-t-il des applications pour la parole ?

Dans le premier chapitre, nous avons montré que, par rapport à l'écrit, l'emploi de la parole dans le cadre de la communication homme-machine pose des problèmes supplémentaires à résoudre. Ceci entraîne en général un surcoût important pour l'application informatique visée. La décision d'y avoir recours ne peut donc être prise à la légère et doit être justifiée soit parce que l'emploi de la parole est tout simplement inévitable, soit parce que les gains auxquels on s'attend l'emportent sur les coûts (Lea, 1983). On constate par exemple que (Biermann et al., 1985 ; Gould, 1982 ; Haton et Pierrel, 1981b ; Lea, 1980a ; Lea, 1983 ; Levinson et Liberman, 1981 ; Mariani, 1985 ; Pierrel, 1987) :

- la communication orale est indispensable dans certaines situations, pour suppléer les autres canaux de communication, lorsque ceux-ci sont soit saturés, soit inopérants ;
- la communication orale libère complètement l'usage de la vue et des mains (contrairement à l'écran et au clavier), laisse l'interlocuteur humain libre de ses mouvements, n'immobilise pas le corps, peut être employée en parallèle avec d'autres vecteurs de l'information et ne nécessite aucun contact visuel ou physique ;
- la communication orale n'est que peu affectée par l'absence de pesanteur, par de fortes accélérations ou d'autres contraintes mécaniques ;

- la communication orale est plus efficace que d'autres modes de communication dans certaines situations de communication ;
- la communication orale est moins sujette à des erreurs, car plus spontanée, que d'autres moyens de communication homme-machine comme par exemple le clavier ;
- la communication orale ne requiert que des dispositifs d'entrée-sortie peu encombrants, elle ne nécessite ni panneaux d'affichage ni instrumentations complexes ;
- la communication orale peut être établie via des moyens simples existants, comme le réseau téléphonique ou les ondes hertziennes ;
- la communication orale est le moyen de communication le plus naturel de l'homme, tout le monde, ou presque, sait parler ;
- la communication parlée ne nécessite aucune formation préalable sauf si le système n'accepte qu'un nombre très limité d'énoncés

Ce constat amène les chercheurs et les industriels à envisager de nombreuses applications, aussi bien dans les domaines civils que militaires. Citons, sans vouloir être exhaustif, quelques applications civiles (Damper, 1984 ; Derouault et Merialdo, 1987 ; Descout, 1984 ; Dubus et al., 1984 ; Firschein et al., 1974 ; Gould et al., 1983 ; Haton et Pierrel, 1981b ; Kohonen, 1988 ; Lea, 1980b ; Mariani, 1985 ; Merialdo, 1987 ; Pierrel, 1981 ; Pierrel et Bennani, 1984 ; Rehm, 1980 ; Schalk et Van Meir, 1983 ; Thompson, 1984) :

- le tri ;
- le contrôle de qualité et l'inspection ;
- la saisie de données ;
- la programmation de machines à commande numérique ou de robots et la commande orale d'ordinateurs ;
- la communication dans les postes de pilotage d'avions ou d'automobiles ;
- la bureautique et plus particulièrement le traitement de textes et la machine à écrire à entrée vocale ;
- l'aide aux handicapés ;
- les transactions bancaires ;
- les systèmes de question-réponse, d'interrogation de bases de données et les centres de renseignements spécialisés ;
- l'identification de locuteurs pour les contrôles de sécurité et d'accès ;
- l'enseignement assisté par ordinateur ;
- etc..

Citons également quelques applications militaires (Castiglione et Goldman, 1984 ; Coler, 1982 ; Descout, 1984 ; DeSipio et Fry, 1983 ; Lea, 1980b ; Mariani, 1985 ; North et Lea, 1982 ; Tarnaud, 1984 ; Young et al., 1989) :

- la cartographie ;
- la formation des contrôleurs aériens ;
- la communication dans les postes de pilotage d'avions ou dans des engins spatiaux ;
- la détection de pannes dans des systèmes complexes ;
- la surveillance de conversations¹ ;
- le commandement et le contrôle par des officiers supérieurs ;
- etc..

Il faut cependant remarquer que certaines fonctions se prêtent mal à l'emploi de la parole (Lea, 1983), notamment :

- les fonctions de contrôle avec des variations continues de niveau ;
- les tâches critiques nécessitant une confiance absolue, aussi bien dans les domaines militaire que civil ;
- et bien sûr les fonctions où ce canal de communication est déjà réservé.

2 Y a-t-il un marché pour la parole ?

Bien sûr ce n'est pas uniquement la perspective de recherches ou d'applications d'un intérêt scientifique certain qui pousse chercheurs et industriels à développer des systèmes de reconnaissance de la parole. Il y a

¹ On lira au sujet de ce point particulièrement épineux J. WEIZENBAUM, *Puissance de l'ordinateur et raison de l'homme*, page 177 : « J'ai incité mes étudiants et collègues à poser une autre question sur ce projet (la reconnaissance automatique de la parole, *note de l'auteur*) : étant donné qu'une machine de reconnaissance de la parole sera extrêmement onéreuse et que seuls les gouvernements et éventuellement quelques très grandes firmes seraient en mesure de la mettre en œuvre, quel sera l'usage qu'ils en feront ? Quelles sont ses utilisations possibles ? A mon avis, il n'existe aucun problème humain impérieux dont la solution soit facilitée par l'existence de ces machines. Mais de telles machines d'écoute, en supposant qu'elles puissent être construites, rendraient bien plus facile qu'aujourd'hui la surveillance des communications phoniques. Il est possible que la seule raison pour laquelle il y a peu de surveillance gouvernementale des conversations téléphoniques dans de nombreux pays est qu'une telle opération exige trop de main-d'œuvre. Chaque conversation d'un appareil sous écoute doit finalement être écoutée par un agent humain. Mais des machines de reconnaissance de la parole pourraient éliminer toutes les conversations « sans intérêt » et ne présenter que les autres à leurs maîtres. (...) ». Cette crainte est partagée par D. DENNETT dans *Artificial Intelligence - Where Are We?*, (D.G. BOBROW, P.J. HAYES Eds.) : « I think 'pure' academic AI also has attendant risks worth pausing to think carefully about. I agree with Joe Weizenbaum that the work on speech recognition, for instance, for all its theoretical importance and fascination as pure science, opens up the technology for vastly increased electronic surveillance. People working in the field should face that issue - and its kin - just as unblinkingly as the nuclear physicists and genetic engineers have faced their issues. The "I'm just a scientist: I can't help what they do with it" line should not sit comfortably on anyone's lips. ». Ce type d'application, de nos jours, est mis en œuvre à grande échelle. On pourra consulter notamment un article récent du magazine « Der Spiegel » consacré aux activités de surveillance de la National Security Agency en Allemagne Fédérale (Spiegel, 1989).

également, ne l'oublions pas, la promesse d'un marché juteux. Certains observateurs semblent assez optimistes et, au vu des résultats obtenus jusqu'aujourd'hui, prévoient des marchés de vingt à soixante milliards de francs en 1990 (Mariani, 1985). D'autres sont plus nuancés et pensent que les techniques de traitement automatique de la parole, à l'heure actuelle du moins, ne se prêtent pas encore à une utilisation généralisée (Descout, 1984)¹. D'autres encore vont jusqu'à se demander s'il existe un besoin quelconque en systèmes de reconnaissance de la parole (Berleur, 1984 ; Neuburg, 1980)². En fait, il est difficile de se prononcer à ce sujet, les études de marché dans le domaine étant plutôt rare. Néanmoins, quelques travaux permettent de donner des éléments de réponse.

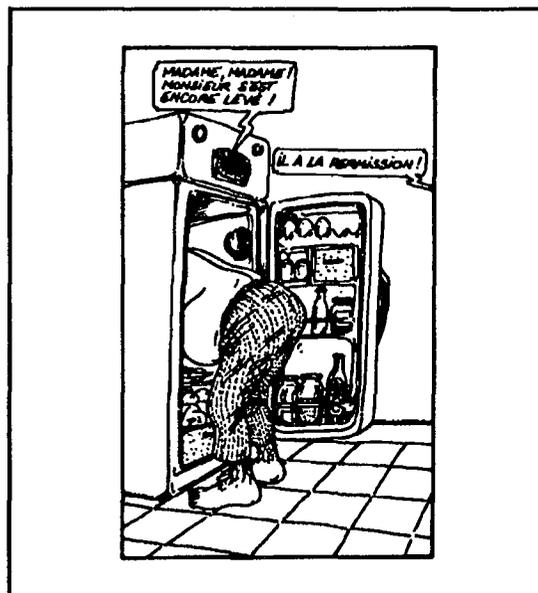


fig. 2.1 : « Contrôle de sécurité et d'accès »
extrait de (Veyron, 1982)

Michael Nye, par exemple, a identifié des applications dans des domaines divers tels que la manipulation et le tri de matériaux, le contrôle de qualité, la saisie de données, la programmation de machines-outils, le contrôle d'appareils médicaux et le contrôle d'engins tactiques militaires (Nye, 1980)³. Ces applications se recoupent en grande partie avec les applications

¹ Dans *Speech Technology Applications in France*, R. DESCOUT écrit : « (...) if a real development of the products of Speech Technology can be observed as predicted, the technology is not yet ready for the wider "Speech Market". ». Il constate notamment qu'à l'heure actuelle la parole coûte encore trop cher pour être utilisée dans des applications domestiques et que, dans le cas des Télécommunications par exemple, la priorité est accordée à des alternatives meilleur marché tel que l'installation de terminaux « Minitel ».

² Alors que E.P. NEUBURG, dans *Needs vs Competence in Speech Recognition*, se pose surtout la question si la reconnaissance de la parole est un problème qu'on cherche à résoudre ou s'il s'agit d'une solution à la recherche d'un problème, J. BERLEUR quant à lui, dans *Compréhension du langage naturel et reconnaissance de la parole par ordinateur* : « est-ce approprié », se demande si les ordinateurs sont « appropriés » à répondre à ce type de problème et si oui, quels sont les acteurs sociaux qui font naître une demande en systèmes de reconnaissance automatique de la parole et tentent de se les « approprier », il pose le problème du besoin « social ».

³ M. NYE, au moment de la publication de cet article, était collaborateur de la Marketing Consultants International, Inc..

citées au paragraphe précédent. Nye est également d'avis que l'intérêt de tels systèmes, qui ont d'ailleurs dépassé le stade du développement, dans des contextes industriels a été prouvé ce qui l'amène à prévoir, dès 1980, une croissance exponentielle, en termes d'unités installées, du marché potentiel de la parole. La figure 2.2 présente les prévisions de Nye.

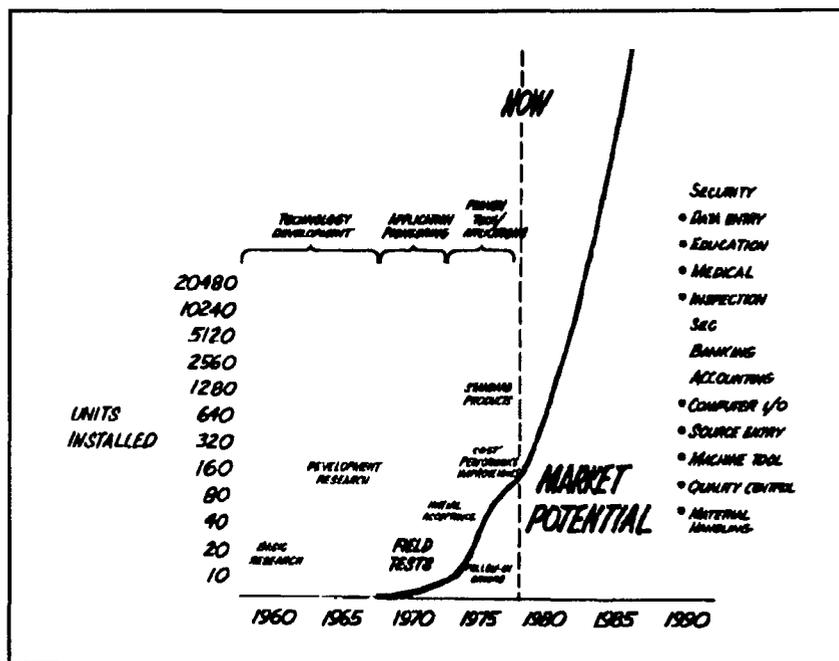


fig. 2.2 : Evolution du marché de la parole selon M. Nye
extrait de (Nye, 1980)

D'autre part, en Allemagne Fédérale, un sondage a été effectué auprès de cadres supérieurs de diverses entreprises, enquête dont l'objectif était de déterminer (Morik, 1984)¹ :

- ce que des clients potentiels considèrent comme un apport des systèmes de traitement du langage naturel et comment les coûts en sont évalués ;
- dans quelle mesure ces clients potentiels pensent que les utilisateurs de tels systèmes seront affectés ;
- si ces clients potentiels investiraient dans des systèmes de traitement du langage naturel (s'ils étaient disponibles) ;
- et finalement sous quelles conditions et pour quelles tâches, ces clients potentiels utiliseraient un système de traitement du langage naturel.

Dans ce contexte il est intéressant de noter que 88,6% des personnes interrogées indiquaient pouvoir utilement employer un système de traitement du langage naturel dans le cadre de leur travail. Pour 65,4%, ce type de système devrait permettre de meilleures prises de décision par un accès facilité aux informations les plus récentes relatives à la gestion de l'entreprise et

¹ Cette étude de marché a été effectuée par K. MORIK (Universität Hamburg, Projekt HAM-ANS, Forschungsstelle für Informationswissenschaft und Künstliche Intelligenz, Mittelweg 179, 2000 Hamburg 13, BRD) et financée par le Ministère de la Recherche et de la Technologie de la République Fédérale Allemande.

16,3% seulement pensaient qu'ils allaient devoir changer leurs habitudes de travail. Malgré le fait que 69,9% des interrogés étaient d'avis que les coûts d'un tel système dépasseraient à court terme ses avantages, 70,8% investiraient dans de tels systèmes s'ils étaient disponibles. 90,8% aimeraient bien disposer d'un système capable de traiter leur langue maternelle, la langue allemande en l'occurrence. Quant aux moyens de communication envisagés, 47,3% des personnes interrogées pensaient que la parole serait utile en entrée contre 20,2% seulement qui pensaient qu'elle serait utile en sortie d'un ordinateur. Enfin les applications visées par la population étudiée se recoupent en partie avec les applications citées au paragraphe précédent à savoir l'interfaçage en langue naturelle de grandes collections d'informations.

En conclusion il semblerait qu'il existe une demande bien réelle en systèmes de reconnaissance de la parole pour une multitude d'applications. Les travaux dans le domaine ne seraient donc pas uniquement dus à une dynamique intrinsèque des centres de recherche et de développement¹.

¹Pour utiliser les termes de J.M. NYE dans *The Expanding Market for Commercial Speech Recognizers* : « The need is clear and the future bright for "humanized" man/machine communications. ».

LES RECHERCHES EN PAROLE, HIER ET AUJOURD'HUI

Le troisième chapitre présente un bref historique des recherches en reconnaissance de la parole, tente de faire le point sur la question et passe en revue les techniques de base qui sont actuellement utilisées dans le domaine.

1 Une histoire riche en rebondissements

La reconnaissance de la parole est un problème par essence pluridisciplinaire. On pourrait en rechercher les origines dans des siècles de travaux sur la langue, le son, la physiologie, la psychologie, les automates etc.. Là n'est cependant pas notre propos. Nous voulons simplement mettre en évidence ici quelques étapes qui nous ont semblé déterminantes. Pour un historique plus détaillé, nous renvoyons le lecteur à (Lea, 1980b).

1.1 Les premiers résultats prometteurs

Le premier dispositif de reconnaissance de la parole, entièrement câblé, a été développé en 1952 par des chercheurs aux laboratoires de la « Bell Telephone » : Davis, Biddulph et Balashek. Il s'agissait d'un appareil capable de reconnaître les chiffres de zéro jusqu'à neuf. La fréquence du signal de parole en entrée était divisée en deux bandes (au-dessous et au-dessus de 900 Hertz). Pour chacun des deux signaux résultants, le nombre de passages par zéro était calculé (Fawe, 1983). La forme ainsi obtenue, un tableau bidimensionnel de nombres, était comparée à dix formes pré-enregistrées,

correspondant chacune à un des chiffres de un à neuf, et le chiffre dont la forme était la plus proche de la forme à reconnaître était sélectionné. Les performances de ce matériel étaient assez surprenantes : la machine réalisait un score de reconnaissance de plus de 97% si les formes de référence provenaient du locuteur lui-même.

En 1958, Dudley et Balashek en présentaient une version améliorée (Dudley et Balashek, 1958). L'originalité de cette réalisation résidait dans le fait que ce nouveau dispositif procédait à une segmentation du signal en unités phonétiques identifiées grâce à leurs caractéristiques spectrales. Encore une fois, si le système était bien réglé pour un locuteur, la reconnaissance était presque totale.

Vers 1959-1960 on voyait apparaître les premiers travaux sur ordinateur. Denes et Mathews (Denes et Mathews, 1960) introduisaient la normalisation temporelle basée entre autre sur des méthodes de programmation dynamique (Bellman, 1957). Il s'agissait de comprimer ou d'étendre, en fonction du temps, des parties de la forme à reconnaître en vue de la ramener à la forme « normalisée » de la forme de référence. Des expériences ont montré que l'utilisation de ces méthodes donnait de meilleurs résultats aussi bien pour un locuteur particulier que pour des locuteurs différents.

Plus tard dans les années soixante, certains concepteurs tentaient de mettre en œuvre des connaissances linguistiques dans leurs systèmes. C'est le cas de Vicens (Vicens, 1969), dont le système utilisait la syntaxe du langage naturel pour reconnaître le langage de contrôle d'un robot, et de Tubach (Tubach, 1970), dont le système utilisait la syntaxe du langage ALGOL pour reconnaître des programmes prononcés mot par mot.

Plusieurs systèmes expérimentaux voyaient encore le jour et il fallait attendre 1972 pour voir apparaître les premiers produits commercialisés par « Scope Electronics, Inc. » et « Threshold Technology, Inc. ». Ces produits dits de première génération étaient implantés sur mini-ordinateur. Les systèmes de deuxième génération, vers 1979, ne nécessitaient déjà plus que des cartes sur base de microprocesseurs et aujourd'hui, les produits de troisième génération tiennent sur quelques circuits VLSI spécialisés (Haton, 1985a). Mais avant cela, les recherches dans le domaine devaient d'abord connaître quelques déboires.

1.2 Désillusion et déception

A la fin des années soixante, les quelques résultats modestes obtenus après presque deux décennies de travaux ne pouvaient cacher le fait que la recherche dans le domaine piétinait. L'optimisme céda la place à la résignation et les réactions ne se firent pas attendre. La plus violente fut certainement l'article « Whither Speech Recognition » de John Pierce des laboratoires Bell en 1969 (Pierce, 1969) mais il faut la situer dans le contexte plus large d'un mouvement de réticence et de scepticisme croissants qui commençait à se faire sentir envers les méthodes et les objectifs de l'intelligence artificielle de façon plus générale et dont les conséquences sont encore sensibles aujourd'hui. C'est également à cette époque que Hubert Dreyfus publia son fameux « What Computers Can't Do » (Dreyfus, 1972) suivi peu

après par « Computer Power and Human Reason » de Joe Weizenbaum (Weizenbaum, 1976). On prenait alors conscience des enjeux culturels que représentait l'informatisation de plus en plus poussée de la société. En plus, les liens étroits qui existaient, et qui existent d'ailleurs toujours, entre la recherche en informatique, en particulier la recherche en intelligence artificielle, et les milieux militaires commençaient à susciter des inquiétudes. Le débat se poursuit d'ailleurs aujourd'hui (Arsac, 1987 ; Berleur, 1984 ; Dreyfus, 1986 ; Gorny, 1985 ; Pullum, 1987 ; Schuler, 1986 ; Weizenbaum, 1984), nous n'avons pas la prétention de le trancher ici. Toujours est-il que ces réactions ne signifèrent pas la fin des recherches en parole.

1.3 La reprise des travaux

Si la lettre de Pierce portait un coup certain au domaine, ce coup n'en fut pas pour autant décisif. Même les laboratoires Bell, après quelques années, s'intéressaient de nouveau à la reconnaissance de la parole. Qui plus est, en 1971 démarrait l'effort de recherche le plus important jamais entrepris dans le cadre de la reconnaissance de la parole. L'ARPA (Advanced Research Project Agency), une des nombreuses agences du Département de la Défense des Etats Unis, lançait le projet SUR (Speech Understanding Research). Ce projet qui s'étendait sur cinq ans avec un budget de quinze millions de dollars devait montrer la faisabilité de la « compréhension automatique de la parole continue » avec un vocabulaire d'un millier de mots (Newell et al., 1973). L'objectif était de construire des systèmes utilisant au maximum des techniques d'intelligence artificielle pour construire la structure linguistique complète des énoncés des locuteurs afin de produire une réponse appropriée. La fin des années soixante-dix vit l'aboutissement de ces recherches avec la réalisation de plusieurs systèmes dont certains seront encore présentés plus en détail par la suite. Les conclusions du comité de coordination du projet ont été rendues publiques dans (Medress et al., 1977)¹ et sont assez optimistes en ce qui concerne l'avenir. Les membres du comité ont même rédigé un rapport fournissant des indications quant aux suites à donner au projet ARPA-SUR (Newell et al., 1976) mais ce dernier n'eut jamais de successeur. Ceci ne signifie pas que les choses devaient en rester là, bien au contraire. Des équipes de recherche de par le monde travaillent aujourd'hui sur les systèmes les plus divers (Castiglione et Goldman, 1984 ; Coler, 1982 ; Descout R., 1984 ; Dubus et al. ; Haton, 1985a ; Kohonen, 1988 ; Mariani, 1982 ; Morin et Pierrel, 1987 ; Mousel et al., 1989b ; Niemann et al., 1985 ; Pierrel, 1982 ; Quinton, 1982 ; Tarnaud, 1984 ; Young et al., 1989).

1.4 Les travaux sur la parole en France

En France, les centres de recherche ne commençaient à s'intéresser à la reconnaissance automatique de la parole que vers la fin des années soixante (Haton, 1980). L'ensemble des recherches est actuellement couvert

¹ Les membres de ce comité affirment dans leur rapport : « We report the successful accomplishment of a significant scientific and technical goal: To demonstrate the feasibility of a computer system's understanding connected speech. (...) There are still many limitations, in performance and in cost. But they do demonstrate that natural speech input to computers is a responsible technical goal to pursue. ».

par cinq grands laboratoires, la coordination en étant assurée depuis 1981 par le GRECO-CNRS (Groupe de REcherches COordonnées) « Communication Parlée » (Pierrel, 1987) :

- le CERFIA (Cybernétique des Entreprises, Reconnaissance des Formes et Intelligence Artificielle) à Toulouse ;
- le CNET (Centre National d'Etudes des Télécommunications) à Lannion ;
- l'ICP (Institut de la Communication Parlée) à Grenoble ;
- le LIMSI (Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur) à Orsay
- et finalement le CRIN (Centre de Recherche en Informatique de Nancy) à Nancy où on s'intéresse depuis une vingtaine d'années aux différents aspects de la reconnaissance et de la compréhension de la parole et dans le cadre duquel notre travail s'est effectué (Haton, 1980 ; Haton et Pierrel, 1981a).

Quelques systèmes développés dans ces laboratoires seront étudiés plus en détail par la suite. A côté de cette recherche universitaire, plusieurs industriels se sont également penchés sur la question, ne citons que Thomson, IBM-France, RENAULT etc..

2 Etat de l'art et tendances

Malgré certains résultats indéniables, tous les problèmes sont loin d'être résolus. Lea, par exemple, pense que ce qui existe aujourd'hui sur le marché est avant tout une collection de dispositifs de reconnaissance de formes plutôt que des dispositifs de compréhension de parole (Lea, 1983) et de façon générale, on sent aujourd'hui le besoin d'aller au-delà de systèmes uniquement capables de transformer en chaîne de caractères une phrase parlée qui leur a été fournie en entrée (Carbonell et al., 1984 ; Young et al., 1989). Le concepteur qui voudrait cependant intégrer des techniques de reconnaissance de parole dans une application informatique aujourd'hui, se trouve encore dans une situation peu confortable. En effet les capacités des systèmes disponibles sur le marché sont encore limitées et chères. En France notamment, on trouve les systèmes suivants (Descout, 1984) :

- MOISE/RM150, système mono-locuteur reconnaissant des mots isolés avec un vocabulaire de 50 mots au prix de 5.000 FF, et MOZART/RME186, système mono-locuteur reconnaissant des mots connectés ou détectant des mots avec un vocabulaire de 100 mots au prix de 30.000 FF, développés par VECSYS ;
- RECONNAISSANCE VOCALE, système mono-locuteur reconnaissant des mots isolés avec un vocabulaire de 100 mots au prix de 15.000 FF, développé par MATRA ;
- SERAPHINE, système mono-locuteur reconnaissant des mots connectés ou phrases brèves respectant une syntaxe restreinte avec un vocabulaire de 100 mots au prix de 30.000 FF.

Aux Etats Unis, le choix est un peu plus diversifié (Lea, 1983). L'offre va de systèmes sur base de circuits VLSI spécialisés, dépendant du locuteur, ne reconnaissant que des mots isolés et coûtant une centaine de dollars (par exemple : IEC VRC 100 etc.) jusqu'à des systèmes complets ou spécialisés, indépendants du locuteur ou reconnaissant des mots connectés, et dont les prix varient aux alentours de cent mille dollars (par exemple : NEC DP100/200 etc.). L'état du marché américain de la parole vers 1983 est présenté sur la figure 3.1.

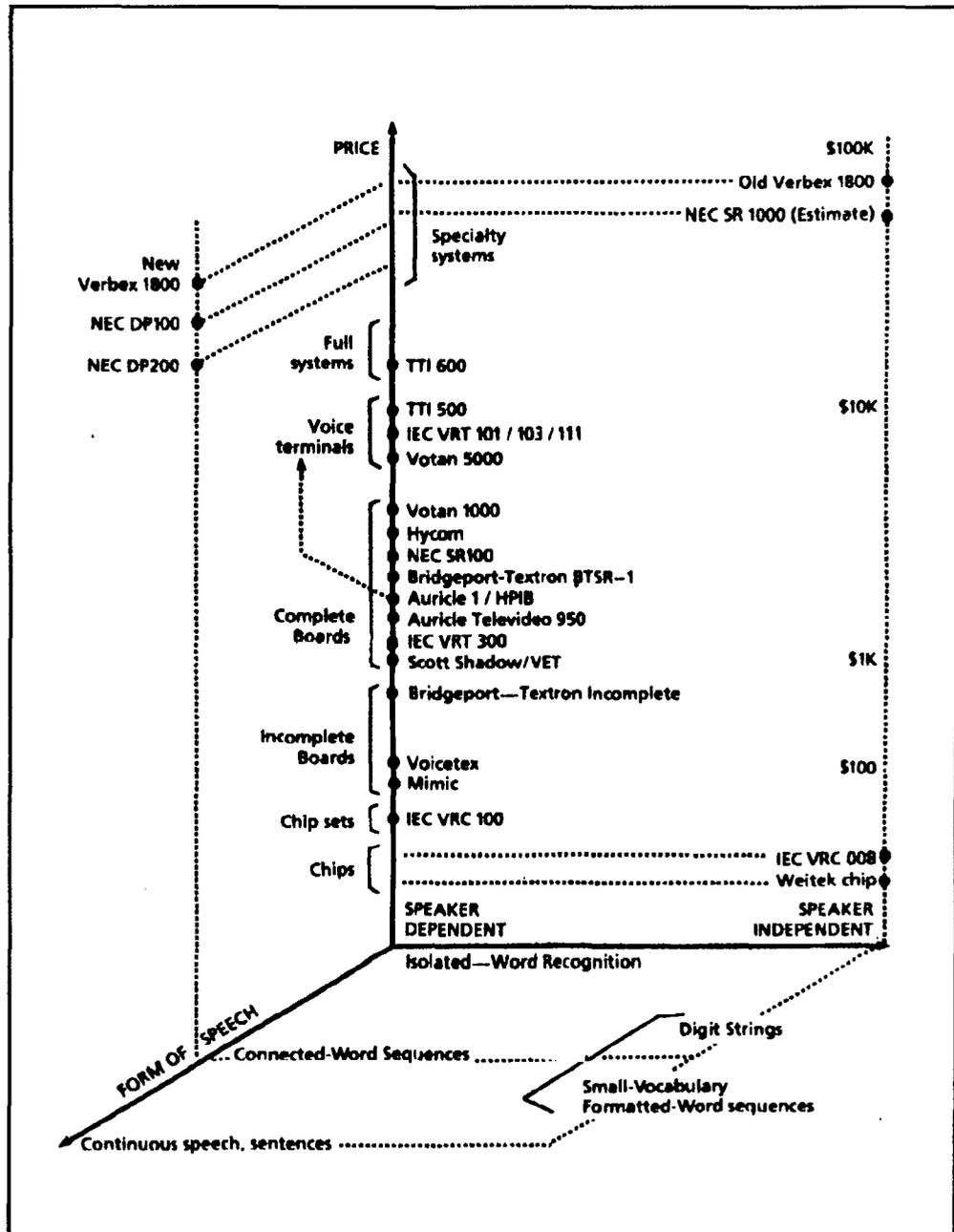


fig. 3.1 : Le marché des systèmes de reconnaissance de parole aux USA, extrait de (Lea, 1983)

Par conséquent, la technologie de la reconnaissance de la parole n'est pas encore tout-à-fait sortie des laboratoires pour entrer dans notre monde quotidien. Parmi les problèmes qui restent à résoudre, il faut compter les

phénomènes de variation inter-locuteurs, la manipulation de vocabulaires de taille importante (quelques milliers de mots), la mise en œuvre de connaissances linguistiques de haut-niveau (syntaxe, sémantique et pragmatique) ainsi que l'intégration de modèles sur l'utilisateur. L'objectif de bon nombre de chercheurs aujourd'hui est d'aller vers des systèmes de dialogue oral pouvant être utilisés par des non-spécialistes afin d'accomplir une tâche (Carbonell, 1984 ; Young, 1989).

3 Techniques et approches

3.1 Reconnaître la parole ou comment maîtriser une quantité énorme d'informations

3.1.1 Différents points de vues pour aborder la reconnaissance de la parole

La parole véhicule des informations très variées, elle renseigne notamment sur *qui* a prononcé *quel message* de *quelle façon* dans *quel environnement*. La tâche d'un système de reconnaissance de la parole est de déterminer automatiquement sur base d'un signal de parole, le message qu'il contient, indépendamment des variations introduites par le locuteur, la façon de parler, l'environnement etc. Au cours de l'histoire, somme toute encore assez courte, de la reconnaissance de la parole se sont profilées certaines approches fondamentales au problème de la conception de systèmes de reconnaissance de parole (Lea, 1980) :

- l'approche « traitement du signal » dont les tenants ne considèrent la parole que comme un cas particulier de phénomène ondulatoire auquel on peut appliquer les techniques habituelles d'analyse de signaux ;
- l'approche « production de la parole » qui suggère que l'on essaie de comprendre la façon dont la parole a été produite par l'appareil phonatoire humain et d'en saisir les aspects déterminants ;
- l'approche « réception sensorielle » qui suggère la duplication du processus de réception auditive humaine afin d'extraire des paramètres et traits de classification à la façon de l'oreille, des nerfs auditifs et des récepteurs sensoriels et enfin
- l'approche « perception de la parole », l'idée étant ici de faire des distinctions catégorielles qui sont considérées comme importantes pour la perception de la parole par l'humain.

Des combinaisons de ces différentes approches sont bien sûr possibles. Dans les paragraphes suivants, nous allons présenter, sans trop entrer dans

les détails techniques, certains traitements appliqués habituellement à la parole et reflétant l'une ou l'autre des approches pré-citées.

3.1.2 La parole, un phénomène continu

Le premier constat qu'on est amené à faire lorsqu'on s'intéresse à la parole est le fait qu'il s'agit d'un phénomène continu. La parole correspond à des variations locales de la pression atmosphérique provoquées par les organes phonatoires et peut être considérée comme une fonction continue du temps $f(t)$ (Haton, 1978 ; Lea, 1980 ; Saito et Nakata, 1985). En tant que telle, elle peut véhiculer une quantité infinie d'informations. Dès lors, le problème de la reconnaissance automatique de la parole est de sélectionner dans cette infinité d'informations possibles celles qui sont pertinentes et qui déterminent la réaction de la machine d'une façon donnée en se basant sur certaines caractéristiques propres à la parole, à savoir qu'il s'agit d'un phénomène périodique et que les organes phonatoires et auditifs humains ont certaines limitations (Lea, 1980). Un exemple de signal de parole continu est présenté sur la figure 3.2.

On pourrait imaginer des dispositifs de reconnaissance de parole travaillant sur base de telles représentations continues ; nous renvoyons le lecteur à (Lea, 1980) pour un exposé plus détaillé. Néanmoins, le traitement de la parole sur calculateur numérique suggère d'autres types de représentations¹.

3.1.3 Digitalisation de la parole

Le traitement de la parole sur ordinateur requiert en effet la digitalisation du signal de parole. Ceci peut être réalisé par un convertisseur analogique-digital qui fournit à des intervalles de temps réguliers la valeur de l'amplitude du signal, opération encore appelée échantillonnage du signal. On peut considérer cet échantillonnage comme une fonction E (Saito et Nakata, 1985) :

$$E : \{nT\} \rightarrow \{f_n\}$$

$$E(nT) = f_n = f(nT)$$

où n est entier, T réel et f la fonction continue à échantillonner.

T est alors appelé la période d'échantillonnage et son inverse $1/T$ la fréquence d'échantillonnage. La figure 3.3 montre un signal de parole échantillonné à 22 kHz. Pour qu'on puisse restituer le signal continu à partir

¹ Comme l'indiquent J.P. HATON et G. PERENNOU dans *Reconnaissance Automatique de la parole* : « L'augmentation de la puissance et de la vitesse de calcul des ordinateurs, ainsi que la baisse du coût des mémoires expliquent la prépondérance croissante des méthodes numériques de traitement sur les méthodes analogiques beaucoup moins souples et moins bien adaptées. ».

du signal échantillonné, la fréquence d'échantillonnage doit valoir au moins le double de la largeur de bande du signal (Nyquist, 1924 ; Shannon et Weaver, 1949). A titre d'indication, pour obtenir une qualité haute-fidélité, la parole doit être échantillonnée à une fréquence se situant entre 16 kHz et 20 kHz. Le codage de chaque échantillon nécessite entre 12 bits et 14 bits, le débit d'informations varie donc entre 192 kbits/s et 280 kbits/s. La plupart du temps on se contente cependant de la qualité d'une ligne téléphonique qui transmet une bande de fréquences située entre 300 Hz et 3400 Hz, la fréquence d'échantillonnage se situe alors entre 8 kHz et 10 kHz, chaque valeur nécessitant de 10 bits à 12 bits. Dans ce cas, le débit d'informations est de 80 kbits/s au moins et de 120 kbits/s au plus. Cette façon de coder le signal de parole est appelée codage PCM linéaire (Boite, 1983 ; Heute, 1988 ; Lea, 1980 ; Saito et Nakata, 1985)¹.

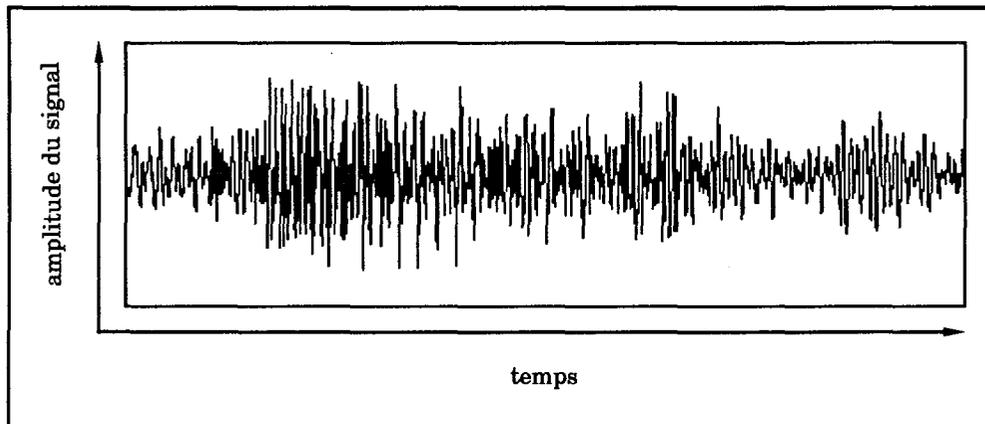


fig. 3.2 : Signal de parole continu correspondant au mot : « Dave »

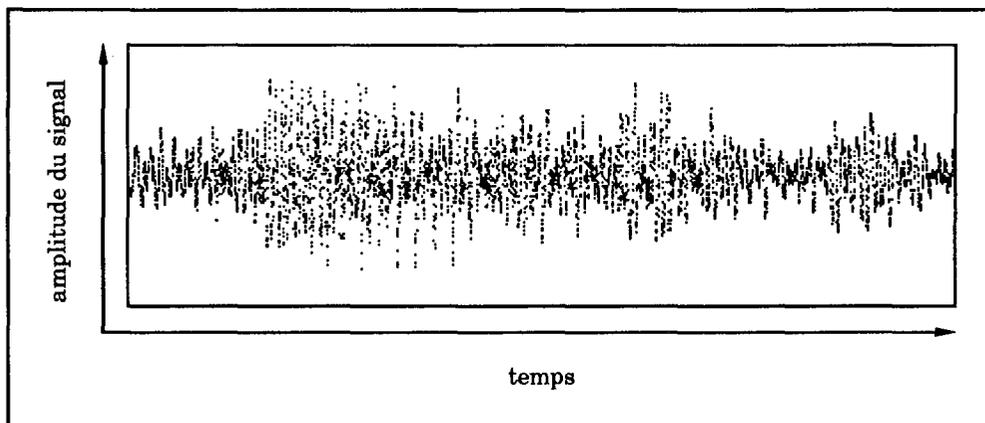


fig. 3.3 : Signal de parole digitalisé correspondant au mot : « Dave »

¹ PCM est l'abréviation de Pulse Code Modulation

3.1.4 Compression du signal de parole

Le codage PCM_{lin} a cependant un inconvénient majeur, le débit très important de données à traiter. Différentes techniques peuvent être employées afin de réduire cette quantité d'informations, le but étant toujours de garder un signal de parole intelligible et, par conséquent, de ne perdre que le minimum d'informations pertinentes sur le signal. Nous n'allons pas présenter ces techniques dans le détail, ce qui sortirait certainement du cadre de cette thèse, mais brièvement les passer en revue :

- Méthodes de codage direct de l'onde vocale : Certaines techniques de codage sont en fait des optimisations du codage PCM_{lin} qui exploitent des caractéristiques particulières du signal vocal, citons les codages DPCM_{lin} (linear Differential PCM), ΔM_{lin} (linear Delta Modulation), PCM_{log} (logarithmic PCM), DPCM_{log}, APCM (Adaptive PCM), $A\Delta M$, DPCM_{AQ} (DPCM with Adaptive Quantization), DPCM_{AP} (DPCM with Adaptive Prediction), ADPCM et APC (Adaptive Predictive Coding). Avec la dernière de ces techniques par exemple, on arrive à réduire le débit d'informations à 32 kbits/s (Boite, 1983 ; Heute, 1988 ; Saito et Nakata, 1985). En exploitant en plus certaines facultés de l'oreille humaine, ce débit peut encore être réduit (Aseo, 1984).
- Codage de la parole par les passages à zéro : Conserver les passages à zéro d'un signal consiste à l'écrêter totalement (Fawe, 1983). Toute information concernant l'amplitude est évidemment perdue dans l'opération, mais une onde beaucoup plus simple est obtenue (figure 3.4). Une propriété, surprenante à première vue, de la parole écrêtée est son intelligibilité qui reste excellente jusqu'à des débits de 8 kbits/s (signal PCM à 1 bit par échantillon).

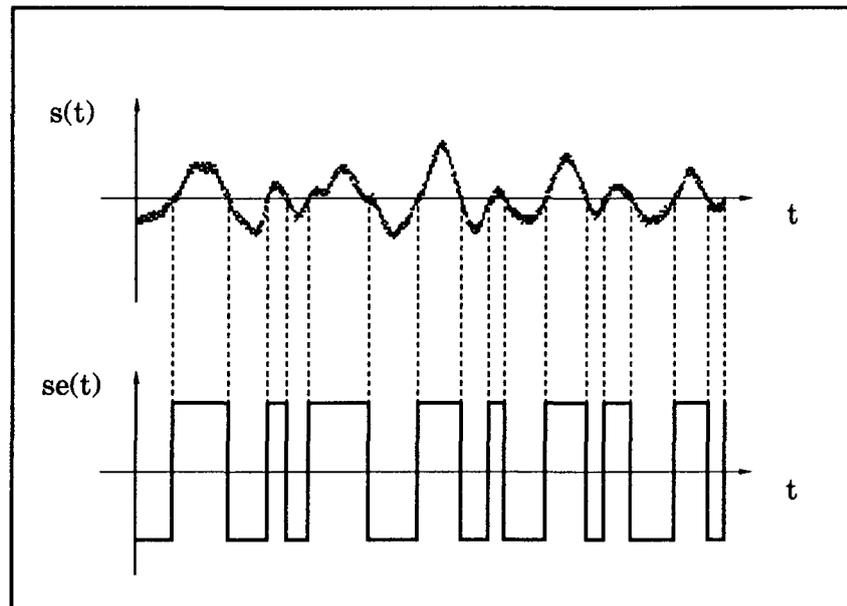


fig. 3.4 : Signal de parole $s(t)$ avec sa version écrêtée $se(t)$

La parole codée de cette façon peut être facilement transmise et synthétisée mais l'interprétation en vue de la reconnaissance reste très com-

plexe. On préfère en général faire subir au signal de parole des traitements permettant d'en dégager des paramètres plus pertinents.

3.1.5 Extraction de paramètres

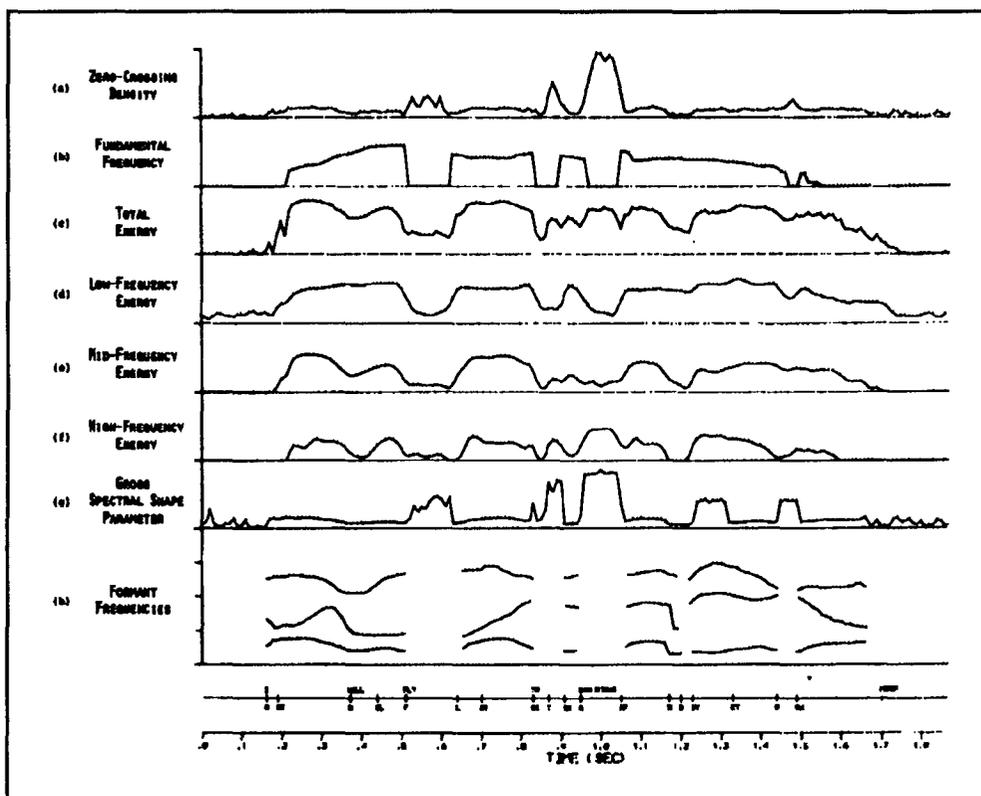


fig. 3.5 : Divers paramètres extraits du signal de parole correspondant à la phrase : « I will fly to San Diego »
extrait de (Zue et Schwartz, 1980)

Les paramètres qu'on essaie de dégager du signal de parole sont en général de deux types (Zue et Schwartz, 1980) :

- des indices temporels, tels que les coefficients de prédiction linéaire dans le cas de la méthode LPC (Boite, 1983 ; Markel et Gray, 1976 ; Saito et Nakata, 1985)¹, la densité des passages à zéro du signal ou de sa dérivée (Fawe, 1983)² etc. ;

¹ Il s'agit d'une technique de codage basée sur un modèle de production de la parole qui permet de réduire la quantité d'informations à traiter à 2,4 kbits/s (Heute, 1988).

² Dans *Représentation de la parole par la densité des passages à zéro (ou fréquence instantanée) en vue de la transmission, du traitement et de la reconnaissance*, A.L. FAWÉ note : « La densité des passages à zéro peut être définie comme le nombre des passages à zéro par unité de temps (fixée fréquemment à 10 msec dans le cas de la parole). (...) Les fluctuations relativement importantes de la densité ne permettent guère une quantification en plus de 4 niveaux. L'extraction de la densité des passages à zéro ramène donc la quantité d'information disponible à quelques 100 bits/s. ».

- des indices fréquentiels, tels que les formants du signal qui peuvent être dégagés à l'aide de bancs de filtres (Haton, 1978) mais, avec la prépondérance croissante des méthodes numériques, sont de plus en plus souvent calculés par FFT (Haton, 1978 ; Saito et Nakata, 1985 ; Silverman, 1977)¹. Une approximation du spectre peut également être obtenue à partir des passages par zéro du signal (Fawe, 1983).

La figure 3.5 présente des paramètres divers qui sont très souvent calculés à partir du signal de parole tandis que la figure 3.6 montre un spectrogramme, c'est-à-dire l'évolution du spectre d'un signal de parole en fonction du temps.

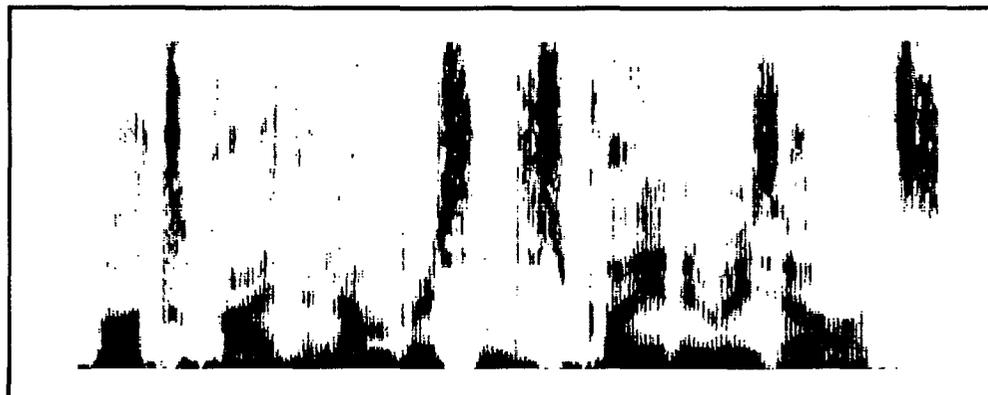


fig. 3.6 : Spectrogramme correspondant à la phrase :
« Multiply the numbers and display the results »
extrait de (Zue et Schwartz, 1980)

3.1.6 Reconnaissance globale ou analytique

Une fois la parole discrétisée et comprimée, une fois les paramètres pertinents dégagés, demeure toujours le problème de la reconnaissance. Différentes approches ont été adoptées par les concepteurs de systèmes pour le résoudre :

- pour les uns, le signal de parole à reconnaître n'est rien d'autre qu'une forme qu'il faut comparer à des signaux de parole pré-enregistrés, c'est-à-dire des formes de références (Denes et Mathews, 1960 ; Fawe, 1983 ; Kaneko et Dixon, 1983 ; Sakoe et Chiba, 1978 ; Sekey, 1984). La forme à reconnaître est alors considérée comme étant identique à la forme de référence la plus « proche » au sens de critères bien déterminés. On parle alors de « reconnaissance globale ». Les formes pré-enregistrées peuvent être des mots ou des phrases complètes. Cette façon de procéder soulève cependant certains problèmes. La première difficulté provient de la variabilité très forte inhérente à la production de parole, même pour un locuteur donné. On y remédie par des techniques de normalisation, le plus souvent inspirées de la programmation dy-

¹ FFT signifie « Fast Fourier Transform » ou « Transformée de Fourier Rapide », il s'agit d'une optimisation de la « Transformée de Fourier Discrète » qui permet, pour un signal discrétisé, de calculer le spectre de ce signal.

namique (Bellman, 1957 ; Saito et Nakata, 1985 ; Sakoe et Chiba, 1978). Une deuxième difficulté provient de l'approche elle-même. Comme on ne fait que comparer des formes à des formes de référence, plus on dispose de formes de référence, plus on est amené à faire de comparaisons. Les conséquences en sont facilement imaginables si on désire traiter un langage un tant soit peu complexe. On y remédie en utilisant des informations statistiques sur les occurrences des formes, en hiérarchisant le processus de reconnaissance etc. (Kaneko et Dixon, 1983 ; Sekey, 1984). Tous ces problèmes font qu'à l'heure actuelle, cette approche n'est utilisée que lorsqu'il faut reconnaître des vocabulaires de taille limitée pour un locuteur particulier. Il faut cependant remarquer que c'est la technique la mieux maîtrisée et que la plupart des systèmes disponibles sur le marché se basent sur cette approche (Lea, 1983).

- pour les autres, le signal de parole est une instanciation d'unités plus abstraites, des phonèmes, des diphonèmes, des syllabes etc., et il faut tout d'abord essayer de dégager ces unités du signal de parole, on parle dans ce cas de « reconnaissance analytique ». Il faut par conséquent qu'on soit capable de localiser les frontières de ces unités sur le signal de parole, c'est le problème de la segmentation, ensuite qu'on puisse déterminer de quelle unité chaque portion de signal est la réalisation, c'est le problème de l'identification. On utilise à cette fin des techniques algorithmiques (Dixon et Silverman, 1977 ; Irumano et al., 1983 ; Tanaka et al., 1983), des techniques d'intelligence artificielle (Carbonell et al., 1983 ; Connolly et al., 1986 ; DeMori et Probst, 1985 ; Fohr et al., 1985 ; Green et Wood, 1984 ; Haton, 1985b ; Mariani, 1983 ; Merialdo, 1987 ; Scagliola, 1983) et même, depuis quelque temps, des techniques basées sur des modèles neuronaux (Kohonen, 1988). Néanmoins, ces problèmes sont loin d'être résolus et il faudra certainement attendre plusieurs années avant de voir apparaître les premiers systèmes commercialisés mettant en œuvre ce type d'approche.

En conclusion, la reconnaissance de la parole demeure un problème extrêmement compliqué dont plusieurs aspects attendent toujours une solution satisfaisante. Si on veut, malgré ces difficultés, réaliser de tels systèmes, on est obligé de s'imposer certaines restrictions.

3.2 Limiter le problème

Les limitations qu'on doit s'imposer afin de pouvoir répondre, en partie du moins, à une demande en facilités d'interaction vocale peuvent porter sur différents axes. Il va de soi que les simplifications qu'on sera finalement amené à faire dépendront des spécifications et des besoins particuliers de l'application informatique dans laquelle le système devra s'intégrer. Dès à présent, nous pouvons constater que les axes principaux sur lesquels on pourra agir sont :

- la complexité du langage ;
- la complexité de l'univers du discours ;
- la forme de l'interaction ;
- le niveau de compréhension ;

- la qualité ergonomique ;
- la dépendance du locuteur.

Cette liste n'est pas forcément exhaustive, nous pensons cependant avoir signalé les facteurs les plus déterminants. Ils seront discutés plus en détail dans les paragraphes suivants.

3.2.1 Langages artificiels ou naturels

Un problème auquel se sont heurtés les concepteurs de systèmes de compréhension de parole, et même du langage naturel de façon générale, est l'extraordinaire diversité des formes et du sens qu'on retrouve dans les énoncés en langage naturel (Chomsky, 1972). Il est apparu très vite que si on voulait réaliser des systèmes utilisables à court terme, le langage servant de support à l'interaction entre l'utilisateur et la machine devait être fortement restreint (Klatt, 1980). La complexité du langage à reconnaître apparaît donc comme une valeur qu'on peut faire varier suivant un axe (figure 3.7) défini par :

- les langages artificiels d'une part : ils possèdent une syntaxe, une sémantique et un lexique définis de façon explicite, le plus souvent dans un formalisme mathématique, et nécessitent toujours une phase importante d'apprentissage par l'homme pour être utilisés efficacement (Pierrel, 1987) ;
- les langages naturels d'autre part : ils ne sont pas définis de façon explicite mais de façon implicite par leur utilisation quotidienne (Levinson et Liberman, 1981) ;
- entre ces deux extrêmes, d'autres catégories de langages ont été distinguées par divers auteurs, ne signalons que les langages quasi-naturels (Desclés, 1982) ou les langages pseudo-naturels (Pierrel, 1981).

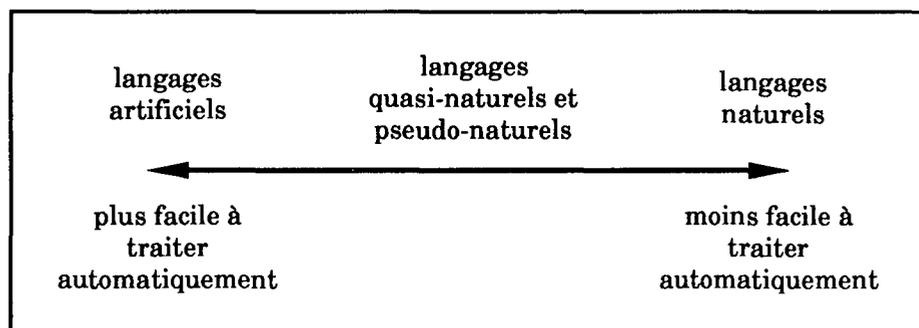


fig. 3.7 : La dimension « complexité du langage »

Pour certaines applications le rapport coûts/bénéfices ne justifie pas le traitement de langages naturels ou proches de langages naturels. C'est le cas de systèmes qui s'adressent à une population de spécialistes, c'est-à-dire des personnes qui utilisent fréquemment ce système. On préférera avoir recours à des langages artificiels, parfois très limités, plutôt que d'alourdir le sys-

tème par des facilités qui seront rapidement perçues comme « gadgets ». Plusieurs réalisations ont montré l'intérêt d'une telle approche (Mariani, 1982 ; Pierrel, 1981).

Pour d'autres applications au contraire, l'emploi de langages naturels comme support de communication entre l'homme et la machine est incontournable¹. En effet, on ne peut exiger de personnes n'utilisant que peu fréquemment un système d'apprendre tout d'abord le langage qui leur permet d'entrer en communication avec cette machine. Ceci est particulièrement vrai dans le contexte d'applications grand public (Carbonell, 1984 ; Cozannet, 1985).

3.2.2 Univers de blocs ou réalité perçue

Une autre difficulté qui surgit lorsqu'on s'attaque à la reconnaissance automatique de la parole est la complexité de la réalité perçue par l'être humain. Dans ses énoncés, l'homme fait référence à son environnement (Levinson, 1983). Si on désire traiter automatiquement ces énoncés, on est obligé de représenter en machine des faits concernant cet environnement. Or, il est impensable d'en faire un inventaire complet. Par conséquent, on est amené à restreindre l'univers du discours dont la complexité apparaît comme une nouvelle dimension (figure 3.8) où :

- d'un côté se situent des univers extrêmement pauvres où les objets ainsi que les relations qu'ils entretiennent sont peu nombreux et facilement identifiables, c'est par exemple le cas des mondes de blocs (Winograd, 1972) ;
- de l'autre côté on trouve la réalité perçue par l'être humain où les objets ainsi que les relations qu'ils entretiennent sont difficilement dénombrables et sont souvent trop flous pour apparaître clairement.
- entre ces deux extrêmes, une infinité d'univers est envisageable, ne citons que les mondes documentaliste (Erman et al., 1980), administratif (Roussanly, 1988), de la marine de guerre (Young et al., 1989) etc..

Ici encore, il ne sert à rien de faire compliqué quand on peut faire simple. La tendance est actuellement de bien délimiter un monde de référence précis, d'intégrer dans un système de reconnaissance de la parole le plus de connaissances pragmatiques possibles et de supposer que le locuteur va être coopératif, c'est-à-dire ne parler que de choses pour lesquelles le système a été

¹ Il faut signaler ici que pour certains, ceci est un objectif qu'on ne pourra jamais atteindre. H. ZEMANEK, par exemple, note à ce sujet dans *The Human Being and the Automation* page 19 : « It is not that natural languages do not have constructional aspects. The philologist comes closest to engineering when he analyses the structure of sentences. Syntax is formal and a set of formal grammatical rules can come very close to the live language. However, exceptions will always remain. The syntax of a constructed language, in contrast, is perfect. The essential difference is on the semantic level. The linguists, up to now, have not been able to approach semantics formally to a satisfactory degree. My opinion is that the semantics of natural language cannot be defined. This is, and will remain for ever, a gap between daily life and science. And this gap has crucial consequences for the computer. (...) ».

conçu, afin de garantir le succès de l'interaction (Roussanaly, 1988 ; Young et al., 1989).

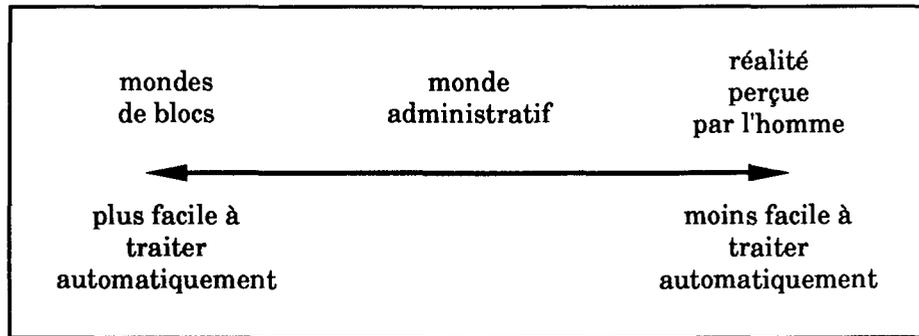


fig. 3.8 : La dimension « complexité de l'univers du discours »

3.2.3 Menus ou initiative partagée

Un autre facteur déterminant la complexité d'un système de reconnaissance de la parole est la forme de l'interaction souhaitée (figure 3.9) :

- dans un système à menus, c'est la machine qui détient l'initiative tout au long de l'interaction ; l'utilisateur, à chaque phase de la communication, n'a qu'un choix limité de possibilités d'action (Mousel-Muller, 1985) ;
- dans un système à initiative partagée par contre, l'initiative appartient tantôt au système tantôt à l'utilisateur (Roussanaly, 1988). Ce type d'interaction, s'il paraît plus naturel, requiert par contre des modèles du dialogue plus sophistiqués.

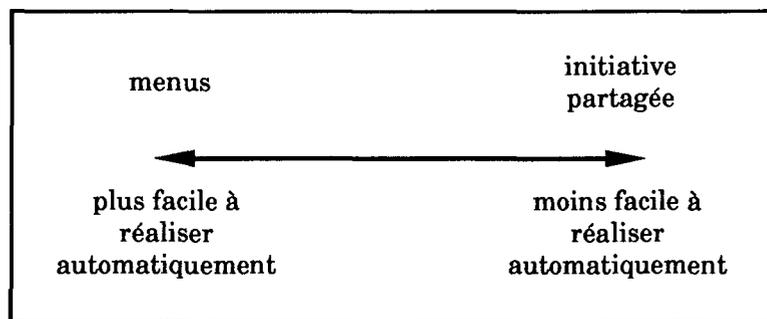


fig. 3.9 : La dimension « forme de l'interaction »

3.2.4 Reconnaître ou comprendre

Il convient également de bien définir le niveau de compréhension requis pour un système de reconnaissance de parole. Nous avons déjà signalé

mots¹ séparés par de courts silences de l'ordre du dixième de seconde, que

- des machines reconnaissant de la parole continue (Bahl et al., 1983 ; Erman, 1980 ; Mariani, 1982 ; Pierrel, 1981 ; Tappert et Dixon, 1974 ; Wolf et Woods, 1980), c'est-à-dire un flot non-interrompu de mots, fortement soumis à des effets de variation contextuelle, de substitution et d'élosion phonétiques.

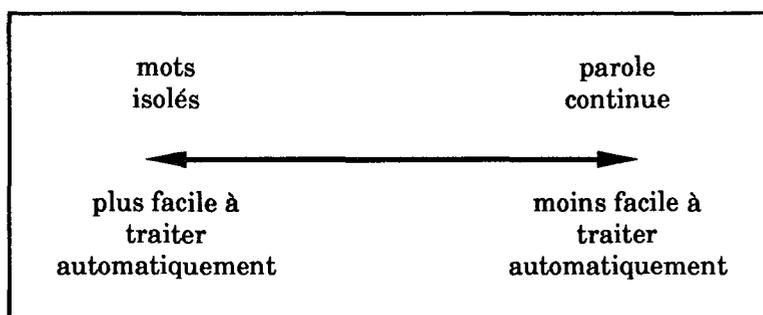


fig. 3.11 : La dimension « qualité ergonomique »

Des études ont montré que des interfaces vocales traitant des mots isolés peuvent raisonnablement être envisagées comme moyens de communication homme-machine efficaces pour de nombreuses applications (Biermann et al., 1985). Néanmoins, ces systèmes exigent de la part du locuteur une certaine adaptation et, par conséquent, semblent mal convenir à des applications grand public dans lesquelles un entraînement préalable des locuteurs est exclu.

3.2.6 Mono-locuteur ou multi-locuteurs

Enfin, le dernier moyen de limiter les problèmes que pose la conception d'un système de reconnaissance de la parole est de restreindre le nombre d'utilisateurs du système. En effet on peut à ce moment faire abstraction des phénomènes de variations inter-locuteurs. Un axe défini par les extrêmes suivants peut être identifié (figure 3.12) :

- d'un côté les systèmes dépendant du locuteur qui ne reconnaissent convenablement des énoncés que lorsque ceux-ci sont prononcés par des locuteurs bien particuliers (Dudley et Balashek, 1958 ; Itakura, 1975) ;
- de l'autre les systèmes indépendant du locuteur qui sont capables de reconnaître des énoncés quelque soit la personne qui les a prononcés (Levinson, 1985 ; Pister-Bourjot et Haton, 1987 ; Rabiner et al., 1979).

¹ Le terme « mot » est utilisé ici dans un sens plus large. En effet, il peut s'agir de groupes de mots, voire même de parties entières de phrases.

COMMUNICATION ORALE HOMME-MACHINE FINALISEE EN LANGAGE NATUREL

Avant de passer à la présentation détaillée de la composante que nous avons développée, il nous a semblé utile de délimiter le cadre méthodologique dans lequel s'est effectué notre projet de recherche. Dans ce chapitre nous allons essayer de définir plus précisément ce que nous entendons par "communication orale homme-machine finalisée en langage naturel".

1 Communication

Le premier terme sur lequel il faut s'entendre est celui de « communication ». Nous tâcherons dans ce paragraphe de cerner cette notion et de proposer un modèle général de la communication (Cherry, 1957 ; Shannon et Weaver, 1949 ; Ungeheuer, 1972). Il sera forcément incomplet mais cependant assez précis pour nous permettre de définir clairement les limites de notre démarche. Les définitions que nous avons adoptées sont inspirées de (Herrlitz et Hundsnurscher, 1971).

Afin de pouvoir définir la notion de « communication », nous sommes obligés de présupposer la connaissance de la notion « d'information ».

Définition

Nous appelons communication l'échange d'informations entre entités capables d'émettre et/ou de recevoir de l'information. L'orientation de cet échange détermine la qualité d'émetteur ou de récepteur des entités impliquées.

On aurait pu définir la notion d'information sur la base de la notion de communication en disant que l'information est ce qui est échangé lors d'une communication. Dans un cas comme dans l'autre on doit présupposer la connaissance de l'un au moins de ces concepts.

A titre d'exemple, considérons un système constitué d'un ordinateur et d'une unité à disques. A un instant donné des informations sont envoyées de l'unité à disques vers l'ordinateur. Il y a échange d'informations, donc communication, entre l'unité à disques et l'ordinateur. L'unité à disques joue le rôle de l'émetteur, l'ordinateur joue le rôle du récepteur.

Pour qu'il puisse y avoir échange d'informations entre émetteurs et récepteurs, certaines conditions sont nécessaires :

- il faut qu'il existe une liaison matérielle, appelée canal, entre l'émetteur et le récepteur ;
- l'émetteur doit transmettre des signaux, porteurs d'informations, via ce canal.

Nous avons ainsi introduit deux nouvelles notions intimement liées, la notion de « canal » et celle de « signal », que nous allons préciser.

Définition

Nous appelons canal la liaison matérielle entre un émetteur et un récepteur capable de transmettre des signaux de l'émetteur vers le récepteur.

Pour pouvoir jouer le rôle d'un canal, une liaison matérielle doit remplir deux conditions :

- l'émetteur doit pouvoir modifier l'état du canal en fonction de l'information à transmettre ;
- le récepteur doit être capable de constater les modifications de l'état du canal.

Définition

Nous appelons signal l'état matériel d'un canal. En tant que tel, le signal est porteur potentiel d'information.

Dans l'exemple que nous avons introduit ci-dessus, le canal serait constitué du câble reliant l'unité à disques à l'ordinateur. Un signal serait l'état électrique de ce câble à un instant donné de la communication.

Pour que le signal puisse remplir sa fonction de porteur potentiel d'information, il faut qu'il existe une relation entre le signal et l'information qu'il est censé véhiculer. C'est ce qui est réalisé par le « signe ».

Définition

Nous appelons signe l'entité qui, à un signal donné, fait correspondre une information donnée. Cette relation est établie par convention.

Ceci signifie qu'il n'y a aucun lien causal entre le signal et son information associée. L'établissement de cette relation est un problème complexe sur lequel nous n'allons pas insister ici.

Si deux entités désirent communiquer, il ne suffit pas par conséquent qu'il existe une liaison matérielle entre les deux, que l'émetteur soit capable de modifier l'état de celle-ci et que le récepteur puisse détecter ces variations, encore faut-il que le récepteur utilise les mêmes signes que l'émetteur de façon à bien associer aux signaux qu'il reçoit les informations que l'émetteur avait l'intention de lui faire parvenir. Cette deuxième liaison entre l'émetteur et le récepteur est garantie par un ensemble commun de signes appelé « répertoire ». Très souvent cependant les signes ne sont pas utilisés isolément mais sont combinés entre eux suivant des règles précises. Ces règles font également partie du répertoire. Lorsque tous ces éléments sont réunis : émetteurs, récepteurs, canal et répertoire, on parle alors d'un système de communication.

Définition

Nous appelons répertoire d'un système de communication l'ensemble contenant les signes de ce système ainsi que les règles de combinaison de ces signes.

Toujours dans le cadre de notre exemple, le répertoire correspondrait au protocole de communication régissant l'échange d'informations entre l'unité à disques et l'ordinateur. Dans notre cas particulier, le protocole est défini par le constructeur du matériel.

Ces considérations nous amènent à représenter graphiquement un système de communication quelconque sous forme du schéma de la figure 4.1 (Hörmann, 1970).

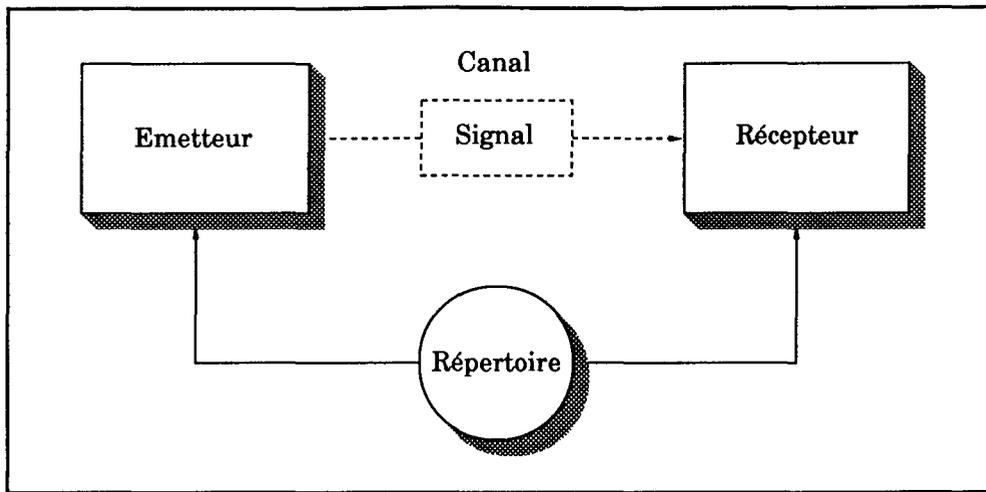


figure 4.1 : Schéma d'un système de communication

2 Communication homme-machine

Dans le cadre de nos recherches, nous avons été amenés à nous intéresser à des cas particuliers de systèmes de communication, les « systèmes de communication homme-machine »¹.

Définition

Nous appelons système de communication homme-machine un système de communication dans lequel un des intervenants (l'émetteur ou le récepteur) est un être humain, l'autre étant une machine.

Les systèmes de communication homme-machine, de nos jours, sont encore caractérisés par un certain nombre de déficiences dues à une conception inadaptée des interfaces homme-machine (Kammersgaard, 1988 ; Scapin, 1986). Une des motivations des recherches dans le domaine du

¹ Certains auteurs pensent que le terme « communication homme-machine » prête à confusion. Citons L.M. BRANSCOMB et J.C. THOMAS dans *Ease of Use: A System Design Challenge* : « People communicate via, not with, computer systems. Alphonse Chapanis, at the Johns Hopkins University, was perhaps the first to point out that the phrase "man-machine communications" is misleading. Instead, as he and others have suggested, it may help to think of a terminal as a channel through which a user communicates with other human beings-those who designed and programmed the system and put data in its memory. Some computer scientists, especially those attempting to build systems that exhibit artificial intelligence, find that view limiting. ». T. WINOGRAD partage entièrement ce point de vue. Dans *Understanding Computers and Cognition* page 123, il écrit : « Of course there is a commitment, but it is that of the programmer, not the program. If I write something and mail it to you, you are not tempted to see the paper as exhibiting language behaviour. It is a medium through which you and I interact. If I write a complex computer program that responds to things you type, the situation is still the same-the program is still a medium through which my commitments to you are conveyed. ». Nous pensons que cette nuance n'est pas particulièrement gênante.

traitement automatique de la parole est l'amélioration qualitative de l'interface homme-machine (Hayes et Reddy, 1983). Il faut cependant noter que les travaux sur la parole ne sont pas les seuls à poursuivre cet objectif (Arthur, 1986 ; Benbasat et Wand, 1984 ; Branscomb et Thomas, 1984 ; Dean M., 1982 ; Dzida, 1983 ; Edmonds, 1982 ; Gaines et Shaw, 1986a ; Gaines et Shaw, 1986b ; Hayes et al., 1981 ; Maguire, 1982 ; Newman, 1981 ; Scapin, 1986 ; Sisson et al., 1986 ; Williamson et Rohlf, 1981)¹. On peut classer ces travaux en deux catégories (Roussanaly, 1988) :

- Les travaux liés au génie logiciel qui ont pour objectif l'amélioration de la convivialité des interfaces homme-machine basées sur l'utilisation de périphériques classiques tels que l'écran, le clavier, la souris etc. en les adaptant aux processus cognitifs humains. Tout en essayant de minimiser les contraintes d'utilisation des systèmes informatiques, ces techniques reposent toutefois sur des modes de communication artificiels qui demandent un effort d'adaptation à un utilisateur non spécialiste.
- Les travaux liés à l'intelligence artificielle qui se proposent de développer des interfaces homme-machine nouvelles tels que des systèmes de reconnaissance de l'écriture manuscrite, des systèmes de compréhension de la parole etc.. Leur finalité est donc de doter les machines de certaines facultés de communication existant chez l'être humain.

Nous situons nos travaux dans la deuxième catégorie. La distinction peut paraître artificielle dans la mesure où les techniques de l'intelligence artificielle sont d'une importance croissante dans le domaine du génie logiciel (Van Lamsweerde, 1982 ; Van Lamsweerde, 1983a ; Van Lamsweerde, 1983b) et inversement (D'Archimbaud, 1985), nous l'avons faite pour la clarté de l'exposé.

3 Communication orale

Les recherches en communication homme-machine peuvent donc regrouper des thèmes aussi variés que le traitement de la parole, le traitement d'images etc.. Nous avons limité notre domaine d'investigation à la « communication orale ».

¹ En effet, comme le signale D. SCAPIN dans son *Guide ergonomique de conception des interfaces homme-ordinateur* page 2 : « L'Interface Homme-Ordinateur (IHO) englobe tous les aspects des systèmes informatiques qui influencent la participation de l'utilisateur à des tâches informatisées. Les aspects immédiatement observables sont bien sûr l'environnement physique du travail et le matériel constituant le poste de travail. Ces aspects sont peut-être les mieux connus, car étudiés depuis longtemps, mais ce ne sont pas les seuls, ni les plus importants. D'autres aspects, plus subtils, concernent les diverses façons dont les informations sont stockées et manipulées. Ceci inclut les procédures informatiques, mais aussi les outils annexes, tels que les documents papier, les aides au travail, etc. ».

Définition

Nous appelons système de communication orale un système de communication dont le répertoire est la parole.

Ce type de communication fait intervenir des facultés très diverses, de la capacité de parler et d'entendre jusqu'à la capacité de comprendre, d'interpréter, de raisonner, de répondre et de mener un dialogue.

4 Communication en langage naturel

Notre objectif étant le développement d'un système présentant une interface orale aussi naturelle que possible, toute limitation à priori sur la complexité du répertoire était inadmissible. Nous avons par conséquent choisi de réaliser un système de communication le plus proche possible du « langage naturel ».

Définition

Nous appelons système de communication en langage naturel un système de communication où le répertoire employé par les intervenants est une langue naturelle.

Nous sommes obligés de supposer que le lecteur a tout au moins une notion intuitive du concept de langue naturelle. Sans trop entrer dans les détails, disons que, pour nous, une langue naturelle est une langue qu'un enfant apprend en grandissant dans son environnement social et dont les humains se servent durant toute leur vie (Herrlitz et Hundsnurscher, 1971).

5 Communication finalisée

Si les notions de communication homme-machine, de communication orale et de communication en langage naturel sont universellement employées et communément admises, il n'en va pas de même de la notion de « communication finalisée ». Ce terme n'étant pas un terme consacré, il convient donc de le préciser. Pour cela, imaginons deux humains communiquant grâce à une langue naturelle. Un des intervenants reçoit des signaux en provenance de l'autre, fait subir un certain traitement à ces signaux et produit finalement une réponse qui se concrétise sous forme de nouveaux signaux. Ce traitement est soumis à l'influence de paramètres très divers, notamment des paramètres linguistiques (la personne appartient à une communauté linguistique bien déterminée), des paramètres ergonomiques (la personne est plus ou moins confortablement installée), des paramètres psychologiques (la personne est plus ou moins calme) etc. Si on se propose de réaliser un système automatique capable strictement des mêmes performances de communication qu'un être humain, il est indispensable de connaître en détail l'influence de tous ces éléments. On se heurte ici aux

limites de nos connaissances actuelles. Dès lors si, à moyen terme, on désire réaliser un système automatique capable de performances au moins analogues, on est obligé de s'imposer certaines limitations. Nous avons présenté au chapitre précédent plusieurs axes de simplification possibles et nous terminerons celui-ci en précisant les limites de la démarche que nous avons choisie. Notre objectif étant de réaliser un système de communication orale homme-machine grand public, certaines limitations sont inévitablement exclues. Dans une application grand public, il est hors de question de limiter à priori la complexité du langage traité. En effet l'apprentissage préalable d'un langage artificiel dans le type d'applications visées est impossible. Pour les mêmes raisons, nous sommes obligés de traiter de la parole continue et de prévoir un système indépendant du locuteur. Les limitations que nous nous sommes imposées se situent au niveau des situations de communication que nous voulons prendre en compte. Nous nous sommes intéressés exclusivement aux dialogues dont le but est la réalisation d'une tâche. La tâche nous permet de limiter l'univers du discours. Par contre, elle exige que notre système dispose de certaines facultés de compréhension. Enfin, nous pensons que le moyen d'interaction le mieux adapté à ce type de communication est un dialogue où l'initiative est partagée.

Définition

Nous appelons communication finalisée une communication dont le but est la réalisation d'une tâche.

Ce chapitre termine la partie Introduction. Nous avons essayé dans cette partie de montrer les motivations et l'évolution des recherches dans le domaine du traitement automatique de la parole, de cerner les problèmes qui se posent dans le cadre du développement d'un système tel que le système du CRIN et de mettre en évidence les particularités qui caractérisent notre approche. Nous allons maintenant aborder des problèmes plus spécifiques au niveau syntaxico-sémantique.

QUELQUES REPERES LINGUISTIQUES

Les traitements et les informations mises en œuvre dans la composante syntaxico-sémantique s'appuient sur divers modèles linguistiques. Le chapitre suivant présente les principaux modèles qui nous ont inspirés dans la conception et la réalisation de cette composante.

1 Des modèles linguistiques, points d'appui à notre étude

Lorsqu'on est confronté à un problème tel que la réalisation d'un système automatique capable de comprendre la langue naturelle parlée et de mener un dialogue intelligent, on est forcément amené à s'intéresser aux sciences du langage (Bátori, 1981 ; Gross, 1981 ; Woods et Makhoul, 1974)¹. En effet, il nous a paru indispensable d'intégrer dans notre système une expertise linguistique. Cette expertise ne peut venir que des experts de la langue, à savoir des linguistes, des psycholinguistes, des psychologues du langage etc..

¹ I.S. BATORI note à ce sujet dans *Die Grammatik aus der Sicht kognitiver Prozesse* page 18 : « (...) Zu beachten ist, daß hier sprachliche Prozesse erörtert werden, die sich im psychologischen Raum abspielen. (...) Diese letztere Einschränkung hat auch positive Aspekte, die Anlehnung an die Psychologie und insbesondere an die Psycholinguistik ermöglicht nämlich die Einengung der möglichen Verarbeitungsmodelle auf die Menge von Modellen, die mit den Ergebnissen der psycholinguistischen Forschung kompatibel sind. Hierfür ist es allerdings erforderlich, sich ein Bild über die psycholinguistische Verstehensprozesse (und über Verstehensprozesse überhaupt) zu verschaffen. » Cette approche, qui peut à priori sembler assez contraignante, a donc des avantages certains. Cependant, une difficulté à laquelle on se heurte est assez bien décrite par M. GROSS dans *La formalisation des langues naturelles* page 103 : « (...) Aujourd'hui, le marché méta-linguistique est saturé de théories, il est devenu matériellement impossible de les comparer. Chacune se développe sans influence sur les autres en engendrant des langages ésotériques. ».

Ces différents experts s'intéressent cependant à beaucoup d'aspects divers et variés du phénomène langage. De plus, leurs approches et leurs préoccupations ont fortement évolué dans le temps. On peut distinguer trois étapes dans cette évolution suivant la démarche qui les caractérise (Bronckart, 1977) :

- la première se caractérise par une démarche structuraliste «de surface» ; on avait comme objectif essentiel de définir les unités linguistiques pertinentes et d'en analyser les relations avec la réalité extralinguistique, qu'elle soit physique, psychologique ou socio-culturelle (Jakobson, 1963 ; Martinet, 1960 ; Sapir, 1953 ; Saussure, 1916 ; Trubetzkoy, 1939) ;
- la deuxième se caractérise par la formulation d'hypothèses concernant les structures non-apparentes de la langue, à savoir les règles sous-jacentes qui «expliquent» les structures de surface (Chomsky, 1957 ; Chomsky, 1965 ; Chomsky, 1971 ; Chomsky et Miller, 1963 ; Fillmore, 1968 ; Fillmore, 1972 ; Fillmore, 1977 ; Hjelmslev, 1943) ;
- la troisième se caractérise par la prise en considération d'unités de langue beaucoup plus larges que les précédentes. Le linguiste se centre désormais sur des séquences d'énoncés, ou discours, tels qu'ils sont émis dans les diverses situations de la vie réelle (Benvéniste, 1966 ; Benvéniste, 1974 ; Culioli et al., 1970 ; Dik, 1978 ; Dik, 1980 ; Dik, 1983a ; Dirven et Fried, 1987 ; Foley et Van Valin, 1984 ; Moeschler, 1985).

Dans le cadre de notre projet nous avons bien sûr été amené à nous intéresser à tous ces problèmes. Cependant, pour le travail de recherche décrit dans cette thèse, nous nous sommes plus particulièrement penchés sur les travaux des deux dernières catégories. De plus plutôt que de présenter une revue complète et exhaustive des modèles linguistiques développés et proposés par les adeptes de cette approche, ce qui vu le nombre de travaux serait bien difficile, nous allons nous limiter aux modèles qui ont eu une influence déterminante sur nos choix de réalisation. Les modèles linguistiques qui ont servi de point d'appui à notre étude sont des types suivants :

- des modèles syntaxiques tels que les grammaires formelles et les automates ;
- des modèles sémantiques telles que les grammaires de cas et les grammaires fonctionnelles ;
- des modèles cognitifs telles que les dépendances conceptuelles.

Dans les paragraphes suivants, nous allons décrire sommairement ces modèles et en discuter les aspects qui nous ont intéressés plus particulièrement dans le cadre de la réalisation de notre système.

1.1 Les modèles syntaxiques

Les modèles syntaxiques présupposent un certain bagage algébrique. Le but n'étant pas ici de faire un cours de théorie des langages ou de théorie

des automates, nous renvoyons le lecteur à (Beckman, 1980 ; Ginzburg, 1969 ; Gross et Lentin, 1967 ; Hopcroft et Ullman, 1979 ; Lewis et Papadimitriou, 1981 ; Pair, 1966) pour un exposé plus détaillé des concepts et des problèmes abordés dans les deux paragraphes suivants. Les questions essentielles auxquelles tentent de répondre les modèles syntaxiques sont résumées dans (Chomsky et Miller, 1963)¹. On peut en dégager deux approches différentes de la description d'un langage :

- soit fournir un mécanisme fini permettant d'engendrer l'infinité de phrases d'un langage donné ;
- soit fournir un mécanisme fini permettant de décider si une phrase donnée appartient à un langage donné ou non.

Le premier mécanisme correspond à ce qui est communément appelé grammaire formelle², le deuxième relève de la théorie des automates abstraits³. Une caractéristique commune aux deux approches est le fait qu'on s'appuie sur des représentations finies pour décrire des langages infinis. On ne pourra donc jamais représenter tous les langages. Le maximum qu'on puisse espérer, c'est de pouvoir en représenter quelques-uns parmi les plus intéressants. La question de savoir si le langage naturel en fait partie résume toute la problématique inhérent à ce type d'approche⁴. Enfin faut-il en-

¹ N. CHOMSKY et G.A. MILLER affirment dans *L'analyse formelle des langues naturelles* page 15 : « Si on veut caractériser un langage avec précision, il importe d'établir un principe qui permette de séparer les séquences d'éléments atomiques qui forment des phrases de celles qui n'en forment pas. Il n'est pas possible, pour réaliser cette distinction, de se contenter d'une simple liste, puisque dans tout système intéressant il n'existe pas de limite à la longueur des phrases. Deux méthodes se présentent à nous pour obtenir ce résultat. Ou bien nous tenterons de développer un test opérationnel d'un genre tel qu'il puisse distinguer les phrases des non-phrases, - ou bien nous essayerons de mettre au point une procédure réursive qui énumère la liste infinie des phrases. »

² M. GROSS et A. LENTIN s'expriment de la façon suivante dans les *Notions sur les grammaires formelles* page 44 : « En d'autres termes le but d'une grammaire est de spécifier, de caractériser les mots (les phrases) appartenant à un langage. Une grammaire formelle apparaît donc comme un algorithme (un ensemble de règles qui permettent de caractériser un langage). »

³ Voir entre autres N. CHOMSKY et G.A. MILLER, *L'analyse formelle des langues naturelles* pp 64-66 : « (...) la conception saussurienne conduit à utiliser pour nos recherches trois sortes de modèles (...) Le mécanisme A est une grammaire qui engendre des phrases avec leurs descriptions structurales ; c'est-à-dire que A représente l'intuition linguistique du sujet parlant, (...) Le mécanisme B (...) représente les processus de perception impliqués dans la détermination de la structure d'une phrase. (...) Le mécanisme C représente la *faculté du langage*, (...) En fait, il est évident que si A est la grammaire d'une langue naturelle L, il n'y a pas de mécanisme de type B qui donnera toujours à la sortie une description structurale correcte si et seulement si une phrase de L est donnée à l'entrée. (...) Vu de cette manière, divers aspects assez importants de la théorie linguistique peuvent être considérés, en principe au moins, comme relevant de la théorie générale des automates (abstrait). (...) »

⁴ H.R. LEWIS et C.H. PAPADIMITRIOU signalent dans *Elements of the Theory of Computation* pages 33-34 : « Let us be somewhat more precise about the notion of "finite representation of a language." The first point to be made is that any such representation must itself be a string, a finite sequence of symbols over some alphabet Σ . Second, we certainly want different languages to have different representations, otherwise the term *representation* could hardly be considered appropriate. But these two requirements already imply that the possibilities for finite representations are severely limited. For the set Σ^* of strings over an alphabet Σ is countably infinite, so the number of possible representations of languages is countably infinite. (This would remain true even if we were not bound to use a particular alphabet Σ , so long as the total number of available symbols

core signaler que la séparation entre génération et analyse, entre grammaires et automates sert ici la clarté de l'exposé. En réalité les deux approches se rejoignent et on peut très bien imaginer des procédures d'analyse basées sur des grammaires ou des procédures de génération s'appuyant sur des automates.

1.1.1 Les grammaires formelles

Dans les trois versions successives de sa théorie générative, la théorie strictement syntaxique (Chomsky, 1957), la théorie standard (Chomsky, 1965) et la théorie standard élargie (Chomsky, 1971), Chomsky tente en vain de formaliser le langage naturel en s'appuyant sur les grammaires formelles (Bronckart, 1977 ; Gross, 1981)¹. Néanmoins celles-ci gardent à nos yeux un intérêt certain. Rappelons brièvement en quoi consiste une grammaire formelle.

Définition

Une grammaire formelle est définie par un quadruplet (T, N, x, P)

où :

T est un ensemble de symboles fini et non vide appelé alphabet (ou vocabulaire) terminal,

N est un ensemble de symboles fini, non vide et disjoint de T appelé alphabet (ou vocabulaire) auxiliaire (ou non-terminal),

x est un élément distingué de N appelé axiome,

P est un ensemble fini et non vide de productions (ou règles de réécriture) telles que :

$$\forall p_i \in P, (p_i) : X_i \rightarrow Y_i$$

avec :

- $X_i \in (N \cup T)^+$ et $Y_i \in (N \cup T)^*$,

was countably infinite.) On the other hand, the set of all possible languages over a given alphabet Σ —that is, 2^{Σ^*} —is uncountably infinite, since 2^{\aleph_0} , and hence the power set of any countably infinite set, is not countably infinite. With only a countable number of representations and an uncountable number of things to represent, we are unable to represent all languages finitely. Thus the most we can hope for is to find finite representations, of one sort or another, for at least some of the more interesting languages. »

¹ Dans *Théories du langage* J.P. BRONCKART fait un bilan modéré de la grammaire générative, pages 236-243 et M. GROSS dans *La formalisation des langues naturelles* argumente de la façon suivante : « Après avoir affirmé que nous ne savons presque pas analyser les phénomènes syntaxiques, examinons ce que nous connaissons mieux, c'est-à-dire les grammaires formalisées. (...) ».

- $(N \cup T)^*$ désignant l'ensemble des chaînes pouvant être construites en concaténant les éléments de $(N \cup T)$ y compris la chaîne vide,
- $(N \cup T)^+$ désignant l'ensemble des chaînes pouvant être construites en concaténant les éléments de $(N \cup T)$ en excluant la chaîne vide,
- \rightarrow désignant la possibilité de réécrire X_i en Y_i .

Chomsky propose une classification des grammaires formelles suivant la complexité de leurs règles de production (Chomsky, 1959) :

- Les grammaires les plus générales sont les grammaires de type 0. Aucune contrainte n'est imposée aux règles de réécriture. Ces grammaires sont encore appelées systèmes semi-thuégiens ou systèmes de réécriture généraux.
- Les grammaires de type 1 sont moins générales que les précédentes, les règles qu'elles contiennent doivent être de la forme :

$$XqY \rightarrow XZY$$

avec $X, Y, Z \in (N \cup T)^*$ et $q \in N$.

Ceci signifie que le symbole q ne peut être réécrit en Z qu'en présence du contexte gauche X et du contexte droit Y . Cette prise en compte du contexte est la raison pour laquelle ces grammaires sont appelées grammaires contextuelles.

- Les grammaires de type 2 sont encore plus contraignantes, elles ne peuvent contenir que des règles de la forme :

$$q \rightarrow X$$

avec $q \in N$ et $X \in (N \cup T)^*$

Cette façon d'écrire signifie que le symbole q peut être réécrit en X quel que soit son contexte. On appelle ces grammaires d'ailleurs grammaires hors-contexte ou grammaires à contexte libre.

- Enfin les grammaires les plus particulières que distingue Chomsky sont les grammaires de type 3, elles ne sont composées que de règles de la forme :

$$q \rightarrow a$$

et

soit $q \rightarrow ar$ (dans le cas d'une grammaire linéaire droite)
soit $q \rightarrow ra$ (dans le cas d'une grammaire linéaire gauche)

avec $a \in T$ et $q, r \in N$

Ce sont les grammaires régulières.

Suivant le type de la grammaire qui a été utilisée pour le générer, on parlera de langage semi-thuéien, de langage contextuel, de langage hors-contexte ou de langage régulier, le problème fondamental étant de savoir quel type de grammaire est nécessaire pour décrire un langage donné. Plus cette grammaire sera générale, plus les procédures qui vont l'exploiter seront complexes. Chomsky pensait pouvoir décrire la langue naturelle grâce à une grammaire contenant des règles contextuelles et hors-contexte, la composante syntagmatique de sa grammaire, ainsi que des règles de production semi-thuéiennes (voir figure 5.1), la composante transformationnelle de sa grammaire (Chomsky, 1957 ; Chomsky, 1965 ; Chomsky, 1971 ; Chomsky et Miller, 1963)¹. D'autres pensent que la langue naturelle est un langage hors-contexte (Salkoff, 1979)². Le débat reste ouvert, nous n'avons pas la prétention de le trancher. Voyons plutôt quels sont les aspects que nous pouvons dégager de ce modèle et qui pourraient être utiles pour des concepteurs de systèmes automatiques de communication orale en langage naturel :

- Une particularité intéressante du modèle des grammaires formelles est son caractère déclaratif. En effet en fournissant un ensemble de règles censées décrire un langage donné, le linguiste ne fait aucune pré-supposition sur les procédures qui vont exploiter ces règles. On peut imaginer des procédures de génération en profondeur d'abord aussi bien que des procédures de génération en largeur d'abord etc. Le caractère déclaratif du modèle permet une mise au point incrémentale de l'ensemble des règles nécessaires dans un système sans avoir à toucher aux procédures de traitement de ces règles (Laurière, 1982 ; Pinson, 1981).
- Les langages algébriques engendrés par des grammaires formelles sont de plus basés sur l'opération de concaténation de symboles. Cette opération est une opération non commutative (Beckman, 1980 ; Gross et Lentin, 1967 ; Hopcroft et Ullman, 1979 ; Lewis et Papadimitriou, 1981). Elle permet d'exprimer des contraintes positionnelles assez fortes, ce qui est intéressant dans le cadre de systèmes tel que le nôtre.
- Finalement ces modèles sont déjà anciens. Les mécanismes y associés sont bien maîtrisés et on en connaît les limites et les possibilités (Kay,

¹ Toujours dans *L'analyse formelle des langues naturelles*, N. CHOMSKY et G.A. MILLER écrivent page 33 : « La question de savoir si des langues naturelles, envisagées comme ensemble de phrases, peuvent être engendrées par tel ou tel type de grammaire de constituants est un problème intéressant et important. Cependant, il n'y a pas de doute que l'ensemble des descriptions structurales que l'on peut associer, par exemple, à la langue française, ne peut pas en principe être engendré par une grammaire de constituants, quelle que soit la complexité de ses règles. » et à la page suivante : « En général, pour chaque type particulier de difficulté qui se rencontre dans une grammaire de constituants, on peut imaginer un palliatif particulier (*ad hoc*) quelconque qui permet de la tourner. Il serait bien préférable, évidemment, d'opérer une révision conceptuelle qu'on pourrait tenir pour réussie si elle parvenait à éviter tout l'ensemble des difficultés d'une manière uniforme, tout en permettant à une simple grammaire de constituants de rendre compte, sans subir d'altération spéciale, de la classe des cas auxquels elle convient, cas qui initialement avaient motivé son développement. Pour autant que nous le sachions, la théorie de grammaires transformationnelles est la seule qui permette d'espérer que ce but puisse être atteint. »

² On lira à ce sujet M. SALKOFF dans son *Analyse syntaxique du Français : grammaire en chaîne* page 141 : « Des résultats nouveaux montrent que la grammaire en chaîne, quoique contenant des règles CS, ne dépasse pas la capacité générative d'une grammaire CF (§ 3.1). Puisque la grammaire en chaîne décrit le français, il s'ensuit qu'une grammaire CF du français, et partant, d'une langue naturelle, peut être écrite. »

1985)¹. Les applications informatiques, des compilateurs (Aho, Sethi et Ullman, 1986) ou générateurs de compilateurs (Johnson, 1975 ; Lesk et Schmidt, 1975) aux systèmes de compréhension de la parole (Pierrel, 1975 ; Mariani, 1982), sont nombreuses.

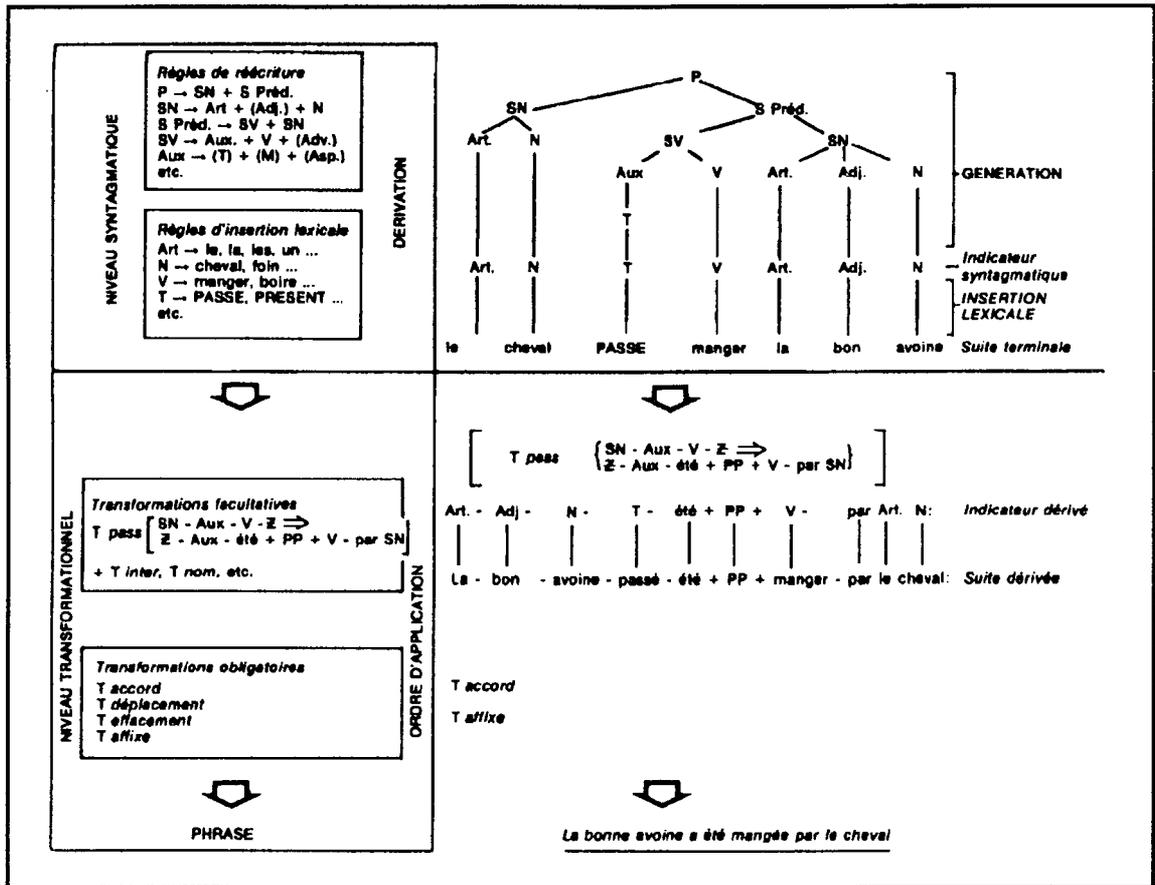


fig. 5.1 : Le modèle strictement syntaxique de Chomsky extrait de (Bronckart, 1977)

Les grammaires formelles ont cependant également certains inconvénients :

- Il s'agit d'un modèle dont le but est la génération de productions (Chomsky et Miller, 1971 ; Gross et Lentin, 1967 ; Lewis et Papadimitriou, 1981). Or, dans un système automatique de communication en langage naturel, on doit résoudre deux problèmes fondamentalement différents faisant intervenir des connaissances linguistiques. Le premier consiste à acquérir les énoncés du locuteur et à transformer ces derniers jusqu'à en obtenir une représentation en machine qui

¹ M. KAY, dans *An Overview of Natural Language Understanding*, souligne plus particulièrement les avantages d'une grammaire à contexte libre par rapport à une grammaire transformationnelle : « From a computational point of view, CFG has the enormous advantage over transformational grammar that there are simple, elegant, and very well understood algorithms for using such grammars both to generate and analyze sentences. Anticipating a later part of this paper, we can ascribe this to the fundamentally *monotonic* nature of this kind of grammar. Transformational grammar lacks this property and, in fact, there are no general analysis methods from this class of grammars. ».

permette à celle-ci de les «interpréter» et, par des mécanismes de «raisonnement», de générer la représentation interne d'une réponse à fournir au locuteur. Le deuxième problème consiste alors à transformer cette nouvelle structure jusqu'à obtenir un énoncé compréhensible pour le locuteur. Si ce dernier est bien un problème de génération, le premier par contre est un problème d'analyse. Dans le cadre de nos recherches nous nous sommes intéressés exclusivement au problème de l'analyse.

- Un autre problème auquel on se heurte en utilisant le modèle des grammaires formelles est celui de la représentation de l'énoncé qu'on obtient. Une grammaire permet en effet de construire, pour une production donnée, ce qu'on appelle son arbre de dérivation. Cet arbre représente la suite des réécritures successives qui ont dû être effectuées à partir de l'axiome pour aboutir à la production (Beckman, 1980 ; Gross et Lentin, 1967 ; Hopcroft et Ullman, 1979 ; Winograd, 1983). Malheureusement ce type de représentation n'est pas d'une grande utilité pour l'interprétation.

Il semble donc que les grammaires formelles ne répondent pas à priori à nos besoins en matière de modèles linguistiques. En effet, nous avons été contraints de faire coopérer ce type de modèle avec d'autres modèles linguistiques pour atteindre notre objectif.

1.1.2 Les automates

Contrairement à une grammaire formelle qui définit un mécanisme de génération de toutes les productions possibles d'un langage, un automate décrit un algorithme capable de décider si oui ou non une production donnée appartient à un langage donné. Il s'agit par conséquent d'un dispositif de reconnaissance de langages (Lewis et Papadimitriou, 1981). On distingue différents types d'automates suivant le type de langages qu'ils doivent reconnaître (Beckman, 1980 ; Ginzburg, 1969 ; Gross et Lentin, 1967 ; Hopcroft et Ullman, 1979 ; Lewis et Papadimitriou, 1981) :

- Les machines de Turing, qui sont des dispositifs composés d'une unité de contrôle, d'une bande support d'information et d'une tête de lecture-écriture. L'unité de contrôle se trouve dans un état précis parmi un ensemble fini d'états possibles. On distingue plusieurs états particuliers, l'état initial ainsi que les états finaux. La bande, limitée à gauche mais infinie vers la droite, est composée de cases pouvant contenir chacune un symbole parmi un ensemble fini de symboles admissibles. Au début, la chaîne de symboles d'entrée se trouve dans les cases les plus à gauche de la bande. La chaîne d'entrée est composée de symboles pris dans un sous-ensemble de l'ensemble des symboles admissibles. Les autres cases contiennent un symbole particulier, le symbole blanc qui n'est pas un symbole d'entrée. Au départ, la machine se trouve dans l'état initial. La tête de lecture-écriture est positionnée sur la case la plus à gauche. Les symboles sont lus par la tête de lecture-écriture ce qui provoque, en fonction de l'état de l'unité de contrôle et du symbole lu, un changement d'état de l'unité de contrôle, le remplacement du symbole sur la bande par un autre symbole et un déplacement de la tête de lecture-écriture d'une case vers la gauche ou vers la droite sachant

que la tête de lecture-écriture ne peut cependant se déplacer au-delà de la limite gauche de la bande. Le processus de lecture-écriture peut ensuite recommencer. Le langage accepté par cette machine est l'ensemble des chaînes de symboles d'entrée qui provoquent le passage de cette machine dans un des états finaux après un nombre fini de transitions d'états. Il est cependant possible que pour des chaînes de symboles non acceptables, la machine ne s'arrête jamais. La figure 5.2 présente une machine de Turing. De façon plus formelle :

Définition

Une machine de Turing est définie par un septuplet $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$

où :

Q est l'ensemble fini des états,

Γ est l'ensemble fini des symboles admissibles,

B est un élément particulier de Γ , le symbole blanc,

Σ est l'ensemble des symboles d'entrée, un sous-ensemble de Γ ne contenant pas l'élément B ,

q_0 est un élément particulier de Q , l'état initial,

δ est une fonction de $Q \times \Gamma$ vers $Q \times \Gamma \times \{L, R\}$, la fonction de transition,

F est l'ensemble des états finaux, sous-ensemble de Q .

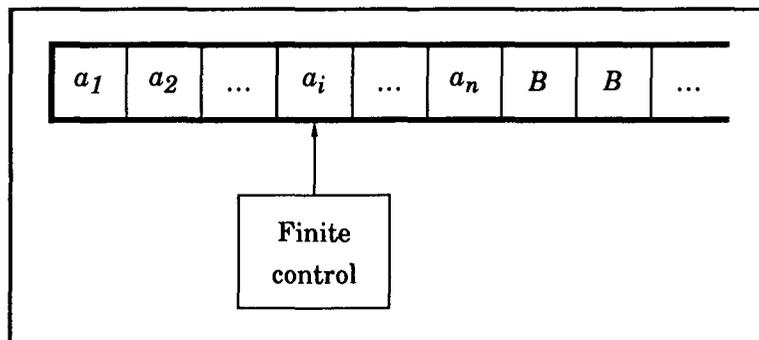


fig. 5.2 : Machine de Turing
d'après (Hopcroft et Ullman, 1979)

On note $\alpha_1 q \alpha_2$ une description instantanée d'une machine de Turing M , où q est l'état courant de M et $\alpha_1 \alpha_2$ la chaîne de Γ^* constituant le contenu de la bande soit jusqu'au symbole non blanc le plus à droite soit jusqu'au symbole à gauche de la tête de lecture-écriture suivant lequel des deux symboles est le plus à droite. Enfin, la tête de lecture-écriture est positionnée sur le symbole à l'extrême gauche de α_2 ou, si α_2 est la chaîne vide, sur un symbole blanc.

Soit $X_1X_2\dots X_{i-1}qX_i\dots X_n$ une description instantanée de M . Si $\delta(q, X_i) = (p, Y, L)$ et si $i > 1$, on peut noter la relation \rightarrow_M :

$$X_1X_2\dots X_{i-1}qX_i\dots X_n \rightarrow_M X_1X_2\dots X_{i-2}pX_{i-1}YX_{i+1}\dots X_n$$

Si $i=1$, une telle relation n'existe pas, la tête de lecture ne pouvant se déplacer au-delà de la limite gauche de la bande. D'un autre côté, si $\delta(q, X_i) = (p, Y, R)$, on peut noter :

$$X_1X_2\dots X_{i-1}qX_i\dots X_n \rightarrow_M X_1X_2\dots X_{i-1}YpX_{i+1}\dots X_n$$

Une description instantanée est le résultat d'une transition d'une autre description instantanée si entre la deuxième et la première il existe une relation \rightarrow_M . Si une description instantanée est le résultat d'un nombre fini de transitions (éventuellement zéro) d'une autre description instantanée, alors la relation \rightarrow_M^* lie la deuxième à la première.

Le langage accepté par M et noté $L(M)$ est l'ensemble des chaînes de Σ^* qui provoquent le passage de M dans un état final lorsqu'elles sont justifiées à gauche sur la bande, M se trouvant dans l'état q_0 et la tête de lecture-écriture étant positionnée sur le symbole le plus à gauche :

$$L(M) = \{w/w \in \Sigma^* \text{ et } q_0w \rightarrow_M^* \alpha_1p\alpha_2 \text{ avec } p \in F \text{ et } \alpha_1 \in \Gamma^* \text{ et } \alpha_2 \in \Gamma^*\}$$

Il est intéressant de noter que les langages acceptés par les machines de Turing sont également des langages engendrés par des grammaires de type 0¹.

Il existe des variantes de cette machine de Turing, les machines de Turing à bande infinie dans les deux sens, les machines de Turing à plusieurs bandes, les machines de Turing indéterministes, les machines de Turing multi-dimensionnelles, les machines de Turing à têtes multiples ainsi que les machines de Turing "off-line". On peut montrer qu'aucune de ces variantes ne dépasse la puissance de description de la machine de Turing, pour tout langage accepté par l'une d'elles, il existe une machine de Turing acceptant ce même langage². Notre but n'étant pas de présenter ici une étude détaillée des machines de Turing, nous n'allons pas nous attarder sur ce sujet. Définissons cependant ce que nous entendons par machine de Turing indéterministe, c'est sur cette définition que s'appuiera la définition du paragraphe suivant. Une machine de Turing indéterministe est une machine de Turing qui, à chaque transition en fonction d'un état donné et d'un symbole lu, a le choix entre plusieurs possibilités. Chaque possi-

¹ J.E. HOPCROFT et J.D. ULLMAN démontrent dans *Introduction to Automata Theory, Languages and Computation* page 221 : « If L is $L(G)$ for unrestricted grammar $G = (V, T, P, S)$, then L is an r.e. language. » et page 222 : « If L is an r.e. language, then $L=L(G)$ for some unrestricted grammar G . ». Page 150 du même ouvrage, les auteurs nous indiquent que : « A language that is accepted by a Turing machine is said to be *recursively enumerable* (r.e.). ».

² Toujours dans *Introduction to Automata Theory, Languages and Computation*, J.E. HOPCROFT et J.D. ULLMAN démontrent les théorèmes suivants, page 159 : « L is recognized by a Turing machine with a two-way infinite tape if and only if it is recognized by a TM with a one-way infinite tape. », page 161 : « If a language L is accepted by a multitape Turing machine, it is accepted by a single-tape Turing machine. », page 164 : « If L is accepted by a nondeterministic Turing machine, M_1 , then L is accepted by some deterministic Turing machine, M_2 . », à la même page : « If L is accepted by a two-dimensional TM M_2 , then L is accepted by a one-dimensional TM M_1 . » et finalement page 165 : « If L is accepted by some k -head TM M_1 , it is accepted by a one-head TM M_2 . »

bilité consiste en un état nouveau, un symbole à écrire et une direction de mouvement de la tête de lecture-écriture. Une chaîne de symboles est acceptée s'il existe une séquence de transitions finies de la machine conduisant à un état final.

- Les automates à mémoire linéairement bornée sont des cas particuliers de machines de Turing indéterministes. Ils répondent aux deux conditions suivantes :
 - 1) l'ensemble des symboles d'entrée contient deux symboles particuliers, φ et $\$$, les marqueurs de fin gauche et droite ;
 - 2) l'automate à mémoire linéairement bornée ne peut déplacer sa tête de lecture-écriture à gauche de φ ou à droite de $\$$, de plus les symboles φ et $\$$ ne peuvent être remplacés par d'autres symboles.

Un automate à mémoire linéairement bornée est une machine de Turing qui, au lieu d'avoir une bande infinie à sa disposition, est limitée à la partie de la bande contenant la chaîne de symboles d'entrée plus les deux positions contenant les marqueurs de fin.

Définition

Un automate à mémoire linéairement bornée est défini par un octuplet $(Q, \Sigma, \Gamma, \delta, q_0, \varphi, \$, F)$

où :

- $Q, \Sigma, \Gamma, \delta, q_0$, et F ont la même signification que pour une machine de Turing indéterministe,
- φ et $\$$ sont des symboles de Σ , les marqueurs de fin gauche et droite.

Le langage $L(M)$ accepté par un automate à mémoire linéairement bornée (M) est :

$$L(M) = \{w/w \in (\Sigma - \{\varphi, \$\})^* \text{ et } q_0 \varphi w \$ \xrightarrow{M^*} \alpha \beta \text{ avec } q \in F\}$$

Les langages acceptés par les automates à mémoire linéairement bornée sont des langages dépendants du contexte¹.

- Les automates à pile sont des dispositifs encore plus particuliers. Ils sont constitués d'une unité de contrôle finie, d'une bande, d'une tête de lecture et d'une pile. La pile est une chaîne de symboles d'un alphabet particulier, l'alphabet de pile. Le symbole le plus à gauche de la pile est considéré comme étant le sommet de la pile. L'automate est indéterministe dans la mesure où dans un état donné, il a un nombre fini de choix de transition, les transitions étant de deux types. Le premier type de transition consomme un symbole d'entrée. Suivant le symbole

¹ La démonstration des théorèmes suivants : « If L is a CSL, then L is accepted by some LBA. » et « If $L=L(M)$ for LBA $M=(Q, \Sigma, \Gamma, \delta, q_0, \varphi, \$, F)$, then $L-\{\varphi, \$\}$ is a CSL. » peut également être trouvée dans *Introduction to Automata Theory, Languages and Computation* page 225 et 226 de J.E. HOPCROFT et J.D. ULLMAN.

d'entrée lu, le sommet de la pile et l'état de l'unité de contrôle, un ou plusieurs choix sont possibles. Chaque choix correspond à un nouvel état de l'unité de contrôle ainsi qu'à une chaîne de symboles (éventuellement vide) destinée à remplacer le sommet de la pile. Après un choix de l'unité de contrôle, la tête de lecture est déplacée d'un symbole vers la droite. Le deuxième type de transition est similaire au premier sauf qu'aucun symbole d'entrée n'est consommé. Après transition, la position de la tête de lecture n'a pas changé. L'automate peut ainsi manipuler la pile sans lire de symboles d'entrée. Le langage accepté par un automate à pile peut être défini de deux façons différentes. La première consiste à dire qu'il s'agit de l'ensemble des chaînes de symboles pour lesquelles l'automate à pile, après un certain nombre de transitions, a vidé sa pile. L'autre consiste à distinguer dans l'ensemble des états possibles de l'automate quelques états finaux et de considérer comme langage accepté par l'automate l'ensemble des chaînes de symboles qui font passer l'automate dans un état final après un nombre fini de transitions. En fait on peut montrer que les deux façons de définir ce langage sont strictement équivalentes. Plus formellement :

Définition

Un automate à pile est défini par un septuplet $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

où :

Q est l'ensemble fini des états,

Σ est l'ensemble fini des symboles d'entrée appelé alphabet d'entrée,

Γ est l'ensemble fini des symboles de pile appelé alphabet de pile,

q_0 est un élément particulier de Q , l'état initial,

Z_0 est un élément particulier de Γ appelé symbole de départ,

F est l'ensemble des états finaux, sous-ensemble de Q ,

δ est une fonction de $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$ vers des sous-ensembles finis de $Q \times \Gamma^*$, ε étant la chaîne de symboles vide.

L'interprétation des transitions se fera de la façon suivante :

$$\delta(q, a, Z) = \{(p_1, \gamma_1), (p_2, \gamma_2), \dots, (p_n, \gamma_n)\}$$

où q et p_i , avec $1 \leq i \leq n$, sont des états, a est un symbole d'entrée, Z est un symbole de pile et γ_i , avec $1 \leq i \leq n$, est une chaîne de symboles de pile ;

signifie que l'automate à pile (M), s'il se trouve dans l'état q , si le symbole lu est a et que le sommet de pile est Z , peut, pour tout i , $1 \leq i \leq n$, entrer dans l'état p_i , remplacer le sommet de pile Z par γ_i et déplacer la tête de lecture d'un symbole vers la droite. Le symbole le plus à gauche de γ_i sera le nouveau sommet de pile.

$$\delta(q, \varepsilon, Z) = \{(p_1, \gamma_1), (p_2, \gamma_2), \dots, (p_n, \gamma_n)\}$$

signifie que l'automate, s'il se trouve dans l'état q et que le sommet de pile est Z , peut, indépendamment du symbole lu, entrer dans l'état p_i et remplacer Z par γ_i , pour tout i , $1 \leq i \leq n$. Dans ce cas, la tête de lecture ne bouge pas.

Une description instantanée est un triplet (q, w, γ) , où q est un état, w une chaîne de symboles d'entrée et γ une chaîne de symboles de pile. On peut noter la relation \rightarrow_M :

$$(q, aw, Z\alpha) \rightarrow_M (p, w, \beta\alpha) \text{ si } (p, \beta) \in \delta(q, a, Z), \text{ avec } a \in \Sigma \cup \{\epsilon\}$$

La fermeture réflexive et transitive de \rightarrow_M est notée \rightarrow_M^* . Le langage accepté par état final $L(M)$ est :

$$L(M) = \{w / (q_0, w, Z_0) \rightarrow_M^* (p, \epsilon, \gamma) \text{ avec } p \in F \text{ et } \gamma \in \Gamma^*\}$$

le langage accepté par pile vide $N(M)$ est :

$$N(M) = \{w / (q_0, w, Z_0) \rightarrow_M^* (p, \epsilon, \epsilon) \text{ avec } p \in Q\}$$

Les deux définitions sont équivalentes¹. Dans les deux cas il s'agit de langages à contexte libre².

- Enfin les automates les plus simples sont les automates finis qui consistent en un ensemble fini d'états et un ensemble de transitions d'un état vers un autre provoqué par la lecture d'un symbole. Pour chaque symbole d'entrée, il y a une et une seule transition d'un état vers un autre (éventuellement le même). Un état particulier est l'état initial, d'autres états particuliers sont les états finaux. L'automate accepte une chaîne de symboles x si la séquence de transitions correspondant aux symboles de x fait passer l'automate de l'état initial dans un état final. Plus formellement :

Définition

Un automate fini est défini par un quintuplet $(Q, \Sigma, \delta, q_0, F)$

où :

Q est l'ensemble fini des états,

Σ est l'ensemble fini des symboles d'entrée appelé alphabet d'entrée,

δ est la fonction de transition de $Q \times \Sigma$ vers Q ,

q_0 est un élément particulier de Q , l'état initial,

¹ J.E. HOPCROFT et J.D. ULLMAN démontrent, toujours dans *Introduction to Automata Theory, Languages and Computation*, page 114 : « If L is $L(M_2)$ for some PDA M_2 , then L is $N(M_1)$ for some PDA M_1 . » et page 115 : « If L is $N(M_1)$ for some PDA M_1 , then L is $L(M_2)$ for some PDA M_2 . »

² Tel que le démontrent J.E. HOPCROFT et J.D. ULLMAN page 115 : « If L is a context-free language, then there exists a PDA M such that $L=N(M)$. » et page 116 : « If L is $N(M)$ for some PDA M , then L is a contexte-free language. » dans *Introduction to Automata Theory, Languages and Computation*.

F est l'ensemble des états finaux, sous-ensemble de Q .

On peut étendre la fonction de transition aux chaînes de symboles en définissant une fonction δ' de $Q \times \Sigma^*$ vers Q tel que :

- $\forall q \in Q, \delta'(q, \varepsilon) = q$
- $\forall w \in \Sigma^*, a \in \Sigma, q \in Q : \delta'(q, wa) = \delta(\delta'(q, w), a)$

où ε est la chaîne vide. Le langage accepté par l'automate fini M est l'ensemble :

$$L(M) = \{x / \delta'(q_0, x) \in F\}$$

Ce langage est un langage régulier¹.

Il n'est pas dans notre propos d'étudier plus en détail ces différents types d'automates, le lecteur trouvera des développements intéressants dans (Beckman, 1980 ; Ginzburg, 1969 ; Gross et Lentin, 1967 ; Hopcroft et Ullman, 1979 ; Lewis et Papadimitriou, 1981). Retenons simplement qu'à chaque langage engendré par un type de grammaire donné correspond un automate d'un certain type capable de décider si oui ou non une production donnée appartient à ce langage. Les équivalences entre grammaires et automates sont résumées dans le tableau de la figure 5.3.

type de grammaire	automate équivalent
système semi-thuéien	machine de Turing
grammaire contextuelle	automate à mémoire linéairement bornée
grammaire hors-contexte	automate à pile
grammaire régulière	automate fini

figure 5.3. : Equivalences entre grammaires et automates

Voyons maintenant quels sont les aspects intéressants pour les concepteurs de systèmes :

- Contrairement aux règles de production, les automates sont des dispositifs destinés essentiellement à l'analyse de langages. Comme nous l'avons montré dans les paragraphes précédents, on leur soumet une chaîne de symboles et ils répondent si cette chaîne appartient ou

¹ Voir toujours *Introduction to Automata Theory, Languages and Computation* de J.E. HOPCROFT et J.D. ULLMAN pour la démonstration des théorèmes suivants : « If L has a regular grammar, then L is a regular set. » et « If L is a regular set, then L is generated by some left-linear grammar and by some right-linear grammar. » pages 218 et 219.

non au langage accepté par l'automate, sauf dans le cas de la machine de Turing où ce problème constitue un problème semi-décidable (Hopcroft et Ullman, 1979).

- Comme les grammaires formelles, les automates sont basés sur l'opération non commutative de concaténation de symboles. Ils permettent donc de prendre en compte des contraintes positionnelles.
- Les automates, tout comme les grammaires formelles, sont étudiés depuis bien des années et les mécanismes y associés sont bien assimilés. Leurs applications sont nombreuses (Aho, Sethi et Ullman, 1986 ; Johnson, 1975 ; Lesk et Schmidt, 1975 ; Thompson, 1968).

D'autre part, on constate que ces modèles ont également certains inconvénients qui rendent leur application telle quelle pour le moins difficile dans le cadre du développement de systèmes de compréhension de langue naturelle parlée :

- Même si ces modèles, contrairement aux grammaires formelles, sont orientés analyse, le résultat de cette analyse est tout de même assez pauvre. En effet dans un système de dialogue intelligent, on ne peut se contenter de savoir si oui ou non un énoncé du locuteur appartient au langage traité. Encore faut-il être capable d'interpréter cet énoncé et de fournir une réponse appropriée. Les automates ne proposent aucune solution à ce problème.
- Un autre désavantage des automates se situe au niveau de la représentation des connaissances. Ils sont plus liés aux procédures de traitement que ne le sont les grammaires formelles. Un automate se modifie par conséquent plus difficilement qu'un ensemble de règles de production par exemple, ce qui constitue un handicap certain pour une mise au point incrémentale d'un système (Laurière, 1982 ; Pinson, 1981).

Par conséquent, les automates n'offrent pas non plus de solution idéale qui se suffirait à elle-même. Nous avons donc été contraint d'étendre notre étude à d'autres modèles linguistiques susceptibles de nous apporter des éléments de réponse.

1.2 Les modèles sémantiques

C'est vers la fin des années soixante que naît un courant en linguistique appelé « sémantique générative ». Ce mouvement, qui se caractérise par une accentuation du rôle de la sémantique dans la grammaire, est à considérer comme une réaction à la grammaire générative chomskyenne. L'approche « sémantique générative » repose sur l'idée que les structures sous-jacentes ultimes des phrases sont des représentations sémantiques définies en termes logiques. Des transformations servent alors à donner une structure syntaxique de surface à ces représentations (Bronckart, 1977). Nous nous sommes plus particulièrement intéressés à la grammaire de cas proposée par Fillmore (Fillmore, 1968)¹ mais nous tenons à signaler qu'il existe

¹ Les idées de C.J. FILLMORE ont assez bien évolué au cours des années. Au début, fortement influencée par la syntaxe transformationnelle, l'approche de FILLMORE est

d'autres représentants de ce même courant d'idées (Lakoff, 1971 ; McCawley, 1971)¹.

1.2.1 Les grammaires de cas

En linguistique traditionnelle, les recherches consacrées aux cas étaient essentiellement orientées vers l'étude des langues sur base des systèmes de cas grec ou latin considérés comme cadre de réflexion universel, vers l'étude de marqueurs morphologiques dans les déclinaisons nominales en tant que manifestation des cas ainsi que vers des études à caractère plutôt historique (Fillmore, 1968).

Fillmore, lui, redéfinit la notion de cas et considère le « cas » comme une relation entre des éléments à l'intérieur d'une phrase. La « forme du cas » est alors l'expression d'une telle relation dans un langage particulier (Fillmore, 1968)². Il considère qu'une phrase simple est composée essentiellement d'un verbe et d'un ou de plusieurs groupes nominaux qui entretiennent des relations casuelles avec le verbe. Par conséquent, il remet en question les catégories traditionnelles « sujet » et « objet » dont il affirme qu'il ne s'agit que de réalisations de surface de fonctions plus profondes (Fillmore, 1972)³. Chacune des relations casuelles est unique à l'intérieur de la phrase bien qu'il puisse y avoir des groupes nominaux composés par con-

devenue de plus en plus « cognitive » (Fillmore, 1977). Notre étude se limite cependant à la grammaire de cas telle qu'elle fut conçue vers la fin des années soixante et le début des années soixante-dix.

¹ Signalons également que dans *Subjects, Speakers, and Roles*, C.J. FILLMORE se réclame de la « sémantique interprétative » plutôt que de la « sémantique générative ».

² Dans *The Case for Case*, C.J. FILLMORE s'exprime ainsi page 21 : « It seems to me that if there are recognizable intrasentence relationships of the types discussed in studies of case systems (whether they are reflected in case affixes or not), that if these same relationships can be shown to be comparable across languages, and that there is some predictive or explanatory use to which assumptions concerning the universality of these relations can be put, then surely there can be no meaningful objection to using the word *case*, in a clearly understood deep-structure sense, to identify these relationships. (...) I shall adopt the usage first proposed, as far as I can tell, by Blake (1930), of using the term *case* to identify the underlying syntactic-semantic relationship, and the term *case form* to mean the expression of a case relationship in a particular language-whether through affixation, suppletion, use of clitic particles, or constraints on word order. ».

³ Dans *Subjects, Speakers, and Roles*, C.J. FILLMORE écrit pages 1, 5 et 6 : « 2. The traditional first task of sentence analysis has been that of understanding and recognizing the highest-level division in a sentence, that between its *subject* and its *predicate*. On the traditional account, the subject of a sentence is that portion of it which indicates 'the person or thing about whom or which a statement is made or a question asked', and its predicate is that portion of the sentence which contains 'the statement or the question asked'. (...) 16. The second grammatical function of NPs which is defined configurationally within the standard theory is that of the direct object. On the traditional account, the role of the direct object in a sentence is that of 'patient' of the action referred to by the verb of the sentence, though deviations from this have long been understood and classified. By its configurational definition, the object NP is identified as that NP which is an immediate constituent of the main VP of the sentence. (...) 18. It would seem however, that linguistic theory ought to provide some way of distinguishing (i) the semantic roles which NPs have with respect to their predicate words, from (ii) facts about their positions in syntactic configurations, on either deep or surface structure levels. (...) ».

jonction. On ne peut cependant joindre des groupes nominaux que s'ils réalisent le même cas.

Sur base de ces considérations, Fillmore définit une grammaire dont les règles sont les suivantes :

- $S \rightarrow M + P$

une phrase (S) se compose de « modalités » (M) et d'une « proposition » (P) ;

- les modalités sont des indications sur la phrase entière du type temps, négation, mode et aspect ;

- $P \rightarrow V + C_1 + C_2 \dots C_n$

une proposition (P) est composée d'un verbe (V) et d'un ou de plusieurs cas (C_1, C_2, \dots, C_n), un cas au moins devant être présent et aucun cas ne devant apparaître plus d'une fois ;

- $C_i \rightarrow NP \mid S$

un cas est réalisé par un groupe nominal (NP) ou une phrase imbriquée (S).

Plusieurs remarques s'imposent :

- Il ne faut pas confondre la notation de Fillmore avec la notation chomskyenne. En effet, ses règles ne contiennent pas d'indication sur l'ordre de réécriture des symboles.
- Les cas, pour Fillmore, sont des concepts universaux dont il pense qu'ils sont probablement innés. Il en fournit une liste : les cas *Agentive*, *Instrumental*, *Dative*, *Factitive*, *Locative* et *Objective* mais précise lui-même que cette liste est susceptible d'être augmentée.
- Le seul cas qui puisse se réécrire sous forme de phrase imbriquée est le cas *Objective*.
- Les problèmes d'insertion lexicale, au niveau du groupe nominal et du verbe, sont traités par des règles supplémentaires. Ces règles spécifient par exemple quels sont les traits sémantiques qu'un cas particulier requiert d'un nom (dans le lexique, il est par conséquent nécessaire de fournir pour tous les noms la liste des traits sémantiques qu'ils possèdent) et quels sont les schémas de cas¹ qu'on est susceptible de rencontrer (dans le lexique, il est par conséquent nécessaire de fournir pour tous les verbes la liste des schémas de cas qui leur sont associés).
- Afin de produire la réalisation de surface de la structure profonde, Fillmore pense qu'il faut également un ensemble de règles qui, en fonction des modalités, de la proposition et des catégories de surface sélectionnées pour réaliser les différents cas déterminent la forme et l'ordre des mots dans la phrase (voir par exemple la figure 5.4).

¹ C.J. FILLMORE parle de « case frame ».

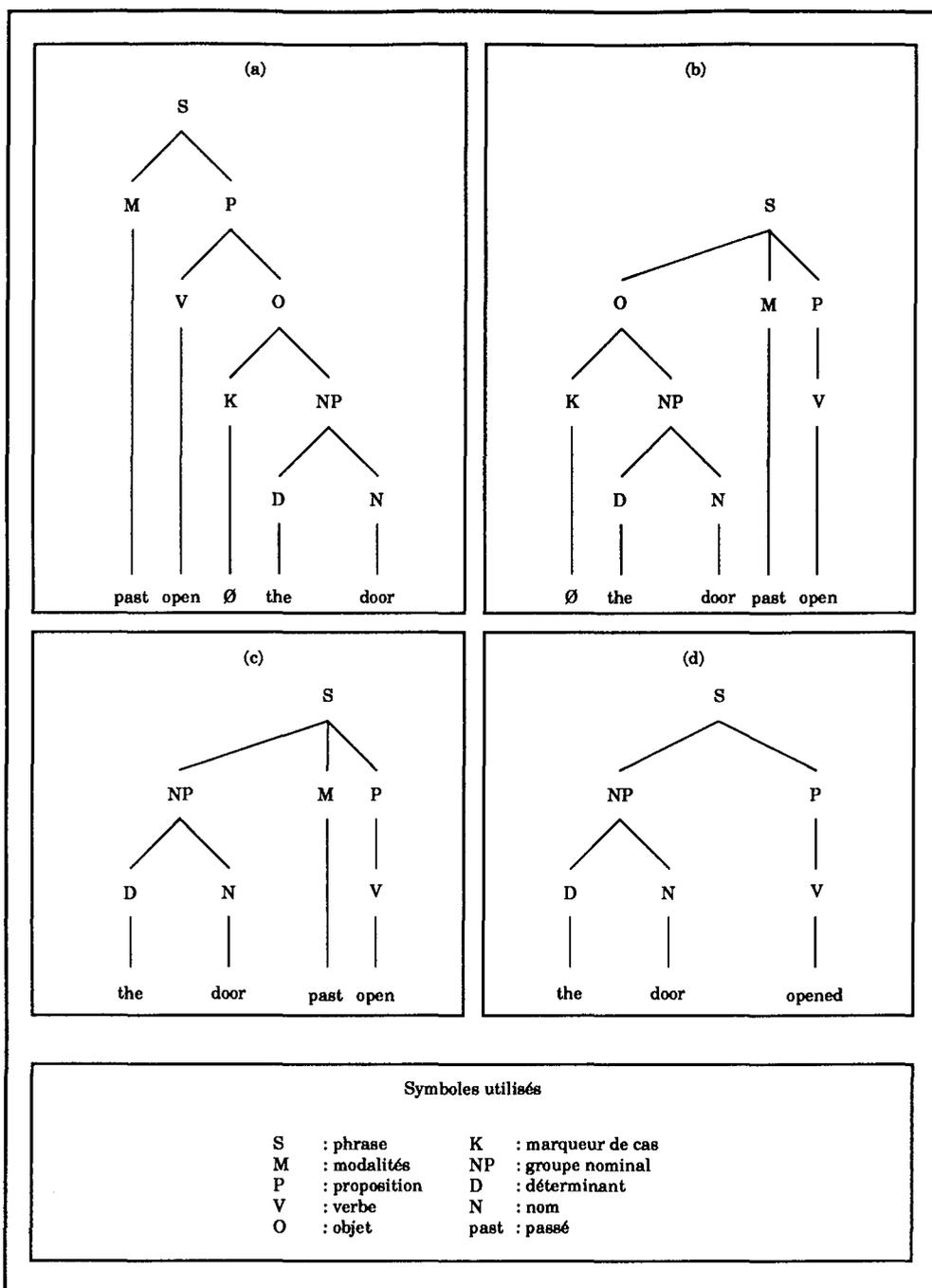


fig. 5.4 : Génération de la structure de surface (d) de la phrase « the door opened » par transformations successives (b, c) à partir de sa structure profonde (a), d'après (Fillmore, 1968)

Nous n'allons pas détailler d'avantage ce modèle, ceci sortirait du cadre de notre travail. Discutons plutôt les aspects qui, pour les informaticiens que nous sommes, semblent intéressants :

- De par sa nature, une grammaire de cas permet d'exprimer des contraintes autres que des contraintes de surface. Elle permet notamment de limiter les combinaisons possibles de mots (essentiellement des noms et des verbes) pouvant apparaître dans une même phrase, in-

dépendamment de leurs positions et en fonction de critères sémantiques (Bruce, 1975).

- Une analyse basée sur une grammaire de cas fournit des structures prédicatives (le verbe étant le prédicat, les groupes nominaux étant les arguments du prédicat) qui paraissent bien adaptées à une interprétation ultérieure (Bruce, 1973, Roussanaly, 1988).

Malheureusement, on pouvait s'en douter, les grammaires de cas présentent également certains points faibles :

- Le noyau du système conçu par Fillmore ne fournit aucune information positionnelle sur les mots. Fillmore parle bien de règles transformationnelles qui déterminent la forme et l'ordre des mots dans la phrase, mais ne semble considérer ce problème que comme un sujet annexe. Si dans un système informatique basé sur le langage naturel écrit (au clavier par exemple) ce fait n'est pas très gênant, dans un système basé sur la parole, une telle approche est vouée à l'échec de par le trop important indéterminisme lié à la langue parlée.
- Enfin, divers reproches sur le plan théorique ont été adressés à Fillmore. Son modèle est loin d'être aussi complet que celui de Chomsky par exemple (Bronckart, 1977), son système de cas n'est pas intégré dans d'autres niveaux de description linguistique et ses cas manquent de critères de test objectifs (Deville, 1989).

La « sémantique générative » a inspiré bon nombre de chercheurs dans le domaine de l'intelligence artificielle. Nous renvoyons le lecteur à (Alterman, 1985 ; Fargues et al., 1986 ; Nash-Webber, 1975 ; Rumelhart, 1973 ; Simmons, 1973 ; Simmons et Slocum, 1972 ; Sowa et Way, 1986) pour plus de détails sur des réalisations informatiques basées sur ces théories.

1.2.2 Les grammaires fonctionnelles

Depuis la fin des années soixante-dix on peut observer le développement d'un courant linguistique qui s'inspire entre autres des idées de Fillmore mais tente de les intégrer dans d'autres niveaux de description linguistique, la linguistique « fonctionnelle » (Dik, 1978 ; Dik, 1980 ; Dik, 1983 ; Dirven et Fried, 1987 ; Foley et Van Valin, 1984). L'approche fonctionnelle considère le langage comme un instrument dont se servent les êtres humains pour atteindre leurs objectifs (Dik, 1980)¹ et requiert d'une théorie linguistique un certain nombre de propriétés (Dik, 1980 ; Dik, 1983b) :

- la capacité de fournir des explications fonctionnelles² ;

¹ Dans *Studies in Functional Grammar* page 2, S. DIK s'exprime de la façon suivante : « Functional Grammar is based on a functional view of the nature of natural language. A view, that is, in which a natural language is first and foremost regarded as an instrument of social interaction by means of which human beings can communicate with each other and thus influence each other's mental and practical activities. ».

² On lira au sujet de ce premier point S. DIK, *Basic Principles of Functional Grammar*, page 5 : « (...) From the functional point of view one wishes, wherever this is possible, to understand why languages are organized the way they are, in the light of the uses

- l'adéquation pragmatique¹ ;
- l'adéquation psychologique².

Contrairement aux grammaires transformationnelles, où les notions les plus importantes sont les constituants, les catégories, l'ordre linéaire et les transformations, les notions fonctionnelles et relationnelles étant considérées comme secondaires, les grammaires fonctionnelles procèdent de la façon inverse et mettent les aspects fonctionnels au premier plan. Une telle grammaire est par conséquent définie en termes de fonctions dont on distingue essentiellement trois types (Dik, 1980 ; Dik, 1983b) :

- les fonctions sémantiques qui définissent les rôles que des participants jouent dans certaines situations (ce concept correspond à la notion de cas chez Fillmore) ;
- les fonctions syntaxiques qui définissent différentes perspectives selon lesquelles des situations peuvent être présentées dans des expressions linguistiques ;
- les fonctions pragmatiques qui définissent le statut informationnel de constituants d'expressions linguistiques selon leur arrangement.

La génération d'une phrase³ se fait essentiellement en trois étapes comme on peut s'en rendre compte sur la figure 5.5 (Dik, 1980 ; Dik, 1983b) :

- il y a tout d'abord sélection d'un schéma de prédicat⁴ ainsi que de termes et insertion de ces termes dans le schéma de prédicat afin de produire une formule⁵ ;
- ensuite, la formule est transformée en formule entièrement spécifiée⁶ par affectation de fonctions syntaxiques et pragmatiques ;

to which they are put: it is the functional, i.e. purpose-related properties of a language which tell us most about its essential nature. (...)».

¹ Toujours dans *Basic Principles of Functional Grammar*, page 6 : « (...) It does not restrict the notion of grammar to isolated sentences, but seeks to clarify how sentences can be integrated into coherent texts, and how linguistic expressions relate to non-linguistic settings. (...)».

² Et enfin page 6 du même *Basic Principles of Functional Grammar* : « (...) Psychological models naturally divide up into models of how speakers go about in constructing and formulating linguistic expressions (production models), and models of how interpreters go about processing and interpreting given linguistic expressions (comprehension models). (...) ...why should a grammar be neutral as between producing and comprehending linguistic expressions, when producing and comprehending such expressions is just what the grammar is there for? (...)».

³ Remarquons que la grammaire proposée par S. DIK est elle aussi orientée vers la production de phrases. DIK en est d'ailleurs conscient, il écrit page 7 de *Basic Principles of Functional Grammar* : « (...) It is written in the productive 'mode', in terms of recipes for constructing linguistic expressions from their basic building blocks, the predicate-frames. This productive mode should be supplemented with an analytic mode, consisting of principles and strategies suitable for arriving at an analysis and interpretation of given linguistic expressions. ».

⁴ DIK parle de « predicate-frame ».

⁵ DIK les appelle des « predications ».

⁶ DIK parle de « fully specified predications ».

- finalement, des règles d'expression sont appliquées à la formule entièrement spécifiée pour fournir comme résultat une expression linguistique.

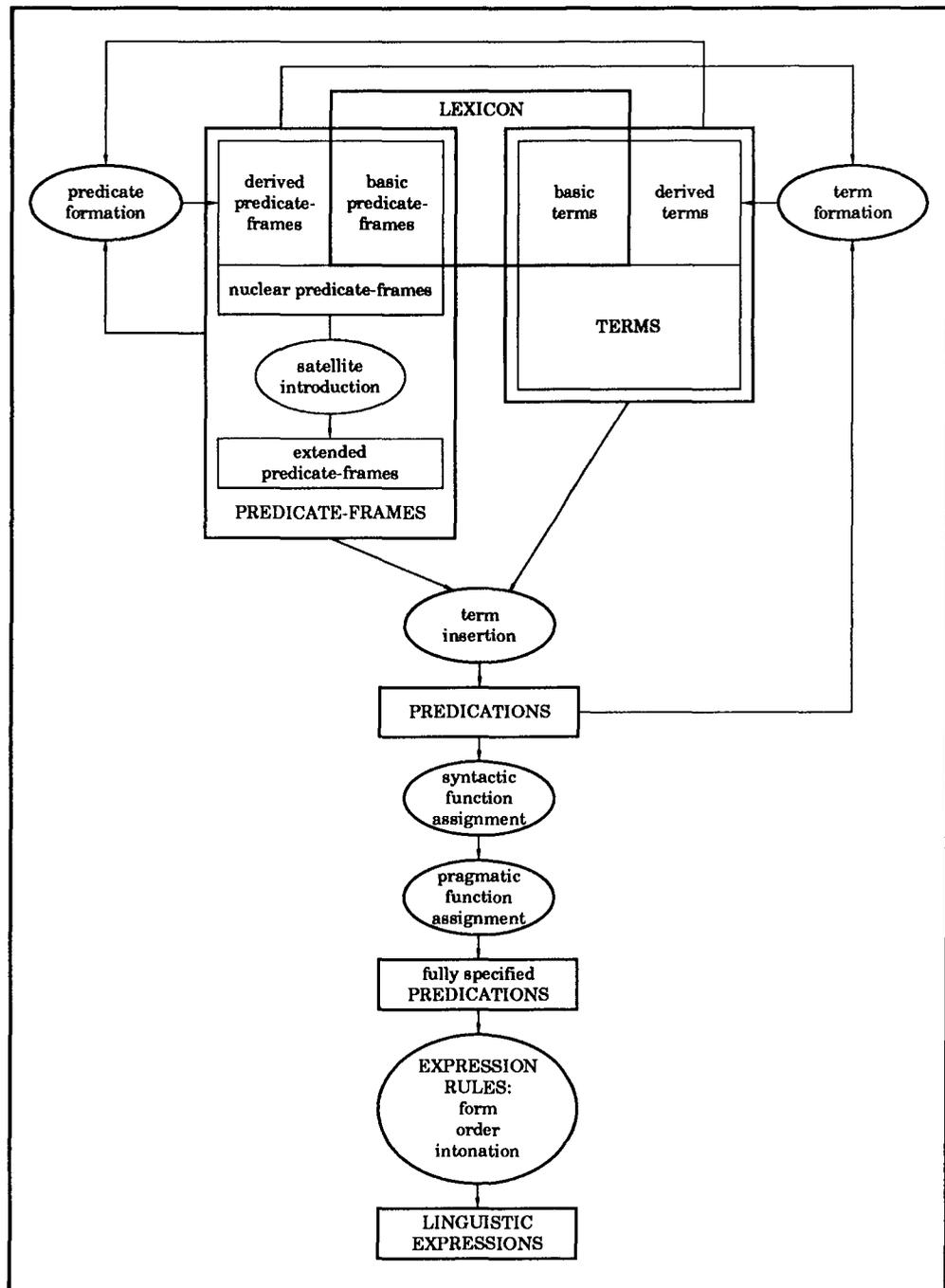


fig. 5.5 : L'organisation d'une grammaire fonctionnelle d'après (Dik, 1983)

Précisons maintenant les différents éléments intervenant dans la génération d'une phrase dans les grammaires fonctionnelles (Dik, 1980 ; Dik, 1983b) :

- les schémas de prédicat désignent des ensembles « d'états de choses »¹ et possèdent la structure suivante :

$$\text{préd}_{\text{cat}} (\text{arg}_1 : \text{restr}_1 (\text{arg}_1))_{\text{fctsém}_1} \dots (\text{arg}_n : \text{restr}_n (\text{arg}_n))_{\text{fctsém}_n}$$

qui spécifie :

- préd : le prédicat concerné ;
- cat : la catégorie du prédicat (verbe, nom, adjectif, ...) ;
- $\text{arg}_1, \dots, \text{arg}_n$: les positions de terme associés à ce prédicat ;
- $\text{fctsém}_1, \dots, \text{fctsém}_n$: les fonctions sémantiques associées à ces positions (agent, bénéficiaire, ...) ² ;
- $\text{restr}_1, \dots, \text{restr}_n$: des restrictions de sélection (humain, animé, ...) imposées par ces positions aux termes, le double point signifie : « tel que » ;

les schémas de prédicat sont classés suivant les traits sémantiques de leurs prédicats (\pm contrôlé et \pm dynamique) en actions (+contrôlé, +dynamique), processus (-contrôlé, +dynamique), positions (+contrôlé, -dynamique) ou états (-contrôlé, -dynamique) ;

- des schémas de prédicat dérivés peuvent être construits à partir de schémas de prédicat de base³, intégrés au lexique, grâce à des règles de formation de prédicats ;
- les deux catégories, schémas de prédicat de base ou dérivés, forment l'ensemble des schémas de prédicat nucléaires⁴ ;
- les schémas de prédicat nucléaires peuvent être étendus par des positions de terme satellites⁵, qui sont liées au schéma de prédicat nucléaire complet plutôt qu'au seul prédicat, pour former des schémas de prédicat étendus ;
- les termes sont des expressions qui font référence à des objets d'univers quelconques ; certains, appelés termes de base, se trouvent tel quel dans le lexique ;
- la plupart du temps cependant, les termes sont des termes dérivés par des opérations de formation de termes⁶, leur structure est la suivante :

$$(\omega x : \varphi_1(x) : \dots : \varphi_n(x))$$

et contient :

¹ DIK parle de « state of affairs ».

² S. DIK laisse le débat ouvert quant au nombre et la nature des fonctions sémantiques. Dans *Basic Principles of Functional Grammar*, page 13, il écrit : « It is an open question which and how many different semantic functions will be necessary and sufficient for a typologically adequate theory of FG for natural languages. ».

³ DIK appelle ce mécanisme « predicate formation »

⁴ DIK parle de « nuclear predicate-frames »

⁵ DIK appelle ce mécanisme « satellite introduction »

⁶ DIK appelle ce mécanisme « term formation »

- ω : un ou plusieurs opérateurs de terme¹ (défini ou indéfini, nombre, ...);
- x : un terme ;
- $\varphi_1(x), \dots, \varphi_n(x)$: des formules ouvertes en x^2 , c'est-à-dire des schémas de prédicat où toutes les positions, sauf celle de x , ont été remplies avec des termes et dont le but est de restreindre l'ensemble des référents potentiels, le double point signifie « tel que » ;

comme les restrictions peuvent elles-mêmes contenir des termes, l'opération de formation de termes est une opération récursive ;

- les formules sont des schémas de prédicat où les positions associées aux prédicats ont été remplies avec des termes ;
- les formules entièrement spécifiées sont des formules où les positions associées aux prédicats se voient affectées, en plus de leurs fonctions sémantiques, des fonctions syntaxiques (sujet, objet, ...) et pragmatiques (thème, centre, ...);
- enfin, les expressions linguistiques sont construites à partir des formules entièrement spécifiées de la façon suivante :
 - la forme des divers éléments de la formule est déterminée ;
 - l'ordre dans lequel ces éléments seront produits est fixé ;
 - les propriétés prosodiques de l'expression linguistique sont définies.

Ce modèle de la langue nous semble extrêmement intéressant pour plusieurs raisons :

- Comme les grammaires de cas, les grammaires fonctionnelles permettent elles aussi d'exprimer des contraintes autres que des contraintes de surface.
- Les structures profondes des énoncés sont des structures prédictives et paraissent bien adaptées à une interprétation ultérieure (Roussanaly, 1988).
- Contrairement à Fillmore, Dik a tenté d'intégrer dans son modèle plusieurs niveaux de description linguistique, à savoir des aspects sémantiques, syntaxiques et pragmatiques.
- Dik propose une classification des schémas de prédicat qui se base sur des critères de test objectifs et linguistiquement pertinents. Nous ne détaillerons cependant pas ce point, le lecteur trouvera plus de précisions à ce sujet dans (Deville, 1989).

¹ Dik parle de « term operators »

² Dik utilise la formulation « open predication in x » et parle de « restrictor »

On pourrait croire qu'il s'agit là d'un modèle linguistique parfaitement adapté aux besoins de concepteurs de systèmes de reconnaissance de la parole. Tel n'est malheureusement pas le cas, en effet :

- Le modèle de Dik, il le constate lui-même, est orienté vers la production d'énoncés. Or notre problème est avant tout un problème d'analyse et d'interprétation.
- La classification des schémas de prédicat que propose Dik, bien que linguistiquement pertinente, nous semble encore trop générale pour être applicable dans un système automatique. Nous pensons que cette classification peut encore être raffinée (Deville, 1989). Dik n'y voit d'ailleurs aucune offense¹.

Enfin, signalons que les grammaires fonctionnelles, tout comme les grammaires de cas, ont servi de fondement théorique à certaines réalisations informatiques (Dik, 1986a ; Dik, 1986b ; Dik, 1987 ; Weigand, 1986 ; Weigand, 1987).

1.3 Les modèles cognitifs

Une dernière famille de modèles qui a retenu notre attention sont les modèles cognitifs, en particulier les « dépendances conceptuelles » de Schank (Schank, 1972). Schank essaie de modéliser le processus de compréhension du langage naturel par l'être humain et se distancie clairement de la sémantique générative (Schank, 1973)². Il s'éloigne des problèmes de linguistique et déplace le centre de gravité de ses préoccupations vers les problèmes de représentation des connaissances (Coulon et Kayser, 1986). Son objectif est, à partir d'énoncés, de construire une structure qui représente leur signification de façon à ce que :

- tous les énoncés qui signifient la même chose, indépendamment de la langue dans laquelle ils sont exprimés, doivent être représentés par une même structure³ ;

¹ Dans *Basic Principles of Functional Grammar*, page 13, S. DIK écrit : « (...) This typology of states of affairs is taken to be universally relevant, but not complete: further subdivisions are required in terms of more refined properties of states of affairs. ».

² A la page 202 de *Identification of Conceptualizations Underlying Natural Language*, R.C. SCHANK se situe par rapport à C.J. FILLMORE en particulier : « The notion of syntactic case is not used at all in conceptual dependency. Traditional linguists' ideas about case are only about so-called case languages. English is not traditionally a case language so the syntactic processor for English has no need of case. Although Fillmore's (1968) remarks on the underlying syntactic cases in English are well taken, his ideas are within the framework of generative grammar and therefore do not pertain here. That is, we are not interested in what is, and what is not, an acceptable utterance in English. Syntactic rules that explain what syntactic cases can be omitted in a syntactic construction shed little light on how the understanding of natural language can best be effected. ».

³ Dans *Identification of Conceptualizations Underlying Natural Language*, page 291, R.C. SCHANK écrit sur ce point : « Thus, the conceptual base is responsible for formally representing the concepts underlying an utterance without respect to the language in which that utterance was encoded. (...) ».

- toute l'information même implicite véhiculée par un énoncé doit apparaître explicitement dans la représentation¹.

La notion de base dans une structure conceptuelle est la notion de « conceptualisation »². Il s'agit d'un ensemble de « concepts » qui entretiennent des « relations ». Les concepts sont de trois types différents : les « noms », les « actions » et les « modificateurs »³. Les relations entre concepts s'appellent « dépendances »⁴. Une syntaxe conceptuelle, c'est-à-dire un ensemble de règles conceptuelles⁵, définit alors, de façon formelle, les dépendances autorisées et la forme qu'elles peuvent revêtir. Les dépendances peuvent être de types très divers, citons en quelques-unes : les dépendances à double sens, les dépendances attributives, les dépendances prépositionnelles⁶ etc.. Certaines dépendances font intervenir la notion de cas : les dépendances d'objet, de bénéficiaire, de direction et les dépendances instrumentales⁷. La syntaxe conceptuelle permet, une fois des dépendances entre certains concepts établies, de prédire l'existence d'autres concepts et d'autres dépendances.

Les concepts de type nom sont notés PP (Picture Producer), les concepts de type action sont notés ACT et les concepts de type modificateur sont notés PA (Picture Aider) ou AA (Action Aider) suivant qu'ils agissent sur les noms (adjectifs) ou les actions (adverbes).

Les concepts de type action sont regroupés en classes appelées « primitives » dont certaines sont : MTRANS (transfert mental ou transfert d'informations), ATRANS (transfert de possession), INGEST (ingestion), CONC (conceptualisation), PTRANS (transfert physique), MBUILD (construction mentale), MOVE (mouvement), PROPEL (propulsion), GRASP (prise) et EXPEL (expulsion) etc..

On peut ensuite relier les conceptualisations par des « relations conceptuelles »⁸ dont les plus importantes sont : la « causalité », le « temps » et

¹ Toujours dans *Identification of Conceptualizations Underlying Natural Language*, page 296, R.C. SCHANK signale : « However, in attempting to uncover the actual conceptualization underlying a sentence, we must recognize that a sentence is often more than its component parts. In fact, a dialogue is usually based on the information that is left out of a sentence but is predicted by the conceptual rules. ».

² R.C. SCHANK parle de « conceptualization ».

³ Appelés respectivement « nominals », « actions », « modifiers » par R.C. SCHANK.

⁴ R.C. SCHANK les appelle « dependencies ».

⁵ Que R.C. SCHANK appelle « conceptual rules ».

⁶ Appelées respectivement « two-way dependencies », « attributive dependencies » et « prepositional dependencies » par R.C. SCHANK.

⁷ R.C. SCHANK distingue les cas « objective », « recipient », « directive » et « instrumental ». Il précise également qu'il convient de ne pas confondre ces cas « conceptuels » avec d'autres cas, dans *Identification of Conceptualizations Underlying Natural Language*, page 298, il écrit : « Linguists who are interested in syntactic well-formedness have begun recently to look into the possibility of using cases in their representations (e.g. Fillmore, 1968). In addition, computational linguists (see Simmons chapter in this volume) have made interesting use of these so-called "semantactic" cases. The cases used by these researchers, however, are only distantly related to what we are referring to here as conceptual cases. Conceptual cases are considered to be part of the ACT upon which they depend and as such are always present whether words expressing them have appeared in a given sentence. (...) ».

⁸ R.C. SCHANK parle de « conceptual relations ».

le « lieu »¹. De plus des modificateurs² peuvent agir sur ces conceptualisations. Schank les note de la façon suivante : p (passé), f (futur), t (transition), t_S (transition qui commence), t_F (transition terminée), k (continuation), ? (interrogative), / (négative), nil (présent) et finalement Δ (atemporel).

Enfin, une sémantique conceptuelle impose alors des limitations sur les dépendances autorisées par la syntaxe. Ces limitations sont exprimées en termes de traits sémantiques (animé, nourriture, ...) associés aux concepts pouvant intervenir dans une conceptualisation.

Un aspect particulier de ce modèle nous a semblé intéressant :

- Le fait de classer les actions en primitives permet de réduire le nombre de règles conceptuelles qui décrivent, pour chaque primitive, les conceptualisations dans lesquelles elles peuvent intervenir.

Cependant :

- Les primitives manquent de critères de classification objectifs, l'approche conceptuelle est une approche fortement « ad-hoc » et est sujette à caution, tant d'un point de vue linguistique que psychologique (Deville, 1989).
- De plus, les préoccupations de Schank sont encore plus éloignées de tous les problèmes de représentation en surface que les modèles sémantiques, d'où un manque de liens explicites entre réalisations de surface et représentations conceptuelles (Deville, 1989). De nouveau l'indéterminisme inhérent à la parole fait que cette approche est difficilement envisageable dans le cadre d'une application de compréhension de la langue parlée.
- Enfin, le réseau conceptuel est censé représenter l'entièreté de l'information véhiculée par un énoncé, y compris les informations implicites. On obtient alors, même à pour des phrases simples, des représentations conceptuelles assez compliquées (voir par exemple la figure 5.6).

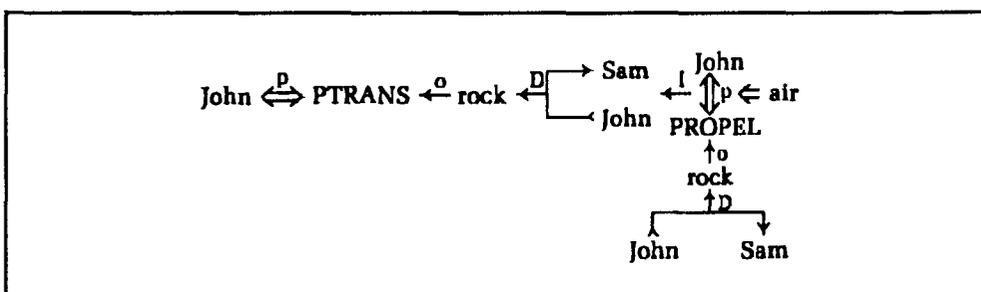


fig. 5.6 : Exemple de réseau conceptuel représentant la phrase
« John threw a rock at Sam »
extrait de (Schank, 1973)

¹ Appelées respectivement « causality », « time » et « location » par R.C. SCHANK.

² R.C. SCHANK parle de « conceptual tenses ».

A part les travaux réalisés par l'équipe autour de Schank (Schank, 1975), certaines notions du modèle des dépendances conceptuelles ont été reprises par d'autres systèmes (Alterman R., 1985 ; Hendrix et al., 1973 ; Nash-Webber, 1975 ; Rumelhart, 1973 ; Sowa et Way, 1986).

1.4 Des modèles linguistiques adaptés au traitement de la parole

Comme on a pu le constater dans les paragraphes précédents, aucun des modèles linguistiques présentés ne satisfait entièrement nos exigences vis-à-vis d'un modèle linguistique. En effet, aucun n'est immédiatement applicable à un problème de compréhension du langage parlé. Nous avons par conséquent ressenti le besoin de proposer des modèles adaptés au traitement de la parole. Il va sans dire qu'ils s'inspirent fortement de ceux exposés dans ce chapitre, plus particulièrement des modèles syntaxiques et des grammaires fonctionnelles. Ils seront présentés dans le huitième chapitre. Signalons toutefois ici que notre objectif est de proposer des modèles doublement validés :

- d'un point de vue linguistique ; nous voulons proposer des modèles qui ne reflètent pas une approche « ad-hoc » et fournissent des solutions au coup par coup suivant l'application à laquelle on s'intéresse ;
- d'un point de vue informatique ; nous voulons proposer des modèles « opérationnels », c'est-à-dire des modèles qui puissent être réellement mis en œuvre dans le cadre d'une application informatique de reconnaissance de la parole.

2 La notion de sous-langage, justification méthodologique de notre approche

2.1 Les sous-langages

Depuis un certain temps il est reconnu par les chercheurs travaillant dans le domaine de la compréhension du langage naturel ou de la parole, que le traitement du langage naturel dans toute sa généralité est extrêmement difficile voire impossible. Ceci est dû au fait qu'un ordinateur nécessite, pour pouvoir effectuer un traitement automatique, une description détaillée et explicite de toutes les situations envisageables. Or, l'état actuel des connaissances sur les phénomènes langagiers ne permet pas d'envisager la modélisation du langage naturel dans ces conditions. C'est la raison pour laquelle plusieurs auteurs se sont intéressés au langage naturel dans des contextes

précis et limités. Des résultats de ces recherches se dégage la notion de sous-langage (Deville, 1989 ; Falzon, 1984 ; Harris, 1968 ; Kittredge, 1982).

Un sous-langage diffère du langage général tant des points de vue lexical, syntaxique, sémantique que pragmatique. La plupart du temps, il ne s'agit pas simplement d'un langage sous-ensemble du langage général mais d'un langage qui peut être décrit selon trois modes par rapport au langage général (Deville, 1989) :

- un mode restrictif, le langage général peut posséder des caractéristiques absentes dans le sous-langage ;
- un mode déviant, le sous-langage peut intégrer des éléments qu'on ne retrouve pas dans le langage général ;
- enfin, un mode préférentiel, complémentaire aux deux modes précédents ; il sert essentiellement à décrire les restrictions et déviations d'un sous-langage par rapport au langage général en termes de distributions probabilistes plutôt qu'en termes binaires de présence ou d'absence (voir figure 5.7).

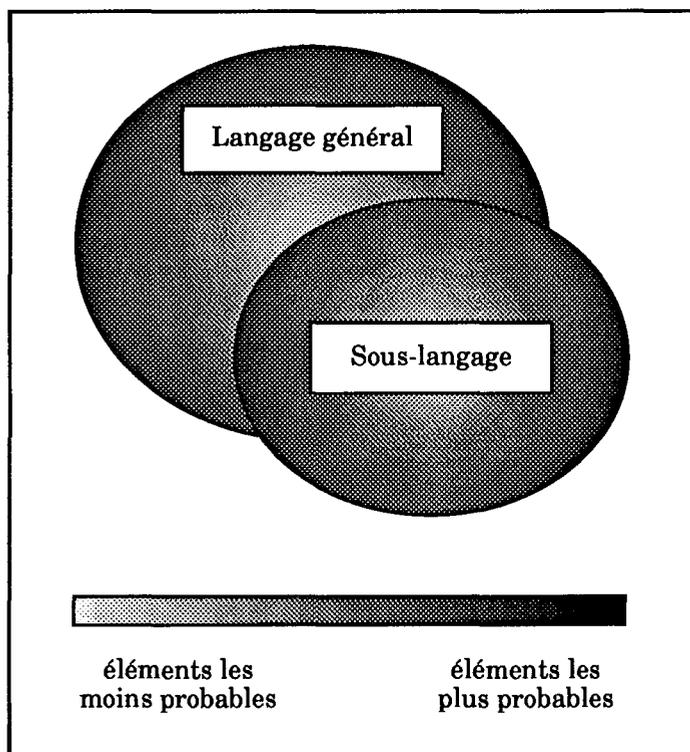


fig. 5.7 : Un sous-langage par rapport au langage général

2.2 Particularités des sous-langages

La description d'un sous-langage selon ces trois modes porte sur tous les niveaux de description linguistique, à savoir les aspects lexicaux, syntaxiques, sémantiques et pragmatiques (Deville, 1989) :

- Les particularités lexicales des sous-langages :

Un sous-langage est caractérisé par la taille fortement réduite de son lexique (Deville, 1989 ; Kittredge, 1982 ; Roussanaly et al., 1986). Contrairement au langage général, le vocabulaire d'un sous-langage apparaît comme virtuellement fini.

Mais le lexique d'un sous-langage est également déviant par rapport au langage général dans la mesure où il peut contenir des mots qui ne sont jamais ou rarement utilisés dans le langage général. Par conséquent, il ne s'agit nullement d'un échantillon représentatif du langage général.

Malheureusement, dans le cas général, on ne peut affirmer avec certitude qu'un mot appartient ou non au lexique d'un sous-langage ou du langage général. Par contre, ce qu'on peut constater, ce sont des fréquences d'apparition de certains mots élevées dans le langage général mais faibles dans le sous-langage et l'inverse pour d'autres mots.

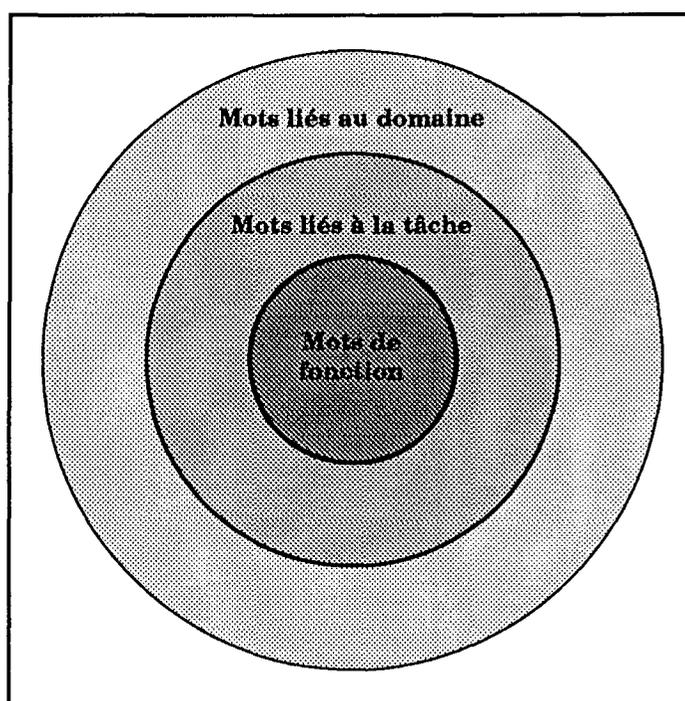


fig. 5.8 : Structure du lexique d'un sous-langage d'après (Deville, 1989)

Ceci est confirmé par des études qui ont montré que peu de sous-langages ont un lexique fermé¹. La propriété de fermeture est à considérer comme relative plutôt que comme binaire. Ce constat a d'ailleurs amené certains linguistes à introduire la notion de « niveau de con-

¹ La propriété de fermeture du lexique est définie de la façon suivante : « If we have a finite set of texts belonging to a certain sublanguage, we can determine its vocabulary and inventory of grammatical constructions. Let us add to this new set new texts of the same sublanguage. If the vocabulary and the inventory of grammatical constructions remains unchanged, the sublanguage is called a closed one - in the opposite case, it is called an open one ». Cette définition de Moskovich est citée dans *Modelization of Task-Oriented Utterances in a Man-Machine Dialogue System* de G. DEVILLE.

fiance » pour qualifier le lexique d'un sous-langage. Le niveau de confiance détermine la probabilité de trouver, dans un nouveau texte du sous-langage, un mot n'appartenant pas au lexique. Deville (Deville, 1989) propose même une décomposition du lexique d'un sous-langage en différentes strates, chacune répondant à la propriété de fermeture jusqu'à un certain degré (voir figure 5.8).

- Les particularités syntaxiques des sous-langages :

Au niveau syntaxique, on observe des comportements similaires. D'une part, un sous-langage est engendré par un ensemble de règles plus restreint que le langage général, d'autre part certaines constructions grammaticales acceptées dans un sous-langage peuvent être agrammaticales dans le langage général.

Enfin, on constate que certaines structures syntaxiques, peu courantes mais néanmoins acceptées dans le langage général, sont utilisées assez fréquemment dans le sous-langage, tandis que d'autres, fréquentes dans le langage général, sont d'un emploi plus rare dans le sous-langage.

- Les particularités sémantiques des sous-langages :

Dans un sous-langage, non seulement le lexique est-il de taille limitée, comparé au vocabulaire du langage général, mais le sens des mots qu'il contient est lui aussi fortement restreint. Si dans le langage général, on observe bon nombre de cas de polysémie, dans les sous-langages par contre, les mots sont le plus souvent monosémiques. Cette caractéristique n'est pas inhérente aux mots eux-mêmes, elle découle plutôt du contexte dans lequel ils sont utilisés.

D'autres mots du sous-langage qui font partie de ce qu'on appelle couramment le « jargon » sont parfois inexistant dans le langage général. Ceci confirme le fait que le lexique d'un sous-langage ne peut être considéré comme un échantillon représentatif du langage général.

Finalement, pour certains mots qui, dans un sous-langage, restent polysémiques, on peut observer un glissement du sens. L'emploi du mot dans le sous-langage modifie la hiérarchie des significations potentielles du mot par rapport à son emploi dans le langage général.

- Les particularités pragmatiques des sous-langages :

Un sous-langage vu sous l'angle pragmatique peut être considéré comme un processus dynamique de transfert de connaissances, plus ou moins complexes dans un domaine, par des moyens linguistiques et dans le but de réaliser une tâche. Or, nous nous sommes essentiellement intéressés aux dialogues finalisés (cf. chapitre 4), c'est-à-dire aux dialogues dont le but est justement la réalisation d'une tâche. On peut observer que ce type de dialogues a une structure proche de celle de la tâche à réaliser. De ce fait, les énoncés dans un tel dialogue ne font pas seulement référence à un domaine limité mais en plus, leur interprétation y est fortement simplifiée. En effet on constate que dans ce type de communication l'interprétation peut se baser sur des modèles de locuteurs fortement restreints et stéréotypés.

Enfin, l'emploi d'un sous-langage est encore caractérisé par des variations lexicales et syntaxiques résultant de l'adaptation au locuteur et de son niveau d'expertise.

Ces considérations amènent G. Deville à définir un sous-langage de la façon suivante :

Définition

Nous appelons sous-langage un ensemble d'énoncés faisant référence à un domaine sémantique limité et bien défini et utilisé pour une fonction spécifique. De tels énoncés sont engendrés par une grammaire spécifique et un vocabulaire spécifique.

2.3 De l'intérêt des sous-langages

La notion de sous-langage est d'une importance capitale pour l'approche que nous avons choisie. En effet, d'une part nous sommes obligés de simplifier le problème de la compréhension de la parole, d'autre part nous voulons imposer le minimum de limitations « à priori » au système que nous voulons construire. La notion de sous-langage nous fournit les moyens de réaliser cet objectif. Si nous réussissons à délimiter un domaine sémantique restreint de façon à ce que des dialogues finalisés dans ce domaine permettent de définir un sous-langage ayant les propriétés pré-citées, les limitations découleront naturellement de l'emploi de ce sous-langage : lexique restreint, syntaxe et sémantique simplifiées ainsi que modèles de la tâche, de l'utilisateur et du dialogue réduits. Les limitations d'un système automatique basé sur cette approche seront une conséquence des situations de communication dans lesquelles le système pourra se trouver. L'utilisation d'un tel système ne nécessitera plus d'apprentissage, il sera accessible au grand public.

QUELQUES REALISATIONS INFORMATIQUES

Après avoir présenté nos sources d'inspiration dans le domaine de la linguistique, nous allons exposer dans ce chapitre quelques systèmes informatiques qui ont eu une influence déterminante sur nos choix au niveau informatique.

1 Un échantillon de systèmes de compréhension de parole

Le domaine de la compréhension automatique de la parole connaît aujourd'hui une telle expansion qu'il serait vain de tenter de faire un inventaire exhaustif de tous les systèmes développés jusqu'à ce jour ou actuellement en cours de développement. C'est la raison pour laquelle nous avons opéré une sélection, sélection qui n'est quand-même pas due au hasard. Nous nous sommes laissés guider par certaines particularités des systèmes qui vont nous servir à justifier nos choix par la suite. Les aspects qui vont nous intéresser plus précisément dans ce chapitre seront les problèmes liés aux architectures des différents systèmes de compréhension de parole. Nous avons donc choisi de présenter KEAL (développé au CNET à Lannion) et MYRTILLE I (développé au CRIN à Nancy) afin d'illustrer des architectures de systèmes hiérarchiques, HWIM (développé par BBN aux Etats Unis) et MYRTILLE II (développé au CRIN à Nancy) pour des architectures de systèmes mixtes et HEARSAY II (développé à la CMU aux Etats Unis) pour une architecture de système non hiérarchique.

2 Le système KEAL

Le système KEAL est un système de reconnaissance de parole continue développé au CNET. Une première version de ce système a été implantée en

1979 (Mercier et al., 1978 ; Quinton, 1980). L'objectif était alors la reconnaissance de phrases. Une deuxième version, présentée en 1981 (Quinton, 1982 ; Siroux et Gillet, 1985 ; Siroux et Mercier, 1984), était augmentée d'un module de gestion de dialogues dont le but était de permettre une communication homme-machine plus complète et naturelle (Nouhen-Bellec et Siroux, 1981). La figure 6.1 montre l'organisation générale de ce système.

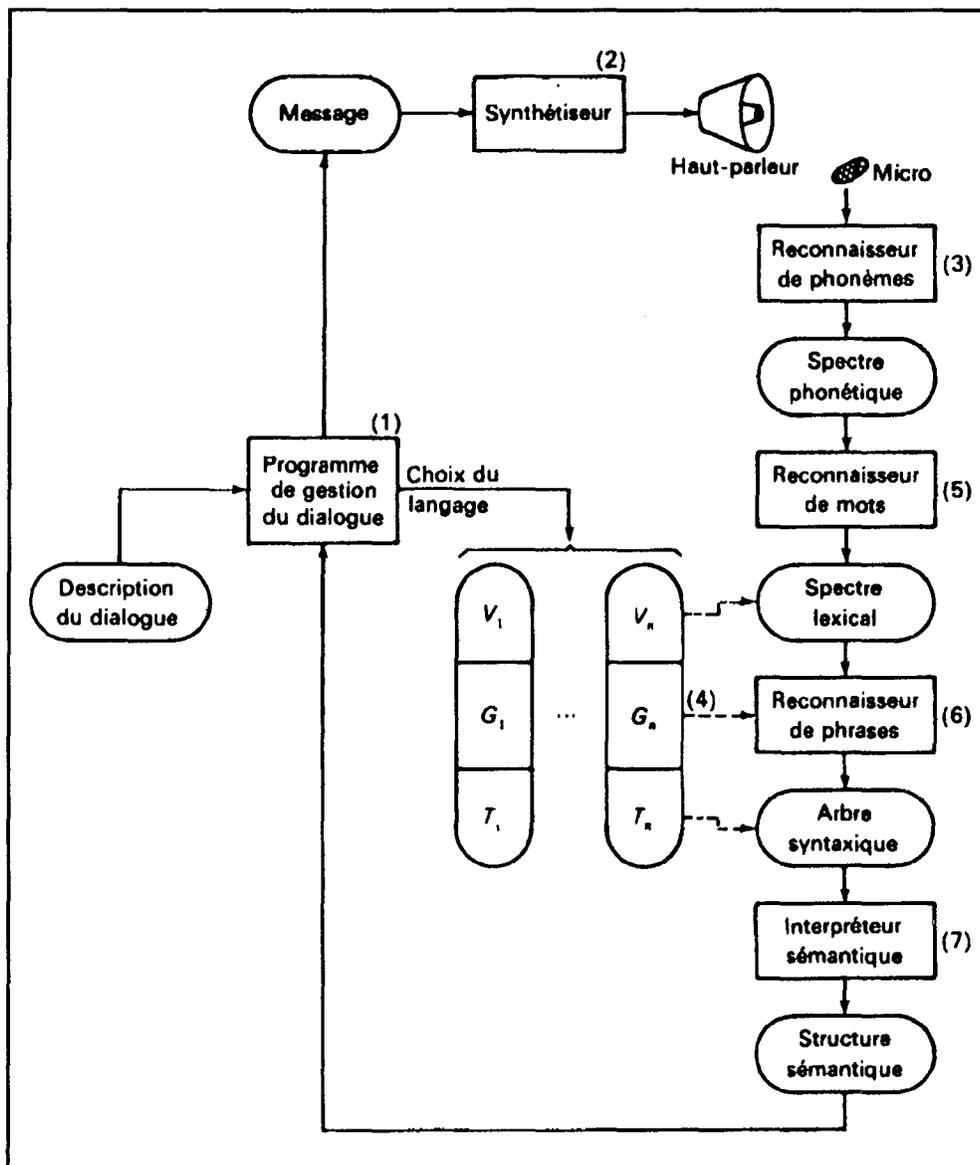


fig. 6.1 : Organisation générale du système KEAL
extrait de (Quinton, 1982)

Le fonctionnement de KEAL peut se résumer de la façon suivante (Quinton, 1982 ; Siroux, 1985) :

- L'analyse acoustique est effectuée par un vocoder et sert à l'extraction des paramètres pertinents pour la reconnaissance phonétique, elle fournit notamment des informations sur la distribution de l'énergie dans les différentes bandes de fréquence ainsi que des informations sur le voisement.

- L'analyse phonétique a deux objectifs : d'une part la segmentation du signal en syllabes, d'autre part l'identification des phonèmes. Elle fournit un « spectre phonétique », c'est-à-dire une séquence de segments dont chacun est constitué d'un score, indiquant sa probabilité d'existence, et d'une collection de phonèmes possibles, ordonnés dans l'ordre décroissant de leur vraisemblance.
- Sur base du spectre phonétique, l'analyse lexicale formule des hypothèses sur les mots. Pour cela elle utilise un dictionnaire contenant la transcription phonétique standard de mots sélectionnés par le module de gestion du dialogue. Toutes ces transcriptions sont comparées au spectre phonétique dans le but de détecter le mot correspondant dans la phrase. Le résultat en est un « spectre lexical » qui consiste en l'ensemble de tous les mots détectés auxquels est associée une « valuation » de la détection.
- Le reconnaiseur de phrases a pour paramètre une grammaire hors-contexte, sélectionnée elle-aussi par le module de gestion du dialogue. Cette grammaire définit l'ensemble des phrases que le système attend de l'utilisateur compte tenu du contexte du dialogue. L'analyseur doit reconstruire la phrase sur base du spectre lexical et fournir l'arbre syntaxique de la phrase.
- L'arbre syntaxique est ensuite traité par un interpréteur sémantique qui a pour paramètres des règles de transformations sélectionnées par le module de gestion du dialogue et associées aux productions de la grammaire qui a permis la reconnaissance de la phrase. Ces règles permettent de générer la structure sémantique de la phrase sous la forme d'une liste de relations appelées « n-uplets ».
- Le module de gestion du dialogue confronte les n-uplets à un graphe représentant le « savoir-faire » du système. Chaque fois qu'un n-uplet peut être apparié à un nœud du graphe, des actions associées au nœud sont déclenchées. Celles-ci provoquent une réponse à l'utilisateur à laquelle sont associés un vocabulaire, une grammaire et un ensemble de transformations déterminant le contexte du cycle suivant du dialogue.

Le système KEAL possède une architecture hiérarchique de type ascendant. Si une telle organisation convient parfaitement à des langages à vocabulaire de taille limitée, il n'en va pas de même pour des vocabulaires plus importants. Le nombre d'hypothèses lexicales devient rapidement très élevé, par manque d'interaction entre les modules de reconnaissance de mots, de phrases et d'interprétation sémantique, et pénalise ainsi le temps de traitement.

3 Le système MYRTILLE I

Le système MYRTILLE I est un système de compréhension automatique du discours continu, développé au CRIN vers le début des années soixante-dix (Pierrel, 1975 ; Pierrel, 1978a ; Pierrel, 1981 ; Pierrel, 1987). L'objectif était de construire un système général de reconnaissance du discours continu capable de traiter tous les langages définis en paramètre par une grammaire hors-contexte. Ici encore, l'effort portait essentiellement sur

la reconnaissance de phrases. Bien que MYRTILLE I intégrait un module de dialogue, Pierrel pensait que cette procédure de dialogue posait, dans le cas général, de nombreux problèmes. C'est la raison pour laquelle il considéra la procédure de dialogue comme un traitement propre à chaque application qu'il fallait ré-écrire entièrement lors de chaque nouvelle application. La figure 6.2 montre l'organisation générale de ce système.

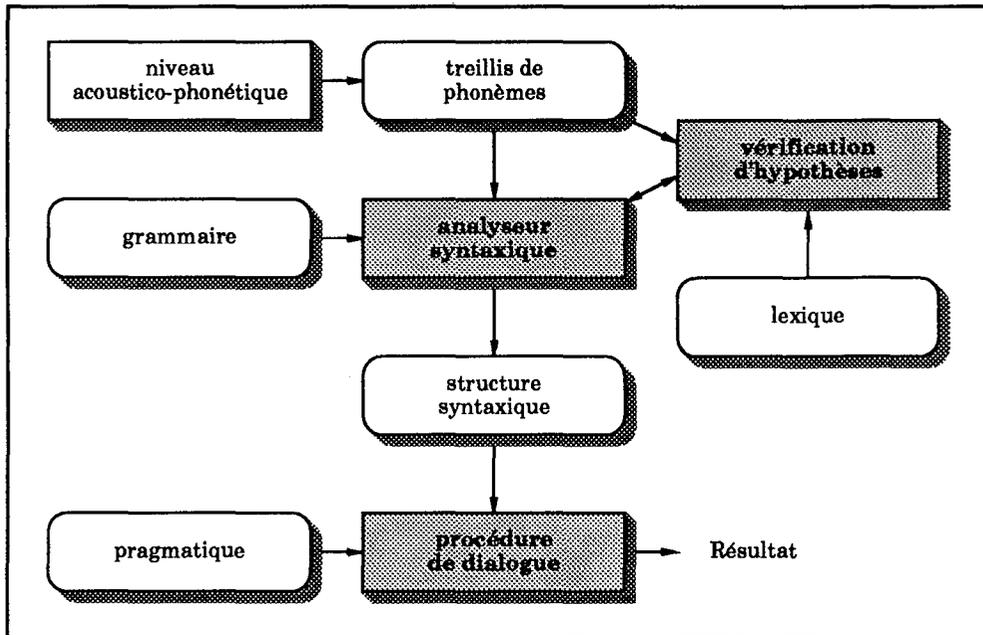


fig. 6.2 : Organisation générale du système MYRTILLE I d'après (Pierrel, 1987)

Le fonctionnement de MYRTILLE I peut se résumer de la façon suivante (Pierrel, 1975 ; Pierrel, 1978a) :

- Le module acoustico-phonétique travaille en trois étapes. Il procède tout d'abord à la segmentation du signal dont le but est de découper le signal vocal en segments minimaux. Sur base du signal segmenté, une chaîne phonémique est produite lors de la phase d'identification des segments. Celle-ci s'appuie sur des méthodes de comparaison dynamique pour résoudre les problèmes de normalisation temporelle. Enfin, la chaîne phonémique est lissée, c'est-à-dire que certaines redondances de la chaîne sont supprimées. Un treillis de phonèmes constitue le résultat du module acoustico-phonétique.
- Le système syntaxico-sémantique reprend comme données les résultats fournis par le niveau acoustico-phonétique. Diverses considérations sur l'importance du niveau syntaxique en compréhension orale ont amené Pierrel à envisager l'analyse syntaxique comme guide du processus de reconnaissance. Une conséquence de ce choix a été de ne considérer la reconnaissance proprement dite que comme un moyen de vérifier les hypothèses émises par le niveau syntaxico-sémantique. De plus, ce vérificateur peut se permettre certaines libertés par rapport à la syntaxe du langage, spécialement en cas d'adjonction ou de suppression d'un mot, si du moins ce dernier n'est pas sémantiquement caractéristique. Pour garantir une certaine indépendance vis-à-vis de l'application, la définition du langage est un paramètre du système.

- La procédure de dialogue doit valider, en dernier recours, les résultats fournis par la reconnaissance, lever les ambiguïtés ou corriger certaines erreurs. Rappelons que pour MYRTILLE I, l'approche à ce niveau est fortement ad-hoc. Pierrel suggère en effet de ré-écrire entièrement cette procédure lors du passage d'une application à une autre.

Le système MYRTILLE I possède une architecture hiérarchique de type descendant. Ce type d'approche convient assez bien à des systèmes traitant un langage artificiel à syntaxe très contrainte. Cependant, dans le cas d'applications grand public où il s'agit de traiter le langage naturel, il convient d'attacher une plus grande importance aux traitements sémantiques et pragmatiques.

4 Les systèmes HWIM ET MYRTILLE II

Les approches ascendantes et descendantes ont assez rapidement montré leurs limites. Les systèmes HWIM et MYRTILLE II constituent une tentative de synthèse entre ces deux approches, ils intègrent une approche mixte. Tous les deux possèdent un module superviseur dont le but est le séquençage de l'activation des autres modules. L'organisation générale de ces deux systèmes est présentée sur les figures 6.3 (HWIM) et 6.4 (MYRTILLE II).

Le fonctionnement de HWIM peut être résumé de la façon suivante (Bolt Beranek and Newman, 1976 ; Wolf et Woods, 1980) :

- L'acquisition du signal de parole est réalisée par le module « RTIME » qui numérise le signal de parole. Le module « PSA » effectue une analyse par prédiction linéaire sur le signal numérisé et en extrait divers paramètres tels que : les formants du signal, la fréquence fondamentale, l'énergie du signal, le nombre de passages par zéro etc..
- Un module de reconnaissance acoustico-phonétique « APR » fournit, sur base de la représentation paramétrique du signal, un treillis phonétique après segmentation du signal, étiquetage des segments et calcul de scores pour chacun des segments. Le treillis phonétique est ensuite utilisé par un module de recherche lexicale « LEXICAL RETRIEVAL » qui lui compare les mots du lexique. La composante lexicale peut rechercher tous les mots du lexique ou seulement un sous-ensemble de celui-ci.
- Cette analyse ascendante est complétée par « VERIFICATION », un mécanisme de vérification descendant qui consiste à générer, à partir d'une hypothèse mot et grâce à un programme de synthèse par règles, une représentation spectrale idéalisée. Cette dernière est comparée à la représentation paramétrique de l'énoncé en utilisant des techniques de programmation dynamique pour la normalisation temporelle.
- La composante syntaxique « SYNTAX » a pour fonctions essentielles de porter un jugement sur la grammaticalité d'une séquence donnée de mots, d'émettre des prédictions sur base d'une séquence d'hypothèses

de type mot et de construire une représentation formelle de l'énoncé. Pour cela, le module syntaxique se base sur des réseaux de transition augmentés « pragmatiques » (Woods, 1970).

- Le contrôle du système est assuré par un module pragmatique, « TRIP ». Il s'agit d'un gestionnaire de base de données qui intègre des connaissances factuelles et pragmatiques. Ces connaissances sont représentées dans un réseau sémantique.
- HWIM est capable de traiter des requêtes écrites ou orales. Dans le cas d'une requête entrée au clavier, le module TRIP active directement le module syntaxique qui doit analyser et interpréter la requête. Dans le cas d'une requête orale par contre, TRIP active un superviseur, la composante « CONTROL STRATEGY ». Le rôle de ce superviseur est de diriger le processus de formation et de prédiction d'hypothèses.

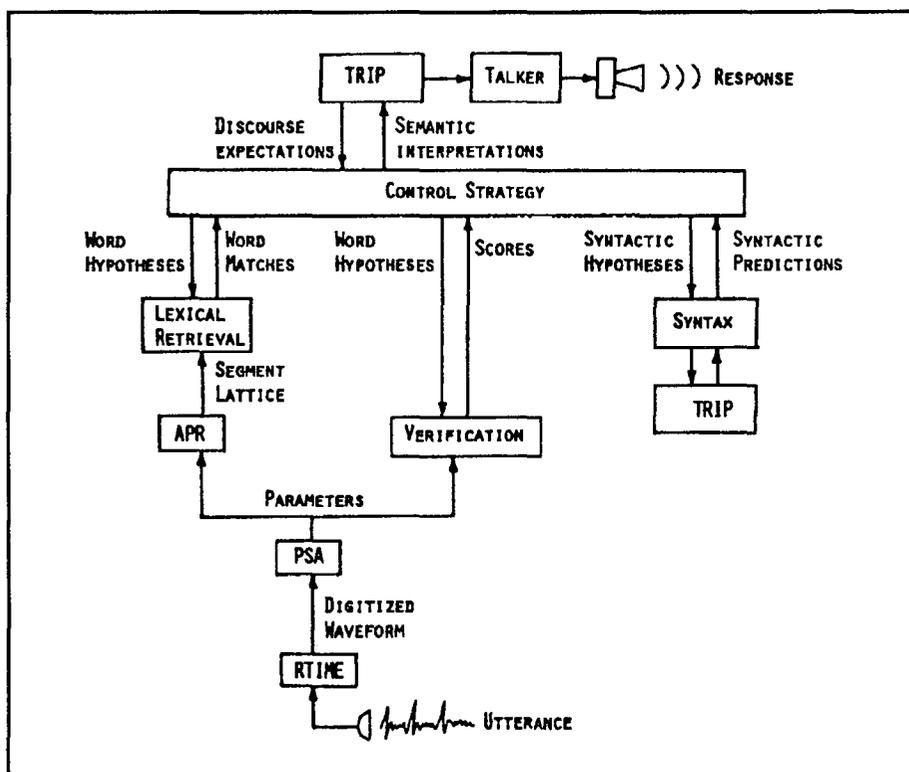


fig. 6.3 : Organisation générale du système HWIM
extrait de (Wolf et Woods, 1980)

Décrivons maintenant le fonctionnement de MYRTILLE II (Haton et al., 1978 ; Pierrel, 1978b ; Pierrel, 1981 ; Pierrel, 1982) :

- Ici encore, un module de « Décodage acoustico-phonétique » est chargé des premiers traitements sur le signal. Son but essentiel est de fournir un treillis de phonèmes qui sera utilisé par la composante de vérification lexicale.
- Le noyau du système est un « SUPERVISEUR » qui est chargé de la gestion des hypothèses, c'est-à-dire de l'ensemble des mots qu'on s'attend à trouver à un endroit déterminé de l'énoncé. Suivant ses besoins, le superviseur active les trois composantes « STRUCTURE »,

« LEXIQUE » ou « PROPHON ». Ceci lui permet de restreindre le nombre des hypothèses mots à émettre vers la vérification lexicale. Afin de restreindre l'ensemble des hypothèses, la composante STRUCTURE met en œuvre la définition syntaxico-sémantique des structures du langage, la composante LEXIQUE la définition syntaxico-sémantique des mots du langage tandis que la composante PROPHON pondère les hypothèses en fonction des traits prosodiques et de la structure phonémique du contexte à reconnaître.

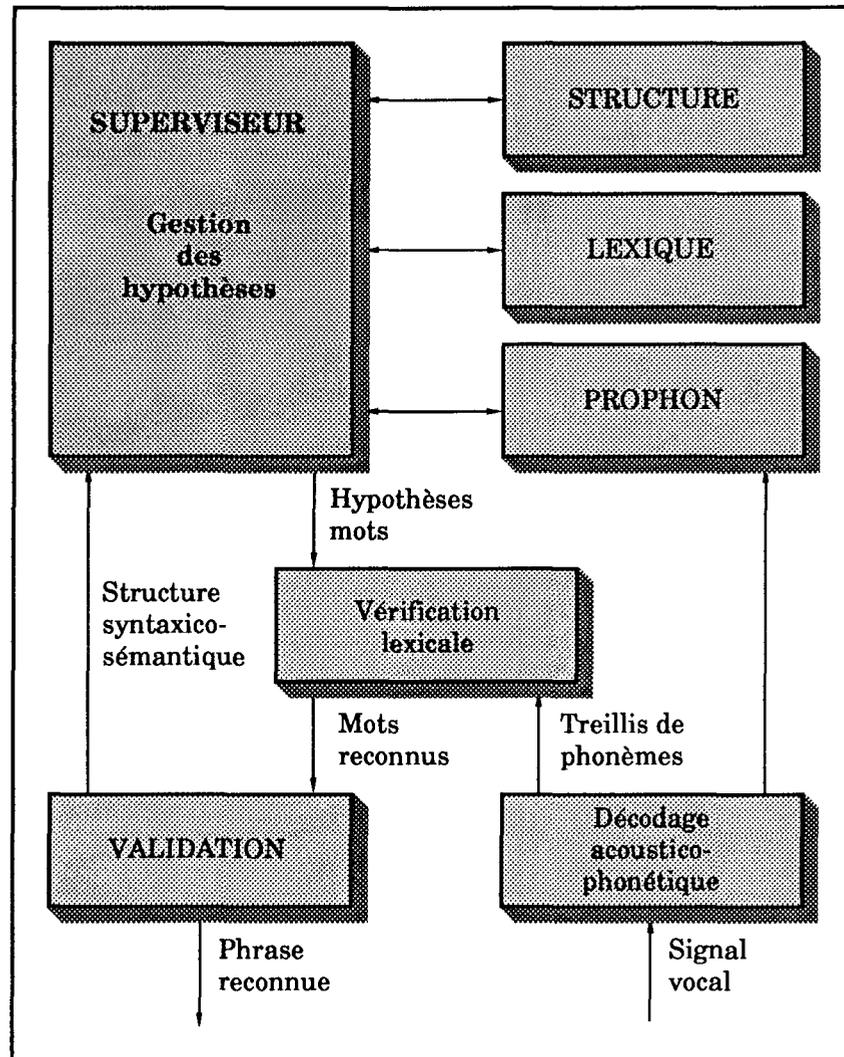


fig. 6.4 : Organisation générale du système MYRTILLE II d'après (Pierrel, 1987)

- La procédure de « Vérification lexicale » procède alors au test des hypothèses par une recherche lexicale des mots. Une procédure de programmation dynamique compare la représentation phonétique de référence du mot à une sous-chaîne du treillis phonétique en entrée.
- Au vu des résultats de la vérification lexicale, un module de « VALIDATION » sélectionne une hypothèse et met à jour la représentation syntaxico-sémantique de l'énoncé.

On constate que HWIM et MYRTILLE II, malgré des différences fondamentales sur lesquelles nous n'allons pas insister ici, présentent cer-

taines similitudes. Les deux sont des systèmes ni purement ascendant ni purement descendant. Ils intègrent tous les deux un mécanisme d'hypothèse-test ce qui en fait des systèmes à architecture mixte. Les deux systèmes s'appuient également sur un module superviseur qui décide, en fonction de l'avancement et des besoins de l'analyse, des processus à activer. S'ils ont tous les deux l'avantage de n'imposer aucune hiérarchie « à priori » aux différentes composantes, ils ont cependant l'inconvénient de séquencer des tâches qui semblent assez indépendantes et par là de leur imposer une hiérarchie « dynamique ».

5 Le système HEARSAY II

Un système qui semble éviter cet inconvénient est le système HEARSAY II, développé à la Carnegie-Mellon University. Dans ce système, toutes les composantes sont considérées comme des modules indépendants. Chacun de ces modules, encore appelés sources de connaissance, communique avec les autres via une structure de données centrale appelée tableau noir ou « blackboard » qui renferme l'ensemble des hypothèses, de n'importe quel niveau du système, sur l'énoncé en cours de traitement. En principe, toutes ces hypothèses sont accessibles à tous les modules. L'activation d'un module est déclenchée par une modification dans le tableau noir. La figure 6.5 montre les différents types d'hypothèses implantés dans HEARSAY II, les sources de connaissance traitant ces hypothèses ainsi que la façon dont ces dernières sont transformées.

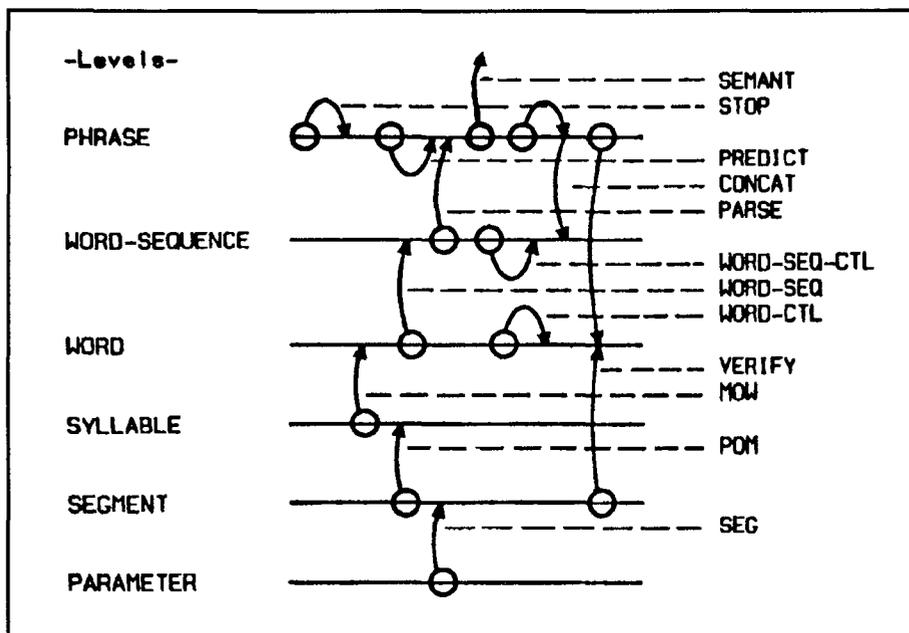


fig. 6.5 : Les sources de connaissances dans HEARSAY II
extrait de (Erman et Lesser, 1980)

Contrairement aux systèmes HWIM et MYRTILLE II, HEARSAY II considère les modules intervenant dans le processus de compréhension comme des modules entièrement indépendants. En fait, cette architecture

définit un modèle valide dans un contexte beaucoup plus général de contrôle de processus parallèles. Elle semble particulièrement bien adaptée au développement incrémental de systèmes. Ce qu'on peut cependant constater d'une part, c'est le fait que l'accès à ce tableau noir doit être géré de même que l'activation des différentes sources de connaissance. Ceci constitue d'une certaine façon un superviseur. D'autre part on constate que les sources de connaissances ainsi que leurs interactions semblent bien maîtrisées aujourd'hui. Il y a peu de chances en effet qu'une composante inconnue jusque maintenant vienne s'ajouter au schéma de la figure 6.5. Cette architecture peut dès lors apparaître comme trop générale et de ce fait peu efficace.

6 Une architecture de système adaptée au dialogue oral homme- machine

Les paragraphes précédents nous permettent de dégager certaines tendances générales. On peut tout d'abord constater que ces systèmes mettent en œuvre des connaissances linguistiques très diverses. Tout le monde semble également plus ou moins d'accord sur la nature des informations linguistiques à prendre en compte, à savoir des informations phonétiques, lexicales, syntaxiques, sémantiques etc.. Cependant aucun des systèmes présentés n'intègre de véritable composante pragmatique. Si elle est présente, elle a été ajoutée lors d'une phase ultérieure du projet, c'est le cas du système KEAL, ou alors il s'agit d'une composante ad-hoc très fruste et rigide dont le passage aisé d'une application vers une autre est loin d'être garanti, c'est le cas de systèmes tels que HWIM ou MYRTILLE I.

De plus, on constate une évolution des architectures fortement hiérarchisées (par exemple celles des systèmes KEAL et MYRTILLE I) vers des architectures non hiérarchiques (par exemple celle du système HEARSAY II) ainsi qu'un renforcement de la coopération entre les différents composants des systèmes. La coopération y est cependant garantie par un contrôle centralisé, soit au niveau des traitements, par un module superviseur, soit au niveau des données, par un tableau noir.

Nous pensons que notre système de dialogue oral homme-machine doit intégrer certaines idées validées par ces réalisations tout en proposant une architecture où :

- la composante pragmatique est intégrée dans le système au même titre que le sont les autres composantes, ceci nous a amenés à redéfinir complètement l'architecture de notre système et non pas à augmenter un système existant ;
- la coopération entre les différentes composantes n'est plus garantie par un contrôle centralisé, mais, afin de tirer pleinement profit de l'architecture non hiérarchique, est réalisée par un contrôle distribué.

L'architecture de notre système sera présentée en détail dans le septième chapitre.

LA MISE EN ŒUVRE DE CONNAISSANCES DANS LE SYSTEME DIAL

Ce chapitre se propose de présenter les raisons qui nous ont amenés à concevoir un nouveau système de dialogue oral homme-machine, le système DIAL. Il situe le développement de ce système dans le cadre des recherches sur la parole au CRIN, en présente les objectifs principaux ainsi que les choix concernant son architecture.

1 Le système DIAL

1.1 La compréhension de phrases dans les systèmes MYRTILLE I et MYRTILLE II

Le développement d'un système de communication orale homme-machine finalisée en langage naturel (le projet DIAL) s'inscrit dans le cadre plus large des recherches sur la parole au CRIN. Il fait suite aux systèmes MYRTILLE I et MYRTILLE II dont les premiers résultats remontent à 1975 et bénéficie bien sûr pleinement de l'expérience acquise lors de la réalisation de ces systèmes (Pierrel, 1975 ; Pierrel, 1981). Le projet DIAL a cependant l'ambition d'aller plus loin.

En effet MYRTILLE I était un système de compréhension de parole continue fortement guidé par la syntaxe. Le langage décrit par la syntaxe utilisée dans MYRTILLE I était assez rigide, Pierrel l'a d'ailleurs classé dans la catégorie des langages artificiels, et son utilisation difficile à envisager dans le cadre d'applications grand public où ce genre de contraintes peut parfois constituer des barrières insurmontables. Néanmoins ce type de

système peut avoir son utilité et Pierrel envisageait des applications dans les domaines du contrôle de processus ou de qualité, de la commande de processus ou de machines-outils, de la bureautique etc.¹.

Partant du constat des limites de MYRTILLE I, fut alors développé MYRTILLE II. L'objectif était d'élargir le champ d'application des systèmes de compréhension de parole continue aux applications grand public. Pour cela il fallait traiter un sous-ensemble plus étendu du langage naturel. C'est ce qui était réalisé dans MYRTILLE II, système capable de traiter un langage pseudo-naturel c'est-à-dire un langage défini par une syntaxe décrivant un sous-ensemble assez vaste du français de base et un lexique spécifique au domaine précis de l'application.

1.2 La compréhension de dialogues dans le système DIAL

Tout comme MYRTILLE I, le système MYRTILLE II était limité à la compréhension d'énoncés isolés. Aucun des deux systèmes n'aborda sérieusement les problèmes de dialogue homme-machine. Or, pour permettre une utilisation effective du langage naturel et de la parole dans la communication homme-machine, comprendre ou reconnaître une phrase ne suffit pas : il faut expliciter le résultat de cette compréhension dans le cadre de l'application mise en œuvre, en d'autres termes, comprendre et gérer l'ensemble du dialogue utilisateur-machine (GRECO, 1985 ; Pierrel, 1987 ; Pierrel, 1988 ; Young et al., 1989). Notre équipe travaille dans ce sens depuis maintenant plusieurs années. Nous avons choisi d'aborder deux familles très différentes d'applications, donc deux types d'interfaces orales nettement différentes :

- d'une part, des applications classiques où l'ordinateur intervient comme exécutant et qui mettent en œuvre un univers pragmatique restreint, la commande de processus : système PARTNER (Morin, 1987) ;
- d'autre part, des applications concernant des activités cognitives plus complexes dans un univers plus riche, le renseignement administratif grand public : système DIAL.

De tels systèmes et plus particulièrement ceux du second type, se caractérisent par l'utilisation de connaissances multiples et variées et par l'importance des informations sémantiques et pragmatiques dans le processus de compréhension. De plus, il serait illusoire de penser que la construction de ces systèmes de dialogue correspond à la juxtaposition d'un modèle de compréhension de phrases et d'un module de gestion de dialogue. Les interactions entre ces deux niveaux sont nettement plus complexes ; les deux processus doivent être intégrés au maximum pour conduire à un véritable système de compréhension et de gestion de dialogues. Cela conduit à mettre en œuvre une coopération forte entre trois niveaux trop souvent dissociés

¹ Signalons que MYRTILLE I a donné naissance à deux projets actuellement en cours de développement au CRIN, les systèmes DIAPASON et PARTNER, qui sont des systèmes de dialogue oral homme-machine mettant en œuvre un univers pragmatique restreint.

dans les architectures hiérarchiques de compréhension de parole : la syntaxe, la sémantique et la pragmatique.

1.3 Les objectifs du système DIAL

Notre objectif, dans le cadre du projet DIAL, est de construire un système informatique susceptible de constituer une interface orale homme-machine efficace et confortable pour l'utilisateur, dans le cadre d'applications informatiques destinées au grand public (Carbonell, 1984). Nous nous proposons de traiter les communications où un utilisateur naïf s'adresse à une machine dans l'intention de réaliser une tâche. Nous partons de l'hypothèse que le sujet adopte un comportement coopératif (Roussanaly, 1988) pour lequel des critères qualitatifs pourraient être les suivants :

- ne donner ni trop, ni trop peu d'informations ;
- être sincère et parler à bon escient ;
- être pertinent ;
- être facilement compréhensible.

Exception faite de ces principes de coopération, nous voulons dispenser le locuteur de toute autre forme de conditionnement préalable. Cela signifie que le locuteur n'effectue pas d'apprentissage et ne reçoit pas non plus de consignes¹. Nous envisageons donc un système qui, à partir :

- d'un dialogue oral finalisé qu'il devra, en partie au moins, gérer et structurer (nous voulons éviter que le dialogue se présente sous forme de menus successifs où tout le contrôle serait attribué à la machine ; pour plus de naturel, il nous semble indispensable que le locuteur puisse également prendre l'initiative lorsqu'il le juge opportun),
- d'une expertise dans un domaine (tâche et univers de connaissances) relativement simple et circonscrit,

soit capable de comprendre et de satisfaire des demandes en provenance d'un nombre élevé d'utilisateurs s'exprimant librement en français. Par conséquent, sur le plan syntaxique et lexical, le système devra accepter un sous-ensemble du français parlé, suffisant pour formuler n'importe quelle requête dans le cadre de l'application. Sur le plan sémantique et pragmatique, l'interprétation d'un énoncé devra se faire en fonction du contexte créé conjointement par la tâche à accomplir et le dialogue en cours. Le locuteur aura donc la possibilité de produire des énoncés anaphoriques et elliptiques.

¹ Contrairement à un système d'interrogation de bases de données classiques, nous voulons nous approcher le plus possible de l'idéal décrit par J. KAMMERSGAARD dans *Four Different Perspectives on Human-Computer Interaction* : « (...) Seen from a dialogue partner perspective the ideal is that a user can 'communicate' with this application in a way similar to the way communication between humans take place; that is to say, based on knowledge about the domain but without knowledge about the domain representation within the computer. ».

Par conséquent notre approche se distingue fondamentalement de celle adoptée dans d'autres systèmes (Biermann et al. 1985).

Les applications ainsi visées pourraient être :

- la consultation de bases de données administratives, météorologiques, téléphoniques ou autres (horaires de la SNCF, des compagnies aériennes etc.) ;
- les demandes de rendez-vous (dans un centre de soins médicaux par exemple) ;
- les demandes de réservation (dans les agences de voyage, les hôtels, les théâtres etc.) ;
- les demandes de location (dans les agences immobilières par exemple) ;
- les commandes d'articles dans les catalogues de vente par correspondance.

1.4 Le choix d'une application test

Nous avons retenu, pour tester notre système, la consultation des informations administratives contenues dans les pages roses de l'annuaire téléphonique des P&T. Un exemple illustrant le type d'informations qu'on peut y trouver est fourni sur la figure 7.1.

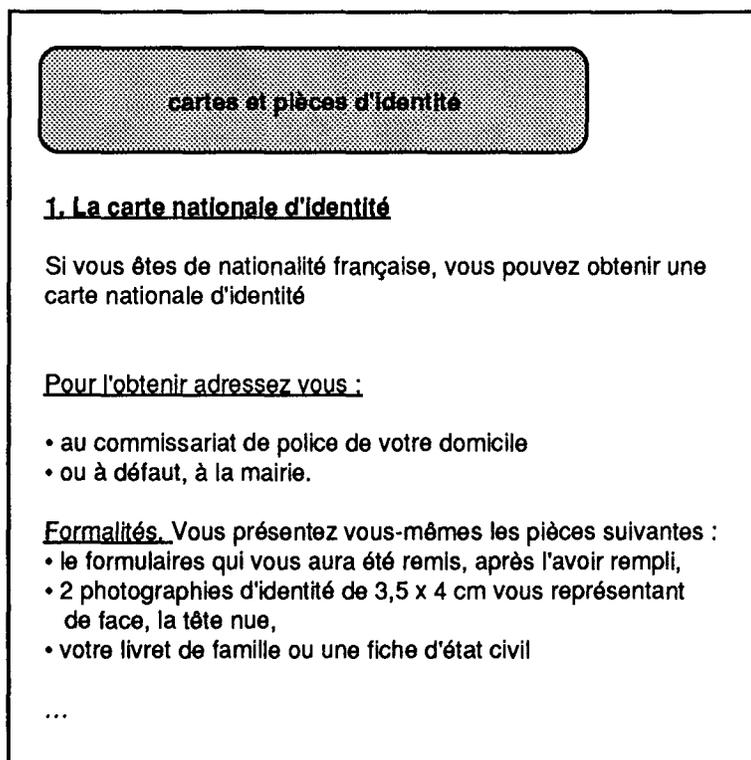


figure 7.1 : Un extrait des pages roses de l'annuaire des P&T

Ce choix est justifié par le fait que cette application engendre des dialogues suffisamment variés pour que l'on puisse tester les différentes capacités attendues du système. En effet :

- la population concernée est le grand public,
- l'application se compose d'une variété de tâches suffisamment simples et relativement bien structurées,
- la taille du lexique, qui avoisine 1000 mots, semble réaliste pour un traitement en temps réel. Cette estimation provient de deux études de corpus effectuées dans le cadre du projet DIAL et dont les conclusions peuvent être trouvées dans (Roussanaly et al., 1986 ; Deville, 1987),
- en se basant sur ce même corpus, nous pouvons raisonnablement nous attendre à un dialogue finalisé avec, en outre, un contrôle partagé du dialogue,
- et enfin, le nombre relativement élevé d'échanges par dialogue laisse présager une grande diversité dans les interventions.

La catégorie de dialogues que nous visons est illustrée par l'échange suivant qui constitue un exemple type de demande de renseignement :

S : *Centre de renseignements administratifs, bonjour.*

U : Bonjour, voilà, je n'ai plus ma carte d'identité. Où peut-on la renouveler?

S : *Est-ce à la suite d'une perte ou d'un vol?*

U : Je l'ai perdue.

S : *Dans ce cas, présentez-vous à la mairie avec une déclaration de perte, deux photos d'identité et votre livret de famille ou une fiche d'état civil.*

U : Comment est-ce-qu'on obtient une déclaration de perte?

S : *Pour une déclaration de perte, vous devez vous adresser à votre mairie.*

U : Merci.

S : *Désirez-vous un autre renseignement?*

U : Non, merci. Au revoir.

S : *Au revoir.*

Il faut cependant noter que ce type d'échange se produit rarement dans la réalité. Nous avons choisi cette situation idéalisée pour la clarté de l'exposé. L'étude de corpus de dialogues en situation réelle montre cependant :

- que la structure des énoncés peut varier fortement en fonction des locuteurs et

- que les reprises et hésitations sont très fréquentes, très difficiles à détecter et à intégrer dans des structures avec les méthodes d'analyse classiques.

Ces spécificités de l'oral exigent la mise en œuvre de processus d'analyse assez souples. Il faut également tenir compte des limites actuelles de la reconnaissance de la parole. Tous ces phénomènes sont source de problèmes divers : erreurs d'interprétation, reconnaissance défectueuse etc.. Notre système doit pouvoir y faire face et être capable de traiter des contestations de la part du locuteur, d'émettre des demandes de reformulation, de précision, de confirmation, de répétition et même, lorsque cela s'avère nécessaire, des consignes en rapport avec la gestion du canal de communication.



fig. 7.2 : « Un système automatique de renseignements administratifs »
extrait de (Veyron, 1982)

2 Le choix d'une architecture

2.1 Les caractéristiques particulières d'un dialogue oral finalisé en langage naturel

Des études ont montré que les dialogues que nous voulons traiter présentent certaines caractéristiques dont il faut absolument tenir compte lors de la conception de systèmes informatiques de communication orale finalisée en langage naturel (Amalberti 1984 ; Angeletti, 1984 ; Bunt, 81 ; Deville, 1986 ; Deville, 1987 ; Morel, 1985). Ces caractéristiques peuvent être résumées de la façon suivante :

- On constate tout d'abord que dans ce type de communication l'initiative est le plus souvent partagée. Il est donc exclu d'envisager un système rigide à menus contrôlant l'échange du début jusqu'à la fin.
- L'emploi de l'oral introduit des problèmes nouveaux par rapport à l'écrit notamment les problèmes d'indéterminisme dus aux limites de la reconnaissance et de l'analyse ainsi que les problèmes liés au canal de communication.
- Dans ce type de communication, le dialogue est finalisé, c'est-à-dire qu'il peut être considéré comme un processus convergeant dont l'objectif est la réalisation d'une tâche.
- Les applications grand public excluent un apprentissage préalable. Ceci nécessite le recours au langage naturel. Cependant l'étude du langage utilisé dans ces applications montre qu'en fait il s'agit de sous-langages qui ont des traits restrictifs et déviants par rapport au langage naturel et ce aussi bien aux niveaux lexical, syntaxique, sémantique que pragmatique.
- Enfin, on observe également une modification du comportement langagier en présence d'un système automatique qui restreint le choix du vocabulaire et la variété de la structure des énoncés mais augmente le nombre des interventions.

De plus, nous pensons qu'il est raisonnable, vu les limites actuelles de nos connaissances sur la nature du phénomène langage, de présupposer que le locuteur soit coopératif (Grice, 1975). Cette hypothèse de travail nous semble sensée. Des résultats de tests confirment en effet que des locuteurs s'adressant à une machine se comportent de façon plus disciplinée que s'ils s'adressent à un humain (Hauptmann et Rudnicky, 1988).

2.2 Une typologie opératoire des connaissances linguistiques

Comme nous l'avons déjà signalé dans les chapitres précédents, nous voulons mettre en œuvre dans notre système de communication toutes les connaissances dont on suppose actuellement qu'elles sont également utilisées par l'être humain dans son activité de communication. La plupart des systèmes de compréhension de parole distinguent les connaissances prosodiques, acoustico-phonétiques, lexicales, syntaxiques, sémantiques et pragmatiques. Cette classification ne tient cependant pas compte de l'aspect dynamique des informations intervenant dans le processus de traitement. Nous voulons modéliser un univers restreint mais facilement adaptable à différentes applications. Il devient alors nécessaire d'étudier plus en détail les facteurs qui interviennent aux différents niveaux de cette modélisation. C'est la raison pour laquelle nous proposons une classification plus opératoire (Carbonell, Charpillat et al., 1985 ; Carbonell, Haton et al., 1985). Nous distinguons notamment les connaissances statiques des connaissances dynamiques. Les connaissances statiques ne varient pas ou peu au cours du temps et ne dépendent que de l'application ou plus généralement de la famille d'applications choisie. Les connaissances dynamiques, elles, varient fréquemment avec le temps et sont liées soit au côté évolutif de la tâche, soit au

dialogue en cours. Ces classes d'informations regroupent chacune divers modèles :

- Les connaissances statiques regroupent :
 - les informations indépendantes de l'application, il s'agit essentiellement des connaissances de bas niveau telles que l'acoustique, la phonétique, la phonologie et la prosodie¹ ;
 - les informations communes à une famille d'applications, en l'occurrence la syntaxe, la sémantique et le modèle du dialogue² ;
 - les informations spécifiques à une application, notamment le lexique et le modèle de la tâche³.
- Les connaissances dynamiques regroupent :
 - les informations liées à l'évolution de la tâche appelées l'univers de la tâche⁴ ;
 - les informations liées au dialogue en cours ; il s'agit de l'historique du dialogue qui s'enrichit au fur et à mesure des échanges⁵ ;
 - les informations liées à l'énoncé en cours de traitement, c'est-à-dire ses représentations acoustiques, phonétiques, prosodiques, lexicales, syntaxiques, sémantiques et pragmatiques⁶.

¹ Dans le *Dialogue oral homme-machine*, J.M. PIERREL les appelle le « modèle de l'utilisateur ». Il peut s'agir d'informations propres à un locuteur (dans le cas de systèmes à usage restreint) ou à une classe de locuteurs (dans le cas d'applications grand public).

² La syntaxe et la sémantique que J.M. PIERREL, toujours dans le *Dialogue oral homme-machine*, regroupe sous le terme de « modèle du langage » sont utilisées pour la reconnaissance, la compréhension et l'interprétation des énoncés du locuteur mais également pour la génération des réponses du système. Le « modèle du dialogue », spécifie les stratégies de gestion de dialogues finalisés en fournissant une description générale des diverses situations de dialogue spécifiques à une famille d'applications.

³ Pour J.M. PIERREL, le lexique fait partie du « modèle du langage » et contient la définition phonétique et syntaxico-sémantique des mots de l'application. Le « modèle de la tâche » renferme une composante sémantico-pragmatique, complément du lexique définissant les objets manipulés et les relations entre ces objets, ainsi qu'une définition en termes de buts ou de sous-buts qui spécifient les chemins d'accès aux données, fonctions ou renseignements de l'application.

⁴ Complément indispensable du modèle de la tâche, l'univers de la tâche décrit, suivant le type de l'application envisagée, l'état d'une base de données, l'état d'un processus, l'état d'un environnement etc. à un moment donné du cycle de vie de cette application.

⁵ L'historique du dialogue que J.M. PIERREL qualifie de « mémoire à long terme » est indispensable pour permettre une interprétation contextuelle des énoncés ainsi que la mise en place d'un véritable dialogue.

⁶ J.M. PIERREL les appelle la « mémoire à court terme ».

2.3 Proposition pour une architecture répartie

Nous avons vu au sixième chapitre que les architectures de système fortement hiérarchisées, qui s'inspirent d'ailleurs de modèles utilisés en linguistique traditionnelle (voir figure 7.3), posent quelques problèmes. Il en va de même des architectures non hiérarchiques à superviseur ou à « blackboard ». C'est la raison pour laquelle nous proposons une architecture non hiérarchique à contrôle distribué. Nous avons choisi une approche de type multi-bases de connaissances (Laasri, 1987), ceci dans un souci de :

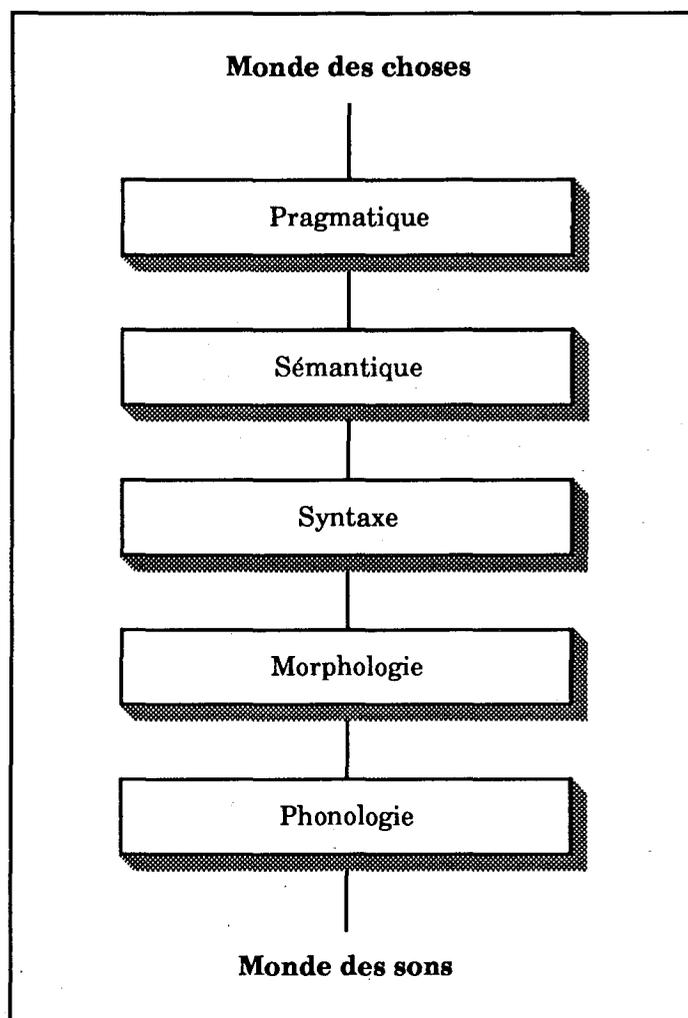


figure 7.3 : Hiérarchie des connaissances en linguistique traditionnelle d'après (Kay, 1985)

- modularité : notre système se compose de plusieurs modules, chacun travaillant sur un type d'informations particulier et utilisant des bases de connaissances spécifiques, ce qui permet de les développer et de les valider de façon indépendante et en parallèle ;

- facilité de mise au point : les différents modules sont chacun un système à base de connaissances ce qui en facilite le développement incrémental¹ ;
- adaptabilité : la séparation très nette entre connaissances d'une part et traitements d'autre part combinée à notre typologie opératoire conduit à un système paramétrique et rend la mise en place d'une nouvelle application plus aisée.

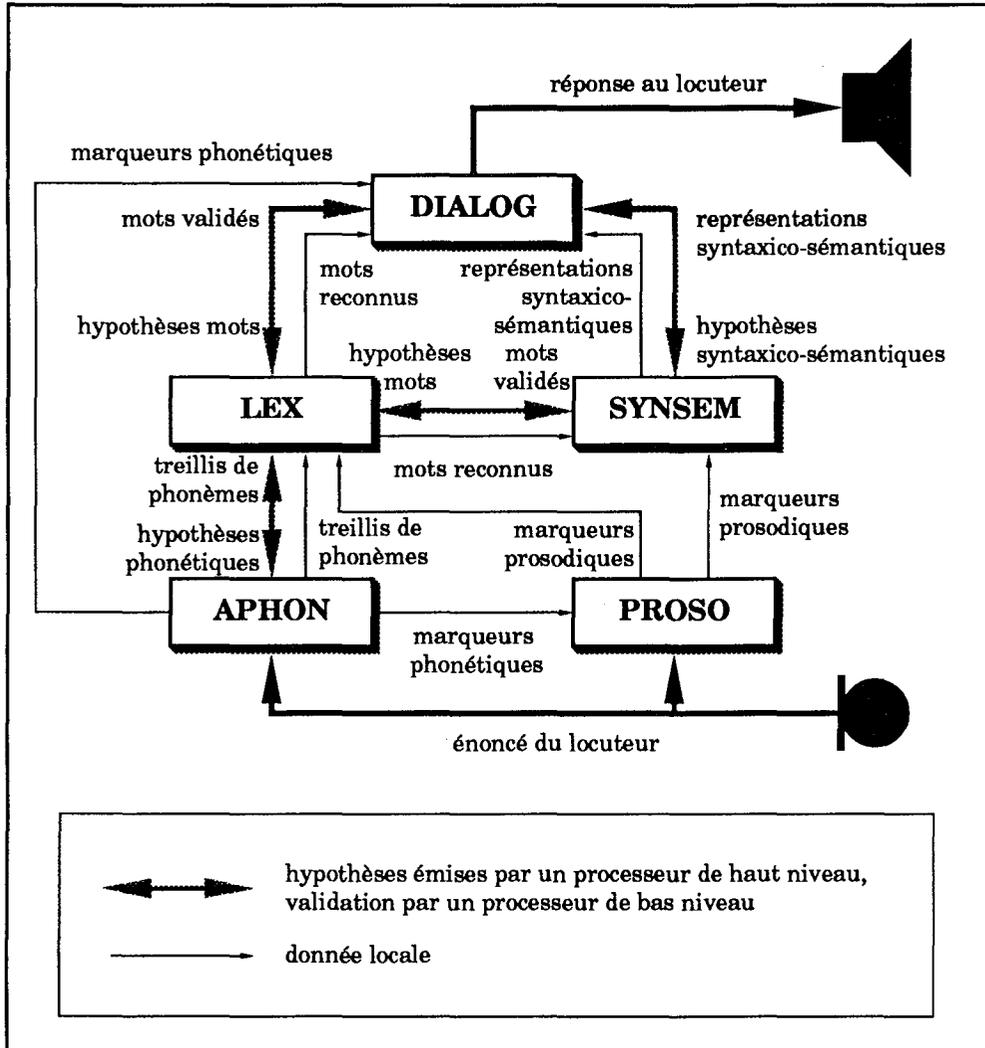


figure 7.4 : Architecture du système DIAL

L'architecture du système DIAL est présentée sur la figure 7.4. Elle intègre une composante de gestion du dialogue au même titre que des composantes de reconnaissance et de compréhension de phrases. La coopération des différents modules, nécessaire à cause de la nature téléologique de la parole (Lindblom, 1983), y est réalisée par un contrôle réparti où les com-

¹ Voir à ce sujet D.G. BOBROW et P.J. HAYES (Eds) *Artificial Intelligence - Where Are We?*, A. SLOMAN y exprime l'opinion que : « ... AI software tools can conquer the world, leaving the software engineers gasping. (For many years to come, rapid prototyping and incremental testing will provide a better guarantee of reliability and robustness than the use of formal specifications, since the specifications will often be bug-ridden.) ».

posantes communiquent deux à deux selon le modèle du producteur/consommateur. Nous avons distingué les modules suivants :

- APHON : le module de décodage acoustico-phonétique (Fohr, 1986 ; Fohr et al., 1985).

Ce module assure deux traitements, la segmentation et l'identification. La segmentation consiste en une étude globale de l'énoncé numérisé du locuteur et est destinée à déterminer les valeurs de ses paramètres : durée moyenne des voyelles, valeur moyenne et variation de l'énergie globale au cours de l'énoncé... ; à effectuer une segmentation grossière en noyaux vocaliques et en groupes consonantiques ; à calculer diverses informations spectrales et temporelles. Elle conduit à une classification des segments qui ne prend pas en compte l'incidence du contexte sur la réalisation des phonèmes et est réalisée, sous forme procédurale, par des algorithmes classiques de traitement du signal. Au cours d'une deuxième étape d'identification des phonèmes, APHON procède alors à une analyse fine, segment par segment, de l'énoncé. Chaque segment est interprété de manière précise en fonction du contexte pour aboutir à la construction d'une représentation de l'énoncé sous forme d'un treillis d'hypothèses phonétiques. L'identification peut éventuellement nécessiter des analyses acoustiques supplémentaires. Elle est actuellement réalisée par un système expert de décodage acoustico-phonétique (Carbonell et al., 1983 ; Carbonell et al., 1986 ; Fohr, 1986 ; Fohr et al., 1985 ; Pierrel, 1987) dont une règle extraite de (Fohr et al., 1985) est présentée à titre d'exemple :

```
SI CONTEXTE_DROIT ( y u w )
SI silence
SI 3000<frequence_burst<4000
ALORS (t)
```

- PROSO : le module prosodique (Pierrel, 1987 ; Carbonell et Bonin, 1988).

Ce module travaille sur l'énoncé numérisé du locuteur ainsi que sur des marques de segmentation fournies par APHON. Il construit les courbes lissées de l'évolution de la fréquence fondamentale et de l'énergie, puis, à partir de l'étude des pics, des vallées et des plateaux de ces courbes, en extrait les marqueurs prosodiques pertinents à savoir des indications sur les frontières de mots ou de syntagmes, et la nature de l'énoncé à traiter de façon à lever certaines ambiguïtés. Considérons par exemple l'énoncé suivant :

« C'est la préfecture qui délivre les visas »

Avec une intonation montante, cette phrase sera interprétée comme une question tandis qu'avec une intonation descendante, elle sera interprétée comme une déclarative. Ce type d'informations peut être très utile pour une analyse syntaxico-sémantique.

- LEX : le module lexical (Haton et al., 1986 ; Pierrel, 1987).

Le module lexical remplit deux fonctions, une fonction de vérification, dans le cas où l'on possède un ensemble d'hypothèses-mots et d'indices temporels précisant leur place probable, ainsi qu'une fonction de sélec-

tion, dans le cas contraire, afin de permettre la construction, à partir du treillis phonétique d'un treillis de mots.

Les problèmes de reconnaissance lexicale sont inhérents à la parole continue et aux limites de la reconnaissance acoustico-phonétique. Pour y faire face, le module lexical utilise différents types de connaissances :

- des règles d'évaluation du processus de segmentation, c'est-à-dire des règles qui permettent d'évaluer les risques de sur- ou de sous-segmentation du signal en fonction de mesures de la composante acoustico-phonétique telles que la longueur des segments etc. ;
- des connaissances phonologiques, c'est-à-dire des règles décrivant les phénomènes d'altération de la prononciation des mots, à savoir les phénomènes d'assimilation, de fusion, de coarticulation et d'anticipation.

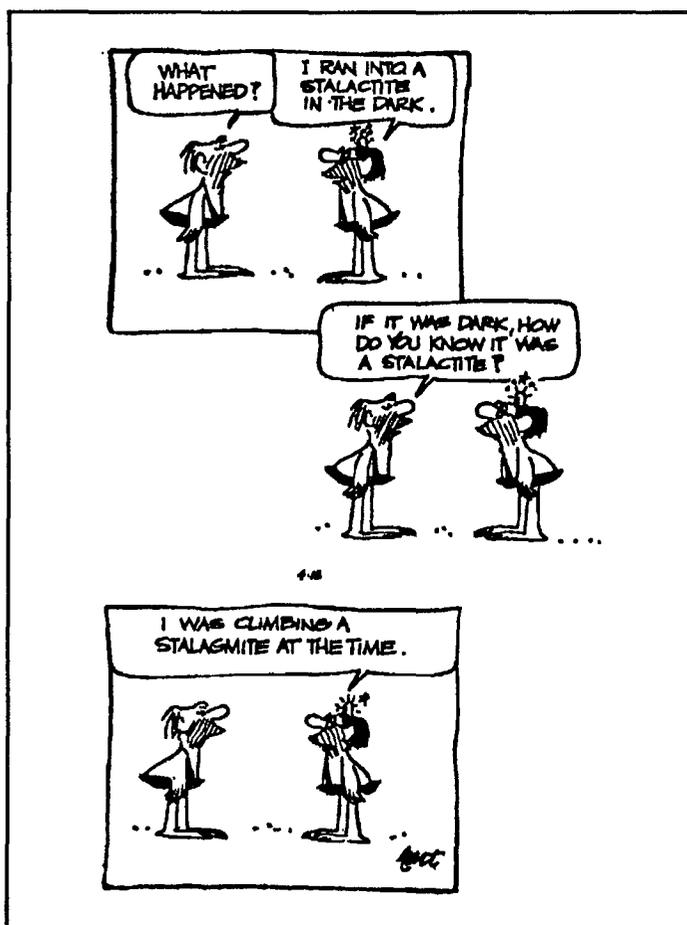


fig. 7.5 : « De l'utilité des connaissances pragmatiques »
extrait de (Hart, 1969)

- SYNSEM : le module syntaxico-sémantique (Mousel et al., 1989a ; Mousel et al., 1989b ; Pierrel, 1987).

Les analyseurs syntaxico-sémantiques sont utilisés pour la reconnaissance des énoncés du locuteur et fournissent comme résultat

une ou plusieurs représentations de l'énoncé sous forme de structures syntaxico-sémantiques grâce à des analyses descendantes gauche-droite ou ascendantes du milieu vers les côtés.

- **DIALOG** : le module pragmatique (Roussanaly, 1988).

La composante dialogue a trois objectifs, elle doit tout d'abord réaliser une interprétation contextuelle des énoncés du locuteur, ensuite elle doit raisonner sur les informations contenues dans ces énoncés et finalement c'est elle qui est chargée de la gestion du dialogue.

Ce sont les deux dernières composantes, que nous qualifions de « composantes de compréhension », qui vont retenir plus particulièrement notre attention par la suite. Nous allons détailler dans les paragraphes suivants le fonctionnement des modules syntaxico-sémantique et pragmatique.

3 La composante syntaxico-sémantique

3.1 Communication avec les autres composantes

L'unité de traitement de la composante syntaxico-sémantique est l'énoncé du locuteur. L'objectif de la composante est de structurer ce dernier sur base de critères syntaxiques et sémantiques. Pour cela, elle échange des informations avec d'autres composantes du système dont :

- **DIALOG**, de laquelle elle reçoit des « hypothèses syntaxico-sémantiques », c'est-à-dire des prédictions sur la structure de l'énoncé à analyser. Elle tente de valider ces hypothèses et fournit à la composante **DIALOG** des « représentations syntaxico-sémantiques » de l'énoncé.
- **LEX**, vers laquelle elle envoie des « hypothèses de type mot » et de laquelle elle obtient des « mots validés » ou simplement des « mots bien reconnus ».
- **PROSO** qui lui fournit des indications sur des frontières de syntagmes. (Dans l'état actuel de notre système, ces informations ne sont pas encore prises en compte par la composante **SYNSEM**).

3.2 Structure de la composante syntaxico-sémantique

Afin de réaliser son objectif qui est la structuration d'un énoncé du locuteur, cette composante intègre plusieurs fonctionnalités :

- Elle doit notamment effectuer différentes analyses de l'énoncé. Nous avons envisagé plusieurs types d'analyse, une analyse qui s'appuie aussi bien sur les « connaissances syntaxiques » que « sémantiques », des « règles de correspondance » sont alors activées pour établir le lien entre les connaissances syntaxiques et sémantiques, ainsi qu'une analyse utilisant exclusivement les connaissances sémantiques. Les analyses peuvent être déclenchées à partir d'hypothèses émises par la composante DIALOG ou à partir de mots reconnus par la composante LEX : dans le premier cas il s'agit « d'analyses descendantes », dans le second il s'agit « d'analyses ascendantes ».

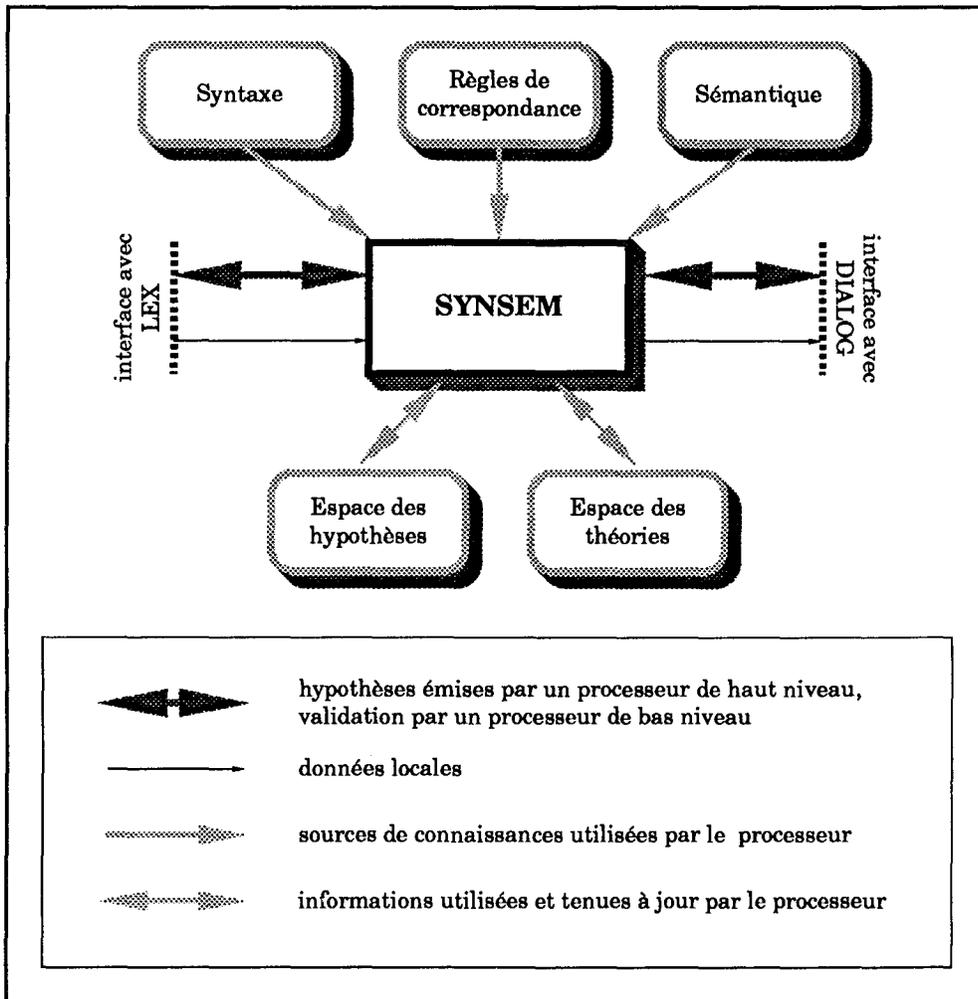


figure 7.6 : Architecture de la composante SYNSEM

- Ces différents types d'analyse nécessitent la gestion d'informations qu'on peut classer en deux catégories : « l'espace des théories » et « l'espace des hypothèses ». L'espace des théories contient l'ensemble des représentations syntaxico-sémantiques de l'énoncé courant que la composante SYNSEM possède à un instant donné. Des théories initiales sont construites sur base des hypothèses syntaxico-sémantiques émises par la composante DIALOG ou sur base des mots bien reconnus par la composante LEX. Ces théories sont alors utilisées par SYNSEM pour émettre des hypothèses mots vers le niveau lexical qui tente de les valider. Avec les mots validés, SYNSEM complète progressivement l'espace des théories. Les représentations syntaxico-sémantiques sont ensuite transmises à la composante DIALOG. L'espace des hypothèses

permet à SYNSEM de garder une trace des hypothèses émises vers la composante LEX et d'éviter ainsi des redondances dans les demandes de validation.

L'architecture de la composante syntaxico-sémantique est présentée sur la figure 7.6. Les connaissances qu'elle met en œuvre seront détaillées ultérieurement, c'est l'objet du huitième chapitre, de même que le seront ses fonctionnalités, c'est l'objet du neuvième chapitre.

4 La composante pragmatique

4.1 Communication avec les autres composantes

L'unité de traitement de la composante DIALOG est le dialogue en cours. Son objectif est de structurer ce dernier sur base de critères pragmatiques. Pour cela elle échange des informations avec d'autres composantes du système dont :

- SYNSEM, vers laquelle elle envoie des « hypothèses syntaxico-sémantiques », c'est-à-dire des prédictions sur la structure du prochain énoncé à analyser. Elle reçoit de cette dernière des « représentations syntaxico-sémantiques » de l'énoncé.
- LEX, vers laquelle elle envoie des « hypothèses de type mot » et de laquelle elle obtient des « mots bien reconnus ».
- APHON qui lui fournit certains « marqueurs acoustico-phonétiques » dont la durée de l'énoncé ainsi que des indications sur l'état du canal de communication.

4.2 Structure de la composante pragmatique

L'objectif de la composante pragmatique étant la structuration du dialogue en cours, elle intègre également plusieurs fonctionnalités :

- En premier lieu, DIALOG effectue une « interprétation contextuelle » de l'énoncé en appliquant des « règles d'interprétation contextuelle » sur la représentation syntaxico-sémantique et sur les mots validés. Le but de cette interprétation est double : d'une part, homogénéiser les différentes représentations et, d'autre part, résoudre les références éventuelles aux énoncés antérieurs à l'énoncé courant.
- L'énoncé interprété va engendrer une phase de « raisonnement » en utilisant un « modèle de la tâche ». Celle-ci consiste à sélectionner les valeurs par défaut, à déduire les informations implicites et à gérer l'état de la tâche, c'est-à-dire déterminer les informations man-

quantantes et détecter les incohérences ainsi que les dépassements des limites du système.

- Le « modèle du dialogue » va, dans la phase suivante de « gestion du dialogue », permettre à la composante de fournir une réponse au locuteur et d'émettre des prédictions sur sa prochaine intervention.
- Ces différents traitements communiquent par l'intermédiaire d'une structure hétérogène, « l'historique du dialogue » qui contient une représentation chronologique du dialogue, une classification typologique des énoncés ainsi que l'état de la tâche.

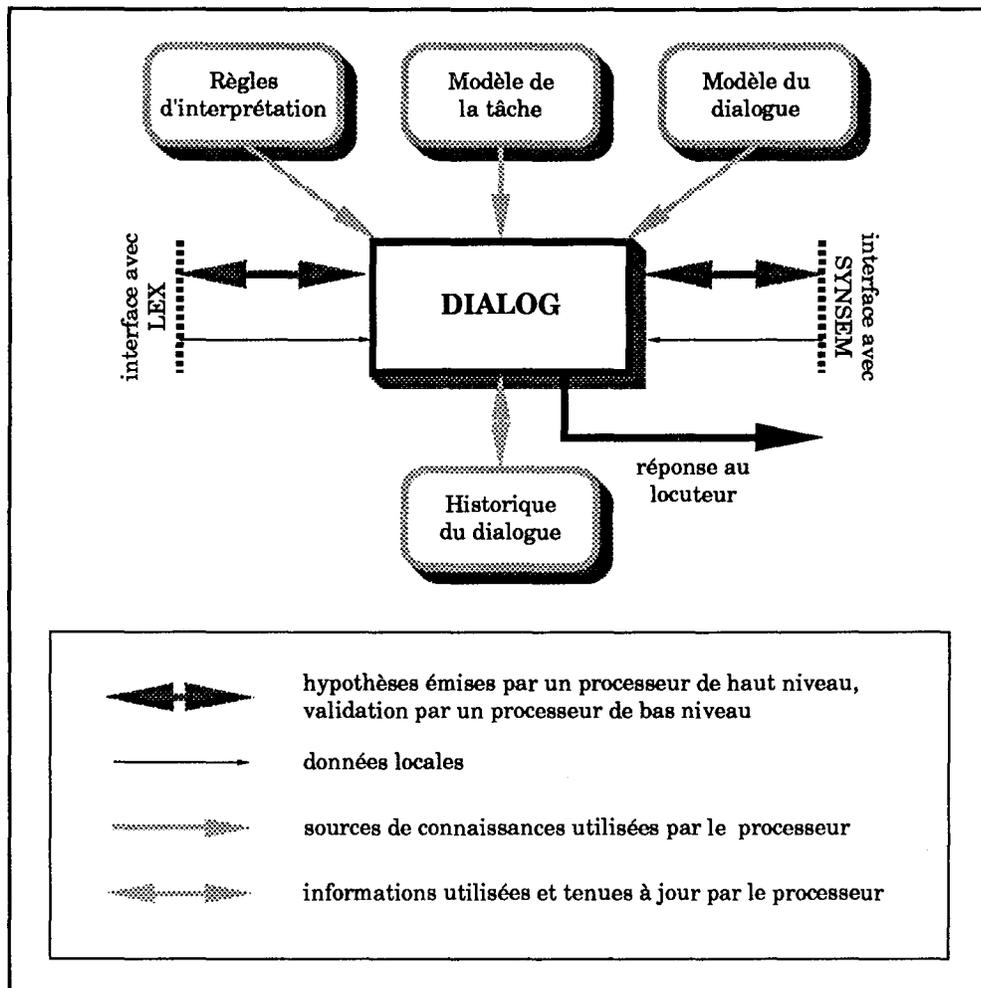


figure 7.7 : Architecture de la composante DIALOG

L'architecture de la composante pragmatique est présentée sur la figure 7.7. Nous n'allons pas la détailler davantage. Cette composante a fait l'objet d'une thèse et nous renvoyons le lecteur à (Roussanaly, 1988) pour tout complément d'information.

LES CONNAISSANCES LINGUISTIQUES DU SYSTEME DIAL

Sur base de l'architecture du système DIAL et plus particulièrement de celle de la composante syntaxico-sémantique présentée au chapitre précédent, nous nous proposons d'exposer dans ce chapitre la nature des différentes connaissances linguistiques qui y sont mises en œuvre.

1 Les Réseaux à Nœuds Procéduraux

1.1 Objectifs des Réseaux à Nœuds Procéduraux

Les Réseaux à Nœuds Procéduraux (notés RNP à partir de maintenant) constituent un modèle de type syntaxique utilisé pour la première fois dans le système MYRTILLE II, développé par J.M. Pierrel au CRIN (Pierrel, 1981). Dans sa Thèse d'Etat, Pierrel fixe quatre objectifs auxquels le modèle des RNP doit répondre. Dans le cadre du système MYRTILLE II, ce modèle doit lui permettre de décrire une structure du langage :

- indépendante de l'application et décrivant un sous-ensemble assez vaste du Français parlé. Ceci n'est plus tout à fait notre objectif dans le cadre du système DIAL. En effet, nos choix méthodologiques nous amènent plutôt à définir une structure de sous-langage indépendante d'une application particulière et portable à l'intérieur d'une famille assez large d'applications. Cet objectif, plus ciblé, reste cohérent avec les objectifs que nous nous sommes fixés au niveau de l'architecture du système DIAL, c'est-à-dire entre autres la paramétrisation du système à différents niveaux. Nous voulons, à l'aide des RNP, pouvoir mo-

déliser la plupart des énoncés du Français parlé dans le contexte d'une application de type "demande de renseignements".

- porteuse d'informations sur la position relative entre les mots. Comme le signale Pierrel, ceci est primordial pour pouvoir restreindre au maximum le nombre d'hypothèses de type mot pour lesquelles sera tentée une reconnaissance acoustico-phonétique sur une partie limitée du treillis phonétique d'entrée.
- permettant de construire en cours de reconnaissance un arbre syntaxico-sémantique, représentant la structure de surface de la partie d'énoncé déjà traitée. Ici aussi, l'objectif a légèrement changé dans la mesure où nous n'utiliserons les RNP que pour construire une représentation syntaxique de l'énoncé, la construction d'une représentation sémantique étant rendue possible par la définition de connaissances sémantiques disjointes mais en relation avec les RNP.
- adaptée au traitement de la parole continue, c'est-à-dire :
 - intégrant des traitements particuliers pour les mots courts de liaison et ne permettant en aucun cas à l'analyse de trop s'appuyer sur eux ;
 - limitant ou plutôt gérant le mieux possible l'indéterminisme dû à la définition syntaxique et aux limites de la reconnaissance des terminaux.

Si, dans le système DIAL, les RNP doivent toujours répondre au deuxième critère, pour des raisons de modularité, nous pensons qu'il est utile de déplacer le problème du traitement des mots courts de liaison vers la composante lexicale.

Pierrel a montré dans sa thèse (Pierrel, 1981) que le modèle des RNP pouvait tout à fait répondre aux quatre objectifs énumérés ci-dessus. Nos objectifs, dans le cadre du système DIAL étant plus restreints, il était tentant pour nous de choisir ce modèle pour définir les structures syntaxiques du sous-langage à traiter.

1.2 Définition des RNP

Les définitions qui vont suivre sont fortement inspirées de (Pierrel, 1981). Les RNP décrivent la syntaxe d'un langage donné à l'aide d'un ensemble de réseaux. Ces réseaux doivent respecter une certaine forme.

Définition

Un **réseau**, au sens RNP, est un graphe orienté où les nœuds et les arcs obéissent à certaines contraintes. A un réseau est associé un symbole. Chaque réseau est constitué d'au moins deux nœuds, appelés respectivement le nœud d'entrée et le nœud de sortie. A l'exception des nœuds d'entrée et de sortie, tous les nœuds sont la destination d'au moins un arc et l'origine d'au moins un arc, le nœud d'entrée n'étant la destination d'aucun arc et le nœud de sortie n'étant l'origine d'aucun arc. A chaque nœud est associé une procédure. A chaque arc orienté est associée une liste ordonnée de symboles.

Il nous faut encore spécifier la sémantique d'une telle représentation. La notion de réseau au sens RNP correspond à la notion de règle de production des grammaires formelles. Un réseau décrit les différentes façons de réécrire, au sens des grammaires formelles, un non-terminal, le symbole qui lui est associé. Chaque chemin dans le graphe que décrit le réseau, ayant comme origine le nœud d'entrée dans le réseau et comme destination le nœud de sortie du réseau, correspond à une façon de réécrire ce non-terminal. La chaîne de symboles produite est fournie par les listes ordonnées de symboles associées aux arcs appartenant au chemin. Les symboles associés aux arcs correspondent soit à des terminaux, soit à des non-terminaux au sens grammaires formelles. Les listes de symboles associées aux arcs étant ordonnées et le graphe étant orienté, on arrive donc parfaitement à exprimer des contraintes positionnelles. Le rôle des procédures associées aux nœuds du réseau était triple à l'origine :

- prendre en compte les phénomènes de type contextuel
- réduire en grande partie l'indéterminisme apparaissant lors du parcours d'un tel réseau en ordonnant les sorties possibles d'une procédure, compte tenu des entrées, du contexte déjà traité, et des résultats de la procédure en cours et
- traiter les mots courts de liaison.

Si les deux premiers points répondent toujours à des besoins dans le système DIAL, il n'en va plus de même en ce qui concerne le troisième. En effet nos choix de réalisation nous amènent à déléguer le traitement des mots courts plutôt à la composante lexicale (cf. § 1.1).

En guise d'illustration, nous allons discuter brièvement un exemple d'un tel réseau utilisé par la composante syntaxico-sémantique de DIAL. L'exemple de la figure 8.1, tiré de (Deville, 1989), décrit la structure du groupe nominal tel qu'il se présente dans l'application test que nous avons choisie. Le nœud d'entrée dans ce réseau est le nœud noté Gn_1, le nœud de sortie en est le nœud noté Gn_6. Les autres nœuds sont notés Gn_2, Gn_3, Gn_4 et Gn_5. Les restrictions principales que nous voulons introduire dans ce réseau sont :

- le fait qu'on ne puisse associer dans notre groupe nominal à un même nom un complément de nom (GP) et une subordonnée relative (RELAT), excluant ainsi les groupes nominaux ambigus (comme dans « la carte de mon fils qui est à Nancy ») ;

- le fait que la coordination soit limitée à deux noms, évitant ainsi des coordinations en cascade (du genre « le passeport et la carte d'identité ou le permis de séjour »).

Si la première restriction est une contrainte hors-contexte (les arcs reliant Gn_4 à Gn_5 expriment bien qu'un nom peut être suivi d'un groupe prépositionnel ou d'une relative mais jamais des deux à la fois), la deuxième, par contre, est une contrainte contextuelle et n'apparaît pas au niveau du réseau. Elle doit être exprimée par une procédure rattachée à un nœud.

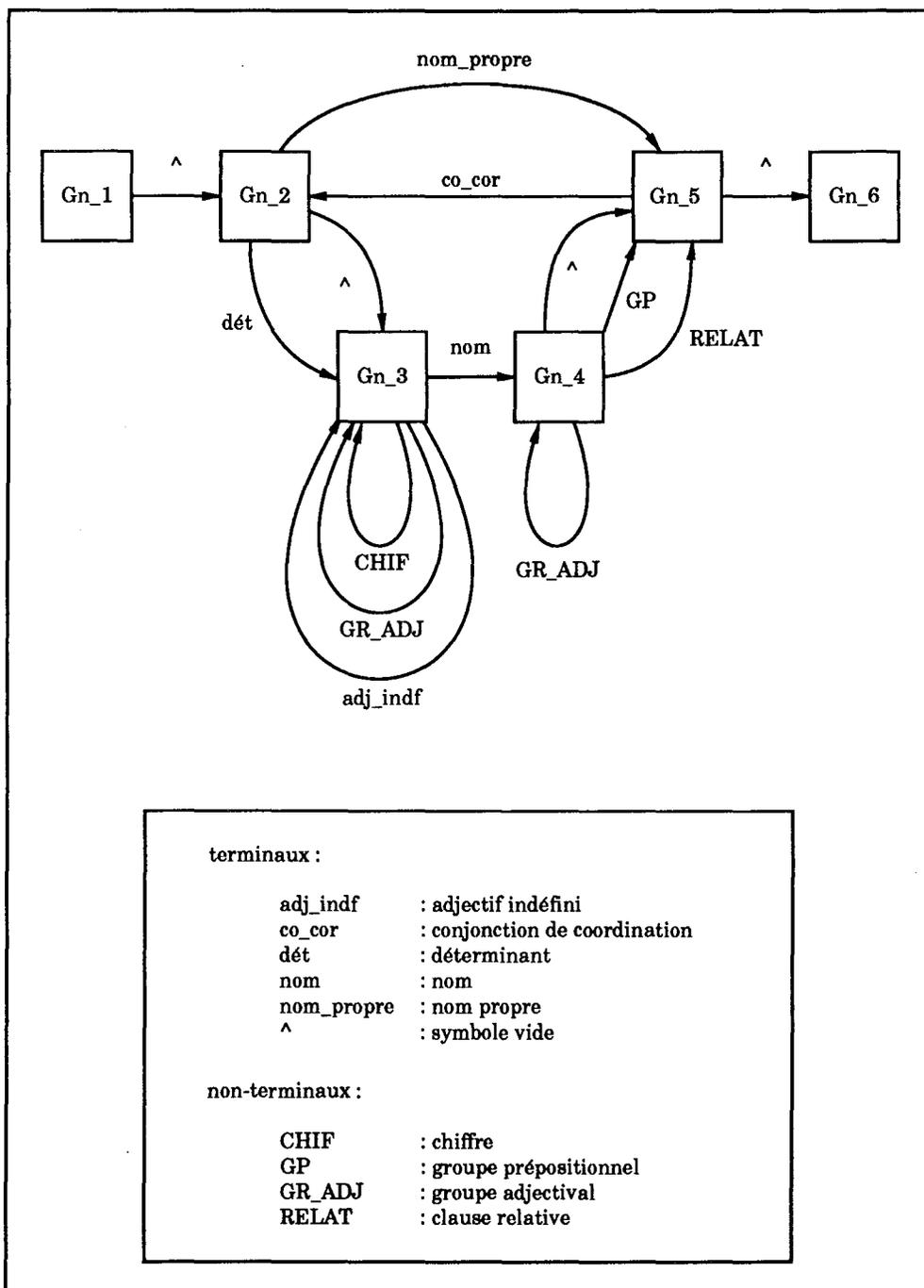


figure 8.1 : Exemple de RNP, le groupe nominal (GN)

Cette procédure associée au nœud Gn_5 de notre exemple pourrait exprimer la contrainte relative à la conjonction de coordination de la façon suivante :

si

lors de l'analyse d'un groupe nominal (GN), on a déjà reconnu une conjonction de coordination (co_cor)

alors

les sorties du nœud dans l'ordre décroissant de leurs plausibilités sont :
la sortie vide (^)

sinon

les sorties du nœud dans l'ordre décroissant de leurs plausibilités sont :
la sortie vide (^)
la sortie conjonction de coordination (co_cor)

1.3 Puissance de description des RNP et comparaison avec d'autres types de représentation

Ce problème a été abordé par Pierrel dans sa Thèse d'Etat (Pierrel, 1981). Les conclusions principales qu'on peut en tirer sont les suivantes :

- tout langage hors-contexte peut être décrit à l'aide de RNP ;
- les RNP permettent de décrire des langages qui ne sont pas à contexte libre¹.

La puissance de description des RNP dépend essentiellement du contenu des nœuds procéduraux et peut atteindre celle des langages de programmation. Par conséquent tout langage engendré par une grammaire formelle peut aussi être décrit par un RNP. Ceci reste vrai pour tous les types de représentations assimilables aux grammaires formelles, par exemple le formalisme de Backus-Naur ou BNF (Naur, 1963). De même, les différents types d'automates trouvant chacun leur équivalent parmi les grammaires formelles, tout langage accepté par un automate peut également être décrit par un RNP. Comme dans le cas des grammaires formelles, ceci reste vrai pour tous les types de représentations assimilables aux automates comme les diagrammes syntaxiques, encore appelés diagrammes de Conway (Jensen et Wirth, 1976), les diagrammes de transition d'état finis et les réseaux de transition récursifs, encore appelés RTN (Woods, 1970), par exemple. Un type de représentation un peu plus particulier sont les réseaux de transition

¹ On pourra lire à ce sujet J.M. PIERREL dans *Etude et mise en œuvre de contraintes linguistiques en compréhension automatique du discours continu* page 173 : « Il est donc possible de représenter des langages autres que des langages à contexte libre à l'aide des réseaux à nœuds procéduraux : on pouvait d'ailleurs intuitivement arriver à cette conclusion en remarquant que : - la structure de réseau est comparable à un graphe général et permet donc de représenter l'ensemble des C-langages ; - les procédures internes permettent de plus de prendre en compte toutes sortes d'informations et donnent donc aux R.N.P. une puissance de description identique aux langages de programmation. En fait, la puissance d'un tel modèle est directement liée aux actions que l'on accepte d'introduire dans les procédures. »

augmentés ou ATN (Woods, 1970). Il s'agit de RTN étendus de trois façons (Barr et Feigenbaum, 1981) :

- un ensemble de registres a été ajouté ; ils peuvent être utilisés pour enregistrer de l'information ;
- les arcs, à côté du fait d'être étiquetés par des classes de mots ou des constructions syntaxiques, peuvent avoir des tests arbitraires associés qui doivent être satisfaits avant que l'arc soit pris ;
- certaines actions peuvent être "attachées" à un arc, pour être exécutées chaque fois qu'il est pris.

Pierrel signale que ses RNP sont assez proches des ATN de Woods (Pierrel, 1987). Ils s'en distinguent cependant par le rôle des procédures :

- celles des ATN ont pour rôle essentiel de construire et de gérer la représentation syntaxico-sémantique de la phrase ;
- dans le cas des RNP, l'obtention et la gestion de cette structure sont prises en charge par une procédure d'analyse externe, le rôle des procédures est alors de déterminer dynamiquement, lors de l'analyse d'une phrase, l'ordre de prise en compte des constructions possibles en supprimant éventuellement certaines possibilités, compte tenu du contexte déjà analysé.

En résumant, on peut affirmer que les RNP de Pierrel ont la même puissance de description que les ATN de Woods avec l'avantage non négligeable de permettre une séparation beaucoup plus nette entre la représentation des connaissances et leur traitement (Lyons et al., 1987)¹. Ceci facilite tout d'abord la réalisation incrémentale de systèmes mais également le développement de systèmes adaptables et portables.

1.4 La modélisation syntaxique du sous-langage de notre application test

Les RNP furent utilisés pour la première fois dans le cadre de l'application de renseignements météorologiques mise en œuvre dans le système MYRTILLE II (Pierrel, 1981). A l'époque, l'objectif de Pierrel était de décrire, à l'aide des RNP, une syntaxe sous-ensemble assez vaste du français de base, le lexique devant être spécifique au domaine précis de l'application. Le langage ainsi décrit était appelé « langage pseudo-naturel » et les principaux avantages que Pierrel attendait de cette modélisation étaient :

¹ Lire également à ce sujet *Computational Linguistics* de G. GAZDAR et C. MELLISH dans *New Horizons in Linguistics* (J. LYONS et al. Eds.) page 231 : « ATN-based parsers were probably the most common kind of parser employed by computational linguists in the 1970s, but they have fallen out of favour in recent years. This is largely because the 'augmentations' destroy the declarative nature of the formalism and because a parser using an ATN is limited in the control strategies it can employ. ».

- la possibilité d'utiliser du langage quasi naturel pour la communication homme-machine ;
- utilisation qui devait être facile et ne nécessiter aucun apprentissage préalable, si toutefois le locuteur se limitait dans ses énoncés au domaine précis de l'application mise en œuvre ;
- un passage facile d'une application à une autre.

La modélisation du français que propose Pierrel, pour laquelle il a conçu quelque quinze sous-réseaux, n'a pu être reprise telle quelle dans le système DIAL pour deux raisons.

Tout d'abord Pierrel s'est appuyé pour ses travaux sur diverses études linguistiques notamment la grammaire de base du français de J. Dubois (Dubois, 1976) ainsi que le précis de grammaire française de M. Grevisse (Grevisse). Il s'agit là d'ouvrages consacrés essentiellement à l'étude du français écrit. L'étude de corpus (Deville, 1987 ; Roussanaly et al., 1986) montre cependant que le français parlé peut en différer assez fortement. Ces études ne sauraient donc servir de bases linguistiques à la modélisation du langage que nous désirons traiter dans le cadre de notre application qui, rappelons-le, s'adresse au grand public (Carbonell, 1984). En particulier, les structures présentatives, très fréquentes dans le langage parlé, ne peuvent être analysées par la grammaire de MYRTILLE II. Par exemple la question : « C'est à vous que je dois m'adresser ? » est un énoncé extrait du corpus enregistré au Ministère de l'Emploi et du Travail à Bruxelles.

Ensuite, la modélisation syntaxique du langage dans MYRTILLE II ne tient pas compte de certaines caractéristiques spécifiques aux sous-langages par rapport au langage général (cf. chapitre 5). Or, le langage que nous désirons prendre en compte possède toutes les caractéristiques d'un sous-langage (Deville, 1989). Ce sous-langage peut, au niveau syntaxique, être structurellement décrit suivant un mode préférentiel (les différences au niveau syntaxique entre langage général et sous-langage ne pouvant pas toujours être exprimées en termes binaires de déviance ou de restriction, il semble parfois plus adapté de les exprimer en termes probabilistes), dans le sous-langage utilisé dans le renseignement administratif, la forme interrogative est notamment beaucoup plus fréquente que dans le langage général. Un ensemble de règles décrivant un sous-ensemble du français de base ne saurait par conséquent modéliser le sous-langage qui nous intéresse (Deville, 1989).

Ces considérations nous ont amenés à redéfinir complètement la syntaxe utilisée par le système DIAL. Ce travail a été effectué par G. Deville dans le cadre de sa thèse doctorale (Deville, 1989) et nous renvoyons le lecteur à cet ouvrage pour une discussion détaillée de la modélisation syntaxique. Actuellement, nous avons défini une trentaine de sous-réseaux, tous sont présentés en annexe (cf. Annexe C). Nous allons terminer en présentant les principales limitations de notre syntaxe :

- Restrictions sur les coordinations. La conjonction est une des constructions les plus complexes du français. Elle pourrait survenir dans pratiquement toutes les structures définies dans notre grammaire provoquant ainsi une explosion combinatoire des structures possibles et de ce fait introduirait trop d'indéterminisme dans l'analyse du lan-

gage. C'est la raison pour laquelle nous limitons pour l'instant la conjonction aux syntagmes nominaux.

- Position des subordonnées. Dans la plupart des cas, les subordonnées et les phrases adverbiales ne pourront se trouver qu'en début ou en fin de la phrase.
- Suppression de plusieurs figures de style. Quelques figures de style comme les phrases interpolées (p. ex. : « Je crois, et je pense que j'ai de bonnes raisons de le croire, qu'il se trompe. ») et les phrases exclamatives (p. ex. : « Mais c'est cher ça ! ») ne sont pas acceptées par notre grammaire.

Signalons encore que, dans un souci de généralité, la description des contraintes syntaxiques ne descend pas jusqu'au niveau des mots qui sont fortement liés aux concepts manipulés dans une application déterminée mais les terminaux de la grammaire se limitent aux catégories syntaxiques¹. Notre modèle devrait donc être assez général pour englober le sous-langage du renseignement administratif indépendamment d'une application particulière. L'étude des deux corpus « pages roses » (enregistré au CRIN) et « action-travail » (enregistré au Ministère de l'Emploi et du Travail à Bruxelles) semble confirmer cet espoir (Deville, 1987 ; Roussanaly et al., 1986). Toutefois, dans le cadre d'une application donnée, il est nécessaire de définir quelque part les liens entre les mots et leurs catégories syntaxiques. Nous avons décidé d'explicitier ces liens dans la composante lexicale (Haton et al., 1986 ; Pierrel, 1987).

2 La grammaire de cas

2.1 Objectifs de la grammaire de cas

La grammaire de cas que nous nous proposons d'exposer dans les paragraphes suivants est un modèle qui a été défini dans le cadre du développement du système DIAL. Elle tente de répondre à deux besoins essentiels :

- d'une part, les connaissances sémantiques peuvent servir à guider l'analyse syntaxique et contribuer à ne sélectionner parmi l'ensemble des analyses possibles que celles qui ont un « sens » (Oden, 1983) ;
- d'autre part, la structure sémantique des énoncés des locuteurs est en fait le résultat principal que nous attendons de la composante syntaxico-sémantique (Roussanaly, 1988).

En effet, la structure syntaxique d'un énoncé n'en décrit que la forme de « surface » et convient assez mal aux traitements ultérieurs qu'on est

¹ Certains terminaux de la grammaire échappent à cette règle, ce sont les terminaux vrais. Nous renvoyons le lecteur à l'annexe C pour une liste exhaustive de ces terminaux vrais.

amené à lui appliquer, à savoir des raisonnements, des inférences etc.. Pour cela, il faut mettre en lumière les relations « profondes » qu'entretiennent les éléments de l'énoncé. C'est ce que nous voulons réaliser à l'aide d'une grammaire de cas.

Dès lors, on pourrait se poser la question pourquoi une telle composante devrait encore intégrer des connaissances syntaxiques. La raison en est que des connaissances sémantiques, représentées par un modèle casuel par exemple, ne véhiculent en général aucune information positionnelle sur les éléments dont ils décrivent les relations. Ce fait, s'il n'est pas particulièrement gênant dans le cas du langage naturel écrit, peut poser des problèmes insolubles dans le cas de la parole. On constate notamment que, contrairement à l'écrit, dans un flot continu de parole, les frontières entre les mots ne peuvent être détectées de façon certaine, les segments minimaux donnent lieu à des interprétations phonétiques non univoques etc.¹. La conséquence en est que si on ne tient compte que des seules contraintes sémantiques, d'un côté le nombre de structures possibles par manque de contraintes positionnelles risque de devenir trop élevé, de l'autre l'identification d'un mot sur le flot de parole peut nécessiter un temps de recherche prohibitif vu le manque d'informations sur la position du mot.

Ces considérations nous ont conduits à envisager la coopération étroite de deux modèles distincts à l'intérieur de la composante syntaxico-sémantique, un modèle syntaxique basé sur les RNP et un modèle sémantique basé sur une grammaire de cas.

2.2 Définition de la grammaire de cas

Le terme « grammaire de cas » que nous avons choisi pour désigner notre modèle sémantique est inspiré de la définition des « cas » que fournit Fillmore : « un ensemble de relations sémantiques universelles entre un prédicat et des arguments » (Fillmore, 1968). Cependant la définition des cas en termes de traits sémantiques primitifs que nous allons proposer est plus proche de la notion de « fonction sémantique » de Dik (Dik, 1978) que de la notion de « cas profond » de Fillmore. Pour une discussion approfondie des aspects linguistiques liés à ces problèmes, nous renvoyons le lecteur à la Thèse de Doctorat de G. Deville (Deville, 1989). L'objectif des paragraphes qui vont suivre est plus limité. Il s'agit de définir les concepts élémentaires mis en œuvre dans notre modèle sémantique - ces définitions s'inspirent d'ailleurs fortement de (Deville, 1989) -, de montrer brièvement quels en sont les aspects originaux et finalement de présenter succinctement la modélisation du sous-langage de notre application test.

La notion de « formule » est une notion centrale à notre modèle².

¹ Dans un texte écrit par contre, les frontières entre les mots sont facilement détectables. De même, les unités minimales qui constituent les mots, c'est-à-dire les lettres, ne posent pas de problèmes d'identification. Le traitement du langage naturel ne rencontre de problèmes analogues à ceux du traitement de la parole que si on essaie de traiter des textes fortement erronnés ou lorsqu'on fait de la lecture optique de textes.

² G. DEVILLE les appelle des « predications ».

Définition

Une formule est une structure qui consiste en un prédicat et un nombre déterminé de termes, arguments du prédicat.

Elle s'appuie sur les notions de « prédicat » et de « terme »¹.

Définition

Un prédicat est une expression (le plus souvent un verbe, un nom ou un adjectif) faisant référence à une entité dans un univers conceptuel de sous-langage et reflétant les propriétés sémantiques de ses arguments ainsi que les relations sémantiques entre ces arguments.

Définition

Un terme est une expression (le plus souvent un groupe nominal ou prépositionnel, un pronom ou un membre) faisant référence à une entité ou un ensemble d'entités dans un univers conceptuel de sous-langage.

Par conséquent, une formule fait référence à une configuration d'univers de sous-langage².

Définition

Une configuration d'univers de sous-langage est une constellation d'entités conceptuelles du sous-langage exprimées en termes de leurs relations mutuelles.

Les entités conceptuelles, dans le cas du système de dialogue du CRIN, sont définies au niveau de la composante pragmatique dans le modèle de la tâche. Elles font référence non seulement à des objets (par exemple « carte d'identité ») mais également à des états (par exemple « valide »), des relations (par exemple « nationalité ») et des actions (par exemple « donner »). De façon générale, les termes font référence à des objets tandis que les prédicats font référence à des états, des relations ou des actions. Comme il s'agit d'entités conceptuelles définies par un sous-langage, l'ensemble des concepts peut être considéré comme un ensemble fini.

Si on voulait définir les contraintes sémantiques sur base des seules notions ci-dessus, le nombre de règles serait trop élevé pour pouvoir être maîtrisé. En effet, il faudrait, pour chaque expression pouvant réaliser un prédicat, fournir la liste des expressions pouvant réaliser un terme pour ce prédicat. De plus, l'ensemble des concepts étant particulier à une application, il faudrait concevoir un ensemble de règles spécifique pour chaque application nouvelle. Nous voulons cependant définir des modèles du langage qui

¹ G. DEVILLE parle de « predicate » et de « term ».

² G. DEVILLE parle de « Sublanguage world Configuration ».

soient portables au moins à l'intérieur d'une famille d'applications, d'où les notions de « primitive de prédicat » et de « cas central »¹.

Définition

Une primitive de prédicat (ou simplement primitive) est une entité prototypique qui englobe les propriétés sémantiques et combinatoires d'un ensemble de prédicats.

Définition

Un cas central (ou simplement cas) est l'expression prototypique d'une fonction sémantique remplie par un terme, argument d'un prédicat, en fonction de la primitive que ce prédicat réalise.

Dès lors, pour définir les contraintes sémantiques, il suffit de fournir, pour chaque primitive, un « schéma de cas »², c'est-à-dire la liste des cas qu'elle requiert, et de choisir des primitives et des schémas de cas suffisamment généraux pour être communs à une famille d'applications. Des traits sémantiques associés aux cas permettent alors d'imposer des contraintes aux termes réalisant ces cas.

Définition

Un schéma de cas est la liste des cas requis par une primitive pour préserver le sens prototypique de cette primitive tel qu'exprimé par un ensemble de configurations d'univers de sous-langage correspondant à la primitive.

Signalons encore qu'un schéma de cas peut être étendu par des « cas périphériques »³.

Définition

Un cas périphérique est un argument qui ne fait pas partie de la définition d'une configuration d'univers de sous-langage mais exprime le contexte spatio-temporel de la configuration, précise les entités secondaires impliquées dans la configuration et fournit des informations sur la cause, les conditions ou le but de la configuration d'univers de sous-langage.

L'exemple de la figure 8.2 montre la représentation sémantique d'une expression linguistique.

Le problème de la modélisation sémantique revient dès lors à une définition et un choix judicieux des primitives et des schémas de cas qui leur sont

¹ Appelées respectivement « predicate primitive » et « central case » par G. DEVILLE.

² G. DEVILLE les appelle des « case frame ».

³ Appelés « peripheral arguments » par G. DEVILLE

associés. Nous allons préciser dans le paragraphe suivant ce que nous entendons par choix judicieux des primitives et des schémas de cas.

2.3 Comparaison avec d'autres modèles sémantiques

Nous avons signalé ci-dessus que notre modèle sémantique s'inspirait des travaux de Fillmore. Malheureusement, un des reproches essentiels qui lui ont été adressés, est le manque de critères objectifs de discrimination de ses relations casuelles. Afin d'éviter ce reproche, nous avons tenu à lier nos cas à une typologie de configurations d'univers de sous-langage qui se reflète dans une typologie de primitives. De ce fait, notre démarche est beaucoup plus proche de celle de Dik que de celle de Fillmore. En effet, nous avons opéré une première classification des prédicats suivant les traits typologiques \pm contrôlé et \pm dynamique en « actes » (+dynamiques et +contrôlés), « processus » (+dynamiques et -contrôlés) et « états » (-dynamiques et \pm contrôlés)¹. Il ne s'agit nullement de traits subjectifs, mais de traits qui peuvent être détectés par des tests syntaxiques. Le tableau de la figure 8.3 fournit les tests syntaxiques qui permettent de décider de l'appartenance d'un prédicat à l'une des trois classes.

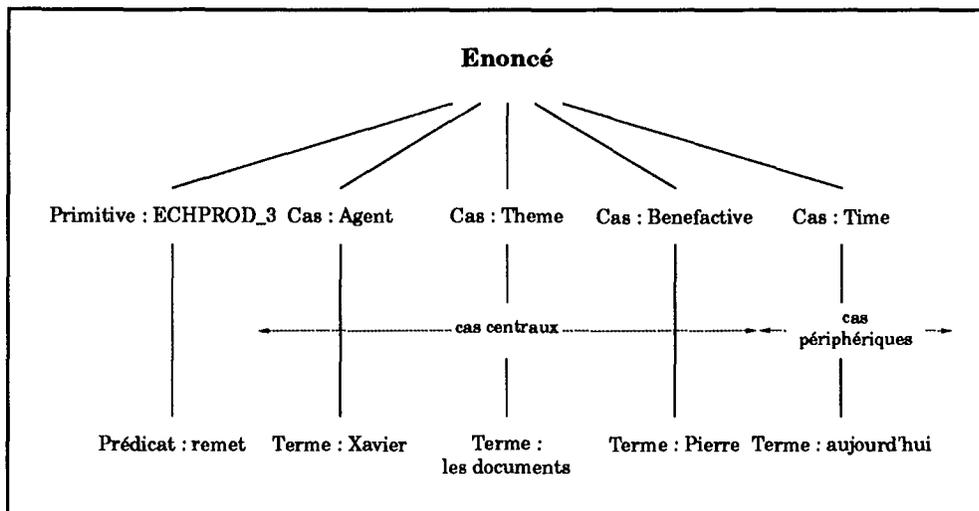


fig. 8.2 : Représentation sémantique de l'énoncé
« Xavier remet les documents à Pierre aujourd'hui »

Cependant, la classification opérée par Dik nous semble encore trop grossière. En effet, nous pensons que si on veut compléter utilement des contraintes syntaxiques par des contraintes sémantiques, cette classification doit être affinée. C'est la raison pour laquelle nous avons introduit les traits sémantiques \pm attributif, \pm spatial et \pm cognitif ainsi que les traits de valence \pm transitif et \pm ditransitif. Ici encore, il ne s'agit nullement de traits choisis arbitrairement mais soit de traits reflétant des dimensions conceptuelles élémentaires qui caractérisent la plupart des configurations d'univers de sous-langage en termes de relations prototypiques entre les entités impliquées dans ces configurations (c'est le cas des traits sémantiques) soit de

¹ En se référant au chapitre 6, le lecteur pourra vérifier que ces traits sémantiques correspondent à ceux proposés par S. DIK.

traits faisant référence à la « valence syntaxique » des prédicats dérivés d'une primitive (c'est la cas des traits de valence).

	+dynamique		-dynamique	
	+contrôlé	-contrôlé	±contrôlé	
	actes	processus	états	
1. peut se combiner avec <i>lentement, graduellement</i>	+	+	-	-
2. peut apparaître à la forme progressive	+	+	+	-
3. peut apparaître dans une construction extraposée	+	+	+	-
4. peut être combinée avec des arguments périphériques de manière	+	-	+	-
5. peut apparaître à la forme impérative	+	-	+	-
6. peut être combinée avec des verbes tels que <i>obliger, persuader etc.</i>	+	-	+	-
7. peut être combinée avec des arguments périphériques de manière tels que <i>consciemment, délibérément etc.</i>	+	-	+	-
8. peut être combinée avec des arguments périphériques de manière tels que <i>avec entrain, enthousiasme etc.</i>	+	-	-	-

fig. 8.3 : Une typologie de configurations d'univers de sous-langage avec les tests syntaxiques correspondants d'après (Deville, 1989)

Sans vouloir entrer dans les détails d'une discussion théorique certes intéressante mais qui dépasserait le cadre de cette thèse, nous allons présenter brièvement les traits que nous avons introduits en commençant par les traits sémantiques :

- une primitive possédant le trait sémantique +attributif exprime une relation de possession, de transfert de possession ou de perte de possession entre les entités d'une classe de configurations d'univers de sous-langage. Des actes (par exemple : délivrer, voler etc.) aussi bien que des processus (par exemple : obtenir, perdre etc.) que des états (par exemple : bénéficiaire, manquer etc.) peuvent posséder le trait sémantique +attributif ;
- une primitive possédant le trait sémantique +spatial exprime une relation spatiale entre les entités d'une classe de configurations d'univers de sous-langage. Des actes (par exemple : arriver, voyager etc.) aussi bien que des processus (par exemple : provenir, aboutir etc.) que des états (par exemple : résider, figurer etc.) peuvent posséder le trait sémantique +spatial ;
- une primitive possédant le trait sémantique +cognitif exprime une relation perceptive, conceptuelle ou émotionnelle entre les entités d'une

classe de configurations d'univers de sous-langage. Des actes (par exemple : attester, exprimer etc.) aussi bien que des processus (par exemple : remarquer, confondre etc.) que des états (par exemple : regretter, présumer etc.) peuvent posséder le trait sémantique +cognitif.

Les traits de valence d'une primitive quant à eux déterminent le nombre d'arguments requis par un prédicat pour préserver son sens prototypique tel qu'exprimé par cette primitive :

- une primitive possédant le trait de valence +transitif fait référence à des prédicats qui, en plus d'un premier argument dont la fonction syntaxique est le sujet, requièrent un deuxième argument dont la fonction syntaxique peut être le complément d'objet direct, le complément d'objet indirect etc. (par exemple : « Je suis allé à la mairie »). Elle exprime une relation sémantique binaire ;
- une primitive possédant le trait de valence +ditransitif fait référence à des prédicats qui, en plus d'un premier argument dont la fonction syntaxique est le sujet, requièrent deux arguments supplémentaires dont les fonctions syntaxiques peuvent être le complément d'objet direct, le complément d'objet indirect etc. (par exemple : « Je lui ai donné mon passeport »). Elle exprime une relation sémantique ternaire.

Les traits typologiques, sémantiques et de valence ainsi définis vont nous permettre de classifier les prédicats en primitives. Nous renvoyons le lecteur à (Chafe, 1970 ; Dik, 1978 ; Dowty, 1979 ; Vendler, 1967) pour une discussion approfondie des problèmes liés à la classification de verbes.

Le regroupement des prédicats en primitives sur base des critères proposés ci-dessus va influencer directement sur les schémas de cas associés aux différentes primitives. En effet, les traits de valence déterminent le nombre des cas obligatoirement associés à une primitive tandis que les traits typologiques et sémantiques en déterminent les fonctions sémantiques.

2.4 La modélisation sémantique du sous-langage de notre application test

Le modèle casuel tel qu'il a été défini dans le cadre du système DIAL constitue un travail original. En particulier, nous avons essayé de tenir compte des caractéristiques de ce sous-langage et de valider empiriquement notre modélisation par des études de corpus enregistrés en situation réelle. La partie linguistique de ce travail a été assurée par G. Deville dans le cadre de sa thèse doctorale (Deville, 1989) et nous renvoyons le lecteur à cet ouvrage pour une présentation détaillée de la modélisation sémantique. Un résumé avec des exemples en est fourni en annexe (cf. Annexe E). La figure 8.4 montre les différents schémas de cas centraux qui ont été associés aux primitives de notre sous-langage.

Rappelons encore que, au vu de notre objectif, les traits typologiques, sémantiques et de valence que nous avons distingués sont pertinents dans le cadre du sous-langage que nous voulons modéliser et ne prétendent donc à

aucune universalité. Nous pensons que des études empiriques supplémentaires sont nécessaires afin de dégager les limites de cette modélisation. De par l'approche que nous avons choisie, cette modélisation devrait cependant être assez générale pour englober le sous-langage du renseignement administratif indépendamment d'une application particulière. L'étude des deux corpus « pages roses » et « action-travail » semblent confirmer cet espoir (Deville, 1987 ; Roussanaly et al., 1986).

Noms des primitives	Configurations de cas
ACT_1	Agent
ACT_2	Agent + Theme
ACT_3	Agent + Theme + Referent
ECHPROD_3	Agent + Theme + Benefactive
MVMT_1	Agent
MVMT_2	Agent + (Source Direction Path)
MVMT_3	Agent + Theme + (Source Direction Path)
COGNT_1	Agent
COGNT_2	Agent + Theme
COGNT_3	Agent + Theme + Cognizant
STATUT_1	Theme
STATUT_2	Theme + Referent
POSSESS_2	Benefactive + Theme
LOCATION_2	Theme + Locative
LOCATION_3	Positioner + Theme + Locative
ST_COGN_2	Cognizant + Theme
PROCESS_1	Processed
PROCESS_2	Effector + Theme
ECHOBT_2	Benefactive + Theme
PR_MVMT_1	Processed
PR_MVMT_2	Processed + (Source Direction Path)
PR_MVMT_3	Effector + Theme + (Source Direction Path)
PR_COGNT_2	Cognizant + Theme

fig. 8.4 : Les schémas de cas centraux associés aux primitives

Enfin, dans le cadre d'une application particulière, il est nécessaire d'explicitier les liens entre les mots de l'application et les catégories sémantiques que nous avons distinguées, à savoir les primitives et les cas. Nous

avons décidé d'expliciter ces liens dans la composante lexicale (Haton et al., 1986 ; Pierrel, 1987). Dans ce contexte, il nous semble important de préciser encore une dernière notion de nature sémantique, liée aux prédicats du lexique, mais intervenant dans les règles de correspondance qui seront décrites ci-dessous. Il s'agit de la notion de « polarité ». En fait une configuration d'univers de sous-langage est censée représenter une partie de la réalité. Différents énoncés peuvent décrire cette même réalité de différents points de vue. Illustrons ceci par les énoncés suivants :

- L'employé vend un timbre à Pierre
- Pierre achète un timbre à l'employé

Les deux prédicats « vend » et « achète » réalisent tous les deux la même primitive ECHPROD_3, à savoir un acte attributif et ditransitif. Dans les deux cas, il y a transfert de possession de l'entité « employé » vers l'entité « Pierre » de l'entité « timbre ». Il s'agit par conséquent de la même configuration qu'il serait préférable de représenter par la même structure :

- vendECHPROD_3 (employé_{Agent}, Pierre_{Benefactive}, timbre_{Theme}) ;
- achèteECHPROD_3 (employé_{Agent}, Pierre_{Benefactive}, timbre_{Theme}).

Dans le premier énoncé où la polarité du prédicat est dite positive, le sujet syntaxique de l'énoncé remplit la fonction sémantique d'agent tandis que le complément d'objet syntaxique remplit la fonction de bénéficiaire. Par contre, dans le deuxième énoncé où la polarité du prédicat est dite négative, le sujet remplit, jusqu'à un certain point, la fonction de bénéficiaire tandis que le complément d'objet remplit, dans cette même perspective, la fonction d'agent.

3 Les règles de correspondance

3.1 Objectifs des règles de correspondance

Le troisième type de connaissances utilisées par la composante syntaxico-sémantique est une base de règles de correspondance qui permet de faire le lien entre les connaissances de nature syntaxique et les connaissances de nature sémantique. Dans la mesure où le résultat le plus important de la composante SYNSEM est une structure sémantique prédicative de l'énoncé, il semble évident que la composante SYNSEM doit disposer des connaissances nécessaires pour transformer une structure syntaxique en structure sémantique. Toutefois, notre objectif est d'aller plus loin. Comme nous l'avons signalé au paragraphe précédent, nous voulons utiliser les connaissances sémantiques pour émettre des contraintes plus fortes que des contraintes exprimées sur base des seules connaissances syntaxiques. Il s'en suit que la structure sémantique de l'énoncé ne pourra pas être simplement construite à partir de sa structure syntaxique une fois l'analyse terminée et grâce à quelques règles de transformation mais devra être cons-

truite, parallèlement à la structure syntaxique, au fur et à mesure de l'avancement de l'analyse. Les deux représentations de l'énoncé vont devoir entretenir des liens étroits. C'est le rôle des règles de correspondance d'explicitier ces liens.

3.2 Définition des règles de correspondance

Les règles de correspondance sont des règles de production du type condition-action où les actions décrivent pour la plupart l'affectation de fonctions sémantiques à certains éléments de l'énoncé (par exemple l'affectation d'un cas à un groupe nominal) mais également les dimensions modales, temporelles et d'aspect de l'énoncé. Les conditions déterminant les actions s'expriment en termes d'informations sur les éléments de l'énoncé à analyser. Ces informations, éventuellement incomplètes, peuvent être :

- des informations sur la fonction syntaxique des éléments ; elles sont dépendantes de la syntaxe et se reflètent dans la structure syntaxique de l'énoncé ;
- des informations sur la catégorie syntaxique des éléments ; elles sont dépendantes de la syntaxe et se reflètent dans la structure syntaxique de l'énoncé ;
- des informations sur la primitive ou les cas liés aux éléments ; elles sont fournies par la sémantique et se reflètent dans la structure sémantique de l'énoncé ;
- des informations sur les contraintes sémantiques imposées aux éléments et exprimées en termes de traits sémantiques ; elles sont fournies par la sémantique et se reflètent dans la structure sémantique de l'énoncé.

3.3 Les règles de correspondance du sous-langage de notre application test

Nous distinguons essentiellement trois types de règles :

- les règles d'affectation de cas centraux, leur but est d'affecter un cas central à des groupes nominaux sur base de la fonction syntaxique du groupe nominal, de sa catégorie syntaxique, de son marqueur de cas, de ses traits sémantiques, de la voix du prédicat auquel il est lié, de la primitive que réalise ce prédicat et, dans certains cas, de la polarité du prédicat ;
- les règles d'affectation de cas périphériques, leur but est d'affecter un cas périphérique aux termes d'un prédicat ou à une formule (dans le cas d'une subordonnée) sur base de la fonction syntaxique du candidat, de sa catégorie syntaxique, de son marqueur de cas, de ses traits sémantiques ainsi que de la primitive que réalise le prédicat auquel il est lié ;

- les règles concernant les prédicats, il s'agit de règles qui déterminent les modalités de l'énoncé (temps, forme, type etc.). Au stade actuel de notre travail, nous n'avons pas encore abordé ces problèmes. Des éléments de réflexion sur l'intégration de ces dimensions dans un système de dialogue oral homme-machine peuvent être trouvés dans (Deville, 1989).

Pour illustrer ceci, fournissons quelques exemples de règles. Les deux premières règles sont des règles d'affectation de cas centraux, la troisième est une règle d'affectation de cas périphérique :

si un « nom », un « pronom sujet » ou un « nom propre »
réalise syntaxiquement un « groupe sujet »
et s'il n'a pas de marqueur de cas
et s'il possède le trait sémantique +animé
et si la voix de l'énoncé est l'actif
et si le prédicat auquel il est lié réalise un « acte »
et si la polarité de ce prédicat n'est pas négative
alors on lui affecte le cas « agent »

si un « pronom objet » réalise
syntaxiquement un « complément d'objet indirect »
et s'il n'a pas de marqueur de cas
et s'il possède le trait sémantique +animé
et si le prédicat auquel il est lié réalise une
primitive « ECHPROD_3 »
et si la polarité de ce prédicat est positive
alors on lui affecte le cas « bénéficiaire »

si un « adverbe »
réalise syntaxiquement un « groupe complément »
et s'il possède le trait sémantique +manière
et si le prédicat auquel il est lié possède les traits
sémantiques +contrôlé ou +dynamique
alors on lui affecte le cas « manière »

Ces quelques exemples terminent le chapitre consacré aux différents types de connaissances linguistiques utilisées par la composante SYNSEM. Nous allons montrer dans le chapitre suivant que non seulement nous avons tenté de proposer des modèles du langage originaux mais que la façon de les mettre en œuvre apporte elle-aussi des éléments de solution nouveaux.

LA COMPOSANTE SYNTAXICO-SEMANTIQUE DU SYSTEME DIAL

Sur base des connaissances linguistiques que nous venons de présenter, la composante SYNSEM doit procéder à l'analyse des énoncés du locuteur. Le chapitre suivant se propose d'exposer les traitements effectués par la composante SYNSEM en insistant plus particulièrement sur les aspects qui ont été implantés.

1 La composante SYNSEM

Comme nous l'avons signalé au chapitre 7, le rôle de la composante syntaxico-sémantique (SYNSEM) consiste à analyser les énoncés des locuteurs. Son objectif est d'arriver à construire la structure syntaxico-sémantique de ces énoncés en utilisant des connaissances syntaxiques et sémantiques.

1.1 Différents types d'analyse

Les points de départ pour l'analyse des énoncés par SYNSEM sont des informations qui sont fournies par les composantes LEX et DIALOG. Ces informations peuvent être :

- des mots bien reconnus par la composante LEX ; il s'agit de mots que LEX détecte sur l'énoncé d'un locuteur de sa propre initiative, soit à cause de leurs propriétés phonologiques bien prononcées, soit à cause de leurs caractéristiques syntaxico-sémantiques ou pragmatiques fortement contraignantes ;

- des hypothèses syntaxico-sémantiques émises par la composante DIALOG en fonction de l'historique du dialogue.

D'un côté, SYNSEM part donc de mots autour desquels elle essaie de construire des structures syntaxico-sémantiques ; il s'agit pour elle de procéder à une analyse ascendante. De l'autre, elle part de structures syntaxico-sémantiques et essaie de voir comment ces structures sont réalisées sur l'énoncé ; il s'agit d'une analyse descendante. Dans les deux cas, elle tente de compléter les structures par un mécanisme d'hypothèse-test en émettant des hypothèses de type mot vers la composante LEX qui va soit les confirmer soit les infirmer sur base de l'énoncé du locuteur. Les résultats de l'analyse sont communiqués à la composante DIALOG sous forme de structures syntaxico-sémantiques.

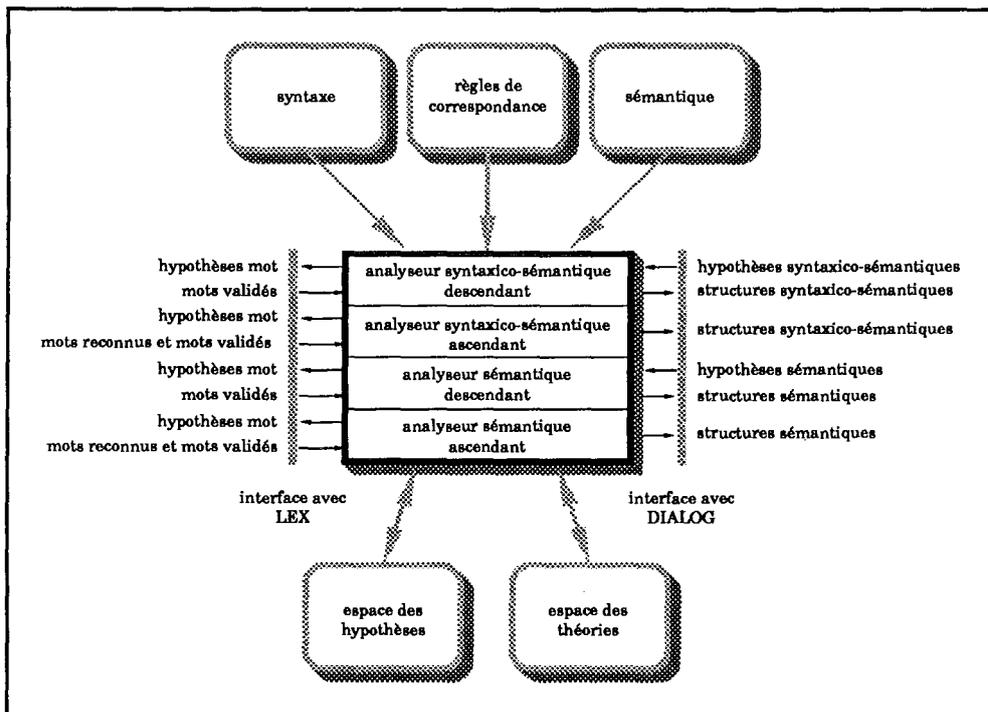


fig. 9.1 : Les différents types d'analyse

Par conséquent, afin de pouvoir remplir pleinement sa tâche, la composante SYNSEM doit intégrer des mécanismes d'analyse descendants aussi bien que des mécanismes d'analyse ascendants.

Un autre aspect de l'analyse des énoncés est lié à la nature des informations mises en œuvre par la composante SYNSEM :

- L'avantage majeur des connaissances syntaxiques est qu'elles sont fortement contraignantes. Elles permettent en particulier d'exprimer des contraintes positionnelles sur les mots et par là d'accélérer la tâche de validation des hypothèses mot par la composante LEX. Un des problèmes auxquels une analyse syntaxique risque cependant toujours d'être confrontée est celui de l'agrammaticalité syntaxique d'un certain nombre d'énoncés.

- Toutefois, des énoncés agrammaticaux, d'un point de vue syntaxique, peuvent toujours véhiculer des informations qui, d'un point de vue sémantique, sont tout à fait cohérentes. Notre objectif n'étant pas de juger de la grammaticalité ou de l'agrammaticalité des énoncés des locuteurs mais de leur fournir, si possible, des réponses, il nous a semblé utile, dans ce cas, de renoncer à l'analyse syntaxique complète de l'énoncé et de n'effectuer que l'analyse sémantique. Il est évident que la validation des hypothèses mot par mot par la composante LEX sera plus fastidieuse dans la mesure où les seules contraintes positionnelles que la composante SYNSEM pourra alors exprimer au sujet de ces hypothèses seront liées aux parties d'énoncé déjà reconnues.

En conséquence, il apparaît que la composante SYNSEM doit être capable de mener une analyse syntaxico-sémantique mais, lorsqu'un énoncé s'avère syntaxiquement incorrect, qu'elle doit également être capable de ne s'appuyer que sur les connaissances sémantiques pour mener à terme son analyse.

Ces diverses considérations nous ont amené à envisager différents types d'analyseurs dans la composante SYNSEM, à savoir : un analyseur syntaxico-sémantique descendant, un analyseur syntaxico-sémantique ascendant, un analyseur sémantique descendant et, enfin, un analyseur sémantique ascendant. Les analyseurs ainsi que les informations qu'ils échangent avec les autres composantes sont présentés sur la figure 9.1.

1.2 Quelques concepts de base

Avant d'aborder les traitements de la composante SYNSEM, il nous a semblé indispensable de préciser auparavant certaines notions fondamentales auxquelles nous ferons référence par la suite ; il s'agit des notions « d'espace des théories » et « d'espace des hypothèses ».

1.2.1 Espace des théories

La notion « d'espace des théories » repose, elle, sur la notion de « théorie ».

Définition

Nous appelons théorie une représentation syntaxico-sémantique ou sémantique recouvrant tout ou une partie de l'énoncé d'un locuteur que la composante syntaxico-sémantique possède à un instant donné de l'analyse de cet énoncé.

L'analyse de la parole s'avère être un processus hautement non déterministe. Par conséquent la composante SYNSEM ne peut se contenter de développer une représentation syntaxico-sémantique unique d'un énoncé mais elle doit au contraire pouvoir développer plusieurs représentations con-

currentes, éventuellement incompatibles, d'un même énoncé. Ces différentes représentations sont regroupées dans une structure d'informations appelée « espace des théories ».

« L'espace des théories » contient toutes les théories produites par les différents analyseurs depuis le début de l'analyse de l'énoncé d'un locuteur jusqu'à un instant donné de cette analyse. Ces théories entretiennent des relations de filiation liées au mécanisme d'analyse de la composante, mécanisme qui peut se résumer de la façon suivante : la composante SYNSEM puise dans « l'espace des théories » des théories qu'elle désire compléter, construit, à partir de ces théories appelées théories mères, de nouvelles théories appelées théories filles et tente de valider ces dernières en émettant des hypothèses vers la composante lexicale.

Définition

Nous appelons espace des théories l'ensemble des théories que la composante syntaxico-sémantique a développé depuis le début de l'analyse de l'énoncé d'un locuteur jusqu'à un instant donné de cette analyse ainsi que les relations de filiation entre ces théories.

La figure 9.2 présente un exemple de théorie.

1.2.2 Le type d'une théorie

De façon générale, les théories de l'espace des théories ont été construites chacune par un analyseur d'un type particulier (syntaxico-sémantique ou sémantique, descendant ou ascendant) et ne peuvent être complétées que par le même type d'analyse. Par conséquent, toute théorie est d'un certain type, celui de l'analyseur qui a servi à la construire :

- syntaxico-sémantique descendant,
- syntaxico-sémantique ascendant,
- sémantique descendant
- ou sémantique ascendant.

Les seules exceptions sont, pour l'instant, les théories qu'on pourrait qualifier de théories initiales et qui sont générées sur base des hypothèses des composantes lexicale et pragmatique et non pas par un mécanisme d'analyse quelconque. En particulier, les hypothèses de DIALOG engendrent des théories initiales syntaxico-sémantiques (ou sémantiques) descendantes, les hypothèses de LEX des théories initiales syntaxico-sémantiques (ou sémantiques) ascendantes.

Lorsqu'une théorie est sélectionnée par la composante SYNSEM dans l'espace des théories afin d'être développée, SYNSEM active l'analyseur compétent suivant le type de cette théorie.

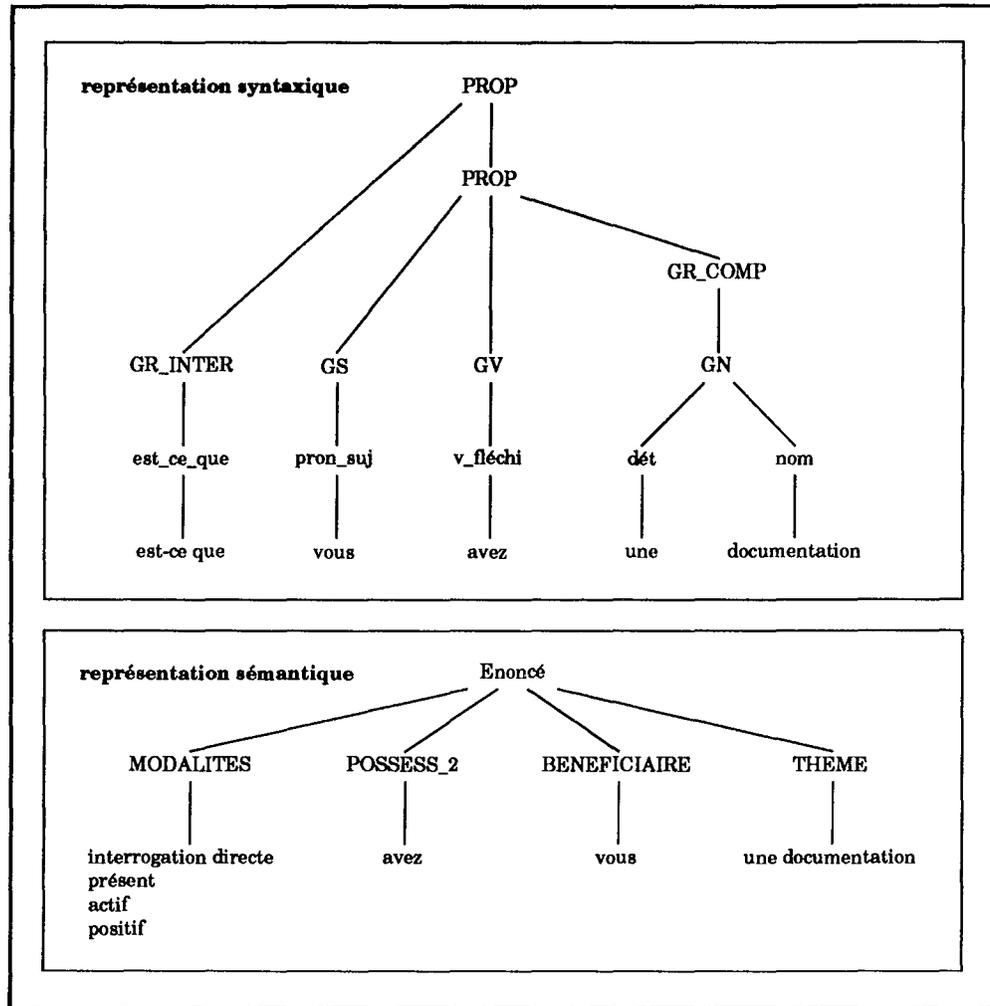


fig. 9.2 : Exemple de théorie construite sur base de l'énoncé
 « Est-ce que vous avez une documentation »

Dans un premier temps, l'espace des théories apparaît donc comme une forêt, au sens de la théorie des graphes, où chaque arbre représente un ensemble de théories qui entretiennent des relations de filiation (voir figure 9.3). La racine de chaque arbre correspond à une théorie initiale créée sur base d'une hypothèse émise par une autre composante, une hypothèse syntaxico-sémantique de la composante DIALOG ou un mot bien reconnu par la composante LEX. Les autres nœuds de l'arbre correspondent à des théories construites sur base d'une théorie mère, soit par mise en œuvre des connaissances linguistiques, soit par validation d'hypothèses de type mot.

Cependant, nous avons encore envisagé d'autres mécanismes de génération de théories que les analyseurs, notamment des mécanismes qui permettent à des théories d'engendrer des théories d'un type différent du leur.

Le premier mécanisme est celui de la « transformation de théories ». En effet, nous avons montré précédemment que le résultat attendu de la composante SYNSEM était en fait la structure sémantique d'un énoncé. Si jamais une théorie syntaxico-sémantique devait être abandonnée à cause de l'agrammaticalité syntaxique de l'énoncé qu'elle tente de structurer, il pourrait être intéressant de continuer l'analyse uniquement sur base de la

structure sémantique déjà reconnue. Une théorie syntaxico-sémantique descendante (respectivement ascendante) peut donc engendrer une théorie sémantique descendante (respectivement ascendante) par simple transformation de la théorie syntaxico-sémantique en théorie sémantique.

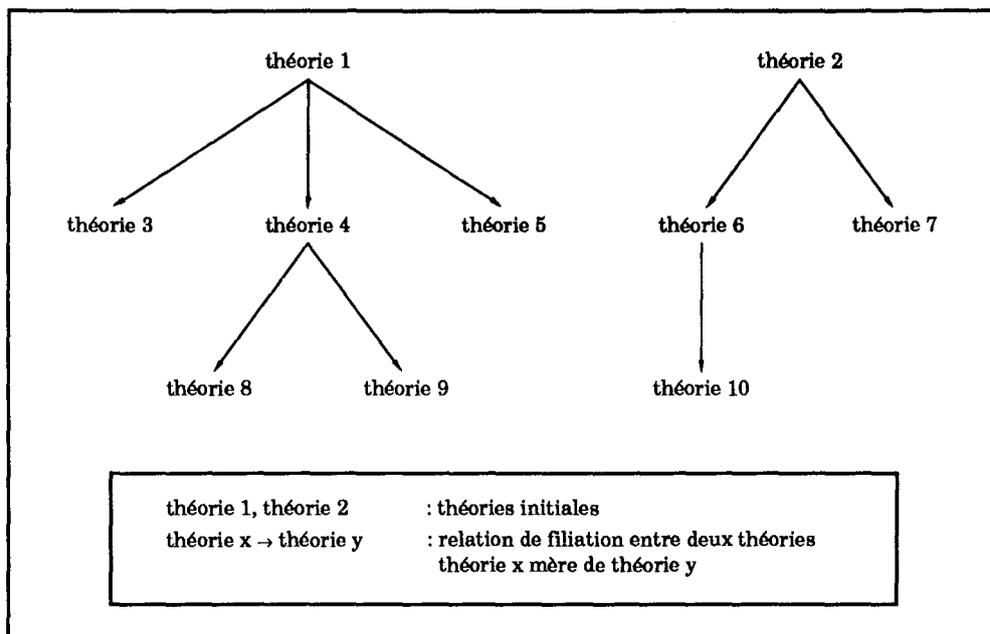


fig. 9.3 : Les relations de filiation entre théories

Le deuxième mécanisme est celui de la « fusion de théories ». Il est possible qu'à un moment donné de l'analyse, une première théorie ascendante décrive une partie d'un énoncé non encore décrite par une deuxième théorie descendante ou ascendante mais que les structures de la première puissent être parfaitement intégrées dans les structures de la deuxième. Nous parlons alors de la fusion de deux théories.

Les différents mécanismes de génération de théories filles à partir de théories mères en fonction du type de celles-ci sont illustrés sur le schéma de la figure 9.4. Dès lors, l'espace des théories apparaît plutôt comme un réseau hiérarchique tel qu'on peut le voir sur la figure 9.5.

1.2.3 Etat d'une théorie

Si d'un côté les mécanismes de génération de théories reflètent le caractère dynamique de l'espace des théories dans son entièreté, de l'autre, les théories considérées de façon isolée possèdent également certains aspects dynamiques.

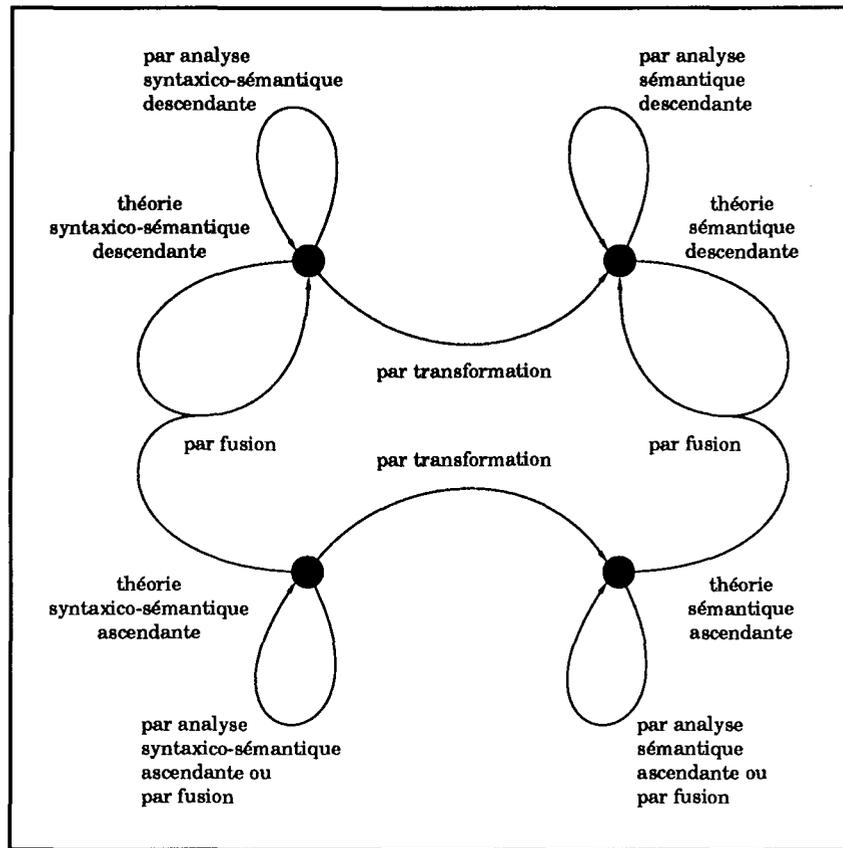


fig. 9.4 : Les différents mécanismes de génération de théories

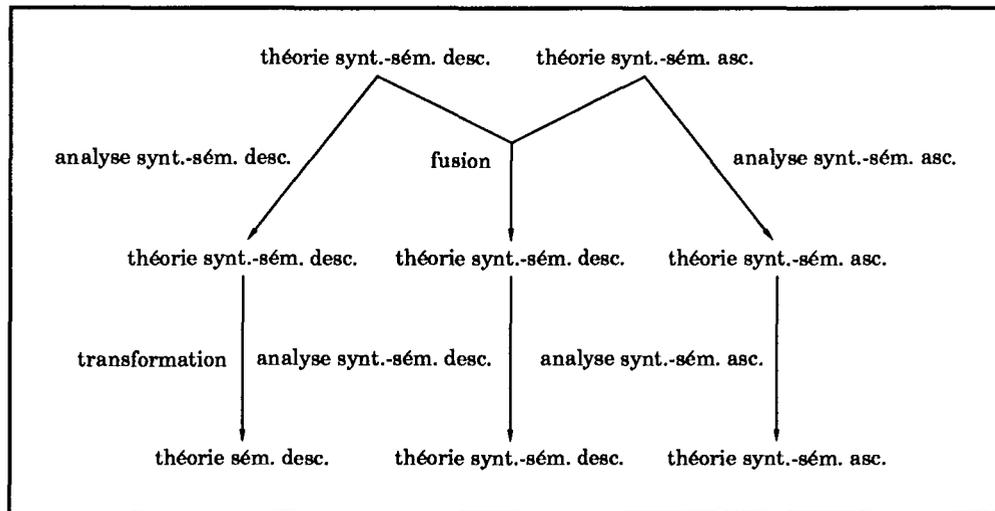


fig. 9.5 : Un espace de théories

Lorsqu'une théorie est créée sur base d'une hypothèse des composantes lexicale ou dialogue, elle est immédiatement prête à être développée. De même, chaque théorie construite sur base d'une théorie existante et n'ayant pas donné lieu à l'émission d'une hypothèse mot est également prête à être développée. Par contre, une théorie construite sur base d'une théorie existante et ayant donné lieu à l'émission d'une hypothèse mot est une théorie en attente de la validation de cette hypothèse. Si l'hypothèse qu'elle a servie à

émettre est validée, la théorie reste en attente d'autres validations éventuelles, si par contre l'hypothèse est infirmée, la théorie se bloque. De la même façon, une théorie prête à être développée qui ne peut pas engendrer de théorie fille en accord avec les connaissances linguistiques se bloque elle-aussi. Par contre, une théorie qui structure entièrement ou en partie un énoncé du locuteur et qui donne lieu à une représentation linguistique complète et en accord avec les connaissances linguistiques devient une théorie complète. Ces considérations nous ont amené à distinguer différents états pour une théorie :

- l'état « prêt » ;
- l'état « en attente » ;
- l'état « bloqué » ;
- et enfin l'état « complet ».

Les différentes transitions d'état possibles sont montrées sur le schéma de la figure 9.6.

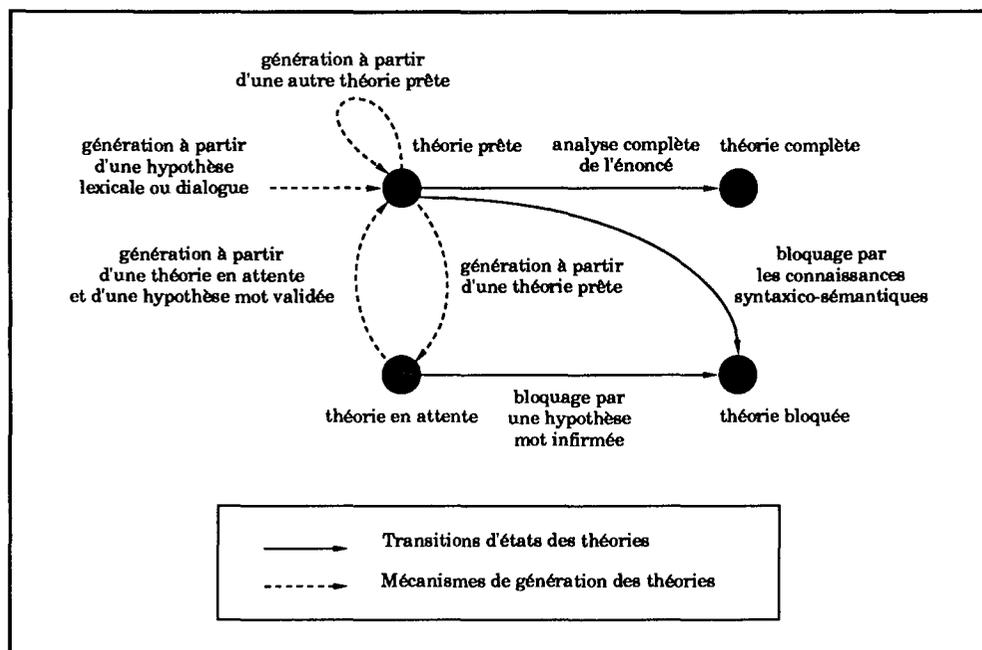


fig. 9.6 : Transitions d'état possibles d'une théorie

1.2.4 Espace des hypothèses

En plus de l'espace des théories, la composante SYNSEM gère, pour des raisons d'efficacité, un « espace des hypothèses mot ». Commençons par définir ce que nous entendons par « hypothèse mot ».

Définition

Nous appelons hypothèse mot une requête de validation de la composante syntaxico-sémantique par laquelle celle-ci demande à la composante lexicale de vérifier la présence éventuelle de mots répondant à certaines contraintes syntaxico-sémantiques ou sémantiques dans l'énoncé d'un locuteur.

Dans la suite, nous parlerons plus simplement d'hypothèses et d'espace des hypothèses.

Une hypothèse est émise par la composante SYNSEM sur base d'une théorie dans l'état « prêt » et de connaissances syntaxico-sémantiques ou sémantiques. Elle donne lieu à la génération d'une théorie dans l'état « en attente ».

Dans la mesure où SYNSEM développe plusieurs théories concurrentes de façon tout à fait indépendante, il se peut très bien que différentes théories soient à l'origine d'une même hypothèse. Dès lors, il semble attrayant de ne demander qu'une seule fois à la composante LEX la validation d'une telle hypothèse. Ceci présuppose la gestion des hypothèses émises par SYNSEM, gestion qui devient possible grâce à l'espace des hypothèses. D'un côté, lorsqu'une hypothèse est validée, il faut pouvoir retrouver toutes les théories en attente de cette validation. De l'autre, lorsqu'une hypothèse peut être émise sur base d'une théorie, il faut pouvoir vérifier si cette hypothèse n'a pas déjà été émise, ou même validée, et traiter la théorie émettrice en conséquence.

Définition

Nous appelons espace des hypothèses l'ensemble des hypothèses émises par la composante syntaxico-sémantique jusqu'à un instant donné de l'analyse de l'énoncé d'un locuteur, ainsi que leurs relations avec les théories en attente de leur validation et leurs relations avec des validations éventuelles déjà effectuées par la composante lexicale jusqu'à cet instant de l'analyse.

1.3 L'analyse d'un énoncé dans SYNSEM

Les deux notions d'espace des théories et d'espace des hypothèses étant définies, nous pouvons aborder maintenant le fonctionnement de la composante SYNSEM.

La structure générale de la composante SYNSEM est présentée sur la figure 9.7. L'exécution de SYNSEM est lancée par la composante DIALOG, le module responsable de la gestion du dialogue. Une fois l'analyse lancée, DIALOG n'intervient plus dans le déroulement de SYNSEM.

SYNSEM est constituée essentiellement d'une boucle. Elle effectue les traitements aussi longtemps qu'elle décide de ne pas abandonner l'analyse. SYNSEM peut par exemple décider de s'arrêter s'il existe au moins une

théorie complète recouvrant tout l'énoncé du locuteur dans l'espace des théories. Avant de suspendre son exécution, elle communique les théories qu'elle a réussi à construire à la composante DIALOG.

Les tâches qui incombent à SYNSEM sont le traitement des informations en provenance des autres composantes et la génération de nouvelles théories. Le traitement des informations communiquées par les autres composantes pose un problème dans la mesure où l'arrivée de ces informations est complètement asynchrone au fonctionnement de SYNSEM. En effet, SYNSEM n'a aucun moyen de savoir exactement à quel moment par exemple la composante LEX a réussi à détecter un mot sur l'énoncé. Deux solutions à ce problème sont alors envisageables :

- Les informations en provenance des autres composantes sont prises en compte par des routines d'interruption qui intègrent les informations reçues dans l'espace des théories et, si nécessaire, dans l'espace des hypothèses. L'inconvénient de cette approche est qu'il est difficile de garantir la cohérence des structures d'informations que sont l'espace des théories et l'espace des hypothèses. Il faut alors imaginer un mécanisme complexe de gestion des interruptions (priorités entre les différents types d'interruptions possibles, masquage des interruptions etc.). C'est la raison pour laquelle nous avons préféré adopter la deuxième solution.
- Les informations en provenance des autres composantes sont prises en compte par un mécanisme de scrutation. A des instants précis de l'analyse, la composante SYNSEM décide de voir si des informations d'autres composantes sont disponibles et, si oui, de les traiter. Dès lors, la question de la cohérence de l'espace des théories et de l'espace des hypothèses se pose en termes beaucoup plus simples et permet d'évacuer tous les problèmes de gestion des interruptions.

Dans l'approche que nous avons choisie, SYNSEM distingue les trois types d'informations qu'elle peut recevoir des autres composantes, à savoir des hypothèses dialogue, des mots bien reconnus et des mots validés. Si des hypothèses dialogue sont disponibles, elle crée des théories initiales descendantes sur base de ces hypothèses, si des mots bien reconnus sont disponibles, elle crée des théories initiales ascendantes sur base de ces hypothèses et, finalement, si des mots validés sont disponibles, elle recherche les théories qui ont émises les hypothèses correspondantes et crée, sur base de ces théories et des mots validés, des théories filles.

L'analyse proprement dite consiste alors à sélectionner, dans l'espace des théories, un sous-ensemble de l'ensemble des théories. Toutes les théories de ce sous-ensemble doivent être traitées par un analyseur qui correspond à leur type. Cet analyseur tente, sur base de la théorie et des connaissances linguistiques, de construire une nouvelle théorie et, éventuellement, d'émettre une hypothèse.

Enfin, si toutes les théories sélectionnées ont été traitées par leur analyseur correspondant, la composante SYNSEM recherche dans l'espace des théories les théories qui pourraient être fusionnées ou qui devraient être transformées, après quoi le cycle des traitements de SYNSEM peut reprendre.

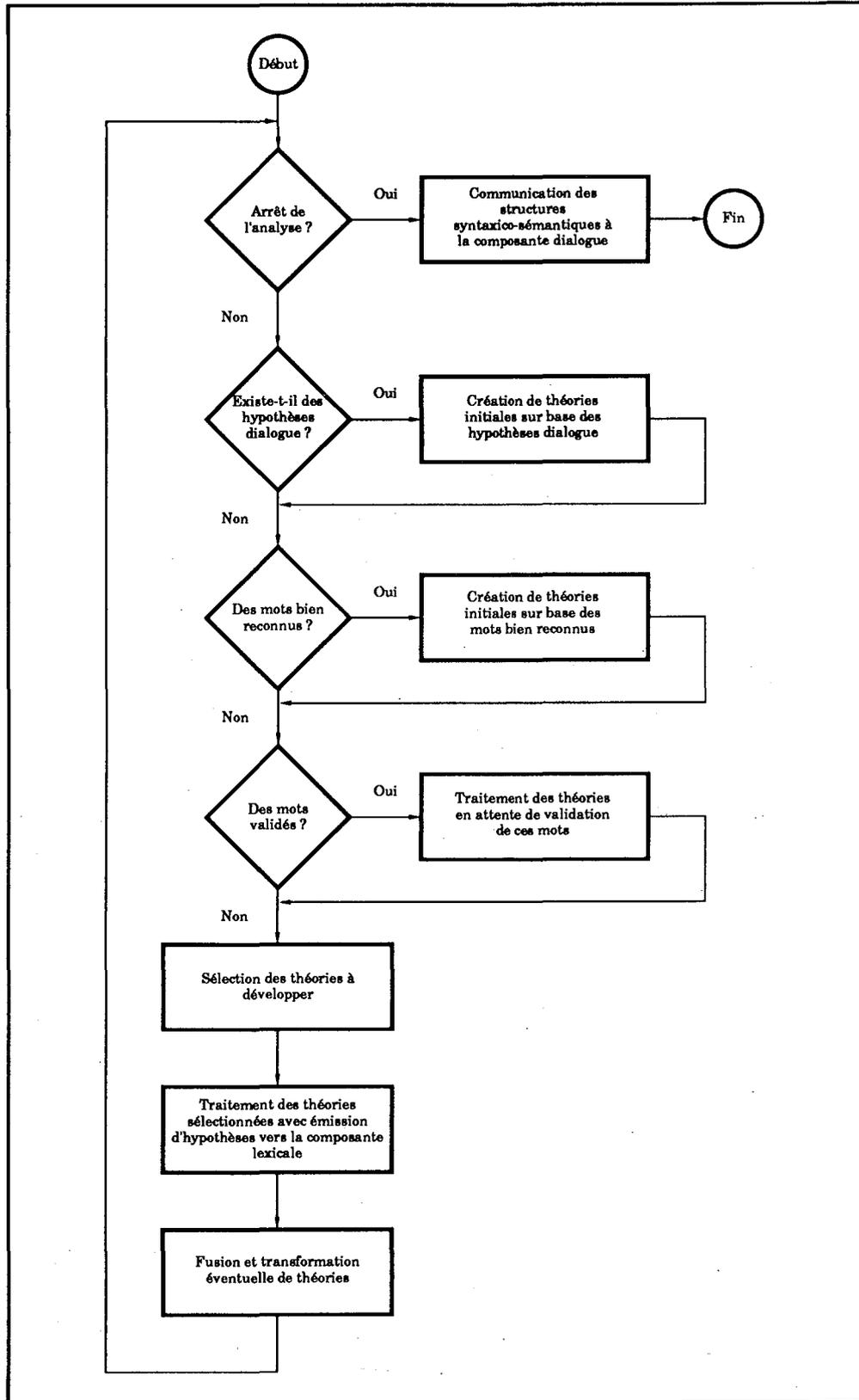


fig. 9.7 : Structure fondamentale de la composante SYNSEM

Ce que nous venons de présenter dans ce paragraphe est le fonctionnement de la composante SYNSEM lorsque celle-ci sera intégrée dans le système DIAL. Actuellement, nous n'en sommes pas encore à ce stade, ce qui

signifie que d'un côté tous les aspects de SYNSEM n'ont pas été implantés mais que de l'autre, nous avons dû implanter des fonctions qui, dans une version intégrée de SYNSEM, n'ont plus de véritable utilité. Il s'agit :

- d'outils autour de la composante SYNSEM destinés à faciliter la conception et la création de bases de connaissances linguistiques ainsi que l'évaluation des résultats de SYNSEM ;
- des fonctions qui font de la composante SYNSEM un système ouvert, paramétré par les connaissances et les stratégies, et offrant diverses facilités de suivi des analyses.

Dans les paragraphes qui vont suivre, nous allons nous intéresser plus particulièrement aux parties de SYNSEM qui ont été implantées. Dans un premier temps nous allons présenter les outils qui ont été réalisés autour de la composante SYNSEM. Ensuite, nous allons décrire la composante SYNSEM proprement dite dans son état actuel. Précisons encore que la composante SYNSEM ainsi que tous les outils annexes ont été réalisés en langage C sous le système d'exploitation UNIX.

2 Autour de SYNSEM

2.1 Outils d'aide à l'étude de corpus linguistiques

Notre approche dans le cadre du développement du système DIAL a été une approche essentiellement empirique. Pour la définition des modèles linguistiques de DIAL, nous nous sommes basé sur des théories linguistiques confirmées que nous avons adaptées aux besoins du problème particulier que constitue la communication orale homme-machine finalisée. Ces modèles ont ensuite été raffinés sur base de deux corpus de dialogues oraux finalisés faisant intervenir des interlocuteurs humains et enregistrés en situation réelle. Les conditions de réalisation ainsi que les objectifs détaillés des deux corpus sont décrits dans l'annexe A. Rappelons que le premier a été enregistré au CRIN (Roussanaly et al., 1986) tandis que le second a été enregistré au Ministère de l'Emploi et du Travail à Bruxelles (Deville, 1987).

La réalisation d'un certain nombre d'outils a été indispensable pour rendre possible l'étude de ces corpus. Nous avons notamment transcrit ces corpus sur support informatique, ce qui nous a amené, en vue de traitements automatiques ultérieurs des corpus, à définir une syntaxe de transcription rigoureuse. Les conventions de transcription complètes peuvent être trouvées en annexe A. A titre d'illustration, nous fournissons ci-dessous un exemple d'une telle transcription :

TR:1.3.

```
<1> Renseignements administratifs. |
| <1> Allo bonjour madame, ..euh.. ma
| carte d'identit( est p(rim(e au
```

<2> Oui.		premier juillet soixante seize ..euh.. quatre vingt six plut [^] t donc ..euh.. il faudrait que je la remplace et je voudrais savoir o ⁺ je dois me renseigner.
<3> Pour obtenir une carte d'identit(vous vous adressez soit au commissariat de police ..euh.. de votre domicile ou @ la mairie de votre commune.		<2> Oui. <3> A la mairie.
<4> Oui, vous souhaitez conna~tre les pi)ces @ fournir?		<4> Oui.
<5> Donc vous vous <H> munissez de votre carte p(rim(e deux photos d'identit(alors les photos d'ite <H> d'identit(c'est ..euh.. trois ..euh.. trois virgule cinq par quatre centim)tres de face et t#te nue hein.		<5> Hem, hem. <6> D'accord.
<6> Et puis un timbre fiscal.		<7> Et le timbre fiscal il est de combien? Quelle somme?
<7> Cent quinze francs.		<8> Cent quinze francs.
<8> Oui.		<9> D'accord. Je vous remercie.
<9> D(sirez vous autre chose?		<10> Non, \$a ira, merci. Au revoir.
<10> [Vous pouvez me donner vos coordonn(es s'il vous pla~t?)		<11> [Oui vingt sept z(ro un ..euh.. au CRIN.]
<11> [Le nom, le nom s'il vous pla~t.]		<12> [Mademoiselle F.]
<12> [Merci beaucoup.]		<13> [De rien.] Au revoir.
<13> Au revoir.		

Certains programmes que nous avons réalisés exploitent directement cette transcription et calculent des résultats bruts sur les corpus tels que des statistiques sur le corpus complet, les énoncés des experts, les énoncés des locuteurs, le lexique utilisé dans le corpus etc.. Nous renvoyons le lecteur à (Deville, 1987 ; Roussanaly et al., 1986) pour les résultats détaillés de l'étude statistique des deux corpus. Afin d'illustrer la représentativité du corpus enregistré au CRIN, quelques résultats sont fournis en annexe A.

D'autres programmes nécessitent des traitements manuels supplémentaires. C'est le cas de LADI (Logiciel d'Aide au Dépouillement de corpus Interactif) qui permet de rechercher de manière ciblée des interventions dans un corpus suivant divers critères, syntaxiques, sémantiques, contextuels etc. (Mousel et Roussanaly, 1988). Pour cela, les interventions des corpus ont du être classifiées manuellement en fonction de ces critères. Les deux corpus ont ainsi été classifiés une première fois suivant des critères syntaxiques, une deuxième fois suivant des critères sémantiques (Deville, 1989 ; Deville et al., 1988). La classification donne lieu à un texte respectant une syntaxe déterminée et qui, avec le corpus, est ensuite utilisée par LADI pour permettre à un utilisateur la consultation interactive des interventions du corpus. Les objectifs ainsi que l'utilisation de LADI sont décrits en détail en annexe B. La figure 9.8 montre la copie d'un écran présenté par LADI à l'utilisateur au cours d'une session de consultation. Les interventions

recherchées dans l'exemple étaient les interventions des locuteurs dialoguant avec l'expert numéro un, dont le sujet était la carte d'identité et dont le type était une question portant sur un coût. LADI signale à l'utilisateur qu'il a trouvé une seule intervention répondant aux critères et précise qu'il s'agit de la septième intervention du locuteur qui était la troisième personne à dialoguer avec l'expert numéro un et que cette intervention concerne le coût d'une carte d'identité. L'intervention elle-même est affichée : « Et le timbre fiscal il est de combien ? Quelle somme ? ».

```

-----
|                                CONSULTATION DE CORPUS                                |
-----
                Voulez-vous continuer la selection? (O/N) o
-----
Numero de l'expert      : 1
Numero de la transaction :
Numero de l'intervention :
Sujet de l'intervention : CID
Type de l'intervention  : COUT
Kwic index              :
-----
Expert: 1, Transaction: 3, Intervention: 7, Sujet: CID Type: COUT
                Nombre d'onces restant a consulter: 0
-----
Et le timbre fiscal il est de combien? Quelle somme?

-----
                Suivant? (O/N) ■
-----

```

fig. 9.8 : Exemple de consultation de corpus avec LADI

2.2 Outils de création de bases de connaissances linguistiques

D'autres outils qui se sont révélés nécessaires sont les outils de création de bases de connaissances linguistiques (Mousel, 1986a). En effet, notre approche du développement de DIAL est une approche incrémentale, ce qui signifie qu'il y a peu de chances que les connaissances linguistiques dont dispose le système DIAL à l'heure actuelle soient dans leur état définitif. Ces connaissances doivent fort probablement encore être affinées. De plus, au vu de nos objectifs, DIAL doit pouvoir être porté vers diverses familles d'applications différentes avec un minimum d'efforts. C'est ce qui nous a amené à considérer les connaissances linguistiques comme un paramètre au système. Dès lors, il faut pouvoir facilement modifier les connaissances de DIAL. Pour garantir cette facilité, les connaissances linguistiques sont décrites par le concepteur du système sous la forme d'une « représentation externe » aisément lisible mais qui doit obéir à une syntaxe précise. Elle est créée par le concepteur à l'aide d'un éditeur de texte quelconque et est ensuite

transformée par un « compilateur de connaissances » en une « représentation interne » qui, elle, va être traitée par la composante SYNSEM. Ainsi, la modification d'une base de connaissances revient à modifier la représentation externe à l'aide d'un éditeur de texte et à recompiler la représentation externe modifiée à l'aide du compilateur de connaissances.

A l'heure actuelle, deux compilateurs de ce type ont été réalisés, un compilateur de connaissances syntaxiques, le programme CRERES, ainsi qu'un compilateur de connaissances sémantiques, le programme CRESEM. Les aspects techniques liés à ces programmes sont décrits dans (Mousel, 1985 ; Mousel, 1986b ; Mousel, 1988). Les objectifs, la syntaxe des formes externes ainsi que l'utilisation de ces programmes sont exposés en détail en annexe D et F. A titre d'illustration, on trouvera ci-dessous deux exemples de description externe, celle du RNP « GN » (Groupe Nominal) ainsi que celle de la configuration de cas de la primitive « echprod_3 ». Tous les deux ont déjà été présentés en exemple au chapitre précédent.

Description externe du RNP « GN » :

Sous-reseau: GN;

Noeud-d-entree: GN_1;

Noeud-de-sortie: GN_6;

Description-des-branches:

Origine: GN_1;	
Composants: ;	Destination: GN_2;
Origine: GN_2;	
Composants: nom_propre;	Destination: GN_5;
Origine: GN_2;	
Composants: det;	Destination: GN_3;
Origine: GN_2;	
Composants: ;	Destination: GN_3;
Origine: GN_3;	
Composants: CHIF;	Destination: GN_3;
Origine: GN_3;	
Composants: GR_ADJ;	Destination: GN_3;
Origine: GN_3;	
Composants: adj_indf;	Destination: GN_3;
Origine: GN_3;	
Composants: nom;	Destination: GN_4;
Origine: GN_4;	
Composants: GR_ADJ;	Destination: GN_4;
Origine: GN_4;	
Composants: ;	Destination: GN_5;
Origine: GN_4;	
Composants: GP;	Destination: GN_5;
Origine: GN_4;	
Composants: RELAT;	Destination: GN_5;
Origine: GN_5;	
Composants: co_cor;	Destination: GN_2;
Origine: GN_5;	
Composants: ;	Destination: GN_6;

Description externe de la configuration de cas associée à la primitive « echprod_3 » :

echprod_3 (AGENT, THEME, BENEFACTIVE)

Au chapitre 8, les RNP ont été définis comme étant des connaissances de type déclaratif mais avec une composante procédurale, les « nœuds procéduraux ». Il s'agit, dans notre cas, tout simplement de fonctions en langage C, la composante SYNSEM étant, elle-aussi, écrite en langage C. Le concepteur du système se voit assigné la tâche de créer ces fonctions. Leur écriture est cependant un travail assez répétitif, surtout dans les premières phases du développement du système lorsque les connaissances linguistiques ne sont pas encore dans un état d'élaboration très avancé. Très souvent, à ce stade du développement, les tests contextuels dans les nœuds procéduraux sont encore inexistantes. Ce constat est à l'origine des programmes NPR et DEFNPR qui sont des générateurs de nœuds procéduraux en langage C. NPR génère une fonction qui constitue l'interface d'appel entre la composante SYNSEM et les nœuds procéduraux, tandis que DEFNPR génère des nœuds procéduraux élémentaires qui ne contiennent encore aucun test contextuel. Les objectifs ainsi que l'utilisation des deux programmes sont décrits en annexe G. Nous fournissons ci-dessous un exemple de nœud procédural généré par le programme DEFNPR. Un tel nœud procédural se limite à fournir, à la fonction qui l'a appelé, la liste de toutes ses sorties possibles :

```
LENT CHIF_4 (nds)

NDS nds;

{
  LENT liste_sorties;
  liste_sorties = LENTINIT;
  creer_lent(&liste_sorties);
  ajout_lent(&liste_sorties,0);
  ajout_lent(&liste_sorties,1);
  return(liste_sorties);
}
```

L'exemple choisi est celui du nœud procédural CHIF_4 du RNP CHIF (on trouvera la description de ce RNP en annexe C). Ce nœud a deux sorties possibles (numérotées à partir de 0). La fonction associée crée une liste de sorties vide, ajoute les deux sorties à la liste des sorties et fournit cette liste comme résultat.

Le concepteur peut ensuite, quand il le désire, modifier les nœuds procéduraux générés par DEFNPR à l'aide d'un éditeur de texte quelconque. Voici un exemple de nœud procédural modifié :

```
LENT CHIF_4 (nds)

NDS nds;

{
  LENT liste_sorties;
  NDS pere;
  int i;
  liste_sorties = LENTINIT;
  creer_lent(&liste_sorties);
  ajout_lent(&liste_sorties,0);
  pere = ndpere_nds(nds);
  for (i=0;
       (i<long_nds(listef_nds(pere))) &&
       (long_chcar(symb_nds(elen_nds(listef_nds(pere),i))) == 0);
       i++);
  if (i<long_nds(listef_nds(pere))) ajout_lent(&liste_sorties,1);
  return(liste_sorties);
}
```

)

L'exemple correspond au même nœud procédural CHIF_4 mais cette fois-ci avec un test contextuel. La fonction n'ajoute la sortie 1 à la liste des sorties que si le RNP CHIF, dont le parcours est à l'origine de l'activation du nœud procédural CHIF_4, a été réécrit en au moins un symbole non vide. On évite ainsi de parcourir le RNP CHIF en ne produisant que des symboles vides (nous renvoyons le lecteur à l'annexe C pour une description du RNP CHIF).

2.3 Outil de consultation de l'espace des théories

Un dernier outil qui a été réalisé et qui ne fait pas non plus partie de la composante SYNSEM proprement dite est le programme LESP (Liste des ESPaces des théories et des hypothèses). Ce programme reprend des espaces des théories et des hypothèses produits par la composante SYNSEM et les présente sous une forme lisible. Il s'agit là d'un outil de validation et d'évaluation indispensable qui permet de suivre au niveau de détail le plus bas et dans le temps l'évolution des deux structures d'informations gérées par SYNSEM. Grâce à l'étude des espaces des théories et des hypothèses, nous pensons pouvoir affiner les connaissances linguistiques (en détectant les théories douteuses) mais également améliorer les stratégies (en suivant la génération des théories via les relations de filiation).

L'utilisation de cet outil est largement illustrée en annexe H. On trouvera ci-dessous un exemple de théorie mise en forme par le programme LESP¹ :

```
Theorie numero: 1057
=====

Type de la theorie   : Theorie synt-sem. descendante
Etat de la theorie  : Theorie complete
Score de la theorie : 85.865463
Mere1 de la theorie : 973
Mere2 de la theorie : aucune
Filles de la theorie: aucune

Arbre syntaxique de la theorie:
-----

Score de l'arbre syntaxique: 85.865463

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
```

¹ Pour l'instant, la composante SYNSEM ne procède encore qu'à une analyse syntaxique des énoncés, c'est la raison pour laquelle LESP ne met en forme que les arbres syntaxiques associés aux théories.

```

nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
  term.: vide num: - sem: - deb: 27.00 fin: 27.00
  term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
    real.: avez num: - sem: - deb: 27.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: det num: - sem: - deb: 45.00 fin: 56.00
    real.: une num: - sem: - deb: 45.00 fin: 56.00
  term.: nom num: - sem: - deb: 56.00 fin: 100.00
    real.: documentation num: - sem: - deb: 56.00 fin: 100.00
  term.: vide num: - sem: - deb: 100.00 fin: 100.00
  term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00

```

3 Réalisation d'un prototype d'évaluation

Il existe actuellement une version de la composante SYNSEM à l'état de prototype d'évaluation. Dans un premier temps, nous avons commencé par implanter l'analyseur syntaxico-sémantique descendant. La partie syntaxique en est terminée, ce qui nous a permis de passer aux premiers tests et à une première évaluation de la composante. L'objectif des paragraphes suivants est de présenter les caractéristiques essentielles du prototype dans l'état présent de son développement.

3.1 Simulation des interfaces

Le système DIAL a été conçu comme un ensemble de modules coopérants, chaque module constituant en lui-même un système à base de connaissance. Cette façon de procéder nous a permis de développer chacune des composantes du système indépendamment des autres. Les composantes n'ont pas été intégrées dès le début du développement et leurs interfaces de communication sont dès lors simulées par des dialogues à l'écran.

Ceci est vrai en particulier pour le prototype de la composante SYNSEM. Celui-ci ne communique pas directement avec une autre composante mais c'est un opérateur humain qui doit remplir le rôle des autres modules du système. Aussi est-ce lui qui fournit à SYNSEM les hypothèses dialogue, les mots reconnus et les mots validés.

L'inconvénient de cette approche est que, pour que le prototype soit utilisable par un opérateur humain faillible, nous avons dû concevoir une interface utilisateur minimale et charger le prototype de certaines vérifications des informations fournies par l'opérateur. Il va de soi que, dans une version intégrée de la composante, ces fonctions n'auront plus aucun sens et vont disparaître complètement. La figure 9.9 présente la copie d'un écran af-

fiché par le prototype de SYNSEM lors de l'émission d'une hypothèse. Par l'émission de cette hypothèse, SYNSEM demande à LEX de vérifier la présence du mot « est-ce que » sur l'énoncé. Dans ce cas particulier, le terminal de la grammaire « est_ce_que » correspond à un seul mot du lexique, il n'en est pas forcément toujours ainsi. Ce mot doit se trouver dans la plage 0-100 sur l'énoncé (les unités employées importent peu ici). L'analyseur étant descendant, il travaille de la gauche vers la droite. Une contrainte forte sur la position du mot est donc la borne gauche de l'hypothèse ; c'est à cet endroit que le début mot doit pouvoir être détecté sur l'énoncé. La borne droite est une contrainte plus faible, elle n'indique qu'une limite sur l'énoncé que le mot ne devrait pas dépasser.

```

*****
*
*                               ANALYSE SYNTAXICO-SEMANTIQUE
*                               -----
*
*****

*****
*
*                               TRAITEMENT DES THEORIES SELECTIONNEES
*
*****

                                Emission d'une hypothese mot.

Terminal   : est_ce_que
debut     : 0.000000 (contrainte forte)
fin       : 100.000000 (contrainte faible)

                                Pour continuer, tapez <RETURN> █

```

fig. 9.9 : Emission d'une hypothèse par SYNSEM

3.2 Paramétrisation par les connaissances

La paramétrisation par les connaissances est une deuxième caractéristique du prototype. Comme nous l'avons déjà signalé précédemment, il est de plus en plus probable que les connaissances dans l'état actuel soient définitives et en outre, le système DIAL doit pouvoir être adapté à d'autres applications. La composante doit donc pouvoir facilement évoluer dans le temps. C'est la raison pour laquelle nous avons tenté de séparer au maximum les connaissances de leur traitement. Cette facilité va de pair avec les compilateurs de connaissances que nous venons de présenter dans les paragraphes précédents. Le prototype de SYNSEM va utiliser les représentations internes produites par ces compilateurs pour procéder aux analyses des énoncés.

Actuellement, à chaque exécution, le prototype demande à l'opérateur de lui fournir le nom du fichier contenant la représentation interne des connaissances. Il est évident que dans une version opérationnelle du système, les connaissances seront intégrées beaucoup plus au système et ne vont pas devoir être fournies de cette façon à chaque exécution de la composante. La figure 9.10 montre l'écran par lequel SYNSEM demande à l'opérateur le nom du fichier contenant la représentation interne des RNP.

```

*****
*
*          CHARGEMENT DES CONNAISSANCES          *
*          -----                               *
*
*****

*****
*
*          CHARGEMENT DES RESEAUX A NOEUDS PROCEDURAUX          *
*
*****

Quel est le nom du fichier contenant le RNP: █

```

fig. 9.10 : Chargement des RNP par SYNSEM

3.3 Paramétrisation par les stratégies

A certains moments, la composante SYNSEM se voit confrontée à des problèmes de stratégie. Nous avons signalé en effet que la composante devait développer plusieurs théories concurrentes pour un même énoncé. Il se pose alors une première question : quelles sont les théories à développer ? De plus, SYNSEM reçoit des informations des autres composantes et utilise des informations linguistiques pour développer des théories. Il se pose alors une deuxième question : comment ces informations doivent-elles être intégrées dans les structures gérées par SYNSEM ? Ces considérations nous ont fait distinguer deux problèmes de stratégie.

3.3.1 Stratégies de sélection

Lorsque le fonctionnement de l'analyse des énoncés par SYNSEM a été exposé dans le paragraphe 1.3, nous avons dit que SYNSEM allait puiser

dans l'espace des théories un sous-ensemble de théories à développer. Certaines des théories ne peuvent pas, de manière évidente, être candidates à être sélectionnées, ce sont :

- les théories bloquées ; une théorie bloquée s'est en effet révélée être une impasse pour l'analyse, il ne sert donc à rien de la sélectionner ;
- les théories complètes ; les théories complètes donnent lieu à une représentation linguistique complète et en accord avec les connaissances linguistiques de l'énoncé et ne sauraient par conséquent être développées davantage ;
- les théories en attente ; elles sont en attente de la validation de mots et, par leur nature, sont inaptes à être développées ;
- et enfin les théories prêtes ayant des théories filles ; elles ont déjà été développées et ne peuvent plus apporter d'éléments nouveaux à l'analyse.

Les seules théories à sélectionner sont dès lors les théories prêtes qui n'ont pas encore de théories filles. Mais faut-il les sélectionner toutes, on risque alors l'explosion combinatoire, ou seulement quelques-unes, il faut alors disposer de critères de sélection. Les théories ont ainsi reçu un attribut supplémentaire qui permet d'appliquer des critères de sélection : un score de vraisemblance. Ce score permet à SYNSEM de sélectionner dans l'espace des théories des théories candidates en appliquant une « stratégie de sélection » donnée. Dans la mesure où diverses stratégies de sélection sont imaginables et qu'il semble impossible de prévoir à priori laquelle va fournir les meilleurs résultats, nous avons fait des stratégies de sélection un paramètre du système. Les stratégies que nous avons envisagées sont les suivantes :

- une stratégie totale ; elle consiste à sélectionner dans l'espace des théories toutes les théories candidates à être développées ;
- une stratégie des n meilleures d'abord avec n fixe ; elle consiste à sélectionner parmi l'ensemble des théories candidates à être développées de l'espace des théories les n théories avec le score le plus élevé (n est à fixer) ;
- une stratégie des n meilleures d'abord avec n variable ; elle consiste à sélectionner parmi l'ensemble des théories candidates à être développées de l'espace des théories toutes celles dont le score se trouve dans un intervalle défini par le score le plus élevé des théories candidates et une certaine distance x (x est à fixer) ;
- une stratégie des n meilleures d'abord avec n variable mais inférieur à un maximum ; il s'agit d'une combinaison des deux stratégies précédentes ;
- enfin une stratégie du meilleur d'abord ; elle consiste à sélectionner parmi l'ensemble des théories candidates à être développées de l'espace des théories la théorie avec le score le plus élevé.

La figure 9.11 montre l'écran par lequel l'opérateur peut fixer l'intervalle acceptable dans le cas d'une stratégie de sélection des n meilleures d'abord avec n variable.

```

*****
*
*          DEFINITION DES STRATEGIES          *
*          -----                            *
*
*****
*****
*
*          DEFINITION DE LA STRATEGIE DE SELECTION          *
*
*****
*****

Quel est l'intervalle acceptable? (un reel entre 0.0 et 100.0) █

```

fig. 9.11 : Définition de l'intervalle acceptable dans la stratégie de sélection des n meilleures d'abord avec n variable

3.3.2 Stratégies de propagation

La sélection des théories n'est cependant pas la seule opération qui nécessite le recours à une stratégie. L'intégration des informations en provenance des autres composantes ainsi que l'utilisation des informations linguistiques en vue de développer les théories sélectionnées y font appel également. Nous avons regroupé ces stratégies sous le terme de « stratégies de propagation ». Ces stratégies définissent en fait le calcul du score associé aux théories à différents moments de l'analyse :

- lorsque la composante **DIALOG** émet une hypothèse dialogue et que **SYNSEM** a créé une théorie initiale sur base de cette hypothèse, **SYNSEM** doit également déterminer un score pour cette théorie ;
- lorsque la composante **LEX** a reconnu un mot et que **SYNSEM** a créé une théorie initiale sur base du mot bien reconnu, **SYNSEM** doit lui associer un score ;
- de la même manière, lorsque la composante **LEX** a validé une hypothèse mot et que **SYNSEM** a créé une nouvelle théorie sur base de la théorie émettrice et du mot validé, il faut calculer le score de cette nouvelle théorie ;
- enfin, lorsque la composante **SYNSEM** crée une nouvelle théorie sur base d'une théorie sélectionnée et des connaissances linguistiques, il faut encore calculer le score de cette nouvelle théorie.

Dans les deux premiers cas, la solution semble assez facile. Les seules informations dont dispose SYNSEM pour créer une théorie initiale sont celles communiquées par DIALOG, respectivement LEX. Un score de vraisemblance est également associé à ces informations. Il suffit dès lors de propager ce score vers les théories initiales.

Dans les autres cas, la réponse est moins évidente. Dans le cas d'une nouvelle théorie basée sur une théorie émettrice et un mot validé, la composante SYNSEM dispose du score associé à la théorie émettrice et du score associé au mot validé par LEX auquel est également associé un score. Les deux scores doivent être combinés pour fournir le score de la nouvelle théorie. A l'heure actuelle, le prototype de SYNSEM ne tient compte que des seules informations syntaxiques ; par conséquent, une théorie est caractérisée par son seul arbre syntaxique. Le problème revient donc à intégrer, dans un arbre syntaxique auquel est associé un certain score (le score de la théorie émettrice), un nouveau nœud syntaxique auquel est associé un autre score (celui du mot validé). Une stratégie de propagation consiste à affecter un poids déterminé à chacun de ces scores. Nous avons envisagé les pondérations suivantes :

- le score du mot validé et le score de l'arbre syntaxique ont tous les deux le même poids ce qui signifie que le score de la nouvelle théorie est tout simplement la moyenne arithmétique des deux scores ;
- le score du mot validé a un poids de un tandis que le score de la théorie émettrice est pondéré :
 - soit par le nombre de nœuds de l'arbre syntaxique (plus il sera développé, plus il aura de poids),
 - soit par le nombre de nœuds correspondant à des mots validés de l'arbre syntaxique (plus sera validé, plus il aura de poids),
 - soit par le nombre de nœuds correspondant à des non-terminaux ou des terminaux de l'arbre syntaxique (plus il sera développé, plus il aura de poids),

le score de la nouvelle théorie est alors la moyenne pondérée des deux scores.

La figure 9.12 montre l'écran par lequel l'opérateur peut fixer la stratégie de propagation de la composante dans le cas de la validation d'un mot.

Dans le dernier cas, la composante SYNSEM dispose de la théorie sélectionnée à laquelle est associée un score de vraisemblance. Les connaissances linguistiques lui permettent de développer cette théorie. Etant donné qu'actuellement, le prototype de SYNSEM ne tient compte que des seules informations syntaxiques, SYNSEM ne peut développer que la représentation syntaxique. Le problème revient donc à ajouter, à un arbre syntaxique auquel est associé un certain score (le score de la théorie sélectionnée), un nouveau nœud syntaxique correspondant à un non-terminal ou un terminal de la grammaire. Deux approches sont possibles :

- on considère que l'ajout de ce nœud n'apporte pas d'information ; dans ce cas le recours à une stratégie de propagation est superflue ;

- on considère, au contraire, que le fait de rajouter un nœud dans la représentation syntaxique modifie celle-ci, ce qui devrait se refléter dans le score de la nouvelle théorie.

```

*****
*
*           DEFINITION DES STRATEGIES           *
*           -----                             *
*
*
*****

*****
*
*           DEFINITION DE LA STRATEGIE DE PROPAGATION *
*
*
*****

Strategie de propagation lors de l'ajout d'un noeud de type
NDSREAL :

Seule la theorie compte                - (1)
Fonction du nombre de noeuds           - (2)
Fonction du nombre de noeuds de type NDSREAL - (3)
Fonction du nombre de noeuds de type NDSTERM ou NDSNTERM - (4)

      Votre choix? (1/2/3/4) █

```

fig. 9.12 : Stratégies de propagation de SYNSEM dans le cas de la validation d'un mot

Le prototype de SYNSEM permet d'évaluer les deux solutions. L'idée qui justifie la deuxième approche est qu'il pourrait être intéressant de disposer d'un mécanisme qui favorise le développement de structures simples par rapport aux structures complexes (complexité en termes du nombre de nœuds d'une structure). Il suffit pour cela d'associer au nœud à ajouter le score le plus bas et ensuite d'appliquer une stratégie de propagation comme dans le cas des mots validés. Nous avons d'ailleurs envisagé les mêmes stratégies de propagation que précédemment :

- le score du nouveau nœud et le score de l'arbre syntaxique ont tous les deux le même poids ce qui signifie que le score de la nouvelle théorie est tout simplement la moyenne arithmétique des deux scores ;
- le score du nouveau nœud a un poids de un tandis que le score de la théorie émettrice est pondéré :
 - soit par le nombre de nœuds de l'arbre syntaxique,
 - soit par le nombre de nœuds correspondant à des mots validés de l'arbre syntaxique,
 - soit par le nombre de nœuds correspondant à des non-terminaux ou des terminaux de l'arbre syntaxique,

le score de la nouvelle théorie est alors la moyenne pondérée des deux scores.

Dans les deux premiers cas, la solution semble assez facile. Les seules informations dont dispose SYNSEM pour créer une théorie initiale sont celles communiquées par DIALOG, respectivement LEX. Un score de vraisemblance est également associé à ces informations. Il suffit dès lors de propager ce score vers les théories initiales.

Dans les autres cas, la réponse est moins évidente. Dans le cas d'une nouvelle théorie basée sur une théorie émettrice et un mot validé, la composante SYNSEM dispose du score associé à la théorie émettrice et du score associé au mot validé par LEX auquel est également associé un score. Les deux scores doivent être combinés pour fournir le score de la nouvelle théorie. A l'heure actuelle, le prototype de SYNSEM ne tient compte que des seules informations syntaxiques ; par conséquent, une théorie est caractérisée par son seul arbre syntaxique. Le problème revient donc à intégrer, dans un arbre syntaxique auquel est associé un certain score (le score de la théorie émettrice), un nouveau nœud syntaxique auquel est associé un autre score (celui du mot validé). Une stratégie de propagation consiste à affecter un poids déterminé à chacun de ces scores. Nous avons envisagé les pondérations suivantes :

- le score du mot validé et le score de l'arbre syntaxique ont tous les deux le même poids ce qui signifie que le score de la nouvelle théorie est tout simplement la moyenne arithmétique des deux scores ;
- le score du mot validé a un poids de un tandis que le score de la théorie émettrice est pondéré :
 - soit par le nombre de nœuds de l'arbre syntaxique (plus il sera développé, plus il aura de poids),
 - soit par le nombre de nœuds correspondant à des mots validés de l'arbre syntaxique (plus sera validé, plus il aura de poids),
 - soit par le nombre de nœuds correspondant à des non-terminaux ou des terminaux de l'arbre syntaxique (plus il sera développé, plus il aura de poids),

le score de la nouvelle théorie est alors la moyenne pondérée des deux scores.

La figure 9.12 montre l'écran par lequel l'opérateur peut fixer la stratégie de propagation de la composante dans le cas de la validation d'un mot.

Dans le dernier cas, la composante SYNSEM dispose de la théorie sélectionnée à laquelle est associée un score de vraisemblance. Les connaissances linguistiques lui permettent de développer cette théorie. Etant donné qu'actuellement, le prototype de SYNSEM ne tient compte que des seules informations syntaxiques, SYNSEM ne peut développer que la représentation syntaxique. Le problème revient donc à ajouter, à un arbre syntaxique auquel est associé un certain score (le score de la théorie sélectionnée), un nouveau nœud syntaxique correspondant à un non-terminal ou un terminal de la grammaire. Deux approches sont possibles :

- on considère que l'ajout de ce nœud n'apporte pas d'information ; dans ce cas le recours à une stratégie de propagation est superflue ;

- on considère, au contraire, que le fait de rajouter un nœud dans la représentation syntaxique modifie celle-ci, ce qui devrait se refléter dans le score de la nouvelle théorie.

```

*****
*
*          DEFINITION DES STRATEGIES          *
*          -----                            *
*
*****
*****
*
*          DEFINITION DE LA STRATEGIE DE PROPAGATION          *
*
*
*
*
Strategie de propagation lors de l'ajout d'un noeud de type
NDSREAL :

Seule la theorie compte                - (1)
Fonction du nombre de noeuds           - (2)
Fonction du nombre de noeuds de type NDSREAL - (3)
Fonction du nombre de noeuds de type NDSTERM ou NDSNTERM - (4)

          Votre choix? (1/2/3/4) █

```

fig. 9.12 : Stratégies de propagation de SYNSEM dans le cas de la validation d'un mot

Le prototype de SYNSEM permet d'évaluer les deux solutions. L'idée qui justifie la deuxième approche est qu'il pourrait être intéressant de disposer d'un mécanisme qui favorise le développement de structures simples par rapport aux structures complexes (complexité en termes du nombre de nœuds d'une structure). Il suffit pour cela d'associer au nœud à ajouter le score le plus bas et ensuite d'appliquer une stratégie de propagation comme dans le cas des mots validés. Nous avons d'ailleurs envisagé les mêmes stratégies de propagation que précédemment :

- le score du nouveau nœud et le score de l'arbre syntaxique ont tous les deux le même poids ce qui signifie que le score de la nouvelle théorie est tout simplement la moyenne arithmétique des deux scores ;
- le score du nouveau nœud a un poids de un tandis que le score de la théorie émettrice est pondéré :
 - soit par le nombre de nœuds de l'arbre syntaxique,
 - soit par le nombre de nœuds correspondant à des mots validés de l'arbre syntaxique,
 - soit par le nombre de nœuds correspondant à des non-terminaux ou des terminaux de l'arbre syntaxique,

le score de la nouvelle théorie est alors la moyenne pondérée des deux scores.

La figure 9.13 montre l'écran par lequel l'opérateur peut fixer la stratégie de propagation de la composante dans le cas de l'ajout d'un nœud correspondant à un non-terminal de la grammaire.

```

*****
*
*                               *
*          DEFINITION DES STRATEGIES          *
*          -----                          *
*
*
*****

*****
*
*          DEFINITION DE LA STRATEGIE DE PROPAGATION          *
*
*
*****

Strategie de propagation lors de l'ajout d'un noeud de type
NDSNTERM:

Seule la theorie compte - (1)
Fonction du nombre de noeuds - (2)
Fonction du nombre de noeuds de type NDSREAL - (3)
Fonction du nombre de noeuds de type NDSTERM ou NDSNTERM - (4)

Votre choix? (1/2/3/4) █

```

fig. 9.13 : Stratégies de propagation de SYNSEM dans le cas de l'ajout d'un nœud correspondant à un non-terminal

Il est évident que la possibilité de paramétrer la composante SYNSEM par les stratégies n'a de sens que dans le contexte de l'évaluation du prototype. Il faudra étudier le comportement des différentes stratégies et, dans la version opérationnelle du système, implanter celle qu'on considère comme la meilleure.

3.4 Facilités

En plus de son caractère ouvert, le prototype de SYNSEM se distingue par certaines options qui rendent plus facile le suivi et l'évaluation de son fonctionnement. De nouveau ces fonctions n'ont de sens que dans le contexte de notre réalisation et seront absentes dans une version opérationnelle.

3.4.1 Abandon et reprise d'une analyse

La première de ces facilités permet à l'utilisateur d'interrompre l'analyse d'un énoncé, de sauvegarder les résultats intermédiaires obtenus et de reprendre l'analyse ultérieurement. Cette possibilité est intéressante dans la mesure où l'analyse d'un énoncé, toutes les interactions avec les

autres composantes étant simulées par un dialogue à l'écran, peut se révéler assez fastidieuse. De plus, l'utilisateur a ainsi la possibilité d'explorer des voies d'analyse différentes à partir d'un instant donné de l'analyse.

La figure 9.14 montre l'écran par lequel l'opérateur peut reprendre une analyse interrompue.

```

*****
*
*          CHARGEMENT DES THEORIES
*          -----
*
*****

*****
*
*          CHARGEMENT DE L'ESPACE DES THEORIES
*
*****

Quel est le nom du fichier contenant l'espace des theories: █

```

fig. 9.14 : Reprise d'une analyse interrompue

3.4.2 Statistiques sur l'évolution de l'espace des théories

Durant l'analyse, le prototype informe de toute façon l'utilisateur sur l'évolution de l'analyse en lui fournissant un résumé du contenu de l'espace des théories. L'utilisateur peut demander une copie sur fichier de ces compte rendus réguliers qui se présentent sous la forme suivante :

```

*** Phase d'analyse numero..... : 19

Nombre de theories completes. : 1
  - recouvrant tout l'annonce. : 0
  - recouvrant une partie.... : 1
Nombre de theories bloquées.. : 476
Nombre de theories en attente : 22
Nombre de theories pretes:... : 390
  - a etre developpees..... : 24
  - deja developpees..... : 366
-----
Nombre total de theories..... : 889

```

L'exemple résume l'état de l'espace des théories au début de la dix-neuvième phase de l'analyse d'un énoncé (une phase correspond à un passage dans la boucle d'analyse).

3.4.3 Trace chronologique des événements

Afin de pouvoir suivre l'évolution d'une analyse dans le temps au niveau de détail le plus fin, il est nécessaire de disposer d'une trace chronologique des événements. Le prototype de la composante SYNSEM fournit cette trace sur demande de l'utilisateur. Elle se présente de la façon suivante :

```
*** Phase d'analyse n.: 1  
  
Creation theorie n.: 0  
Traitement theorie n.: 0  
Creation theorie n.: 1  
Creation theorie n.: 2  
Creation theorie n.: 3  
Creation theorie n.: 4  
Creation theorie n.: 5  
Creation theorie n.: 6  
Emission hypothese theorie n.: 6  
Creation theorie n.: 7
```

L'exemple décrit les événements qui se sont passés lors de la première phase de l'analyse d'un énoncé.

3.4.4 Sauvegarde automatique de l'espace des théories

Nous pensons qu'il est intéressant non seulement de connaître régulièrement l'état de l'espace des théories et de savoir, dans l'ordre chronologique, quels sont les événements qui ont marqué l'analyse mais également de pouvoir consulter l'espace des théories lui-même pendant l'analyse. L'utilisateur a donc la possibilité de demander la sauvegarde automatique, à des intervalles réguliers, de l'espace des théories sur un fichier. Le fichier ainsi produit peut être transformé par le programme LESP (présenté dans un des paragraphes précédents) en un texte facilement lisible par l'utilisateur.

La figure 9.15 montre l'écran par lequel l'utilisateur peut fixer la fréquence des sauvegardes automatiques.

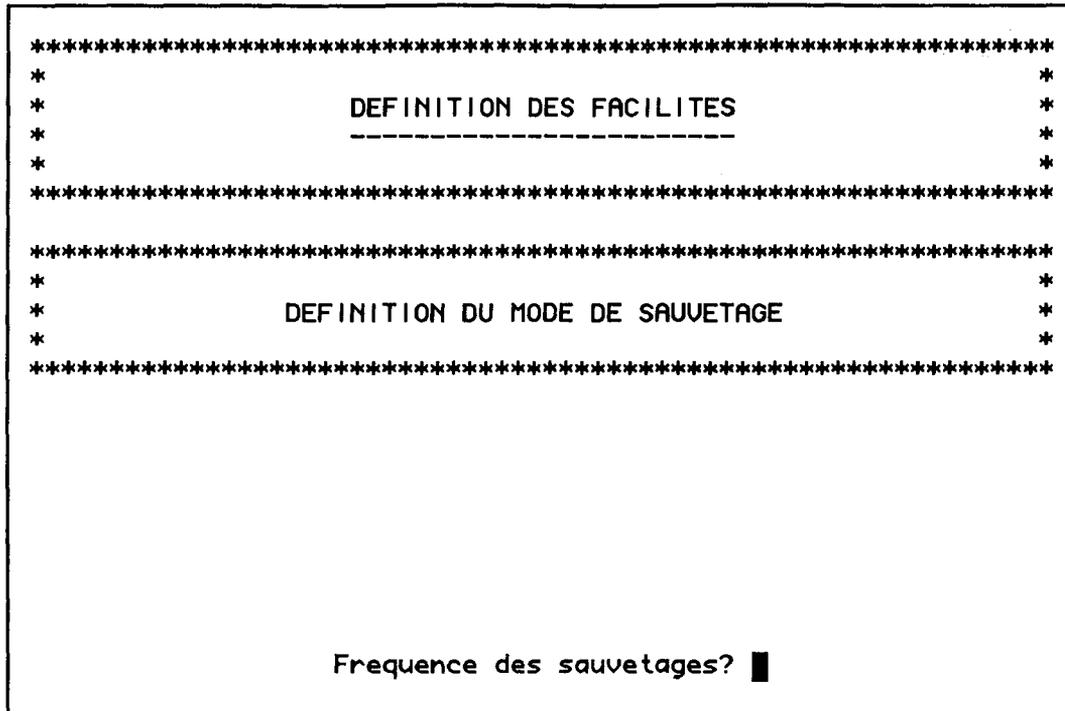


fig. 9.15 : Sauvegarde automatique de l'espace des théories

3.4.5 Nettoyage automatique de l'espace des théories

Une dernière facilité est liée au problème de l'accroissement de l'espace des théories. Dans une perspective d'évaluation, il est bien sûr intéressant de mémoriser toutes les théories jamais produites en cours d'une analyse. On risque cependant de toucher aux limites du matériel dont on dispose aujourd'hui. Afin d'éviter le dépassement de la capacité en mémoire du système, l'utilisateur peut demander le nettoyage automatique de l'espace des théories. Au cours de ce nettoyage vont être détruites toutes les théories qui ne peuvent plus servir à générer de nouvelles théories et ne sont d'aucune utilité, à savoir les théories bloquées et les théories prêtes mais déjà développées.

4 Premiers résultats

Le prototype que nous avons réalisé nous a permis de procéder aux premiers tests de la composante SYNSEM et d'en tirer les premières conclusions :

- La notion de théorie telle que nous l'avons définie est une notion tout à fait viable dans le contexte d'un système de dialogue oral homme-machine. En effet, des analyses que nous avons effectuées sur divers énoncés nous ont montré que des résultats peuvent être obtenus dans les limites du matériel dont nous disposons actuellement.

- Des essais ont également montré que suivant les stratégies adoptées, on arrive à améliorer sensiblement le comportement de la composante en termes de nombre de théories dans l'espace des théories et en termes de nombre de phases d'analyse nécessaires pour reconnaître la structure d'un énoncé.

Les quelques tests que nous avons pu effectuer jusque maintenant semblent indiquer que la stratégie de sélection la plus appropriée soit une stratégie des n meilleurs d'abord avec n variable tandis que les stratégies de propagation les mieux adaptées sont apparemment :

- affaiblissement des théories lors de l'ajout de nœuds non-terminaux dans l'arbre syntaxique et pondération du score de la théorie par le nombre de nœuds de l'arbre ;
- pas d'affaiblissement des théories lors de l'ajout de nœuds terminaux dans l'arbre syntaxique ;
- propagation du score d'un mot validé en pondérant le score de la théorie émettrice par le nombre de nœuds de type « réalisation ».

Des résultats détaillés sont présentés en annexe H, on y trouvera deux exemples d'analyses d'énoncés qui confirment nos conclusions. Il faut cependant encore signaler que notre étude est loin d'être terminée :

- Dès les premiers essais, nous avons dû revenir sur la définition des connaissances linguistiques. En particulier, des tests contextuels ont été introduits dans quelques nœuds procéduraux afin d'endiguer la génération de certaines théories indésirables (cf. annexe G).
- Seules des études comparatives systématiques sur base d'énoncés extraits des corpus nous permettront de savoir quelles sont les stratégies les mieux adaptées à notre problème et de vérifier l'adéquation des connaissances linguistiques.

ORIENTATIONS FUTURES

Dans ce travail, nous avons tenté de montrer les grandes lignes qui nous ont guidées dans le développement du système de dialogue oral homme-machine DIAL ainsi que différentes réalisations qui ont été effectuées dans ce contexte. Nous allons essayer de faire le bilan des enseignements que nous avons pu tirer de notre expérience et de proposer des voies de réflexion futures.

1 Ce qui a été fait

Tout au long des chapitres précédents, nous avons tenté d'exposer les principes généraux qui nous ont guidés lors de la conception et du développement du système DIAL et, en particulier, de sa composante syntaxico-sémantique, la composante SYNSEM.

Dans une première partie, nous avons montré que notre domaine d'investigation était celui de la compréhension de la parole mais avec des limitations précises. Nous nous sommes intéressés exclusivement aux dialogues oraux homme-machine finalisés en langage naturel.

Une deuxième partie nous a permis d'exposer les bases théoriques et empiriques qui ont servi de fondement à notre système. Rappelons que l'objectif était d'éviter la conception d'un système ad-hoc et d'en appuyer la définition sur des théories linguistiques confirmées. Non seulement avons-nous voulu tirer pleinement profit des enseignements de la linguistique théorique mais en plus, nous avons essayé d'exploiter l'expérience acquise grâce à des systèmes de compréhension de parole antérieurs.

Enfin, dans une troisième partie, nous avons présenté la façon dont le problème a été abordé dans le cas particulier de la composante syntaxico-sémantique SYNSEM. La composante a été située à l'intérieur d'un contexte plus large, celui du système DIAL complet. Nous avons montré comment les théories linguistiques sont appliquées dans SYNSEM et donnent lieu à des modèles opératoires de la langue. Finalement, le lecteur aura pu se rendre

compte des outils que nous avons réalisés et qui nous permettent de valider notre approche d'un point de vue opérationnel.

Pour conclure, faisons le résumé de ce qui a été fait :

- le sous-langage « demande de renseignement » (le langage de notre application test) a été modélisé syntaxiquement aussi bien que sémantiquement ;
- des outils facilitant la définition et permettant la prise en compte aisée de ces connaissances linguistiques ont été réalisés ;
- un prototype d'évaluation de la composante SYNSEM, se limitant actuellement à la mise en œuvre de connaissances syntaxiques par un analyseur descendant, a également été développé.

Les résultats obtenus grâce à ces réalisations semblent confirmer la validité de notre approche.

2 Ce qu'il reste à faire

L'étude que nous avons entamé est cependant loin d'être close. Les étapes suivantes nous semblent indispensables pour arriver à la définition d'un système de dialogue oral homme-machine robuste et fiable :

- SYNSEM ne peut se limiter à la prise en compte des seules informations syntaxiques ; il faudra, à terme, qu'elle prenne en compte également les connaissances sémantiques ;
- les autres types d'analyse qui ont été présentés devront, eux-aussi, être implantés ;
- il faudra effectuer une évaluation systématique de la composante SYNSEM complète, étudier le comportement de SYNSEM en fonction des stratégies et affiner les connaissances linguistiques ;
- avec les autres composantes, SYNSEM devra être intégrée dans un système complet où la communication entre les modules ne sera plus simulée ;
- on pourra alors passer à l'évaluation du système DIAL complet ;
- la portabilité du système DIAL devra ensuite être vérifiée par l'application à des domaines divers et variés.

En plus de ces travaux fondamentaux qu'il faudra absolument aborder, nous pensons que d'autres tâches sont moins primordiales mais présentent néanmoins un intérêt certain :

- l'utilisation de plus en plus généralisée de stations de travail graphiques remet partiellement en cause l'approche choisie pour la représentation externe des connaissances basée essentiellement sur une description textuelle. La convivialité de bon nombre d'outils de

définition des connaissances ou d'évaluation des résultats pourra très certainement être amélioré par le recours à des moyens graphiques ;

- des études linguistiques approfondies doivent encore être entreprises pour déterminer les limites de validité de nos modèles linguistiques actuels. Pour l'instant, ils ne prétendent à aucune validité universelle mais seule des études ciblées permettront d'obtenir des réponses à ces questions.

3 Quelques remarques en guise de conclusion

Pour terminer, nous pensons qu'il est important de rappeler encore certains points afin d'éviter tout malentendu.

Ce travail s'inscrit dans le cadre du développement d'un système de dialogue homme-machine finalisé en langage naturel à composante orale. Dans ce contexte, la notion de « dialogue » a une signification très limitée¹ et nous ne prétendons nullement traiter « les » dialogues de façon générale.

De même, notre préoccupation n'a pas été l'étude du langage naturel parlé dans toute sa généralité et en tant que moyen de communication chez l'homme et nous n'avons pas utilisé l'ordinateur comme outil d'étude des comportements langagiers humains². Notre approche a plutôt été inverse ; nous nous sommes inspiré des théories du langage afin de réaliser des systèmes dont l'interface soit la plus naturelle possible.

¹ Sur la notion de « dialogue » on pourra consulter *The Concept of Dialogue*, par B. GORANZON, M. FLORIN et P. SALLSTROM. Page 280, les auteurs pensent que : « It has become almost fashionable to speak of dialogue in discussions of any kind of communicative interaction between people, between man and machine or between interconnected systems in general. But then, considering the historical and philosophical perspectives, does this not mean that we are deliberately overlooking an important distinction, i.e. the distinction between 'simple communication' (or 'information transfer') and 'authentic dialogue'. Not all conversations between two people are necessarily dialogues in a true sense. Any talk may, of course, be empty words. »

² Au sujet de ce problème, voir également J. WEIZENBAUM *On the Impact of the Computer on Society* page 610 : « We must be clear about the fact that a computer is nothing without a program. A program is fundamentally a transformation of one computer into another that has autonomy and that, in a very real sense, behaves. Programming languages describe dynamic processes. And, most importantly, the processes they describe can be actually carried out. Thus we can build models of any aspect of the real world that interests us and that we understand. And we can make our models work. But we must be careful to remember that a computer model is a description that works. Ordinarily, when we speak of A being a model of B, we mean that a theory about some aspects of the behavior of B is also a theory of the same aspects of the behavior of A. It follows that when, for example, we consider a computer model of paranoia, like that published by Colby *et al.* (2), we must not be persuaded that it tells us anything about paranoia on the grounds that it, in some sense, mirrors the behavior of a paranoiac. After all, a plain typewriter in some sense mirrors the behavior of an autistic child (one types a question and gets no response whatever), but it does not help us to understand autism. A model must be made to stand or fall on the basis of its theory. (...) ».

Nos recherches, ainsi que d'autres travaux touchant aux mêmes problèmes (Deville, 1989 ; Roussanaly, 1988), semblent confirmer que ce type de limitation est incontournable si on envisage l'intégration du langage naturel parlé dans une application informatique à court ou à moyen terme.

BIBLIOGRAPHIE

- Aho A.V., Sethi R., Ullman J.D. (1986)
Compilers - Principles, Techniques, and Tools
Reading, Mass. : Addison-Wesley Publishing Company
- Alterman R. (1985)
A Dictionary Based on Concept Coherence
Artificial Intelligence, vol. 25
- Amalberti R., Carbonell N., Falzon P. (1984)
Stratégie de contrôle du dialogue en situation d'interrogation téléphonique
Nancy : Actes du séminaire GRECO "Dialogue oral homme-machine à composante orale", 11-12 Octobre
- Angeletti T., Beysson B. (1984)
Recueil d'un groupe de questions écrit dans le cadre d'un projet d'interrogation de bases de données textuelles en langage naturel
Nancy : Actes du séminaire GRECO "Dialogue oral homme-machine à composante orale", 11-12 Octobre
- Arnold D. (1986)
Eurotra: A European Perspective on MT
Proceedings of the IEEE, vol. 74, n° 7
- Arsac J. (1987)
Les machines à penser - Des ordinateurs et des hommes
Paris : Editions du Seuil
- Arthur J.D. (1986)
A Descriptive/Prescriptive Model for Menu-based Interaction
International Journal of Man-Machine Studies, vol. 25
- Aseo J. (1984)
Speech Compression Brings Voice Messaging down to Earth
Computer Design, July

- Bahl L.R., Jelinek F., Mercer R.L. (1983)
A Maximum Likelihood Approach to Continuous Speech Recognition
IEEE Transactions on Pattern Analysis and Machine Intelligence,
vol. PAMI-5, n° 2
- Barr A., Feigenbaum E.A. (Eds.) (1981)
The Handbook of Artificial Intelligence
Los Altos, Cal. : William Kaufmann
Trad. : *Le manuel de l'intelligence artificielle - tome 1*, Paris :
Eyrolles, 1986
- Bátori I.S. (1981)
Die Grammatik aus der Sicht kognitiver Prozesse
Tübingen : Gunter Narr Verlag
- Beckman F.S. (1980)
Mathematical Foundations of Programming
Reading, Mass. : Addison Wesley Publishing Company
- Bellman J. (1957)
Dynamic Programming
Princeton : Princeton University Press
- Benbasat I., Wand Yair (1984)
A Structured Approach to Designing Human-Computer Dialogues
International Journal of Man-Machine Studies, vol. 21
- Benvéniste E. (1966)
Problèmes de linguistique générale vol. I
Paris : Gallimard
- Benvéniste E. (1974)
Problèmes de linguistique générale vol. II
Paris : Gallimard
- Berleur J. (1984)
Compréhension du langage naturel et reconnaissance de la parole par ordinateur : « est-ce approprié ? »
Namur : Actes des Deuxièmes Journées de Réflexion sur
l'Informatique, 30-31 Août et 1er Septembre
- Biermann A.W., Ballard B.W., Sigmon A.H. (1983)
An Experimental Study of Natural Language Programming
International Journal of Man-Machine Studies, vol. 18
- Biermann A.W., Rodman R.D., Rubin D.C., Heidlage J.F. (1985)
Natural Language with Discrete Speech as a Mode for Human-to-Machine Communication
Communications of the ACM, vol. 28, n° 6

- Bisiani R. (1985)
Computer Systems for High-Performance Speech Recognition
Versailles : Actes du séminaire INRIA "Principes de la communication homme-machine : Parole, vision et langage naturel", 28 mai - 7 juin
- Bobrow D.G., Hayes P.J. (Eds.) (1985)
Artificial Intelligence - Where Are We?
Artificial Intelligence, vol. 25
- Boite R. (1983)
Introduction au traitement de la parole
Louvain-la-Neuve : Actes de la journée ATHENA "Analyse, synthèse et reconnaissance de la parole", 17 Mai
- Bolc L, Kowalski A., Kozłowska M., Strzalkowski T. (1985)
A Natural Language Information Retrieval System with Extensions towards Fuzzy Reasoning
International Journal of Man-Machine Studies, vol. 23
- Bolt Beranek and Newman Inc. (1976)
Speech Understanding Systems-Final Report
Cambridge, Mass. : B.B.N. Report n° 3438
- Bonnet A. (1984)
L'intelligence artificielle - Promesses et Réalités
Paris : InterEditions
- Branscomb L.M., Thomas J.C. (1984)
Ease of Use: A System Design Challenge
IBM Systems Journal, vol. 23, n° 3
- Bresnan J., Kaplan R. (1981)
Lexical Functional Grammars; a Formal System for Grammatical Representation
in Bresnan J. (Ed.), *The Mental Representation of Grammatical Relations*
Cambridge, Mass. : MIT Press
- Bronckart J.P. (1977)
Théories du langage
Bruxelles : Pierre Mardaga, éditeur
- Bruce B. (1973)
Case Structure Systems
Stanford, Cal. : Proceedings of the International Joint Conference on Artificial Intelligence "IJCAI-73"
- Bruce B. (1975)
Case Systems for Natural Language
Artificial Intelligence, vol. 6

- Bunt H. (1981)
Conversational Principles in Question-Answer Dialogues
 in Krallman et Stickel (Eds.), *Zur Theorie der Frage*
 Tübingen : Gunter Narr Verlag
- Carbonell N., Bonin J.J. (1988)
Utilisation d'informations prosodiques en reconnaissance de la parole continue
 Nancy : Actes des dix-septièmes "Journées d'étude sur la parole", 20-22
 Septembre
- Carbonell N., Charpillat F., Haton J.P., Mangeol B., Mousel P.,
 Pierrel J.M., Roussanaly A. (1984)
Dialogue oral homme-machine : bilan du projet MYRTILLE et perspectives
 Nancy : Actes du séminaire GRECO "Dialogue homme-machine à
 composante orale", 11-12 Octobre
- Carbonell N., Charpillat F., Haton J.M., Mangeol B., Mousel P.,
 Pierrel J.M., Roussanaly A. (1985)
Utilisation d'informations linguistiques dans un système de dialogue oral homme-machine
 Paris : Actes des quatorzièmes "Journées d'étude sur la parole", 10-13
 Juin
- Carbonell N., Damestoy J.P., Fohr D., Haton J.P. (1986)
APHODEX: Design and Implementation of an Acoustic-Phonetic Decoding Expert System
 Proceedings of IEEE ICASSP-86
- Carbonell N., Haton J.P., Mangeol B., Mousel P., Pierrel J.M.,
 Roussanaly A. (1985)
Les sources de connaissance dans un système de dialogue oral homme-machine
 Grenoble : Actes du cinquième congrès AFCET "Reconnaissance des
 Formes et Intelligence Artificielle", 25-29 Novembre
- Carbonell N., Haton J.P., Pierrel J.M. (1983)
Elaboration d'un système expert pour le décodage phonétique automatique de la parole
 Speech Communication, vol. 2, n° 2-3
- Castiglione D., Goldman J. (1984)
Speech and the Space Station
 Speech Technology, August/September
- Chafe W. (1970)
Meaning and the Structure of Language
 Chicago : University of Chicago Press
- Chapman G. (1985)
Airland Battle Doctrine and the Strategic Computing Initiative
 The CPSR Newsletter, vol. 3, n° 4

- Charpillat F. (1985)
Un système de reconnaissance de la parole continue pour la saisie de textes lus
Nancy : Thèse de Doctorat de l'Université de Nancy I
- Cherry C. (1957)
On Human Communication
New York : John Wiley and Sons
- Chomsky N. (1957)
Syntactic Structures
Den Haag : Mouton
Trad. : *Structures syntaxiques*, Paris : Editions du Seuil, 1969
- Chomsky N. (1959)
On Certain Formal Properties of Grammars
Information and Control, 2
- Chomsky N. (1965)
Aspects of the Theory of Syntax
Cambridge, Mass. : M.I.T. Press
Trad. : *Aspects de la théorie syntaxique*, Paris : Editions du Seuil, 1971
- Chomsky N. (1971)
Deep Structure, Surface Structure and Semantic Interpretation
in Jacobovitz L.A. et Steinberg D.D. (Eds.), *Semantics*
Cambridge : Cambridge University Press
- Chomsky N. (1972)
La forme et le sens dans le langage naturel
in Jakobson R., Halle M., Chomsky N., *Hypothèses*
Paris : Editions Seghers - Laffont
- Chomsky N., Miller G. A. (1963)
Introduction to the Formal Analysis of Natural Languages
in Halla M., Lunt H. et MacLean H. (Eds.), *Handbook of Mathematical Psychology*, vol. II
New York : John Wiley and Sons
Trad. : *L'analyse formelle des langues naturelles*, Paris : Mouton/Gauthier-Villars, 1968
- Colby K.M., Watt J.B., Gilbert J.P. (1966)
A Computer Method of Psychotherapy: Preliminary Communication
The Journal of Nervous and Mental Disease, vol. 142, n° 1
- Coler C.R. (1982)
Helicopter Speech Command Systems: Recent Noise Tests Are Encouraging
Speech Technology, September/October
- Connolly J.H., Edmonds E.A., Guzy J.J., Johnson S.R., Woodcock A. (1986)
Automatic Speech Recognition Based on Spectrogram Reading
International Journal of Man-Machine Studies, vol. 24

- Coulon D., Kayser D. (1986)
Informatique et langage naturel : Présentation générale des méthodes d'interprétation des textes écrits
Technique et Science Informatiques, vol. 5, n° 2
- Cozannet A. (1985)
Interrogation des pages roses en langue naturelle
Dijon : Actes des Journées "Bases de données" de l'AFCEP, 14-15 Novembre
- Culioli A., Fuchs C., Pécheux C. (1970)
Considérations théoriques à propos du traitement formel du langage
Paris : Documents du Centre de Linguistique Quantitative de la Faculté des Sciences de l'Université de Paris
- Damper R.I. (1984)
Voice-Input Aids for the Physically Disabled
International Journal of Man-Machine Studies, vol. 21
- Danlos L. (1985)
Génération automatique de textes en langue naturelle
Paris : Masson
- D'Archimbaud L.D. (1985)
Panorama des outils de génie logiciel pour le développement d'applications de l'intelligence artificielle
Revue Génie Logiciel, n° 2
- Dean M. (1982)
How a Computer Should Talk to People
IBM Systems Journal, vol. 21, n° 4
- DeMori R., Probst D. (1985)
Knowledge-Based Computer Recognition of Continuous Speech
Versailles : Actes du séminaire INRIA "Principes de la communication homme-machine : Parole, vision et langage naturel", 28 mai - 7 juin
- Denes P., Mathews M.V. (1960)
Spoken Digit Recognition Using Time-Frequency Patterns Matching
Journal of the Acoustical Society of America, vol. 32
- Derouault A.-M., Merialdo B. (1987)
Natural Language Modeling for Phoneme-to-Text Transcription
Paris : Actes du séminaire INRIA "Recherche et développement dans les industries de la langue", 7-11 Décembre
- Desclés J.P. (1982)
Langages quasi naturels et catégories grammaticales
Pont-à-Mousson : Actes du colloque ARC "Domaine et objectifs de la recherche cognitive"

- Descout R. (1984)
Speech Technology Applications in France
Speech Communication, vol. 3, n° 4
- DeSipio R.G., Fry E. (1983)
Avionics System Plays 'Ask and Tell' with Its Operator
Speech Technology, January/February
- Deville G. (1987)
Corpus de dialogues oraux finalisés en situation réelle : Demande de renseignements auprès du ministère de l'emploi et du travail (Cellule Action-Travail)
Namur : Facultés Universitaires Notre Dame de la Paix
- Deville G. (1989)
Modelization of Task-Oriented Utterances in a Man-Machine Dialogue System
Antwerpen : Proefschrift ter verkrijging van de graad van Doctor in de Letteren en Wijsbegeerte aan de Universitaire Instelling Antwerpen
- Deville G., Mousel P., Roussanaly A. (1988)
Classification d'énoncés de corpus de dialogues oraux finalisés
Nancy : Rapport interne n° 88-R-151
- Deville G., Paulussen H. (1986)
A Case Grammar as an Original Linguistic Model for the Semantic Representation of Utterances in a Man-Machine Dialog System
Antwerpen : Thesis for the Post-Graduate Degree of Computational Linguistics, Universitaire Instellingen Antwerpen
- Dik S. (1978)
Functional Grammar
Amsterdam : North-Holland Publishing Company
- Dik S. (1980)
Studies in Functional Grammar
London : Academic Press
- Dik S. (Ed.) (1983a)
Advances in Functional Grammar
Dordrecht, Holland : Foris
- Dik S. (1983b)
Basic Principles of Functional Grammar
in Dik S. (Ed.), *Advances in Functional Grammar*
Dordrecht, Holland : Foris
- Dik S. (1986a)
Linguistically Motivated Knowledge Representation
Amsterdam : Working Papers in Functional Grammar, n. 9,
University of Amsterdam

- Dik S. (1986b)
Two Papers on the Computational Application of FG
Amsterdam : Working Papers in Functional Grammar, n. 18,
University of Amsterdam
- Dik S. (1987)
Functional Grammar - Computational Model of the Natural Language User
Amsterdam : Paper presented at the Workshop "Functional Grammar and the Computer", Institute for Functional Research on Language and Language Use, June
- Dirven R., Fried V. (Eds.) (1987)
Functionalism in Linguistics
Amsterdam : John Benjamins Publishing Company
- Dixon N.R., Silverman H.F. (1977)
The 1976 Modular Acoustic Processor (MAP)
IEEE Transactions on Acoustics, Speech, and Signal Processing,
vol. ASSP-25, n° 5
- Dowty D. (1979)
Meaning and Montague Grammar - The Semantics of Verbs and Time in Generative Semantics and in Montague PTQ
Dordrecht : Reidel
- Dreyfus H.L. (1972)
What Computers Can't Do: A Critique of Artificial Reason
New York : Harper and Row
- Dreyfus H.L., Dreyfus S.E. (1986)
Mind over Machine
New York : Macmillan/The Free Press
Trad. : *Künstliche Intelligenz - Von den Grenzen der Denkmaschine und dem Wert der Intuition*, Reinbek bei Hamburg : Rowohlt
Taschenbuch Verlag, 1987
- Dubois J. (1976)
Grammaire de base du français
Paris : Larousse
- Dubus D., Hella F., Sfez E. (1984)
Dialogue homme-machine à bord d'un véhicule automobile
Nancy : Actes du séminaire GRECO "Dialogue homme-machine à composante orale", 11-12 Octobre
- Dudley H., Balashek S. (1958)
Automatic Recognition of Phonetic Patterns in Speech
Journal of the Acoustical Society of America, vol. 30
- Dzida W. (1983)
IFIP-Modell - Das IFIP-Modell für Benutzerschnittstellen
Sonderheft Office Management

- Edmonds E. (1982)
The Man-Computer Interface: A Note on Concepts and Design
International Journal of Man-Machine Studies, vol. 16
- Erman L.D., Hayes-Roth F., Lesser V.R., Reddy D.R. (1980)
The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty
Computing Surveys, vol. 12, n° 2
- Erman L.D., Lesser V.R. (1980)
The HEARSAY II Speech Understanding System: A Tutorial
in Lea W.A. (Ed.), *Trends in Speech Recognition*
Englewood Cliffs, N.J. : Prentice-Hall Inc.
- Falzon P. (1984)
Les langages opératifs
Nancy : Actes du séminaire GRECO "Dialogue homme-machine à composante orale", 11-12 Octobre
- Fargues J., Landau M.C., Dugourd A., Catach L. (1986)
Conceptual Graphs for Semantics and Knowledge Processing
IBM Journal of Research and Development, vol. 30, n° 1
- Fawe A.L. (1983)
Représentation de la parole par la densité des passages par zéro (ou fréquence instantanée) en vue de la transmission, du traitement et de la reconnaissance
Louvain-la-Neuve : Actes de la journée ATHENA "Analyse, synthèse et reconnaissance de la parole", 17 Mai
- Fillmore C.J. (1968)
The Case for Case
in Bach E. et Harms R.T. (Eds.), *Universals in Linguistic Theory*
New York : Holt, Rinehart & Winston
- Fillmore C.J. (1972)
Subjects, Speakers, and Roles
in Davidson et Harman (Eds.), *Semantics of Natural Language*
Dordrecht, Holland : D. Reidel Publishing Company
- Fillmore C.J. (1977)
Scenes-and-Frames Semantics
in Zampolli A. (Ed.), *Linguistic Structures Processing*
Amsterdam : North-Holland Publishing Company
- Finin T.W., Joshi A.K., Webber B.L. (1986)
Natural Language Interactions with Artificial Experts
Proceedings of the IEEE, vol. 74, n° 7
- Firschein O., Fischler M.A., Coles L.S., Tenenbaum J.M. (1974)
Intelligent Machines Are on the Way
IEEE Spectrum, July

- Fohr D. (1986)
APHODEX : un système expert de décodage acoustico-phonétique de la parole continue
Nancy : Thèse de Doctorat de l'Université de Nancy I
- Fohr D., Carbonell N., Haton J.P. (1985)
SYSTEXP, un système expert pour le décodage acoustico-phonétique de la parole
Avignon : Actes des cinquièmes journées internationales sur les systèmes experts et leurs applications, 13-15 Mai
- Foley W.A., Van Valin R.D. (1984)
Functional Syntax and Universal Grammar
Cambridge : Cambridge University Press
- Gaines B.R., Shaw M.L.G. (1986a)
From Timesharing to the Sixth Generation: The Development of Human-Computer Interaction. Part I
International Journal of Man-Machine Studies, vol. 24
- Gaines B.R., Shaw M.L.G. (1986b)
Foundations of Dialog Engineering: The Development of Human-Computer Interaction. Part II
International Journal of Man-Machine Studies, vol. 24
- Gazdar G., Klein E., Pullum G., Sag I. (1985)
Generalized Phrase Structure Grammar
Oxford : Basil Blackwell
- Ginzburg A. (1969)
Algebraic Theory of Automata
New York : Academic Press
- Göranzon B., Florin M., Sällström P. (1988)
The Concept of Dialogue
AI & Society, vol. 2, n° 4
- Gorny P. (1985)
Informatik und Militär
in Butte W. (Ed.), *Militarisierte Wissenschaft*
Reinbek bei Hamburg : Rowohlt Taschenbuch Verlag
- Gould J.D. (1982)
Writing and Speaking Letters and Messages
International Journal of Man-Machine Studies, vol. 16
- Gould J.D., Conti J., Hovanyecz T. (1983)
Composing Letters with a Simulated Listening Typewriter
Communications of the ACM, vol. 26, n° 4

- GRECO (1985)
Dialogue oral homme-machine en situation orientée par l'action
Groupe "Dialogue homme-machine" du GRECO "Communication
Parlée"
Grenoble : Actes du 5^{ème} congrès AFCET RF-IA
- Green P.D., Wood A.R. (1984)
***Knowledge-Based Speech Understanding: Towards a Representational
Approach***
in O'Shea T. (Ed.), *ECAI-84: Advances in Artificial Intelligence*
Amsterdam : Elsevier Science Publishers B.V. (North-Holland)
- Grevisse M.
Précis de grammaire française
Collection Grevisse
- Grice H.P. (1975)
Logic and Conversation in Syntax and Semantics 3: Speech Acts
E. Cole et J.L. Morgan (Eds.)
New York : Academic Press
- Gross M., Lentin A. (1967)
Notions sur les grammaires formelles
Paris : Gauthier-Villars
- Gross M. (1981)
La formalisation des langues naturelles
Pour la Science, Septembre
- Guida G., Mauri G. (1986)
***Evaluation of Natural Language Processing Systems: Issues and
Approaches***
Proceedings of the IEEE, vol. 74, n° 7
- Halliday M., Fawcett R. (1985)
New Developments in Systemic Linguistics
London : Batsford
- Harris Z. (1968)
Mathematical Structure of Language
New York : John Wiley and Sons
Trad. : *Structures mathématiques du langage*, Paris : Dunod, 1971
- Haton J.P. (1974)
***A Practical Application of a Real-Time Isolated-Word Recognition
System Using Syntactic Constraints***
IEEE Transactions on Acoustics, Speech, and Signal Processing,
vol. ASSP-22, n° 6
- Haton J.P. (1980)
Speech Recognition Work in Western Europe
in Lea W.A. (Ed.), *Trends in Speech Recognition*
Englewood Cliffs, N.J. : Prentice-Hall Inc.

- Haton J.P. (1984)
Compréhension de la parole et intelligence artificielle : état des recherches
 Nancy : Actes du séminaire GRECO "Dialogue homme-machine à composante orale", 11-12 Octobre
- Haton J.P. (1985a)
Intelligence artificielle en compréhension automatique de la parole : état des recherches et comparaison avec la vision par ordinateur
 Technique et Science Informatiques, vol. 4, n° 3
- Haton J.P. (1985b)
Knowledge-Based and Expert Systems in Automatic Speech Recognition
 Versailles : Actes du séminaire INRIA "Principes de la communication homme-machine : Parole, vision et langage naturel", 28 mai - 7 juin
- Haton J.P., Mangeol B., Pierrel J.M. (1986)
Organisation et fonctionnement d'une composante lexicale dans un système de dialogue oral homme-machine
 Toulouse : Actes du séminaire GRECO CNRS Communication parlée et GALF "Lexique et traitement automatique des langages", Université Paul Sabatier, janvier
- Haton J.P., Messenet G., Pierrel J.M., Sanchez C. (1978)
La chaîne de compréhension parlée du système MYRTILLE II
 Paris : Actes du congrès AFCET "Théorie et Techniques de l'Informatique", 13-15 Novembre
- Haton J.P., Pérennou G. (1978)
Reconnaissance automatique de la parole
 Namur : Huitième école internationale d'été d'informatique de l'AFCET, 10-22 Juillet
- Haton J.P., Pierrel J.M. (1981a)
Analyse et reconnaissance automatique de la parole au C.R.I.N. : Approches, motivations et résultats obtenus
 Montréal : Actes des douzièmes "Journées d'étude sur la parole", Mai
- Haton J.P., Pierrel J.M. (1981b)
Applications actuelles et problèmes ouverts en reconnaissance - compréhension de la parole
 Nancy : Rapport interne 81-R-024, CRIN
- Hart J. (1969)
B.C. Cave in
 London : Fawcett Publications Inc.
- Hauptmann A.G., Rudnicky A.I. (1988)
Talking to Computers: An Empirical Investigation
 International Journal of Man-Machine Studies, vol. 28

- Hayes P., Ball E., Reddy R. (1981)
Breaking the Man-Machine Communication Barrier
Computer, March
- Hayes P.J., Reddy D.R. (1983)
Steps towards Graceful Interaction in Spoken and Written Man-Machine Communication
International Journal of Man-Machine Studies, vol. 19
- Hendrix G.G., Sacerdoti E.D., Sagalowicz D., Slocum J. (1978)
Developing a Natural Language Interface to Complex Data
ACM Transactions on Database Systems, vol. 3, n° 2
- Hendrix G.G., Thompson C.W., Slocum J. (1973)
Language Processing via Canonical Verbs and Semantic Model
Stanford, Cal. : Proceedings of the International Joint Conference on Artificial Intelligence "IJCAI-73"
- Herrlitz W., Hundsnurscher F. (1971)
Eine Einführung in die moderne Linguistik
Weinheim : Verlag Julius Beltz
- Heute U. (1988)
Medium-Rate Speech Coding - Trial of a Review
Speech Communication, vol. 7, n° 2
- Hjelmslev L. (1943)
Omkring Spragteoriens grunlæggelse
København : Munksgraad
Trad. : *Prolegomènes à une théorie du langage*, Paris : Editions de minuit, 1968
- Hopcroft J.E., Ullman J.D. (1979)
Introduction to Automata Theory, Languages and Computation
Reading, Mass. : Addison Wesley Publishing Company
- Hörmann H. (1970)
Psychologie der Sprache
Berlin : Springer Verlag
Trad. : *Introduction à la psycholinguistique*, Paris : Larousse, 1972
- Irumano T., Kanasashi H., Akiba K., Mafune Y. (1983)
A Spoken Word Recognition System Based on Phoneme Recognition
Speech Communication, vol. 2, n° 2-3
- Itakura F. (1975)
Minimum Prediction Residual Principle Applied to Speech Recognition
IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-23

- Jakobson R. (1963)
Essais de Linguistique générale
 Paris : Editions de minuit
- Jensen K., Wirth N. (1976)
Pascal User Manual and Report (2ème édition)
 Berlin : Springer Verlag
 Trad. : *Pascal manuel de l'utilisateur*, Paris : Eyrolles, 1980
- Johnson S.C. (1975)
Yacc: Yet Another Compiler Compiler
 Murray Hill, N.J. : Computing Science Technical Report no. 32,
 AT&T Bell Laboratories
- Kammersgaard John (1988)
Four Different Perspectives on Human-Computer Interaction
 International Journal of Man-Machine Studies, vol. 28
- Kaneko T., Dixon N.R. (1983)
A Hierarchical Decision Approach to Large-Vocabulary Discrete Utterance Recognition
 IEEE Transactions on Acoustics, Speech, and Signal Processing,
 vol. ASSP-31, n° 5
- Kay M. (1985)
On Overview of Natural Language Understanding
 Versailles : Actes du séminaire INRIA "Principes de la communication homme-machine : Parole, vision et langage naturel",
 28 mai - 7 juin
- Kayser D. (1985)
Des machines qui comprennent notre langue
 La recherche, vol. 16, n° 170
- Kittredge R. (1982)
Variation and Homogeneity in Sublanguages
 in Kittredge R. et Lehrberger J. (Eds.), *Sublanguage: Studies of Language in Restricted Semantic Domains*
 Berlin : De Gruyter
- Klatt D.H. (1980)
Overview of the ARPA Speech Understanding Project
 in Lea W.A. (Ed.), *Trends in Speech Recognition*
 Englewood Cliffs, N.J. : Prentice-Hall Inc.
- Kohonen T. (1988)
The "Neural" Phonetic Typewriter
 Computer, March
- Laasri H., Maître B., Haton J.P. (1987)
ATOME : outil d'aide au développement des systèmes multi-experts
 Antibes : Actes du sixième congrès AFCET "Reconnaissance des Formes et Intelligence Artificielle"

- Lakoff G. (1971)
On Generative Semantics
in Jacobovitz L.A. et Steinberg D.D. (Eds.), *Semantics*
Cambridge : Cambridge University Press
- Laurière J.-L. (1982)
***Représentation et utilisation des connaissances (deuxième partie :
Représentation des connaissances)***
Technique et Science Informatiques, vol. 1, n° 2
- Lea W.A. (1980a)
The Value of Speech Recognition Systems
in Lea W.A. (Ed.), *Trends in Speech Recognition*
Englewood Cliffs, N.J. : Prentice-Hall Inc.
- Lea W.A. (1980b)
Speech Recognition: Past, Present, and Future
in Lea W.A. (Ed.), *Trends in Speech Recognition*
Englewood Cliffs, N.J. : Prentice-Hall Inc.
- Lea W.A. (1983)
Selecting the Best Speech Recognizer for the Job
Speech Technology, January/February
- Ledgard H., Whiteside J.A., Singer A., Seymour W. (1980)
The Natural Language of Interactive Systems
Communications of the ACM, vol. 23, n° 10
- Lesk M.E., Schmidt E. (1975)
Lex: A Lexical Analyzer Generator
Murray Hill, N.J. : Computing Science Technical Report no. 39,
AT&T Bell Laboratories
- Levinson S. (1983)
Pragmatics
Cambridge : Cambridge University Press
- Levinson S. (1985)
Statistical Methods for Speaker Independence
Versailles : Actes du séminaire INRIA "Principes de la
communication homme-machine : Parole, vision et langage naturel",
28 mai - 7 juin
- Levinson S., Liberman M. (1981)
La reconnaissance de la parole par ordinateur
Pour la Science, Juin
- Lewis H.R., Papadimitriou C.H. (1981)
Elements of the Theory of Computation
Englewood Cliffs, N.J. : Prentice-Hall, Inc.

- Lindblom B. (1983)
On the Teleological Nature of Speech Processes
 Speech Communication, vol. 2, n° 2-3
- Lyons J., Coates R., Deuchar M., Gazdar G. (Eds.) (1987)
New Horizons in Linguistics
 London : Pelican
- McCawley J.D. (1971)
Interpretative Semantics Meets Frankenstein
 Foundations of Language, vol. 7
- Maguire M. (1982)
An Evaluation of Published Recommendations on the Design of Man-Computer Dialogues
 International Journal of Man-Machine Studies, vol. 1982
- Mariani J. (1982)
ESOPE : un système de compréhension de la parole continue
 Paris : Thèse d'état de l'Université de Paris 6
- Mariani J. (1983)
Reconnaissance phonétique par diphonèmes
 Speech Communication, vol. 2, n° 2-3
- Mariani J. (1985)
La reconnaissance de la parole
 La recherche, vol. 16, n° 170
- Markel J.D., Gray A.H. (1976)
Linear Prediction of Speech
 Berlin : Springer Verlag
- Martinet A. (1960)
Eléments de linguistique générale
 Paris : Armand Colin
- Martin T.B. (1976)
Practical Applications of Voice Input to Machines
 Proceedings of the IEEE, vol. 64, n° 4
- Medress M.F., Cooper F.S., Forgie J.W., Green C.C., Klatt D.H., O'Malley M.H., Neuburg E.P., Newell A., Reddy D.R., Ritea B., Shoup-Hummel J.E., Walker D.E., Woods W.A. (1977)
Speech Understanding Systems - Report of a Steering Committee
 SIGART Newsletter, n° 62
 réimprimé dans : Artificial Intelligence, vol. 9, 1978
- Mercier G., Quinton P., Vires R. (1978)
KEAL : un système pour un dialogue oral avec une machine
 Paris : Actes du congrès AFCET "Théorie et Techniques de l'Informatique", 13-15 Novembre

- Merialdo B. (1987)
Multi-level Decoding for Very Large Size Dictionary Speech Recognition
Paris : Actes du séminaire INRIA "Recherche et développement dans les industries de la langue", 7-11 Décembre
- Moeschler J. (1985)
Argumentation et conversation. Eléments pour une analyse pragmatique du discours
Paris : Hatier
- Morel M.A., et al. (1985)
Analyse linguistique d'un corpus d'oral finalisé
Paris : Rapport final d'étude, CNRS-GRECO "Communication Parlée"
- Morik K. (1984)
Customers' Requirements for Natural Language Systems: Results of an Inquiry
International Journal of Man-Machine Studies, vol. 21
- Morin P., Pierrel J.M. (1987)
PARTNER un système de dialogue oral homme-machine
Paris : Actes du congrès COGNITIVA
- Morrison P. (1973)
The Mind of the Machine
Technology Review, January
- Mosanto P. (1988)
La compréhension des langages naturels : libre tribune
Journ'ALMIN revue de l'Association des Licenciés et Maîtres en Informatique de Namur, n° 10
- Mousel-Muller J. (1985)
Un système de messagerie vocale utilisant un éditeur de sons
Namur : Mémoire de Licence et Maîtrise en Informatique des Facultés Universitaires Notre-Dame de la Paix
- Mousel P. (1985)
Le programme de création des réseaux à noeuds procéduraux
Nancy : Rapport interne n° 85-R-092, CRIN
- Mousel P. (1986a)
Logiciels d'aide à la création de bases de connaissances linguistiques
Nancy : Rapport interne n° 86-R-029, CRIN
- Mousel P. (1986b)
Implémentation des réseaux à noeuds procéduraux sous UNIX BSD 4.2 sur le VAX 11/785 du CRIN
Nancy : Rapport interne n° 86-R-123, CRIN

- Mousel P. (1988)
Implémentation d'une grammaire de cas sous UNIX BSD 4.2 sur le VAX 11/785 du CRIN
 Nancy : Rapport interne n° 88-R-023, CRIN
- Mousel P., Pierrel J.M., Roussanaly A. (1989a)
Cooperation and Representation of Syntactic-Semantic and Pragmatic Knowledge in a Natural Language Task Oriented Spoken Dialogue System
 Paris : Actes de la "European Conference on Speech Communication and Technology", 26-28 Septembre
- Mousel P., Pierrel J.M., Roussanaly A. (1989b)
Coopération entre syntaxe, sémantique et pragmatique dans un système de dialogue oral homme-machine
 Paris : Actes du septième congrès AFCET "Reconnaissance des Formes et Intelligence Artificielle", 29 Novembre - 1 Décembre
- Mousel P., Roussanaly A. (1988)
Ladi : Logiciel d'Aide au Dépouillement de corpus Interactif
 Nancy : Rapport interne n° 88-R-024
- Nagao M., Tsujii J.I., Nakamura J.I. (1986)
Machine Translation from Japanese into English
 Proceedings of the IEEE, vol. 74, n° 7
- Nash-Webber B. (1975)
The Role of Semantics in Automatic Speech Understanding
 in Bobrow D.G. et Collins A. (Eds.), *Representation and Understanding: Studies in Cognitive Science*
 New York : Academic Press
- Naur P., (Ed.) (1963)
Revised Report on the Algorithmic Language ALGOL 60
 Communications of the ACM, vol. 6, n° 1
- Neuburg E.P. (1980)
Needs vs. Competence in Speech Recognition
 in Lea W.A. (Ed.), *Trends in Speech Recognition*
 Englewood Cliffs, N.J. : Prentice-Hall Inc.
- Newell A., Barnett J., Forgie J.W., Green C., Klatt D., Licklider J.C.R., Munson J., Reddy D.R., Woods W.A. (1973)
Speech Understanding Systems: Final Report of a Study Group
 Amsterdam : North-Holland/American Elsevier
- Newell A., Cooper F.S., Forgie J.W., Green C., Klatt D., Medress M.F., Neuburg E.P., O'Malley M.H., Reddy D.R., Ritea R., Shoup-Hummel J.E., Walker D.E., Woods W.A. (1976)
Considerations for a Follow-on ARPA Research Program for Speech Understanding Systems
 Pittsburgh, Pa. : Computer Science Department, Carnegie-Mellon University

- Newman J. (1981)
Human Factors Requirements for Managerial Use of Computer Message Systems
in Uhlig R.P. (Ed.), *Computer Message System*
Amsterdam : North-Holland Publishing Company/IFIP
- Niemann H., Brietzmann A., Mühlfeld R., Regel P., Schukat G. (1985)
The Speech Understanding and Dialog System EVAR
in DeMori R. et Suen C.Y. (Eds.), *New Systems and Architectures for Automatic Speech Recognition and Synthesis*
Berlin : Springer Verlag
- North R.A., Lea W.A. (1982)
Application of Advanced Speech Technology in Manned Penetration Bombers
Honeywell Systems and Research Center report AFWAL-TR-82-3004 to Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio
- Nouhen-Bellec A., Siroux J. (1981)
CADI : Constructeur Automatique de Dialogue Intelligent intégré à un système de reconnaissance et de compréhension de la parole continue
Rennes : Thèse de troisième cycle de l'Université de Rennes I
- Nye J.M. (1980)
The Expanding Market for Commercial Speech Recognizers
in Lea W.A. (Ed.), *Trends in Speech Recognition*
Englewood Cliffs, N.J. : Prentice-Hall Inc.
- Nye J.M. (1982)
Human Factors Analysis of Speech Recognition Systems
Speech Technology, April
- Nyquist H. (1924)
Certain Factors Affecting Telegraph Speed
Bell Systems Technical Journal, vol. 3
- Oden G.C. (1983)
On the Use of Semantic Constraints in Guiding Syntactic Analysis
International Journal of Man-Machine Studies, vol. 19
- Pair C. (1966)
La formalisation des grammaires
Cahiers du CRAL, 1ère série, n° 1
- Pereira F.C.N., Warren D.H.D. (1980)
Definite Clause Grammars for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks
Artificial Intelligence, vol. 13
- Pierce J.R. (1969)
Whither Speech Recognition
Journal of the Acoustical Society of America, vol. 46, n° 6

- Pierrel J.M. (1975)
Contribution à la compréhension automatique du discours continu
Nancy : Thèse de troisième cycle de l'Université de Nancy I
- Pierrel J.M. (1978a)
Un système de compréhension automatique du discours continu utilisant des contraintes morphologiques, syntaxiques et sémantiques
R.A.I.R.O. Informatique/Computer Science, vol. 12, n° 2
- Pierrel J.M. (1978b)
MYRTILLE II : Un système de compréhension du discours continu
Namur : Huitième école internationale d'été d'informatique de l'AFCEC, 10-22 Juillet
- Pierrel J.M. (1981)
Etude et mise en œuvre de contraintes linguistiques en compréhension automatique du discours continu
Nancy : Thèse d'état de l'Université de Nancy I
- Pierrel J.M. (1982)
Utilisation de contraintes linguistiques en compréhension automatique de la parole continue : le système Myrtille II
Technique et Science Informatiques, vol. 1, n° 5
- Pierrel J.M. (1987)
Dialogue oral homme-machine
Paris : Hermès
- Pierrel J.M. (1988)
Dialogue homme-machine en langage naturel écrit et oral
Paris : Actes des 1^{ères} journées du PRC Communication Homme-Machine, EC2 Editeur
- Pierrel J.M., Bennani M. (1984)
Prototype pour l'adjonction d'entrées-sorties vocales à un système d'EAO
Nancy : Rapport interne n° 84-R-041
- Pinson S. (1981)
Représentation des connaissances dans les systèmes experts
R.A.I.R.O. Informatique/Computer Science, vol. 15, n° 4
- Pister-Bourjot C., Haton J.P. (1987)
Automatic Learning: An Approach to the Adaptation of a Speech Recognition System to One or Several Speakers
Speech Communication, vol. 6, n° 1
- Pullum G.K. (1987)
Natural Language Interfaces and Strategic Computing
AI & Society, vol. 1, n° 1

- Quinton P. (1980)
Contribution à la reconnaissance de la parole. Utilisation d'heuristiques pour la reconnaissance de phrases.
Rennes : Thèse d'état de l'Université de Rennes
- Quinton P. (1982)
Utilisation de contraintes syntaxiques pour la reconnaissance de la parole continue
Technique et Science Informatiques, vol. 1, n° 3
- Rabiner L.R., Levinson S.E., Rosenberg A.E., Wilpon J.G. (1979)
Speaker-Independent Recognition of Isolated Words Using Clustering Techniques
IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-27, n° 4
- Rehm B.E. (1980)
Physical Access Control Using Automatic Speaker Verification
San Antonio, Texas : Conference Digest "Engineering & Society Interfaces", April 21-23
- Roussanaly A., Mousel P., Carbonell N., Mangeol B., Pierrel J.M. (1986)
Réalisation d'un corpus de dialogues oraux, application aux renseignements administratifs
Nancy : Rapport interne CRIN no. 86-R-083
- Roussanaly A. (1988)
DIAL : la composante dialogue d'un système de communication orale homme-machine finalisée en langage naturel
Nancy : Thèse de Doctorat de l'Université de Nancy I
- Rumelhart D.E., Norman D.A. (1973)
Active Semantic Networks as a Model of Human Memory
Stanford, Cal. : Proceedings of the International Joint Conference on Artificial Intelligence "IJCAI-73"
- Sabah G. (1988)
L'intelligence artificielle et le langage : représentations des connaissances
Paris : Hermès
- Sabah G. (1989)
L'intelligence artificielle et le langage: processus de compréhension
Paris : Hermès
- Saito S., Nakata K. (1985)
Fundamentals of Speech Signal Processing
Tokyo : Academic Press Japan, Inc.

- Sakoe H., Chiba S. (1978)
Dynamic Programming Algorithm Optimization for Spoken Word Recognition
 IEEE Transactions on Acoustics, Speech, and Signal Processing,
 vol. ASSP-26, n° 1
- Salkoff M. (1979)
Analyse syntaxique du Français : grammaire en chaîne
 Etudes en linguistique française et générale n° 2
 Amsterdam : John Benjamins Publishing Company
- Sammet J.E. (1966)
The Use of English as a Programming Language
 Communications of the ACM, vol. 9, n° 3
- Sapir E. (1953)
Le langage
 Paris : Payot
- Saussure F. de (1916)
Cours de linguistique générale
 Paris : Payot
 Rééd. : *Cours de linguistique générale*, Paris : Payot, 1974
- Scagliola C. (1983)
Continuous Speech Recognition without Segmentation: Two Ways of Using Diphones as Basic Speech Units
 Speech Communication, vol. 2, n° 2-3
- Scapin D. L.(1986)
Guide ergonomique de conception des interfaces homme-ordinateur
 Rapport INRIA
- Schalk T.B., Van Meir E.L. (1983)
Terminals, Listen up, Speech Recognition is a Reality
 Computer Design, September
- Schank R.C. (1972)
Conceptual Dependency: A Theory of Natural Language Understanding
 Cognitive Psychology, vol. 3
- Schank R.C. (1973)
Identification of Conceptualizations Underlying Natural Language
 in Schank et Colby (Eds.), *Computer Models of Thought and Language*
 San Francisco : W.H. Freeman and Company
- Schank R.C. (Ed.) (1975)
Conceptual Information Processing
 Amsterdam : North-Holland Publishing Company

- Schuler D. (1986)
A Responsible Computing Initiative - Rescuing Artificial Intelligence from the Military
The CPSR Newsletter, vol. 4, n° 4
- Schwarz R.M., Burger J.F., Simmons R.F. (1970)
A Deductive Question-Answerer for Natural Language Inference
Communications of the ACM, vol. 13, n° 3
- Sekey A. (1984)
Building a Model for Large Vocabulary Isolated Word Recognition
Speech Technology, August/September
- Shannon C.E., Weaver W. (1949)
A Mathematical Theory of Communication
Urbana, Ill. : University of Illinois Press
- Shapiro S.C., Kwasny S.C. (1975)
Interactive Consulting via Natural Language
Communications of the ACM, vol. 18, n° 8
- Shelley M. (1818)
Frankenstein or the Modern Prometheus
London
Trad. : *Frankenstein ou le Prométhée des temps modernes*, Paris :
Editions Albin Michel, 1984
- Silverman H.F. (1977)
An Introduction to Programming the Winograd Fourier Transform Algorithm (WFTA)
IEEE Transactions on Acoustics, Speech, and Signal Processing,
vol. ASSP-25, n° 2
- Simmons R.F. (1970)
Natural Language Question-Answering Systems: 1969
Communications of the ACM, vol. 13, n° 1
- Simmons R.F. (1973)
Semantic Networks: Their Computation and Use for Understanding English Sentences
in Schank et Colby (Eds.), *Computer Models of Thought and Language*
San Francisco : W.H. Freeman and Company
- Simmons R., Slocum J. (1972)
Generating English from Semantic Networks
Communications of the ACM, vol. 15, n° 10
- Siroux J., Gillet D. (1985)
A System For Man-Machine Communication Using Speech
Speech Technology, vol. 4, n° 4

- Siroux J., Mercier G. (1984)
Dix ans de communication orale et de dialogue entre l'homme et la machine avec le système KEAL
Nancy : Actes du séminaire GRECO "Dialogue oral homme-machine à composante orale", 11-12 Octobre
- Sisson N., Parkinson S.R., Snowberry K.
Considerations of Menu Structure and Communication Rate for the Design of Computer Menu Displays
International Journal of Man-Machine Studies, vol. 25
- Slocum J. (1986)
Machine Translation: An American Perspective
Proceedings of the IEEE, vol. 74, n° 7
- Sowa J.F., Way E.C. (1986)
Implementing a Semantic Interpreter Using Conceptual Graphs
IBM Journal of Research and Development, vol. 30, n° 1
- Spiegel (1989)
NSA: Amerikas großes Ohr
Der Spiegel, n° 8, Février
- Tanaka A., Togawa F., Ueda T., Hakaridani M., Iwahashi H., Nishioka Y., Kobayashi T., Kinpara S., Yamashita K. (1983)
A Study of Syllable Oriented Recognition of Continuous Speech
Speech Communication, vol. 2, n° 2-3
- Tappert C.C., Dixon N.R. (1974)
A Procedure for Adaptive Control of the Interaction Between Acoustic Classification and Linguistic Decoding in Automatic Recognition of Continuous Speech
Artificial Intelligence, vol. 5
- Tarnaud P. (1984)
Intégration du dialogue vocal dans un environnement réel complexe
Nancy : Actes du séminaire GRECO "Dialogue homme-machine à composante orale", 11-12 Octobre
- Thompson H. (1984)
Speech Transcription: An Incremental, Interactive Approach
in O'Shea T. (Ed.), *ECAI-84: Advances in Artificial Intelligence*
Amsterdam : Elsevier Science Publishers B.V. (North-Holland)
- Thompson K. (1968)
Regular Expression Search Algorithm
Communications of the ACM, vol. 11, n° 6
- Trubetzkoy N.S. (1939)
Grundzüge der Phonologie
Prague : Travaux du Cercle linguistique de Prague, VII
Trad. : *Principes de phonologie*, Paris : Klincksieck, 1949

- Tubach J.P. (1970)
Reconnaissance automatique de la parole
Grenoble : Thèse d'état de l'Université de Grenoble
- Ungeheuer G. (1972)
Sprache und Kommunikation
Hamburg : Buske
- Van Lamsweerde A. (1982)
Automatisation de la production de logiciels d'application : quelques approches. 1^{re} Partie
Technique et Science Informatiques, vol. 1, n° 6
- Van Lamsweerde A. (1983a)
Automatisation de la production de logiciels d'application : quelques approches. 2^{ème} Partie
Technique et Science Informatiques, vol. 2, n° 1
- Van Lamsweerde A. (1983b)
Automatisation de la production de logiciels d'application : quelques approches. 3^{ème} Partie
Technique et Science Informatiques, vol. 2, n° 2
- Vendler Z. (1967)
Linguistics in Philosophy
Ithaca : Cornell University Press
- Veyron M. (1982)
Un nègre blanc le cul entre deux chaises
Paris : Editions Albin Michel
- Vicens P.J. (1969)
Aspects of Speech Recognition by Computer
Stanford, Cal. : Ph.D. Dissertation, Stanford University
- Waltz D.L. (1978)
An English Language Question Answering System for a Large Relational Database
Communications of the ACM, vol. 21, n° 7
- Weigand H. (1986)
Natural Language Semantics in a Conceptual Language
Amsterdam : Papers of the Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam
- Weigand H. (1987)
Conceptual Models: Creation or Evolution
Amsterdam : Papers of the Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam

- Weischedel R.M. (1986)
Knowledge Representation and Natural Language Processing
Proceedings of the IEEE, vol. 74, n° 7
- Weizenbaum J. (1966)
ELIZA - A Computer Program For the Study of Natural Language Communication Between Man and Machine
Communications of the ACM, vol. 9, n° 1
- Weizenbaum J. (1967)
Contextual Understanding by Computers
Communications of the ACM, vol. 10, n° 8
- Weizenbaum J. (1972)
On the Impact of the Computer on Society
Science, vol. 176, n° 12
- Weizenbaum J. (1976)
Computer Power and Human Reason
San Francisco : W.H. Freeman and Company
Trad. : *Puissance de l'ordinateur et raison de l'homme*,
Boulogne-sur-Seine : Editions d'Informatique, 1981
- Weizenbaum J. (1984)
Kurs auf den Eisberg
Zürich : Pendo Verlag
- Williamson H., Rohlf S. (1981)
The User Interface Design Process
in Uhlig R.P. (Ed.), *Computer Message System*
Amsterdam : North-Holland Publishing Company/IFIP
- Winograd T. (1972)
Understanding Natural Language
New York : Academic Press
- Winograd T. (1983)
Language as a Cognitive Process
Reading, Mass. : Addison Wesley Publishing Company
- Winograd T., Flores F. (1986)
Understanding Computers and Cognition
Reading, Mass. : Addison-Wesley Publishing Company
- Winston P.H. (1984)
Artificial Intelligence
Reading, Mass. : Addison-Wesley Publishing Company, Second Edition

- Wolf J.J., Woods W.A. (1980)
The HWIM Speech Understanding System
in Lea W.A. (Ed.), *Trends in Speech Recognition*
Englewood Cliffs, N.J. : Prentice-Hall Inc.
- Woods W.A. (1970)
Transition Network Grammars for Natural Language Analysis
Communications of the ACM, vol.13, n° 10
- Woods W.A., Makhoul J. (1974)
Mechanical Inference Problems in Continuous Speech Understanding
Proceedings of the "International Joint Conference on Artificial
Intelligence", Stanford
- Young S.R., Hauptmann A.G., Ward W.H., Smith E.T., Werner P. (1989)
High Level Knowledge Sources in Usable Speech Recognition Systems
Communications of the ACM, vol. 32, n° 2
- Zemanek H. (1975)
***The Human Being and the Automation - Selected Aspects of a
Philosophy of Information Processing***
in E. Mumford et H. Sackman (Eds.), *Human Choice and Computers*
Amsterdam : North-Holland Publishing Company
- Zue V.W., Schwartz R.M. (1980)
Acoustic Processing and Phonetic Analysis
in Lea W.A. (Ed.), *Trends in Speech Recognition*
Englewood Cliffs, N.J. : Prentice-Hall Inc.

NOM DE L'ETUDIANT : MOUSEL Pierre

NATURE DE LA THESE : Doctorat de l'Université de NANCY I en Informatique



VU, APPROUVE ET PERMIS D'IMPRIMER

NANCY, le 23 MARS 1990 ~°641

LE PRESIDENT DE L'UNIVERSITE DE NANCY I



M. BOULANGE

Résumé

Cette thèse se propose d'exposer la composante syntaxico-sémantique d'un système de dialogue oral homme-machine finalisé en langage naturel (le système DIAL) actuellement en cours de développement au CRIN. Dans un premier temps, nous essayons de mettre en évidence les liens étroits qui existent entre l'informatique et les sciences de la langue. Une deuxième partie nous permet d'exposer les théories linguistiques desquelles notre travail s'inspire. La troisième partie est alors consacrée à notre réalisation, la composante syntaxico-sémantique (la composante SYNSEM) du système DIAL. La quatrième et dernière partie nous permet de faire le bilan et de présenter les perspectives ouvertes par notre travail.

Mots-clefs

Dialogue homme-machine ; Langage naturel ; Parole ; Syntaxe ; Sémantique

Syntaxe et sémantique dans un système de dialogue oral homme-machine finalisé en langage naturel



- Annexes -

Thèse

présentée et soutenue publiquement le 6 avril 1990

pour l'obtention du

Doctorat de l'Université de Nancy I

(Mention Informatique)

par

Pierre MOUSEL

Composition du Jury

Président :

Jean-Paul HATON

Rapporteurs :

Daniel COULON

Gérard SABAH

Directeur de thèse :

Jean-Marie PIERREL

Examineurs :

Jean-Claude ASSELBORN

Jacques BERLEUR

TABLE DES MATIERES

TABLE DES MATIERES	i
ANNEXE A : L'étude de corpus linguistiques, validation empirique de notre approche	1
1 Objectifs de l'étude de corpus linguistiques	1
2 Conditions de réalisation	2
3 Conventions de transcription	3
4 Exemple de transcription	4
5 Quelques statistiques sur le corpus CRIN	5
ANNEXE B : LADI, un outil d'aide au dépouillement de corpus	7
1 Objectifs de LADI	7
2 Utilisation de LADI	8
2.1 Restrcription du corpus	8
2.2 Partitionnement du corpus	8
2.3 Classification des transactions et des interventions du corpus	8
2.3.1 Syntaxe de la classification	9
2.3.2 Exemple de classification	11
2.4 Traitement de la classification	13
2.5 Consultation du corpus	14
2.5.1 Enchaînement du dialogue avec ladi	14
2.5.2 Les écrans de ladi	17
3 Exemple d'utilisation	25
ANNEXE C : Modélisation syntaxique du sous-langage	31
1 Symboles utilisés dans les RNP	31
1.1 Symboles non-terminaux (font référence à un sous-réseau de la grammaire)	31
1.2 Symboles terminaux	32
1.2.1 Catégories syntaxiques	32
1.2.2 Terminaux vrais	33
2 Les RNP	33
2.1 Le sous-réseau CENTAINE	33
2.2 Le sous-réseau CHIF	33
2.3 Le sous-réseau DATE	34
2.4 Le sous-réseau DECLAR	34
2.5 Le sous-réseau DIZAINE	35
2.6 Le sous-réseau ENONCE	35
2.7 Le sous-réseau GN	36
2.8 Le sous-réseau GN_INTER	36

2.9	Le sous-réseau GOI	37
2.10	Le sous-réseau GP	37
2.11	Le sous-réseau GR_ADJ	38
2.12	Le sous-réseau GR_COMP	38
2.13	Le sous-réseau GR_INTER	39
2.14	Le sous-réseau GR_THEM	39
2.15	Le sous-réseau GS	39
2.16	Le sous-réseau GV	40
2.17	Le sous-réseau INDIR	41
2.18	Le sous-réseau INFINITIVE	42
2.19	Le sous-réseau MILLIER	42
2.20	Le sous-réseau MILLION	43
2.21	Le sous-réseau PREST_1	43
2.22	Le sous-réseau PREST_2	44
2.23	Le sous-réseau PREST_INT	44
2.24	Le sous-réseau PROP	45
2.25	Le sous-réseau QUEST	45
2.26	Le sous-réseau RELAT	46
2.27	Le sous-réseau SGV	46
2.28	Le sous-réseau SUB_CIRC	47
ANNEXE D : Le programme CRERES		49
1	Objectif du programme CRERES	49
2	Syntaxe du texte traité par CRERES	50
3	Exemple d'utilisation	51
3.1	Les données	52
3.2	Les résultats	54
ANNEXE E : Modélisation sémantique du sous-langage		69
1	Symboles utilisés dans le modèle fonctionnel	69
1.1	Les traits sémantiques des primitives	69
1.2	Les primitives	70
1.2.1	Les actes	70
1.2.2	Les états	71
1.2.3	Les processus	71
2	Le modèle fonctionnel de notre sous-langage	72
2.1	La typologie des primitives	72
2.2	Les schémas de cas associés aux primitives	73
ANNEXE F : Le programme CRESEM		75
1	Objectif du programme CRESEM	75
2	Syntaxe du texte traité par CRESEM	76
3	Exemple d'utilisation	76
3.1	Les données	77
3.2	Les résultats	78
ANNEXE G : La génération automatique de nœuds procéduraux		83
1	Objectif du programme NPR	83
2	Objectif du programme DEFNPR	84
3	Utilisation des générateurs	84
4	Exemple d'utilisation	85
4.1	Résultat de DEFNPR	86
4.2	Post-traitement des résultats de DEFNPR	89
4.3	Résultat de NPR	90

ANNEXE H : Résultats de la composante SYNSEM	95
1 Objectif du prototype d'évaluation	95
2 Un exemple d'analyse	96
2.1 Les conditions de l'analyse	96
2.2 L'espace des théories construit lors de l'analyse	96
2.3 L'évolution de l'espace des théories	104
2.4 La chronologie des événements	107
3 Un autre exemple d'analyse	109
3.1 Les conditions de l'analyse	109
3.2 L'espace des théories construit	109
3.3 L'évolution de l'espace des théories	141
3.4 La chronologie des événements	144
4 Conclusions	145

ANNEXE A : L'étude de corpus linguistiques, validation empirique de notre approche

L'annexe A fournit un complément d'informations au sujet des corpus de dialogues oraux que nous avons été amenés à enregistrer dans le cadre du développement du système DIAL. Elle décrit les objectifs de ces corpus, les conditions de réalisation du corpus enregistré au CRIN, les conventions de transcription ainsi qu'un exemple de transcription d'un dialogue.

1 Objectifs de l'étude de corpus linguistiques

L'enregistrement de corpus de dialogues oraux dans le cadre d'une application telle que le système DIAL du CRIN poursuit deux objectifs :

- d'une part, les concepteurs d'un tel système doivent disposer d'un matériel linguistique qui leur permette d'effectuer certaines études (lexicales, syntaxiques, sémantiques, pragmatiques etc.) afin de définir les sources de connaissances à mettre en œuvre dans le système ;
- d'autre part, les réalisateurs ont besoin d'un ensemble de dialogues test en vue de la validation des modèles linguistiques et des choix de réalisation aux niveaux lexical, syntaxique, sémantique et pragmatique.

Nous avons par conséquent été amenés à enregistrer deux corpus linguistiques, un premier réalisé au CRIN (Roussanaly et al., 1986) ainsi qu'un

deuxième réalisé au Ministère de l'Emploi et du Travail à Bruxelles (Deville, 1987). Vu l'application test que nous avons choisie, les deux corpus concernent le renseignement administratif. Le corpus enregistré au CRIN doit être considéré comme une expérience pilote. Celle-ci nous a permis de tester un protocole de réalisation, de définir des conventions de transcription et de développer quelques outils d'exploitation de tels corpus. Nous n'allons pas décrire en détail ces deux corpus, cela a été fait dans (Roussanaly et al., 1986 ; Deville, 1987). Nous allons plutôt nous limiter à la description des conditions de réalisation de l'expérience pilote menée au CRIN¹, à la définition de nos conventions de transcription et à la présentation d'un exemple de dialogue enregistré au cours de l'expérience effectuée au CRIN et retranscrit suivant nos conventions.

2 Conditions de réalisation

Les membres du groupe de travail « Dialogue » au CRIN ont très tôt ressenti le besoin d'exploiter du matériel linguistique enregistré en situation réelle pour les raisons présentées ci-dessus. Malheureusement, devant la réticence de certains organismes, nous étions d'abord dans l'impossibilité d'enregistrer un corpus de dialogues oraux dont les interlocuteurs auraient été un expert réel dans un domaine administratif et un utilisateur ayant un besoin réel en renseignements. C'est la raison qui nous a amené à envisager dans un premier temps une expérience pilote effectuée au CRIN. Nos problèmes se sont ensuite réglés, nous avons obtenu les autorisations nécessaires pour une expérience à grande échelle en milieu réel au Ministère de l'Emploi et du Travail à Bruxelles.

La population visée par l'expérience pilote menée au CRIN était de ce fait fortement limitée. En effet, les seules personnes pouvant entrer en compte étaient les membres du CRIN. Leur participation bénévole a cependant permis d'enregistrer une quarantaine de conversations permettant ainsi une première série d'études significatives.

Nous avons réalisé deux séances d'enregistrement. Une première le 25 juin 1986 de 15.30h à 19.00h et une deuxième le 2 juillet 1986 de 9.00h à 12.00h et de 14.30h à 19.00h.

Quelques jours avant la première séance, toutes les personnes du CRIN ont été averties, par une note distribuée dans leurs casiers, de l'existence d'un centre de renseignements administratifs au CRIN en vue de la réalisation d'un corpus linguistique. Elles ont été invitées à téléphoner au poste 2505 du CRIN pour y exposer leurs problèmes administratifs actuels ou passés. Après la première séance d'enregistrement, deux notes différentes ont été distribuées, l'une s'adressant aux personnes qui n'avaient pas encore téléphoné et leur rappelant l'existence du centre de renseignement, l'autre s'adressant aux personnes qui avaient déjà téléphoné lors de la première journée et leur demandant d'encourager leurs collègues de travail de téléphoner au centre de renseignement.

¹ Comme nous n'avons pas participé personnellement à la réalisation du corpus enregistré au Ministère de l'Emploi et du Travail à Bruxelles, nous renvoyons le lecteur à (Deville, 1987) pour plus de détails au sujet de ce corpus.

Pour procéder aux enregistrements, nous avons utilisé un magnétophone TELEX (réglage d'enregistrement automatique, moniteur de contrôle) et une capsule intégrant un micro et pouvant être fixée à l'écouteur du téléphone.

L'opératrice était une personne possédant une certaine expérience dans le renseignement téléphonique mais pas dans le domaine d'application choisi. Il a été impossible d'obtenir une opératrice expérimentée dans le renseignement administratif en raison de la rareté d'une telle compétence : un centre de renseignements administratifs par région. L'opératrice volontaire a donc accepté de s'initier au domaine d'application et d'acquérir les connaissances contenues essentiellement dans les pages roses de l'annuaire, ce qui a représenté une journée de travail environ.

Compte tenu du caractère limité de l'expérimentation, nous n'avons pu procéder à aucun entraînement en situation réelle de l'opératrice. Ce fait explique la variabilité temporelle de son comportement qui se traduit par l'absence de reprises, d'hésitations et de tournures familières dans les premières conversations et par leur fréquence importante à partir de la deuxième demi-journée d'enregistrement. Cette évolution a pu influencer sur le comportement des locuteurs et doit être pris en compte au niveau de l'exploitation du corpus.

3 Conventions de transcription

Sur base des enregistrements sur cassette, nous avons procédé à une transcription orthographique en vue d'études linguistiques ultérieures. La transcription a été faite avec des conventions rigides de façon à faciliter un traitement automatique par la suite.

Les principales conventions sont les suivantes :

- les différentes transactions sont numérotées et commencent toutes par l'entête suivante :

TR: <num>

où TR: est un mot clef et <num> un entier, identifiant l'opératrice, suivi d'un point (.), d'un deuxième entier, identifiant la transaction pour cette opératrice, et d'un point ;

- toutes les interventions, celles de l'opératrice et celles des locuteurs, sont également numérotées et commencent par :

<num>

où <num> est le numéro de l'intervention, de l'opératrice ou du locuteur, dans la transaction courante ;

- les mots entre accolades (()) signalent des ambiguïtés d'interprétation ;

- les énoncés entre crochets ([]) signalent des énoncés non significatifs (en particulier la déclinaison de l'identité du locuteur qui, dans une application de ce type, n'a aucun sens, mais que l'opératrice a demandée à des fins de contrôle) ;
- le mot clef <S> signale un silence prolongé ;
- le mot clef <H> signale une courte hésitation ;
- le mot clef <?> signale une partie de texte qui n'a pu être interprétée ;
- le mot clef ..euh.. signale un hésitation prolongée ;
- les énoncés des différents intervenants dans une transaction ont été transcrits en parallèle (les énoncés de l'opératrice se trouvent systématiquement à gauche, ceux des locuteurs à droite) et sont séparés par une barre verticale (|) ;
- enfin, pour représenter les caractères accentués, nous avons utilisé des caractères spéciaux permettant le traitement automatique des accents, les correspondances étant les suivantes :

à	→	@
â	→	&
é	→	(
è	→)
ê	→	#
î	→	~
ï	→	`
ô	→	^
ù	→	+
û	→	%
ü	→	"
ç	→	\$

4 Exemple de transcription

L'exemple qui suit est la retranscription de la troisième transaction enregistrée lors de l'expérience pilote au CRIN.

TR:1.3.

```

<1> Renseignements administratifs. |
| <1> Allo bonjour madame, ..euh.. ma
| carte d'identit( est p(rim(e au
| premier juillet soixante seize
<2> Oui. |
| ..euh.. quatre vingt six plut^t
| donc ..euh.. il faudrait que je la
| remplace et je voudrais savoir o+
| je dois me renseigner.
<3> Pour obtenir une carte |
| d'identit( vous vous adressez |
| soit au commissariat de police |
| ..euh.. de votre domicile ou @ <2> Oui.
| la mairie de votre commune. |
| <3> A la mairie.
<4> Oui, vous souhaitez conna-tre |

```

les pi)ces @ fournir?		
		<4> Oui.
<5> Donc vous vous <H> munissez de		
votre carte p(rim(e		<5> Hem, hem.
deux photos d'identit(alors		
les photos d'ite <H>		
d'identit(c'est ..euh.. trois		
..euh.. trois virgule cinq par		
quatre centim)tres de face et		
t#te nue hein.		
		<6> D'accord.
<6> Et puis un timbre fiscal.		
		<7> Et le timbre fiscal il est de
		combien? Quelle somme?
<7> Cent quinze francs.		
		<8> Cent quinze francs.
<8> Oui.		
		<9> D'accord. Je vous remercie.
<9> D(sirez vous autre chose?		
		<10> Non, ça ira, merci.
<10> [Vous pouvez me donner vos		Au revoir.
coordonn(es s'il vous pla~t?)		
		<11> [Oui vingt sept z(ro un ..euh..
<11> [Le nom, le nom s'il vous		au CRIN.]
pla~t.]		
		<12> [Mademoiselle F.]
<12> [Merci beaucoup.]		
		<13> [De rien.]
<13> Au revoir.		Au revoir.

5 Quelques statistiques sur le corpus CRIN

Nous présentons ci-dessous quelques chiffres bruts qui donnent des indications sur la représentativité du corpus enregistré au CRIN, des résultats plus détaillés sont fournis dans (Roussanaly et al., 1986) :

nombre de transactions du corpus	: 42
nombre d'énoncés de l'opératrice	: 555
nombre d'énoncés des locuteurs	: 598
nombre de mots du corpus	: 13136
nombre de mots du lexique (formes différentes)	: 1377
nombre moyen d'énoncés par transaction	: 27
nombre de mots du corpus (locuteurs)	: 5848
nombre de mots du lexique (loc.)	: 868
nombre moyen d'énoncés par transaction (loc.)	: 14
nombre de mots du corpus (opératrice)	: 7288
nombre de mots du lexique (op.)	: 937
nombre moyen d'énoncés par transaction (op.)	: 13

ANNEXE B : LADI, un outil d'aide au dépouillement de corpus

L'annexe B fournit un complément d'informations au sujet du Logiciel d'Aide au Dépouillement de corpus Interactif LADI. Outre un bref rappel des objectifs de cet outil, elle décrit de façon détaillée l'utilisation de LADI ainsi que quelques exemples.

1 Objectifs de LADI

LADI est un outil destiné à assister un linguiste dans le dépouillement de corpus, tâche souvent fastidieuse mais nécessaire dans le cadre du développement d'applications informatiques traitant le langage naturel¹. Dans le cadre du développement du système DIAL, nous avons été amené à étudier des corpus de conversations entre un opérateur et un usager. LADI est un outil interactif de consultation facilitant l'étude linguistique des interventions dans de tels dialogues. La consultation des interventions se fait en contexte, le contexte étant défini par la combinaison libre de plusieurs facteurs tels qu'un intervenant, une transaction, une intervention, un sujet de transaction, un type d'intervention ou un mot (respectivement une racine de mot). Pour pouvoir utiliser LADI correctement, l'utilisateur doit effectuer certaines opérations.

¹ Signalons que cet outil a été réalisé sur demande pour un linguiste, G. Deville, qui l'a utilisé afin d'étudier des corpus de dialogues réalisés dans le cadre du projet DIAL.

2 Utilisation de LADI

2.1 Restrcription du corpus

La première étape par laquelle l'utilisateur doit passer obligatoirement est la retranscription manuelle du corpus de conversations. Les conventions de retranscription de corpus de dialogues oraux sont définies dans (Mousel et Roussanaly, 1988) et rappelées dans l'annexe A.

2.2 Partitionnement du corpus

Le résultat de cette retranscription est un fichier séquentiel dont la taille est généralement assez importante si le corpus est représentatif d'un point de vue linguistique. Afin de permettre une consultation rapide du corpus, ce fichier doit être divisé en un ensemble de fichiers plus petits (voir figure B.1). Cette séparation est effectuée automatiquement par un premier programme, le programme « separ ». Le programme « separ » crée un fichier par transaction et par intervenant (opérateur ou locuteur) du corpus. Les noms des fichiers résultats sont obtenus par combinaison du nom du fichier contenant le corpus, de l'identification de la transaction et de l'identification de l'intervenant. Chaque fichier est constitué d'une suite d'enregistrements formés de deux parties, un numéro d'intervention ainsi que la retranscription de l'intervention.

C'est ainsi, par exemple, que le fichier contenant les interventions du locuteur ayant conversé avec le premier opérateur lors de la cinquième transaction de cet opérateur va s'appeler `crinTR:1.5.1oc` si le corpus était retranscrit sur le fichier `crin`.

Le programme « separ » prend comme unique paramètre le nom du fichier contenant le corpus à partitionner. Exemple d'utilisation du programme « separ » sous UNIX :

```
$ separ crin
$
```

2.3 Classification des transactions et des interventions du corpus

Une deuxième opération manuelle que l'utilisateur doit effectuer est la classification des transactions et des interventions du corpus retranscrit. La classification tient compte du fait qu'une transaction peut concerner plusieurs « sujets » et qu'une intervention est d'un « type » donné.

L'utilisateur peut définir les sujets et les types qu'il juge pertinents. La classification donne lieu à un texte qui obéit à une syntaxe déterminée.

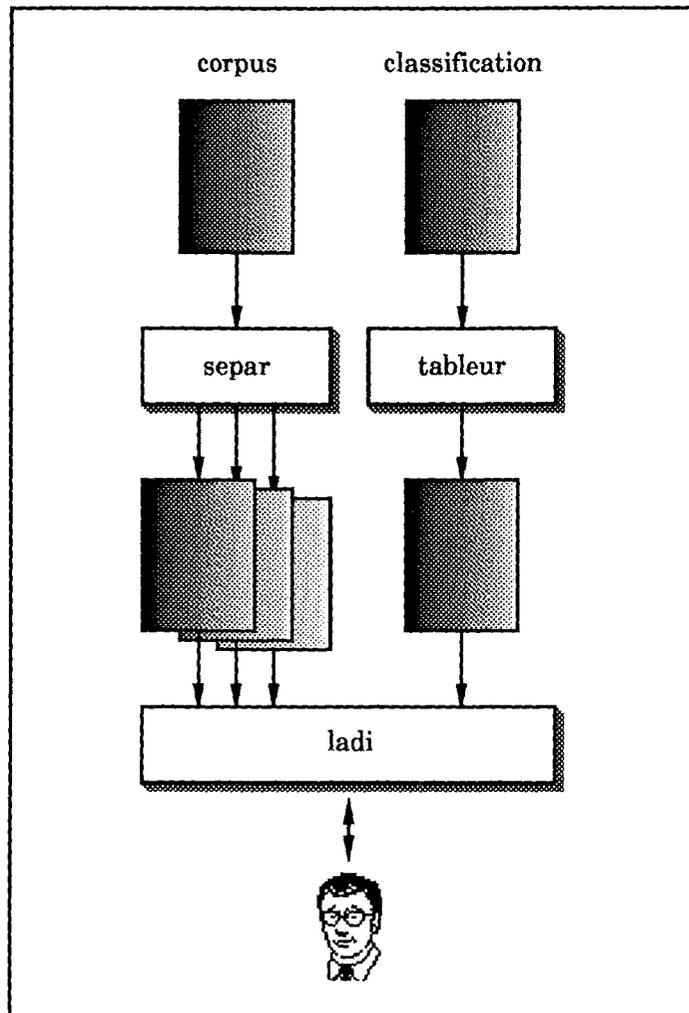


fig. B.1 : Utilisation de LADI

2.3.1 Syntaxe de la classification

Pour décrire formellement cette syntaxe, nous avons utilisé les conventions suivantes :

- les non-terminaux sont entre crochets '<' et '>';
- les terminaux sont en caractères gras ou entre simples quotes ;
- la production est indiquée par '::=' ;
- les alternatives de production sont indiquées par '|' ;
- les éléments optionnels d'une production sont indiqués entre crochets '[' et ']' ;

- la fin d'une règle est signalée par un point-virgule ';'.

La syntaxe que nous avons élaborée est la suivante :

```

<texte classification> ::=
    <déclaration sujets> <déclaration types>
    [<entête>] <liste classifications>;

<déclaration sujets> ::=
    Sujets-possibles: <liste sujets>;

<déclaration types> ::=
    Types-possibles: <liste types>;

<entête> ::=
    Transaction Sujet Intervention Type-de-question;

<liste classifications> ::=
    <classification> |
    <liste classifications> <classification>;

<liste sujets> ::=
    Identificateur |
    <liste sujets> Identificateur;

<liste types> ::=
    Identificateur |
    <liste types> Identificateur;

<classification> ::=
    '>' <expert> '.' <transaction> '.' <sujet> <typologies>;

<expert> ::=
    Entier;

<transaction> ::=
    Entier;

<sujet> ::=
    Identificateur;

<typologies> ::=
    <typologie> |
    <typologies> <typologie>;

<typologie> ::=
    <liste interventions> <type>;

<liste interventions> ::=
    Entier |
    <liste interventions> Entier;

<type> ::=
    Identificateur;

```

Les terminaux peuvent s'écrire indépendamment en majuscules ou en minuscules. Certains sont des mots clés, il s'agit de :

- **Sujets-possibles:** Ce mot clé peut encore s'écrire **SP** ;
- **Types-possibles:** Ce mot clé peut encore s'écrire **TP** ;
- **Transaction** Ce mot clé peut encore s'écrire **Tr** ;
- **Sujet** ;

INFO COGN SPA			
TR	SUJET	INTERVENTION	TYPE-DE-QUESTION
>1.1.	NAT	1	REQ
		2	TEMP
>1.2.	CIV	2	TEMP
>1.3.	CID	1	REQ
		1	LIEU
		1	TOPIC
		7	COUT
>1.4.	CID	1	TOPIC
		1 13	FORMA
>1.5.	CID	1	TOPIC
		4	FORMA
>1.6.	CID	2	FORMA
>1.7.	TER	1 2	TOPIC
		4	MVMT
		4	SPA
		7	STAT
		11	FORMA
		12	TEMP
>1.8.	DEC	6	FORMA
>1.9.	MAR	1	STAT
		2	STAT
		2	TEMP
		4 6	FORMA
>1.10.	JUD	1	STAT
		1	TOPIC
		1 7	REQ
>1.11.	CID	1	STAT
		1	TOPIC
		1 3	FORMA
		4	STAT
>1.12.	CIV	1	TOPIC
		2	TEMP
>1.13.	VOT	1	TOPIC
		1 5	FORMA
>1.14.	FIS	2	TOPIC
		2	FORMA
>1.15.	CID	1	TOPIC
		1	STAT
		1	FORMA
>1.16.	CAD	1	REQ
		1	TOPIC
		3	TEMP
>1.17.	VOT	1	TOPIC
		1	STAT
		1	FORMA
		4	TEMP
		12	LIEU
>1.19.	HLM	1	COND
		6 7	LIEU
>1.20.	CID	4	STAT
		4	LIEU
>1.21.	CID	1	TOPIC
>1.22.	CID	1	TOPIC
		4 22	FORMA
		4 25 27	TEMP
		7	LIEU
>1.23.	NAT	1	TOPIC
		1	STAT
		5	FORMA
		5	COND
>1.24.	CID	1	TOPIC
		2 22	TEMP
		11 15	REQ
>1.25.	CID	2	TOPIC
		3	STAT

		4	FORMA
>1.26.	CID	1	MVMT
		1	FORMA
		3	STAT
>1.27.	CID	4	TOPIC
		16	FORMA
>1.28.	CID	1	TOPIC
		2	STAT
		3 6	FORMA
>1.31.	BAP	1	MVMT
		1 4 5	COND
		1 2	STAT
		8	REQ
>1.32.	CID	1	TOPIC
		1	STAT
		1	FORMA
		5	REQ
>1.33.	MAR	1	TOPIC
		1 3	FORMA
>1.34.	CID	5	TOPIC
		8	FORMA
		12	TEMP
		15	REQ
>1.35.	CID	2	STAT
		3 7	FORMA
>1.36.	CIV	2	TOPIC
		2	FORMA
		3	COUT
>1.39.	CID	1	TOPIC
		1 4	FORMA
>1.40.	NAI	1	TOPIC
		2	FORMA
>1.41.	JUD	3	TOPIC
		4 9 11	FORMA
		12	COUT
>1.42.	NAI	2	STAT
		3	LIEU
		8	FORMA

2.4 Traitement de la classification

La classification effectuée par l'utilisateur est à enregistrer sur un fichier. Si le texte de ce fichier respecte la syntaxe définie, il peut être traité par un deuxième programme, le programme « tableur » (voir figure B.1). Le programme « tableur » transforme ce texte en une structure de données interne qu'il enregistre sur un fichier dont le nom est le nom du fichier contenant la classification suffixé de « .tableur ». Il prend comme paramètre le nom du fichier contenant la classification et peut être utilisé avec deux options :

- l cette option fournit un listing source sur le fichier standard en sortie ;
- d cette option fournit, sur le fichier standard en sortie, une version lisible de la structure de données construite par « tableur ».

Exemple d'utilisation du programme « tableur » sous UNIX :

```
$ tableur action-travail
$
```

2.5 Consultation du corpus

Si l'utilisateur a retranscrit son corpus et traité la retranscription avec « separ » et s'il a classifié les interventions du corpus et traité la classification avec « tableur », alors il peut passer à la phase de consultation proprement dite du corpus (voir figure B.1). Le programme « ladi » se présente sous la forme d'un logiciel interactif. Il permet de consulter des corpus linguistiques sur base de certains critères d'accès. Le but est de visualiser les interventions du corpus qui répondent à ces critères de sélection. Les critères d'accès actuellement prévus et définis sur base de conventions de transcription sont :

- l'identification d'un opérateur (un numéro) ;
- l'identification d'une transaction (un numéro) ;
- l'identification d'une intervention (un numéro) ;
- le sujet d'une transaction (un identificateur) ;
- le type d'une intervention (un identificateur) ;
- un mot clé devant être contenu dans l'intervention (une suite de caractères)

La recherche peut se faire sur aucun, un ou plusieurs des critères de sélection. Les interventions sélectionnées sont affichées à l'écran. L'utilisateur a la possibilité d'enregistrer les interventions sélectionnées sur un fichier de façon à garder une trace de la consultation. L'exécution du programme « ladi » est lancée par la commande suivante sous UNIX :

```
$ ladi
```

2.5.1 Enchaînement du dialogue avec ladi

La figure B.2 montre l'enchaînement du dialogue sous ladi. Les noeuds de la figure représentent les différents écrans qui sont présentés à l'utilisateur. Les flèches indiquent les conditions de passage d'un écran à l'autre. Les conditions de transition sont les suivantes :

- **continue** : l'utilisateur a répondu avec un caractère quelconque ;
- **tableur inexistant** : il n'existe pas de fichier dont le nom est celui entré au clavier par l'utilisateur ;
- **non type tableur** : le nom de fichier introduit au clavier par l'utilisateur n'est pas suffixé par « .tableur » ;
- **tableur OK** : le fichier dont le nom a été introduit au clavier par l'utilisateur existe et son nom possède le suffixe « .tableur » ;
- **corpus inexistant** : il n'existe pas de fichier dont le nom est celui entré au clavier par l'utilisateur ;

- **trace** : l'utilisateur a répondu « o » ou « O » à la question posée ;
- **non trace** : l'utilisateur a répondu « n » ou « N » à la question posée ;
- **erreur trace** : le fichier dont le nom a été introduit au clavier par l'utilisateur n'a pas pu, pour une raison quelconque, être créé ;
- **trace OK** : le fichier dont le nom a été introduit au clavier par l'utilisateur a pu être créé ;
- **consultation suivante** : l'utilisateur a répondu « o » ou « O » à la question posée ;
- **erreur sujet** : la chaîne de caractères entrée au clavier par l'utilisateur ne correspond pas à un sujet de la classification ;
- **erreur type** : la chaîne de caractères entrée au clavier par l'utilisateur ne correspond pas à un type de la classification ;
- **sujet OK** : la chaîne de caractères entrée au clavier par l'utilisateur correspond à un sujet de la classification ;
- **type OK** : la chaîne de caractères entrée au clavier par l'utilisateur correspond à un type de la classification ;
- **critères OK** : les critères entrés au clavier par l'utilisateur sont valides ;
- **suivant** : l'utilisateur a répondu « o » ou « O » à la question posée ;
- **non suivant** : l'utilisateur a répondu « n » ou « N » à la question posée ;
- **arrêt** : l'utilisateur a répondu « n » ou « N » à la question posée ;

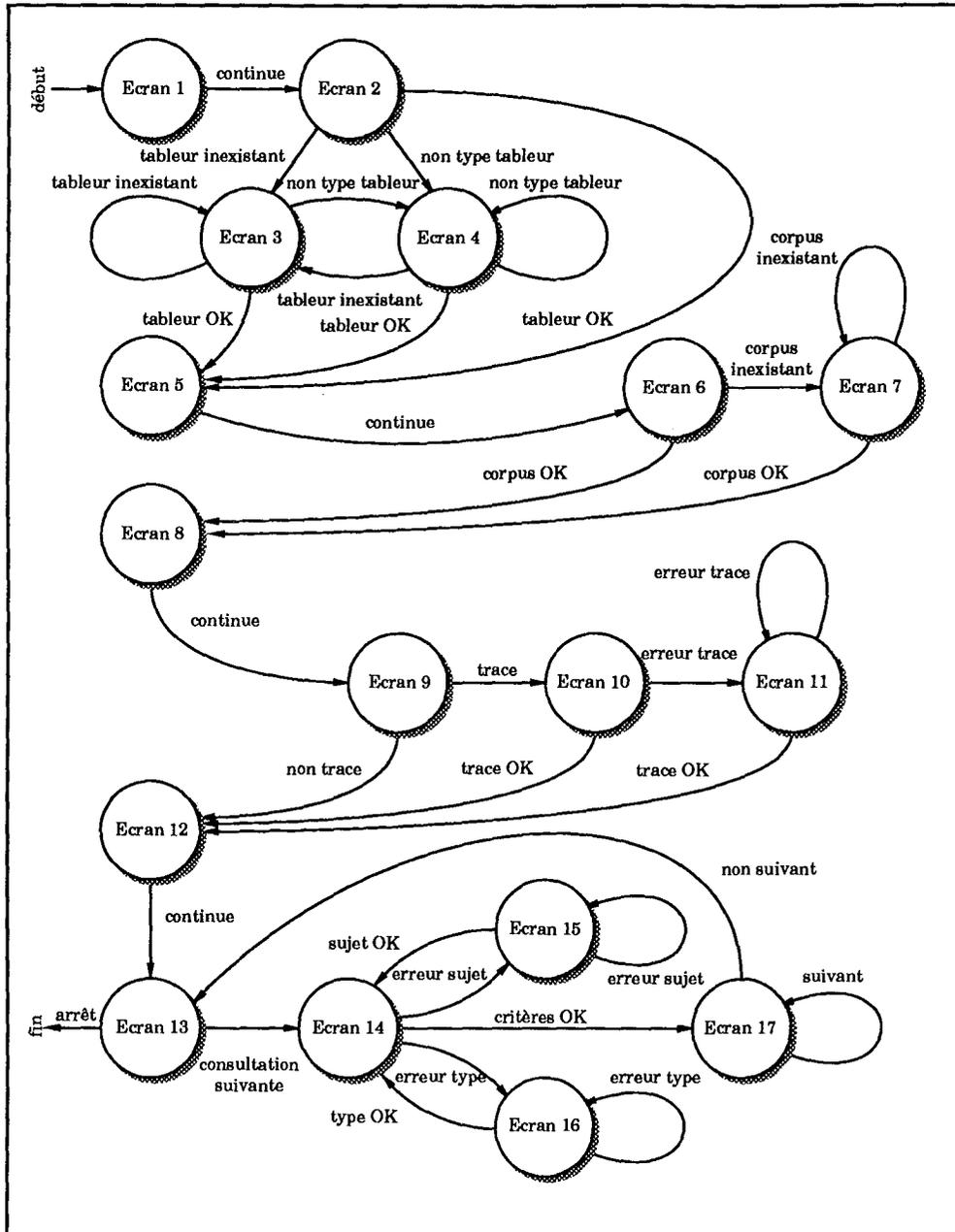


fig. B.2 : Enchaînement du dialogue avec ladi

Les écrans présentés à l'utilisateur sont montrés sur les figures B.3 à B.19.

2.5.2 Les écrans de ladi

```
*****
*
*          CONSULTATION DE CORPUS          *
*          -----                          *
*
* Ce programme permet de faire de la consultation de corpus. *
* Pour cela le corpus a du etre saisi avec les conventions du *
* CRIN (rapport interne 86-R-083). Il a du etre classifie, *
* classification qui a du etre traitee par le compilateur *
* 'tableur', et separe, separation qui a du etre realisee par *
* le programme 'separ' (rapport interne 88-R-024). *
*
*
*          (c) 1988 Roussanaly-Mousel      *
*
*          version 1.0                      *
*
*****

Pour continuer, tapez <RETURN> █
```

fig. B.3 : Ecran 1

```
-----
|          CHARGEMENT DU TABLEUR          |
-----

Quel est le nom du fichier contenant le tableur: █
```

fig. B.4 : Ecran 2

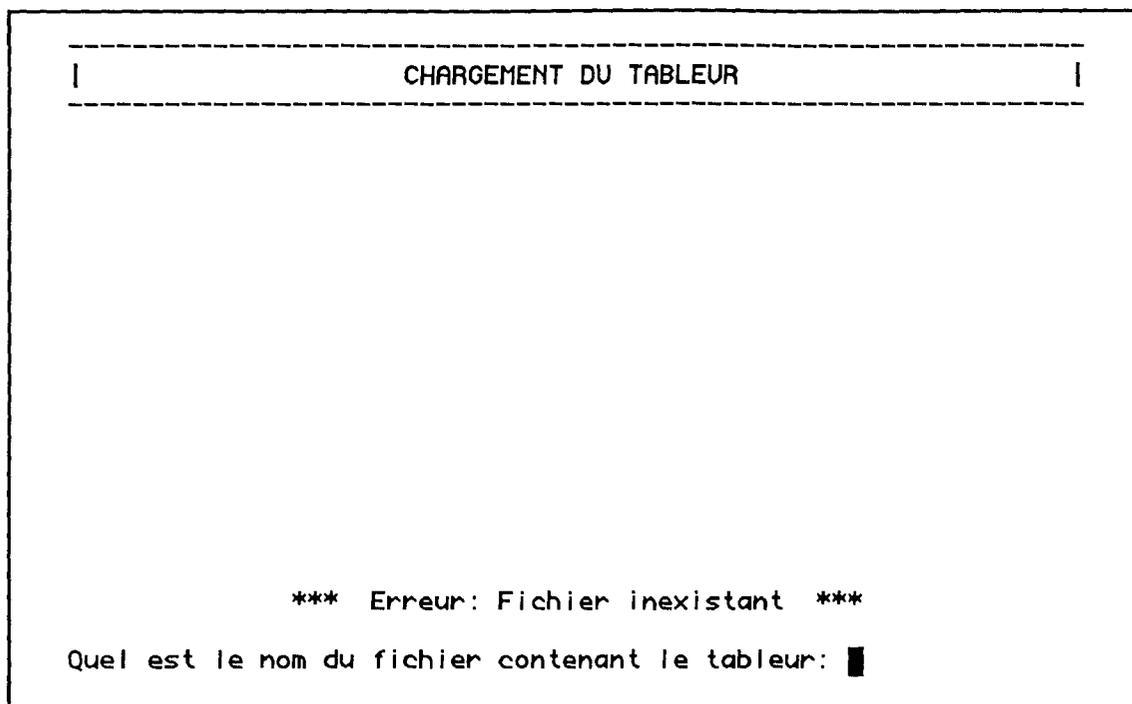


fig. B.5 : Ecran 3

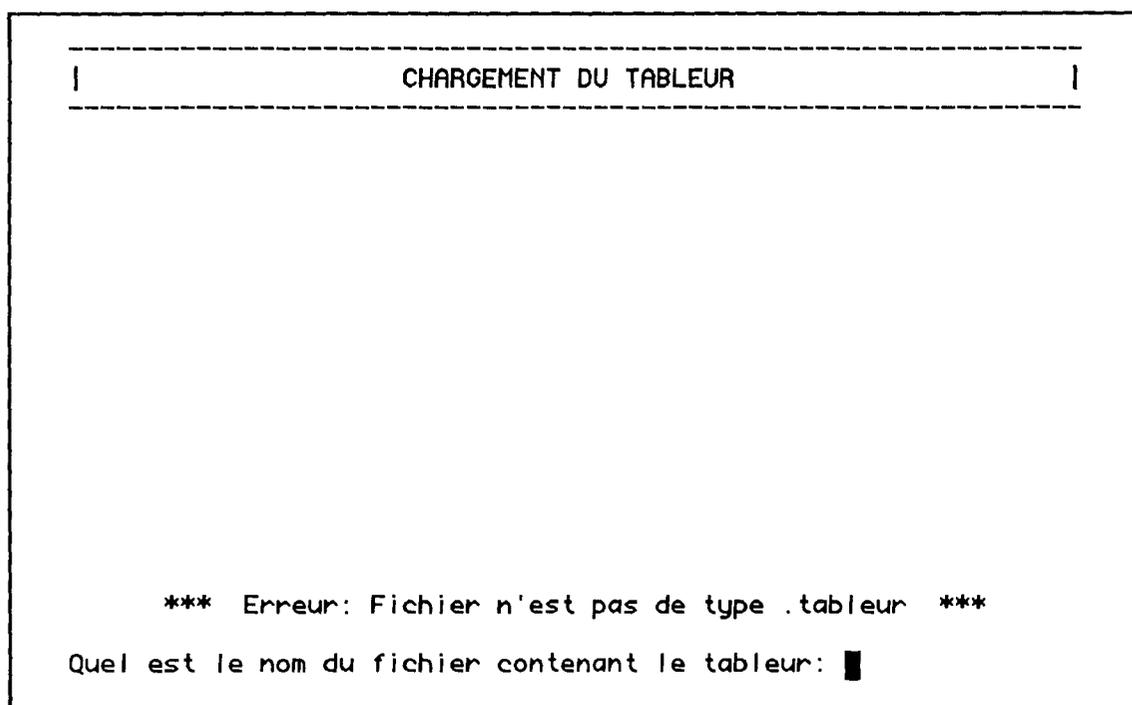


fig. B.6 : Ecran 4

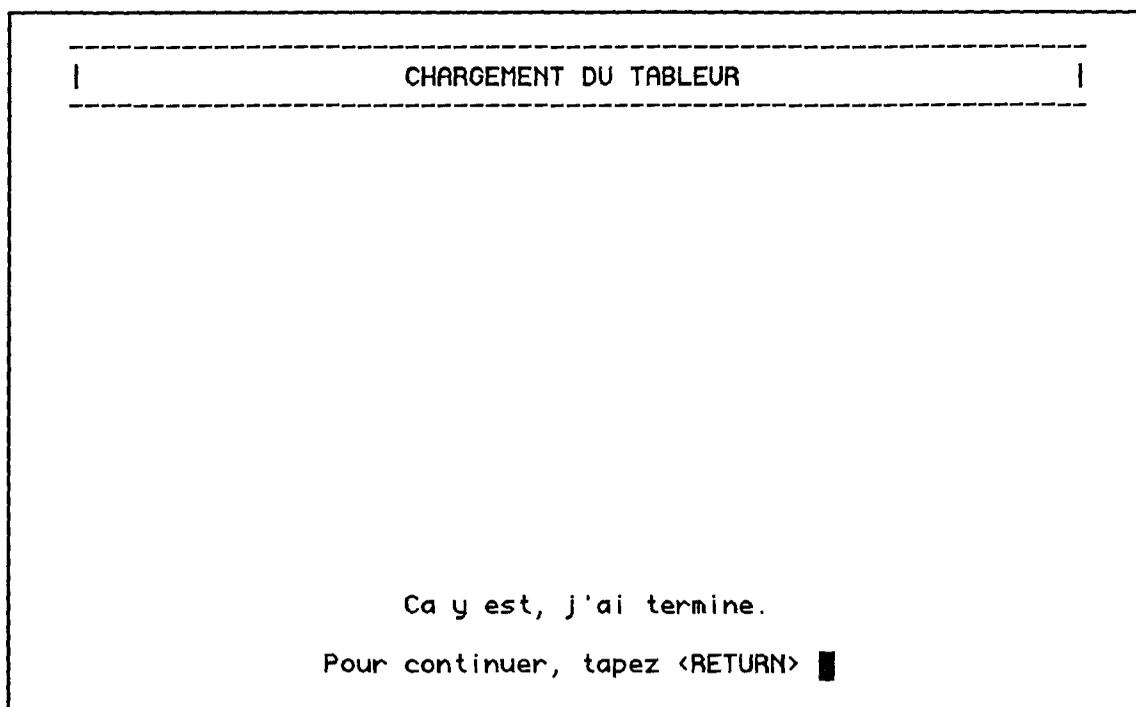


fig. B.7 : Ecran 5

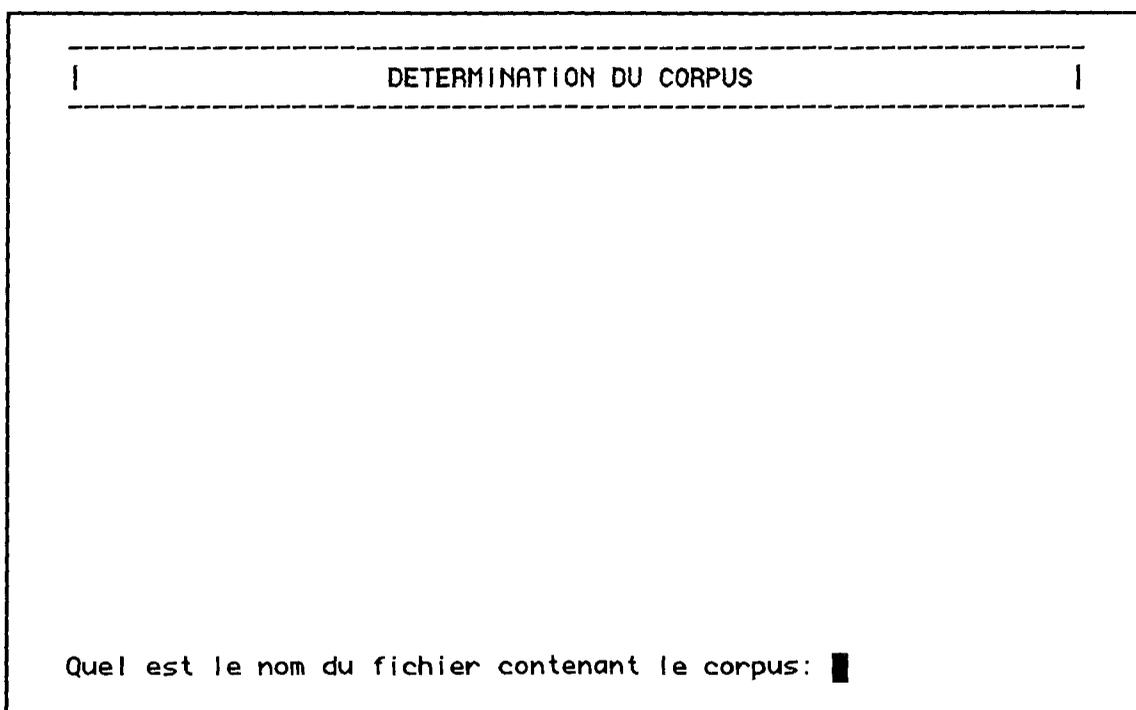


fig. B.8 : Ecran 6

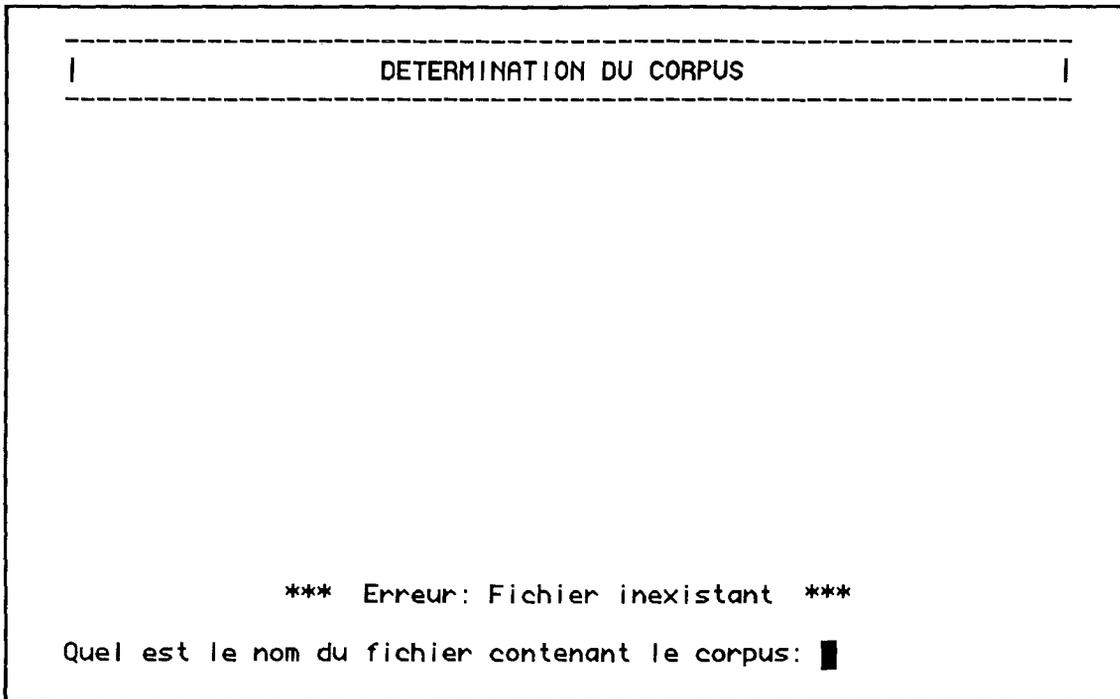


fig. B.9 : Ecran 7

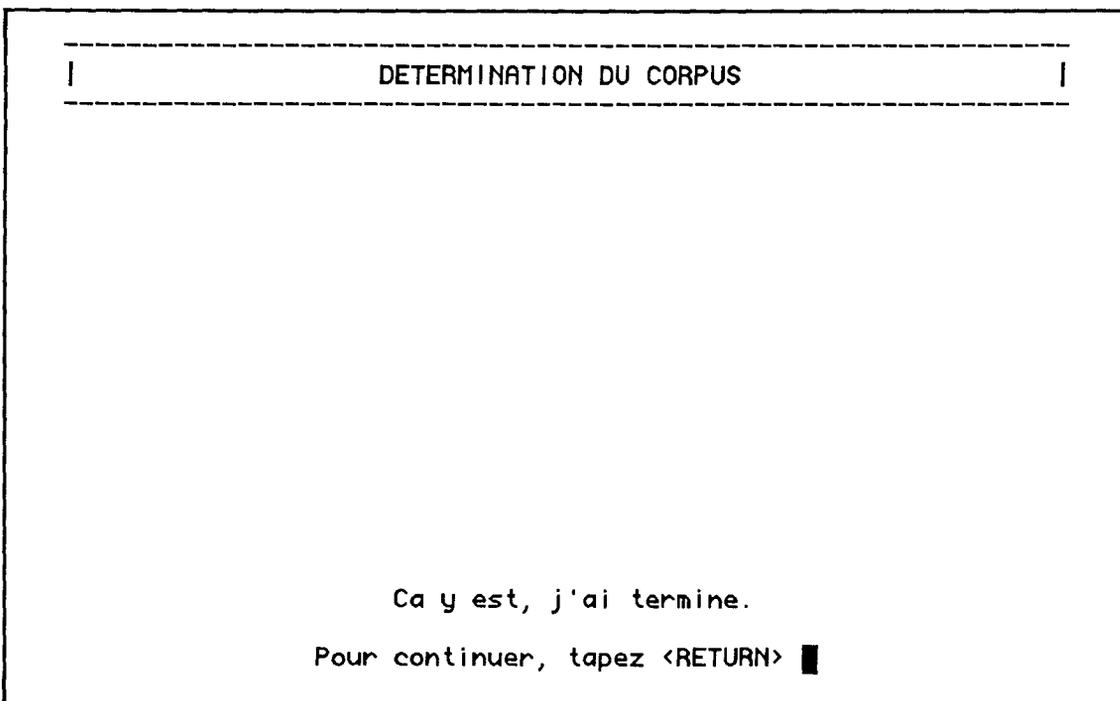
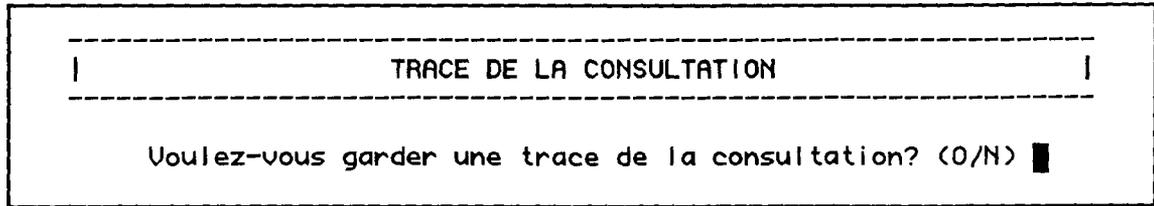


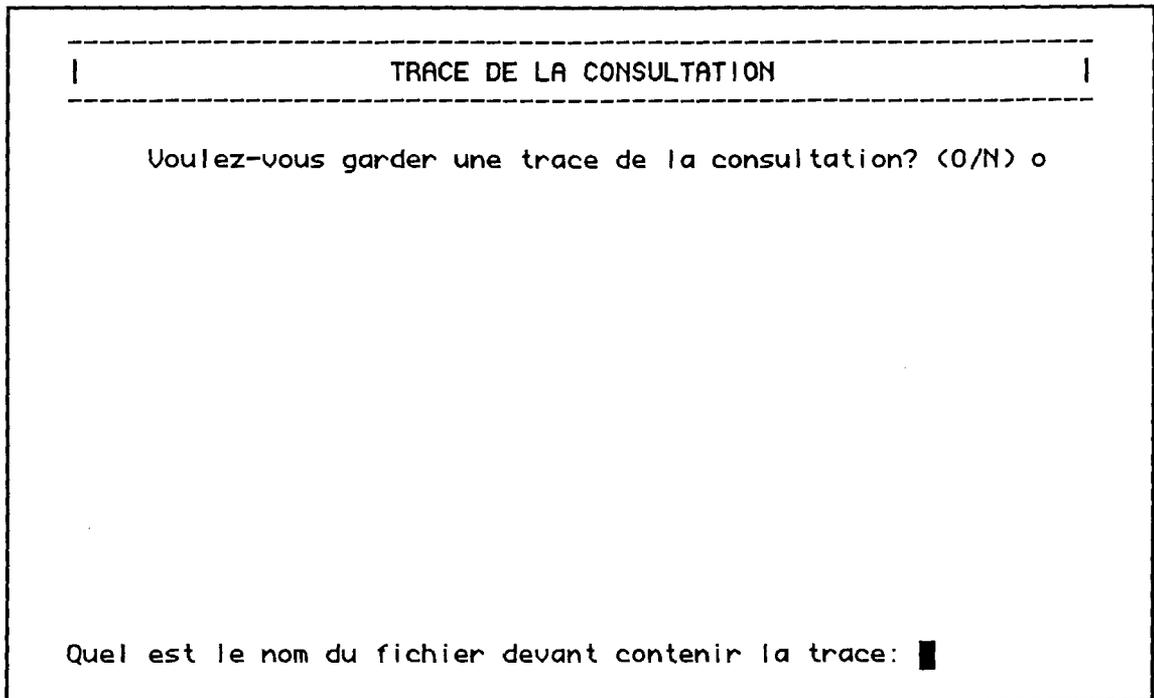
fig. B.10 : Ecran 8



| TRACE DE LA CONSULTATION |

Voulez-vous garder une trace de la consultation? (O/N) █

fig. B.11 : Ecran 9



| TRACE DE LA CONSULTATION |

Voulez-vous garder une trace de la consultation? (O/N) o

Quel est le nom du fichier devant contenir la trace: █

fig. B.12 : Ecran 10


```
-----  
| CONSULTATION DE CORPUS |  
-----  
Voulez-vous continuer la selection? (O/N) █
```

fig. B.15 : Ecran 13

```
-----  
| CONSULTATION DE CORPUS |  
-----  
Voulez-vous continuer la selection? (O/N) o  
-----  
Numero de l'expert      : █  
Numero de la transaction :  
Numero de l'intervention :  
Sujet de l'intervention :  
Type de l'intervention  :  
Kwic index              :  
-----
```

fig. B.16 : Ecran 14

```
-----  
| CONSULTATION DE CORPUS |  
-----  
Voulez-vous continuer la selection? (O/N) o  
-----  
Numero de l'expert      :  
Numero de la transaction :  
Numero de l'intervention :  
Sujet de l'intervention : █  
Type de l'intervention  :  
Kwic index              :  
-----  
  
*** Erreur: Sujet inexistant ***
```

fig. B.17 : Ecran 15

```
-----  
| CONSULTATION DE CORPUS |  
-----  
Voulez-vous continuer la selection? (O/N) o  
-----  
Numero de l'expert      :  
Numero de la transaction :  
Numero de l'intervention :  
Sujet de l'intervention  :  
Type de l'intervention   : █  
Kwic index              :  
-----  
  
*** Erreur: Type inexistant ***
```

fig. B.18 : Ecran 16

```
-----  
| CONSULTATION DE CORPUS |  
-----  
Voulez-vous continuer la selection? (O/N) o  
-----  
Numero de l'expert      :  
Numero de la transaction :  
Numero de l'intervention :  
Sujet de l'intervention  :  
Type de l'intervention   :  
Kwic index              :  
-----  
Expert: 1, Transaction: 1, Intervention: 2, Sujet: PCR Type: STAT  
Nombre d'onces restant a consulter: 498  
-----  
je beneficie d'une interruption de carriere, hein  
  
-----  
Suivant? (O/N) █
```

fig. B.19 : Ecran 17

3 Exemple d'utilisation

Dans ce paragraphe, nous présentons un exemple complet illustrant l'utilisation de ladi, nous montrons un exemple de dialogue suivi d'un exemple de trace construite par ladi.

- Dès qu'il a lancé le programme ladi, l'utilisateur voit apparaître l'écran de la figure B.20.
- Il doit ensuite indiquer quel est le fichier qui contient la classification du corpus qu'il désire consulter. Ceci est réalisé sur l'écran de la figure B.21. Ce fichier est alors chargé par le programme ladi, le succès de cette opération étant signalé par l'écran de la figure B.22.
- Une fois le chargement terminé, l'utilisateur doit fournir au programme ladi le nom du fichier qui contient le corpus non séparé. Sur l'exemple de la figure B.23, c'est le corpus enregistré au CRIN (concernant les pages roses) qui va être consulté. Si ce corpus existe, l'utilisateur en est averti par l'écran de la figure B.24.
- L'utilisateur peut ensuite spécifier s'il désire garder une trace de la consultation et si oui, sur quel fichier cette trace devra être enregistrée. L'écran de la figure B.25 indique que le fichier trace s'appellera « trace ». Si la trace a pu être créée, l'utilisateur verra apparaître l'écran de la figure B.26.
- Il peut ensuite passer à la consultation proprement dite. Dans le cas de l'écran de la figure B.27, l'utilisateur veut consulter les interventions des locuteurs ayant parlé à l'expert numéro un, dont le sujet est la carte d'identité et qui portent sur un coût. Le programme ladi signale à l'utilisateur qu'il existe une telle intervention en fournissant les indications permettant de localiser très exactement cette intervention dans le corpus. De plus, ladi communique à l'utilisateur le nombre d'interventions répondant aux mêmes critères et restant à consulter. L'utilisateur peut alors décider de les consulter un à un ou d'interrompre la consultation de cette liste d'interventions.
- S'il arrive qu'aucune intervention ne réponde aux critères exigés, l'utilisateur voit s'afficher l'écran de la figure B.28. Dans le corpus CRIN, il semble qu'il n'existe aucune intervention dont le sujet soit la carte d'identité et qui contienne la chaîne de caractères « merci ».

```

*****
*
*          CONSULTATION DE CORPUS
*          -----
*
*   Ce programme permet de faire de la consultation de corpus.
*   Pour cela le corpus a du etre saisi avec les conventions du
*   CRIN (rapport interne 86-R-083). Il a du etre classifie,
*   classification qui a du etre traitee par le compilateur
*   'tableur', et separe, separation qui a du etre realisee par
*   le programme 'separ' (rapport interne 88-R-024).
*
*
*          (c) 1988 Roussanaly-Mousel
*
*          version 1.0
*
*****

Pour continuer, tapez <RETURN> █

```

fig. B.20 : Lancement de ladi

```

-----
|          CHARGEMENT DU TABLEUR          |
-----

Quel est le nom du fichier contenant le tableur: p_rose.tableur█

```

fig. B.21 : Chargement du tableur

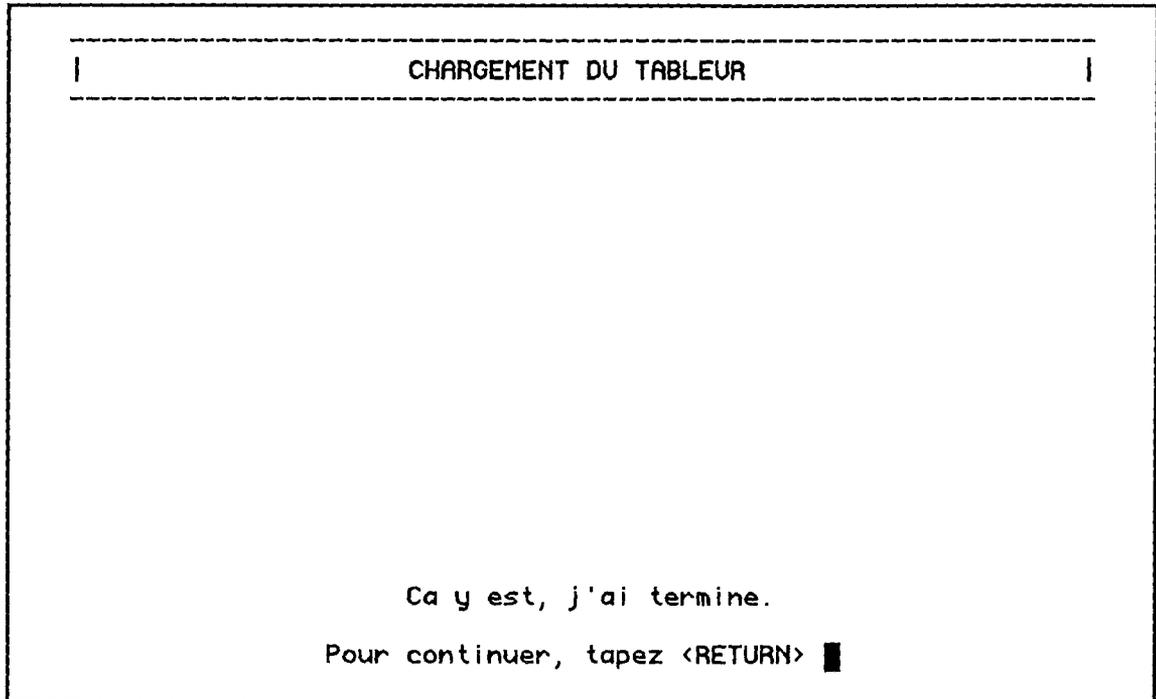


fig. B.22 : Fin du chargement du tableur

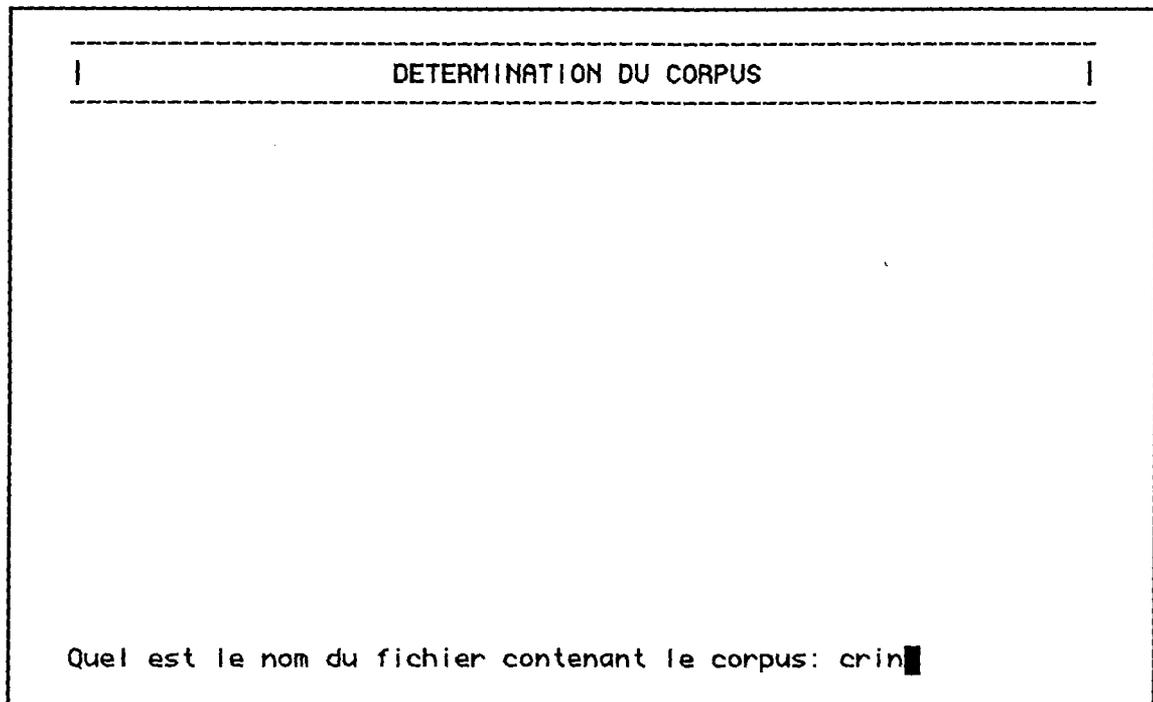


fig. B.23 : Détermination du corpus

DETERMINATION DU CORPUS	
<p>Ca y est, j'ai termine.</p> <p>Pour continuer, tapez <RETURN> █</p>	

fig. B.24 : Fin de la détermination du corpus

TRACE DE LA CONSULTATION	
<p>Voulez-vous garder une trace de la consultation? (O/N) o</p>	
<p>Quel est le nom du fichier devant contenir la trace: trace█</p>	

fig. B.25 : Détermination de la trace


```

-----
|                CONSULTATION DE CORPUS                |
-----
Voulez-vous continuer la selection? (O/N) o
-----
Numero de l'expert      :
Numero de la transaction :
Numero de l'intervention :
Sujet de l'intervention : CID
Type de l'intervention  :
Kwic index              : merci
-----

Aucun enonce n'a pu etre selectionne.

Pour continuer, tapez <RETURN> █

```

fig. B.28 : Sélection non réussie

Lorsque l'utilisateur a terminé la consultation du corpus qui l'intéresse, il arrête l'exécution du programme ladi et peut retrouver le résultat de cette consultation dans le fichier trace qu'il a créé. Dans le cas de l'exemple présenté, ce fichier a le contenu suivant :

Criteres de selection:

```

Sur l'expert      : 1
Sur la transaction: aucun
Sur l'intervention: aucun
Sur le sujet      : CID
Sur le type       : COUT
Sur le mot cle    : aucun

```

```

Expert      n. 1
Transaction n. 3
Intervention n. 7

```

Et le timbre fiscal il est de combien? Quelle somme?

Criteres de selection:

```

Sur l'expert      : aucun
Sur la transaction: aucun
Sur l'intervention: aucun
Sur le sujet      : CID
Sur le type       : aucun
Sur le mot cle    : merci

```

ANNEXE C : Modélisation syntaxique du sous-langage

L'annexe C présente la syntaxe du sous-langage de l'application test « demande de renseignements » telle qu'elle a été définie par G. Deville dans sa thèse de doctorat (Deville, 1989). Nous nous sommes limités ici à la présentation des RNP. Nous renvoyons le lecteur à la thèse de G. Deville pour une discussion plus détaillée du sujet.

1 Symboles utilisés dans les RNP

1.1 Symboles non-terminaux (font référence à un sous-réseau de la grammaire)

CENTAINES	:	centaines
CHIF	:	groupe entier
DATE	:	date
DECLAR	:	phrase déclarative
DIZAINES	:	dizaines
ENONCE	:	énoncé complet
GN	:	groupe nominal
GN_INTER	:	groupe nominal interrogatif
GOI	:	groupe objet indirect
GP	:	groupe prépositionnel
GR_ADJ	:	groupe adjectival
GR_COMP	:	groupe complément
GR_INTER	:	groupe interrogatif
GR_THEM	:	groupe thématique
GS	:	groupe sujet
GV	:	groupe verbal
INDIR	:	phrase interrogative indirecte

INFINITIVE	:	phrase infinitive
MILLIER	:	milliers
MILLION	:	millions
PREST_1	:	phrase présentative (thème préposé)
PREST_2	:	phrase présentative (thème postposé)
PREST_INT	:	phrase présentative interrogative
PROP	:	proposition
QUEST	:	phrase interrogative
RELAT	:	phrase relative
SGV	:	suite groupe verbal
SUB_CIRC	:	phrase subordonnée

1.2 Symboles terminaux

1.2.1 Catégories syntaxiques

adj	:	adjectif
adj_indf	:	adjectif indéfini
adj_inter	:	adjectif interrogatif
adv	:	adverbe
adv_inter	:	adverbe interrogatif
aux	:	auxiliaire
aux_inf	:	auxiliaire à l'infinitif
co_cor	:	conjonction de coordination
co_sub	:	conjonction de subordination
det	:	déterminant
dizaine_1	:	'dix', ... 'dix-neuf'
dizaine_2	:	'vingt', ... 'cinquante'
dizaine_bel	:	'septante', 'nonante'
dizaine_fr	:	'soixante', 'quatre-vingt'
jour	:	'lundi', ... 'dimanche'
mois	:	'janvier', ... 'décembre'
nom	:	nom
nom_propre	:	nom propre
pr	:	pronom personnel
prep	:	préposition
prep_obj	:	préposition dans un groupe objet
pr_inter	:	pronom interrogatif
pron_obj1	:	pronom personnel non-fléchi
pron_obj2	:	pronom personnel fléchi
pron_suj	:	pronom sujet
pr_relat	:	pronom relatif
unité	:	'un', ... , 'neuf'
v_fléchi	:	verbe fléchi
v_indir_fléchi	:	verbe fléchi introduisant une phrase interrogative indirecte
v_indir_n_fléchi	:	verbe non fléchi introduisant une phrase interrogative indirecte
v_n_fléchi	:	verbe non fléchi

1.2.2 Terminaux vrais

aussi	million
cent	moins
ce_que	ne
c_est	par
de	pas
est_ce_que	premier
est_ce_qui	que
et_un	si
le_moins	un
le_plus	zéro
mille	

2 Les RNP

2.1 Le sous-réseau CENTAINE

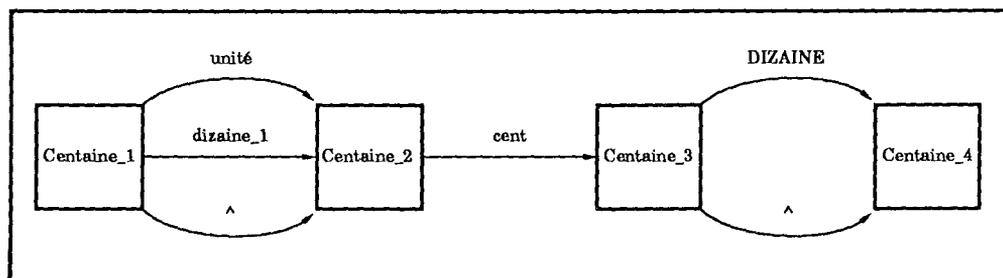


fig. C.1 : Le sous-réseau CENTAINE

2.2 Le sous-réseau CHIF

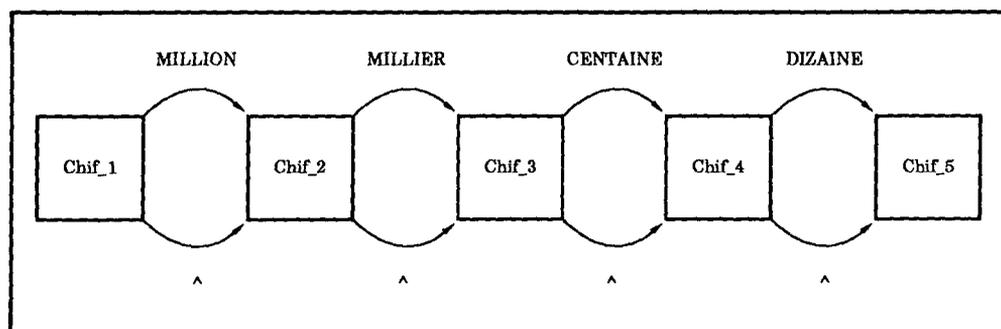


fig. C.2 : Le sous-réseau CHIF

2.3 Le sous-réseau DATE

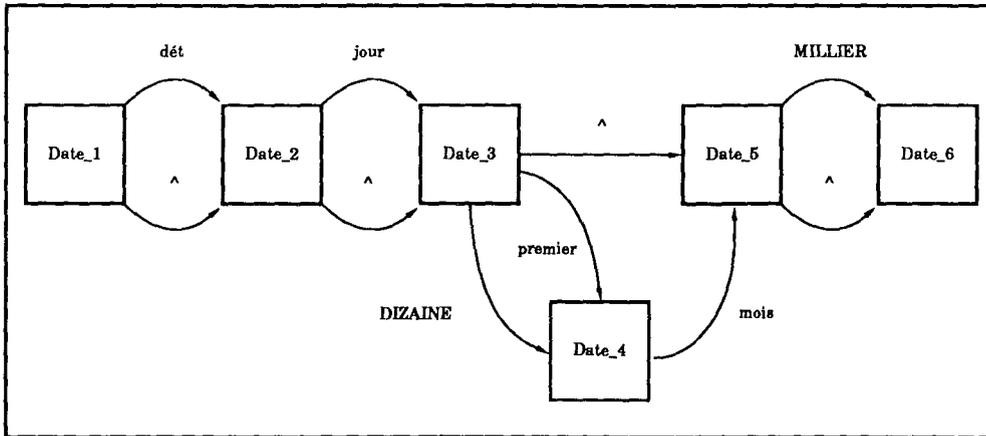


fig. C.3 : Le sous-réseau DATE

2.4 Le sous-réseau DECLAR

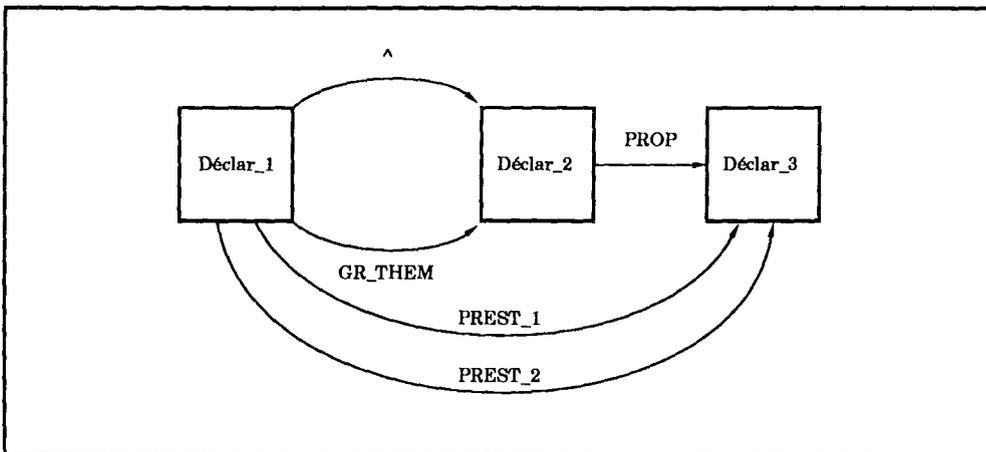


fig. C.4 : Le sous-réseau DECLAR

2.5 Le sous-réseau DIZAINES

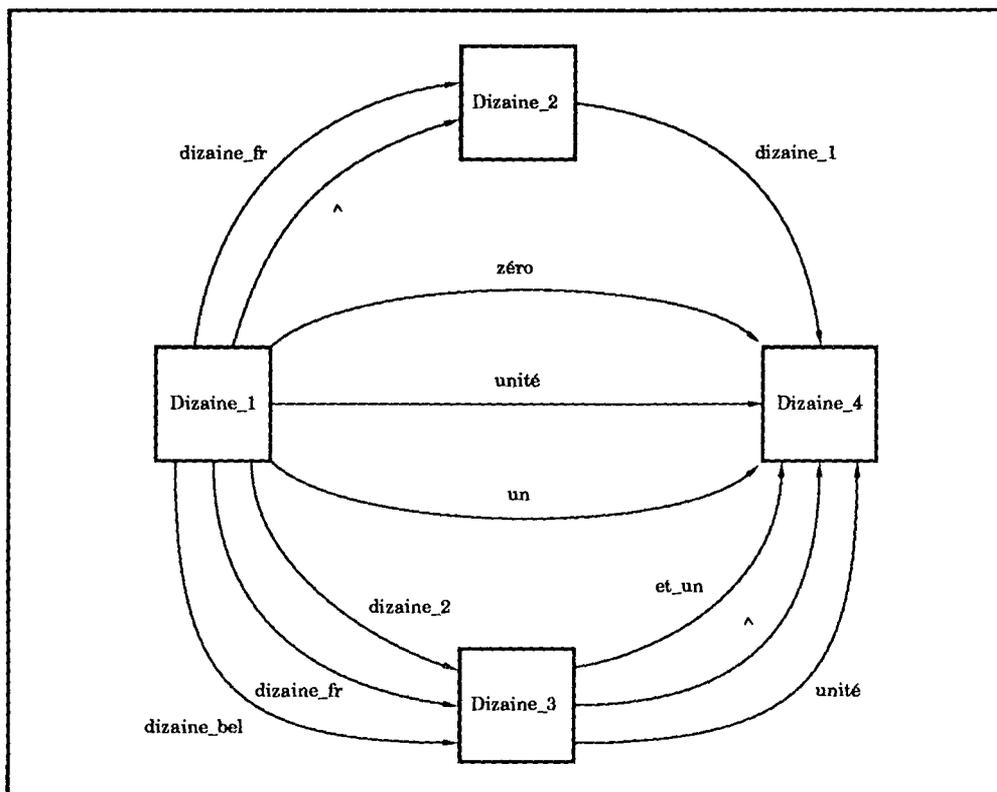


fig. C.5 : Le sous-réseau DIZAINES

2.6 Le sous-réseau ENONCE

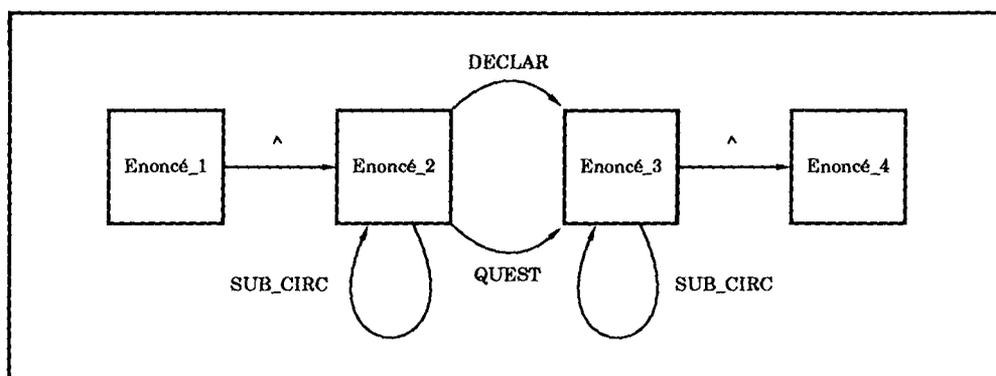


fig. C.6 : Le sous-réseau ENONCE

2.7 Le sous-réseau GN

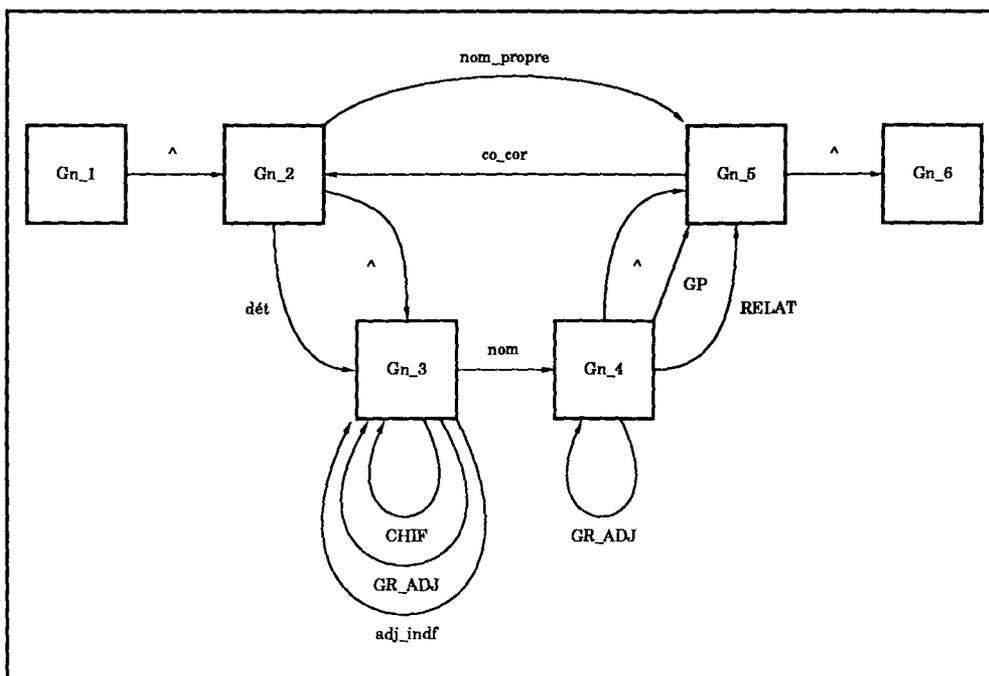


fig. C.7 : Le sous-réseau GN

2.8 Le sous-réseau GN_INTER

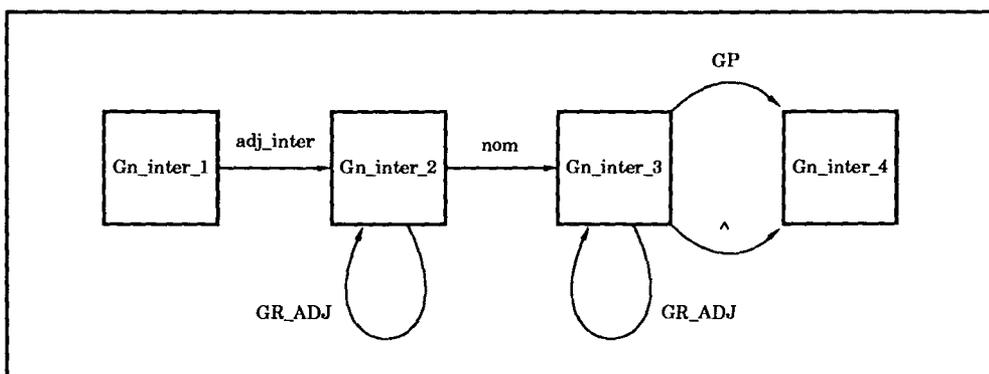


fig. C.8 : Le sous-réseau GN_INTER

2.9 Le sous-réseau GOI

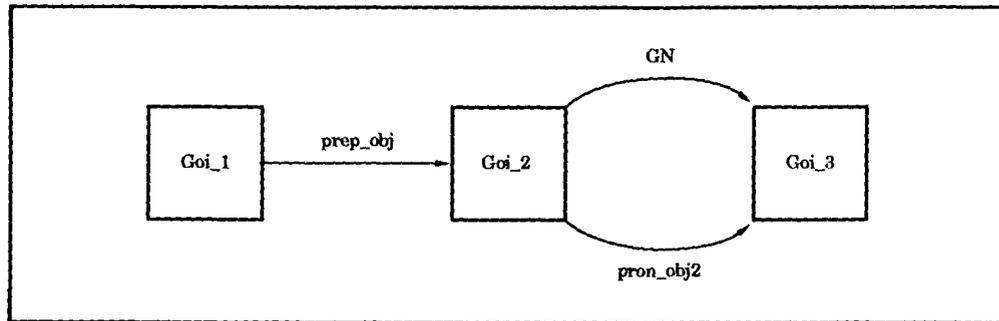


fig. C.9 : Le sous-réseau GOI

2.10 Le sous-réseau GP

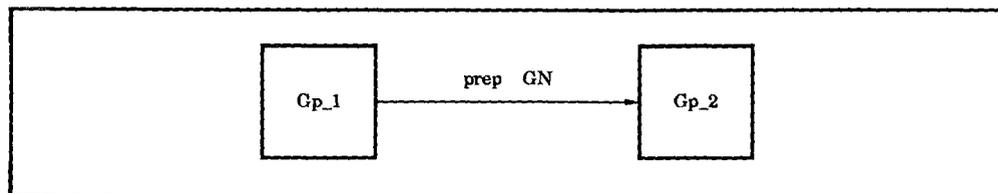


fig. C.10 : Le sous-réseau GP

2.11 Le sous-réseau GR_ADJ

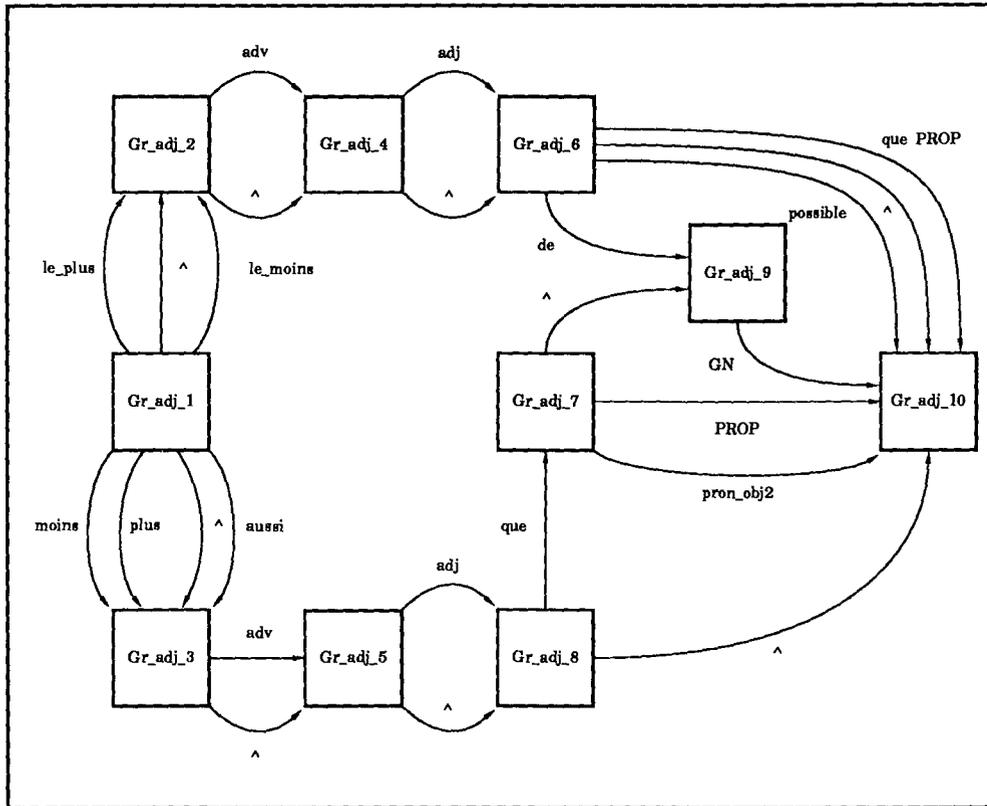


fig. C.11 : Le sous-réseau GR_ADJ

2.12 Le sous-réseau GR_COMP

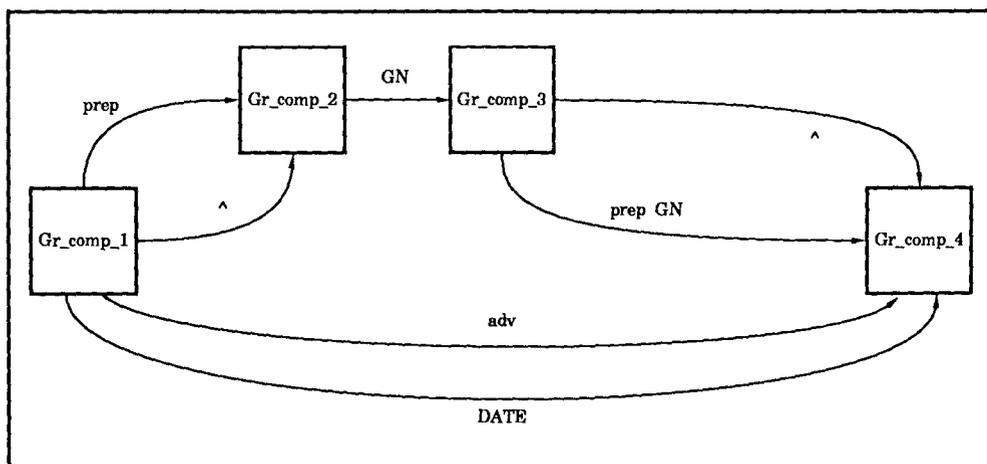


fig. C.12 : Le sous-réseau GR_COMP

2.13 Le sous-réseau GR_INTER

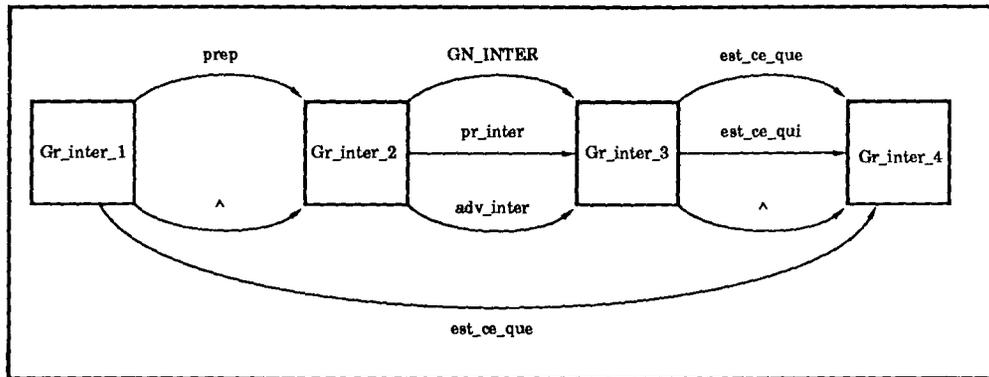


fig. C.13 : Le sous-réseau GR_INTER

2.14 Le sous-réseau GR_THEM

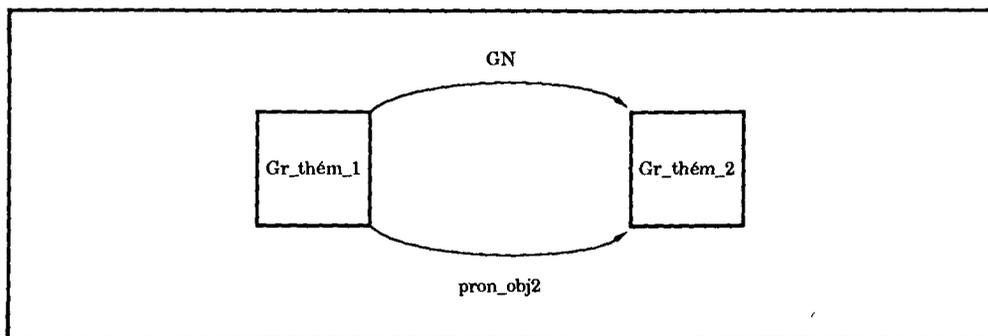


fig. C.14 : Le sous-réseau GR_THEM

2.15 Le sous-réseau GS

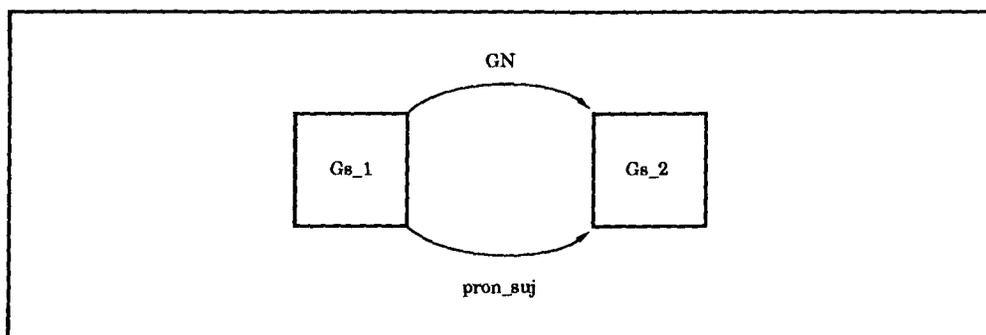


fig. C.15 : Le sous-réseau GS

2.16 Le sous-réseau GV

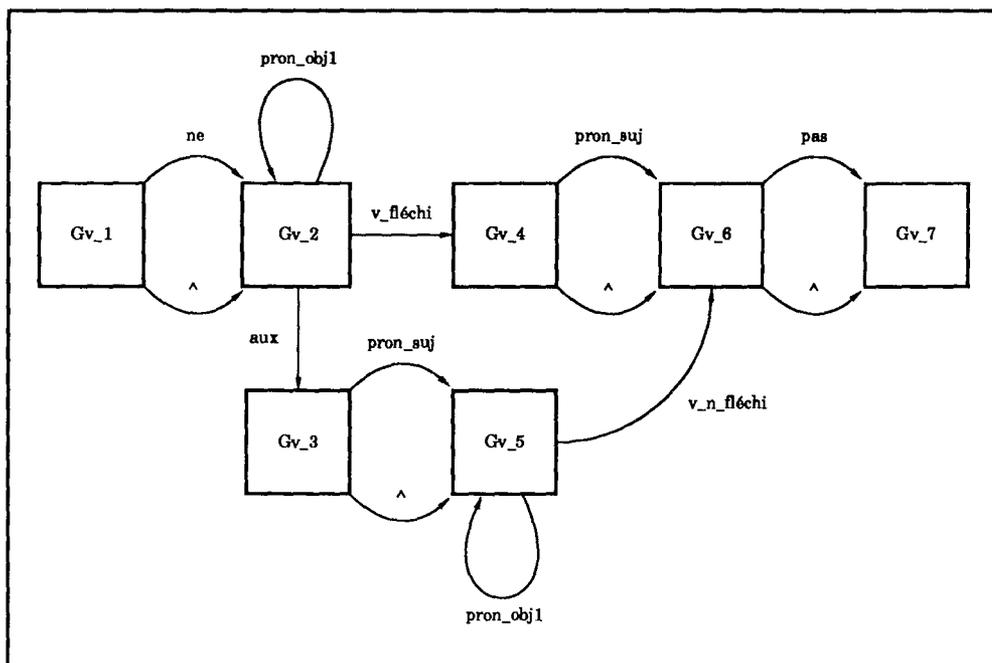


fig. C.16 : Le sous-réseau GV

2.17 Le sous-réseau INDIR

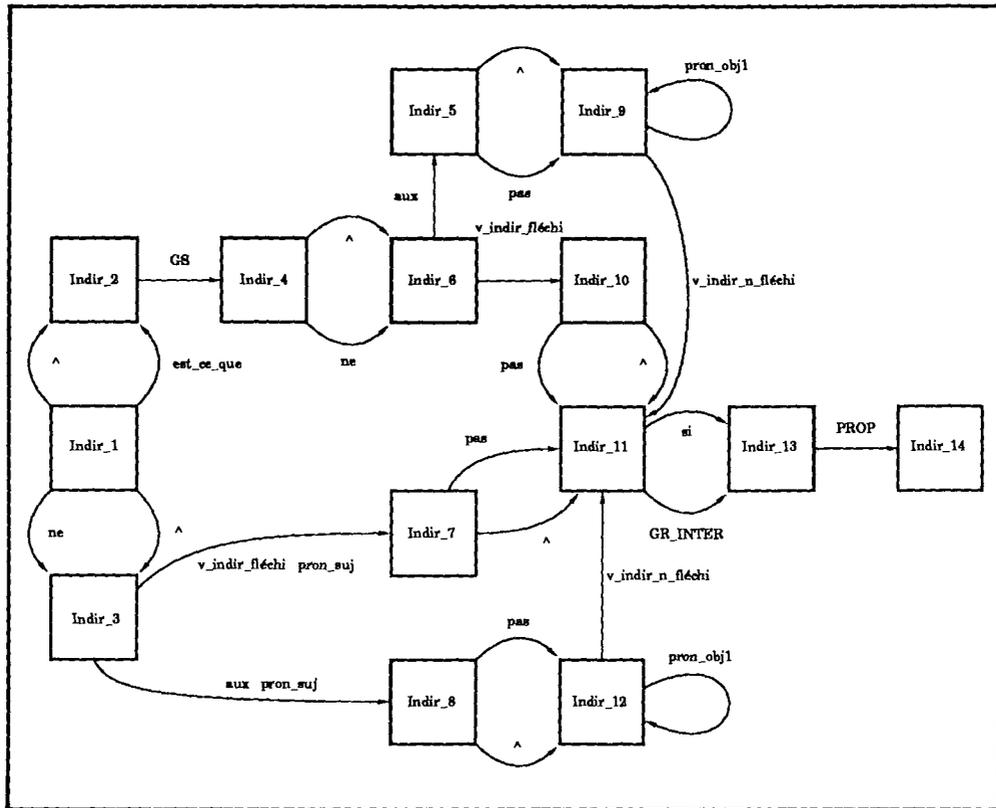


fig. C.17 : Le sous-réseau INDIR

2.18 Le sous-réseau INFINITIVE

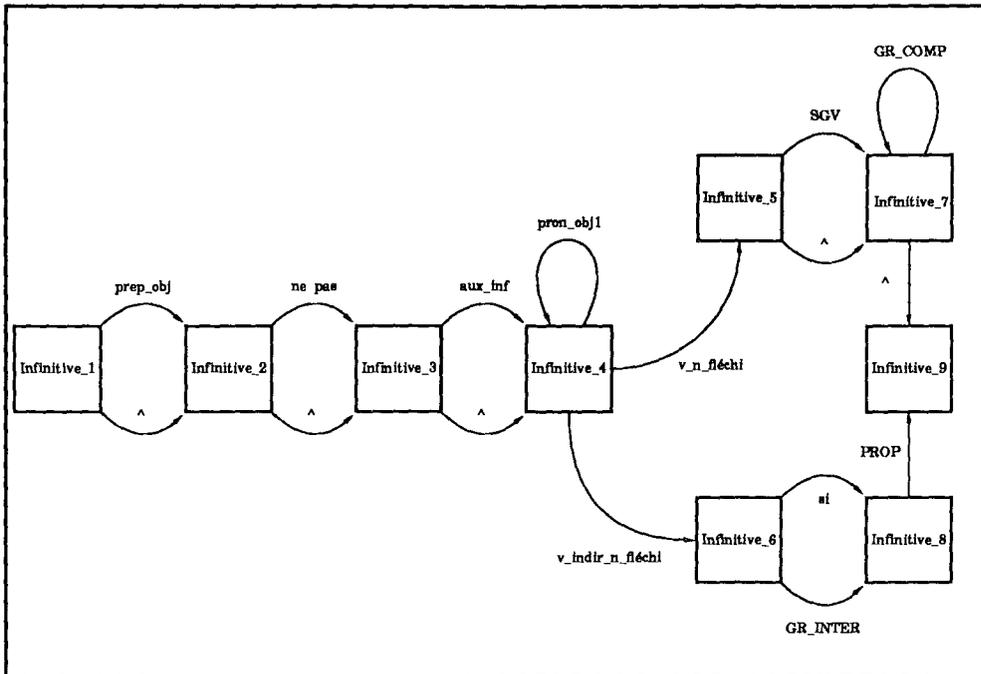


fig. C.18 : Le sous-réseau INFINITIVE

2.19 Le sous-réseau MILLIER

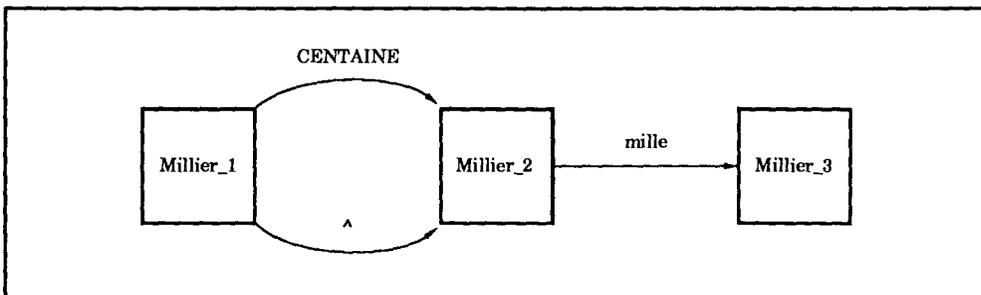


fig. C.19 : Le sous-réseau MILLIER

2.20 Le sous-réseau MILLION

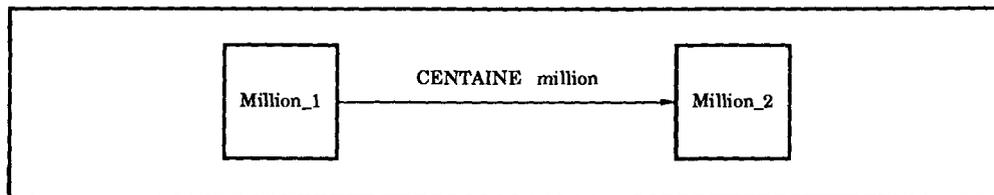


fig. C.20 : Le sous-réseau MILLION

2.21 Le sous-réseau PREST_1

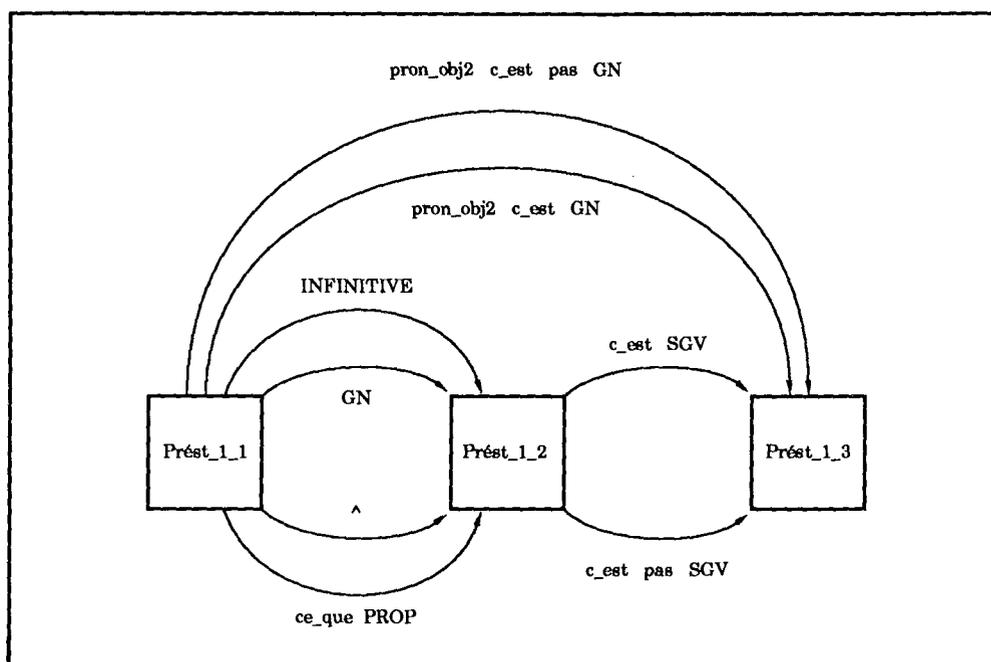


fig. C.21 : Le sous-réseau PREST_1

2.22 Le sous-réseau PREST_2

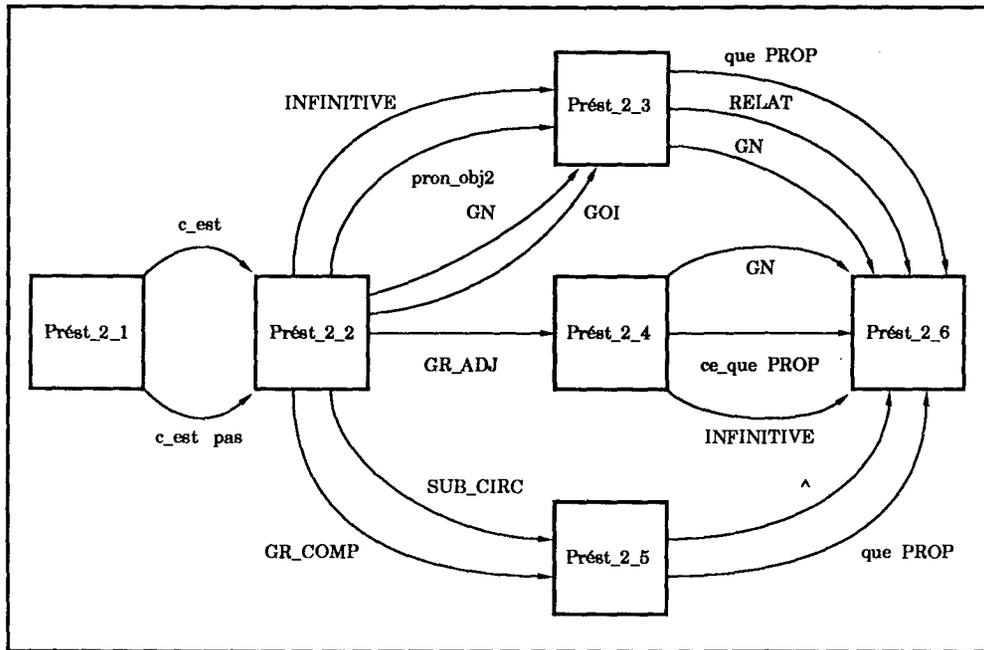


fig. C.22 : Le sous-réseau PREST_2

2.23 Le sous-réseau PREST_INT

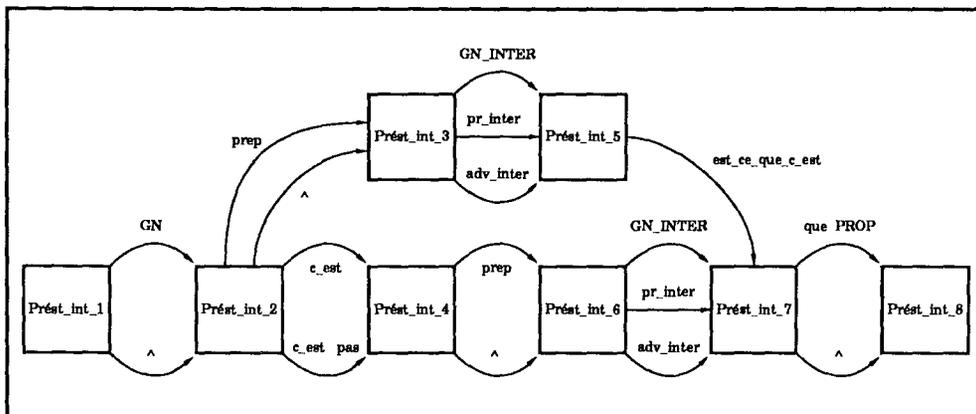


fig. C.23 : Le sous-réseau PREST_INT

2.24 Le sous-réseau PROP

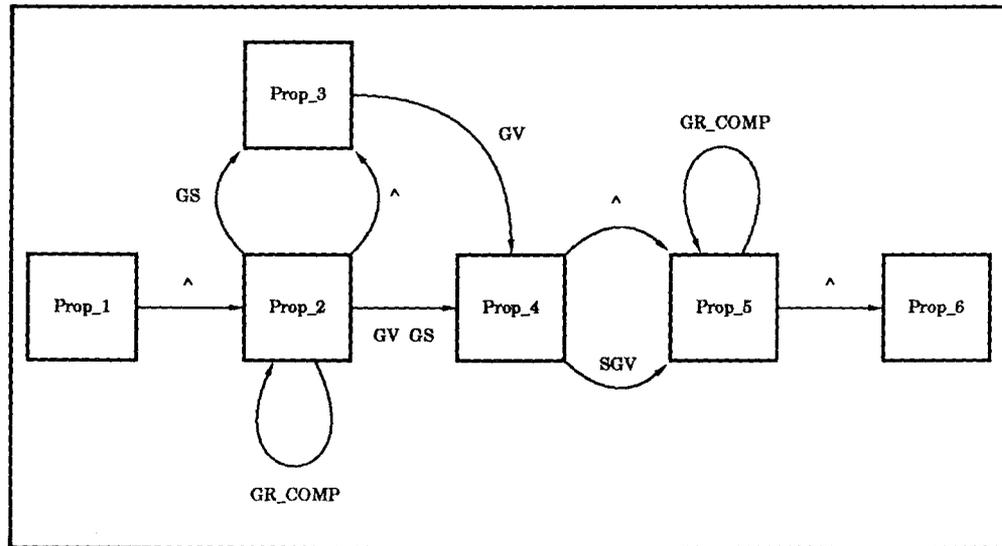


fig. C.24 : Le sous-réseau PROP

2.25 Le sous-réseau QUEST

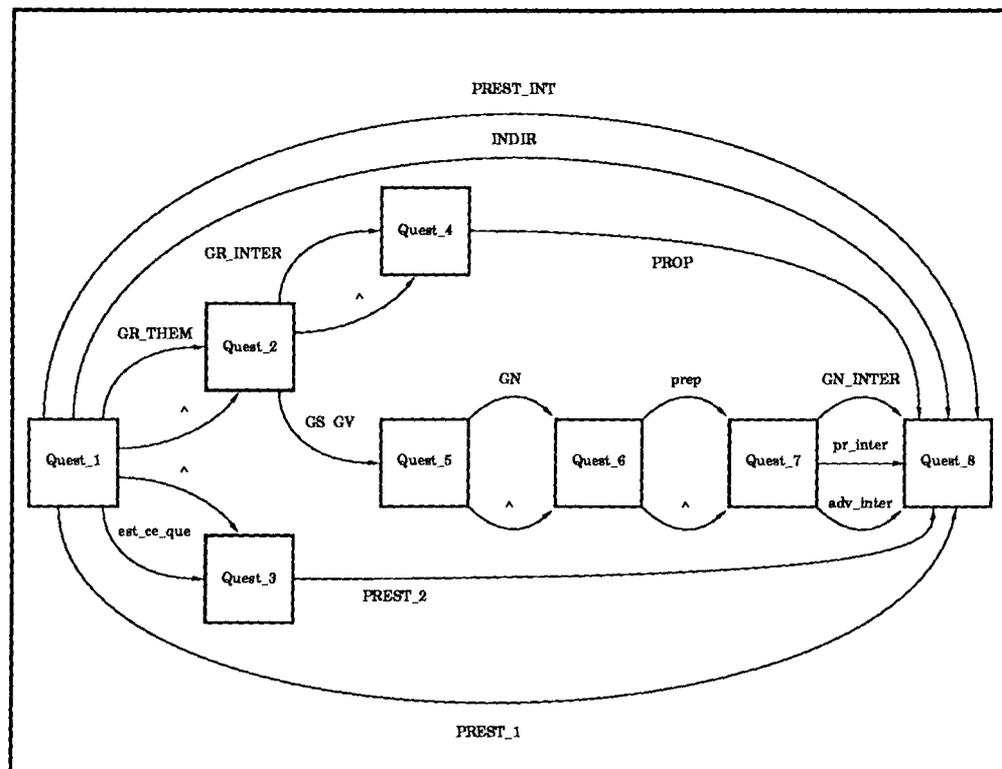


fig. C.25 : Le sous-réseau QUEST

2.26 Le sous-réseau RELAT

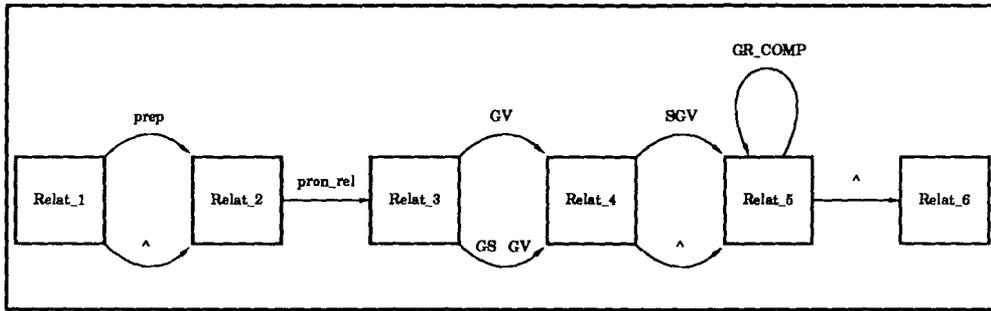


fig. C.26 : Le sous-réseau RELAT

2.27 Le sous-réseau SGV

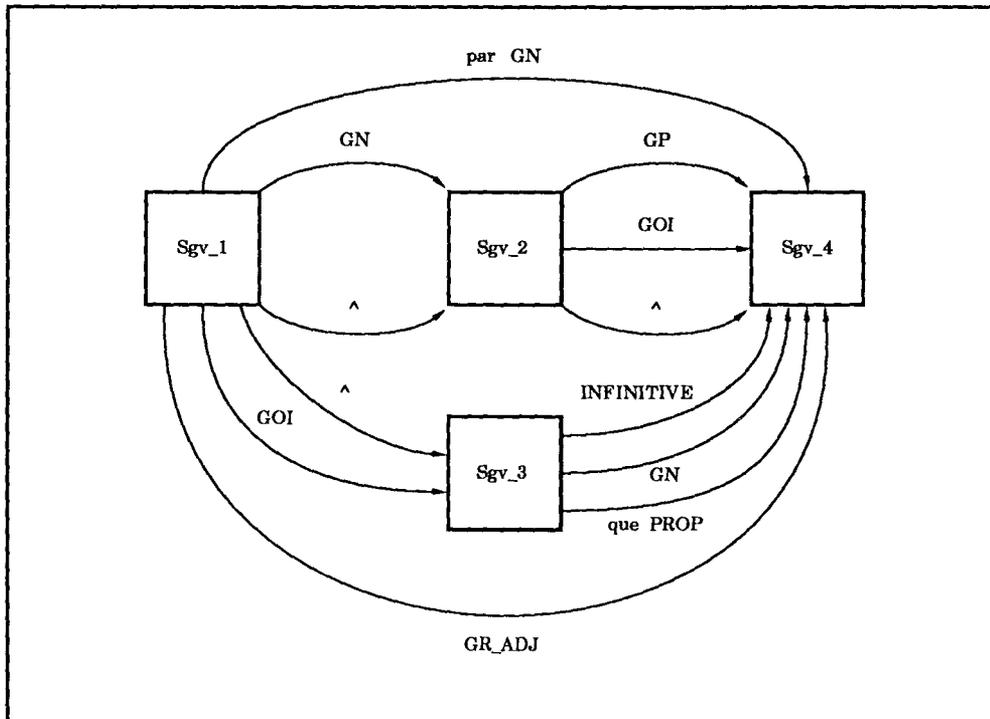


fig. C.27 : Le sous-réseau SGV

2.28 Le sous-réseau SUB_CIRC

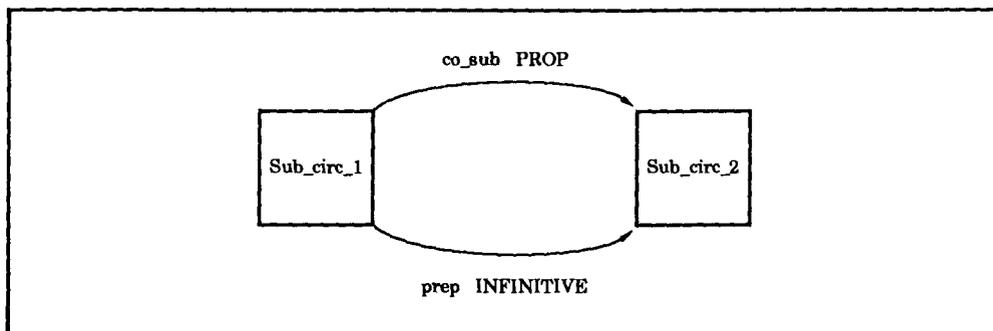


fig. C.28 : Le sous-réseau SUB_CIRC

ANNEXE D : Le programme CRERES

L'annexe D fournit un complément d'informations au sujet du programme de création de réseaux à nœuds procéduraux CRERES. Outre un bref rappel des objectifs de ce programme, elle contient la syntaxe complète d'un texte traité par CRERES. Elle présente également un exemple complet d'utilisation de ce programme.

1 Objectif du programme CRERES

Le programme de création des réseaux à noeuds procéduraux a pour objectif de permettre un passage aisé d'une description externe de la syntaxe d'un langage vers sa représentation interne. La description externe est fournie par l'utilisateur sous forme d'un texte respectant une certaine présentation (définie dans le paragraphe suivant) qui est alors traité par le programme CRERES.

L'utilisateur active le programme CRERES et lui fournit certains paramètres qui peuvent être des options et/ou un fichier. Les deux options possibles sont '-l' si l'utilisateur désire un listing source et '-d' s'il désire une version lisible de la représentation interne construite. Le fichier contient le texte à traiter. Si aucun fichier n'est spécifié, CRERES attend les données sur le fichier standard en entrée.

Si le fichier spécifié n'existe pas, CRERES termine son exécution de façon anormale. Si le fichier existe mais que le texte ne respecte pas la syntaxe acceptée par CRERES, les erreurs sont signalées et aucune représentation interne n'est construite. Sinon, la représentation interne est créée sur un fichier de même nom que le fichier en entrée suffixé par '.rnp'.

Deux rapports (Mousel, 1985) et (Mousel, 1986b) présentent en détail ce programme. La version datant de 1985 a cependant été fortement modifiée et

la version actuelle est celle de 1986. Les paragraphes suivants se rapportent à cette dernière version.

2 Syntaxe du texte traité par CRERES

Pour décrire la syntaxe du texte destiné à être traité par le programme CRERES, nous avons utilisé les conventions suivantes :

- les non-terminaux sont entre crochets '<' et '>';
- les terminaux sont en caractères gras ou entre simples quotes ;
- la production est indiquée par '::=' ;
- les alternatives de production sont indiquées par la barre verticale '|';
- la fin d'une règle de production est indiquée par le point-virgule ';';
- le symbole vide est indiqué par l'accent circonflexe '^'.

La syntaxe est la suivante :

```

<texte> ::=
  <définition-non-terminaux>
  <définition-terminaux>
  <définition-réseau>;

<définition-non-terminaux> ::=
  Non-terminaux: <liste-non-terminaux> ';';

<définition-terminaux> ::=
  Terminaux: <liste-terminaux> ';';

<définition-réseau> ::=
  Description-des-sr: <liste-descr-sous-réseaux>

<liste-non-terminaux> ::=
  <liste-non-terminaux> ',' Identificateur |
  Identificateur;

<liste-terminaux> ::=
  <liste-terminaux> ',' Identificateur |
  Identificateur;

<liste-descriptions-sr> ::=
  <liste-descr-sous-réseaux> <description-sous-réseau> |
  <description-sous-réseau>;

<description-sous-réseau> ::=
  Sous-reseau: Identificateur ';'
  Noeud-d-entree: Identificateur ';'
  Noeud-de-sortie: Identificateur ';'
  Description-des-branches: <liste-descr-branches>;

<liste-descr-branches> ::=
  <liste-descr-branches> <descr-branche> |
  <description-branche>;

<description-branche> ::=
  Origine: Identificateur ';'
  Composants: <liste-composants> ';'

```

```

Destination: Identificateur ';' ;

<liste-composants> ::=
  <liste> | ^ ;

<liste> ::=
  <liste> ',' Identificateur |
  Identificateur ;

```

Les terminaux peuvent s'écrire indépendamment en majuscules ou en minuscules. Certains sont des mots clés, il s'agit de :

- **Non-terminaux:** Ce mot clé peut encore s'écrire **Nt** ;
- **Terminaux:** Ce mot clé peut encore s'écrire **T** ;
- **Description-des-sr:** Ce mot clé peut encore s'écrire **Dsr** ;
- **Sous-reseau:** Ce mot clé peut encore s'écrire **Sr** ;
- **Noeud-d-entree:** Ce mot clé peut encore s'écrire **Ne** ;
- **Noeud-de-sortie:** Ce mot clé peut encore s'écrire **Ns** ;
- **Description-des-branches:** Ce mot clé peut encore s'écrire **Db** ;
- **Origine:** Ce mot clé peut encore s'écrire **Or** ;
- **Composants:** Ce mot clé peut encore s'écrire **Comp** ;
- **Destination:** Ce mot clé peut encore s'écrire **Dest**.

Enfin, un **Identificateur** est composé d'une lettre suivie de lettres, chiffres ou de caractères '_' (souligné). Il faut encore remarquer que deux identificateurs, ne se distinguant que par l'écriture majuscule ou minuscule de certains des caractères les composant, sont considérés comme différents.

3 Exemple d'utilisation

L'exemple suivant illustre l'utilisation du programme CRERES. Le texte fourni en entrée décrit les RNP présentés en Annexe C, définissant ainsi la syntaxe de notre application test. Ce texte est rédigé à l'aide d'un éditeur de texte quelconque. Un listing en est fourni dans le paragraphe 'Les données'. Traité par le programme CRERES avec les options '-l' (listing source) et '-d' (démonstration), il donne lieu à deux types de résultat, un fichier sur disque (suffixé par '.rnp') contenant la représentation interne des RNP et un texte affiché à l'écran (sortie standard redirigeable) composé d'un listing source et d'une version lisible de la représentation interne. Un listing du texte affiché à l'écran est fourni dans le paragraphe 'Les résultats'.

Pour des raisons de place, nous nous sommes limités à la description des trois premiers RNP, CENTAINE, CHIF et DATE. Cependant, la liste des occurrences des symboles construite par CRERES étant assez concise, nous l'avons fournie intégralement.

3.1 Les données

Non-terminaux:

```

CENTAINE,
CHIF,
DATE,
DECLAR,
DIZAINE,
ENONCE,
GN,
GN_INTER,
GOI,
GP,
GR_ADJ,
GR_COMP,
GR_INTER,
GR_THEM,
GS,
GV,
INDIR,
INFINITIVE,
MILLIER,
MILLION,
PREST_1,
PREST_2,
PREST_INTER,
PROP,
QUEST,
RELAT,
SGV,
SUB_CIRC;

```

Terminaux:

```

adj,
adj_indf,
adj_inter,
adv,
adv_inter,
aux,
aux_inf,
co_cor,
co_sub,
det,
dizaine_1,
dizaine_2,
dizaine_bel,
dizaine_fr,
jour,
mois,
nom,
nom_propre,
prep,
prep_obj,
pr_inter,
pron_obj1,
pron_obj2,
pron_suj,
pr_relat,

```

unite,
 v_flechi,
 v_indir_flechi,
 v_indir_n_flechi,
 v_n_flechi,

aussi,
 cent,
 ce_que,
 c_est,
 de,
 est_ce_que,
 est_ce_qui,
 et,
 et_un,
 le_moins,
 le_plus,
 mille,
 million,
 moins,
 ne,
 par,
 pas,
 plus,
 possible,
 premier,
 que,
 si,
 un,
 zero;

Description-des-sr:

Sous-reseau: CENTAINE;

Noeud-d-entree: CENTAINE_1;
 Noeud-de-sortie: CENTAINE_4;

Description-des-branches:

Origine: CENTAINE_1; Composants: unite;	Destination: CENTAINE_2;
Origine: CENTAINE_1; Composants: dizaine_1;	Destination: CENTAINE_2;
Origine: CENTAINE_1; Composants: ;	Destination: CENTAINE_2;
Origine: CENTAINE_2; Composants: cent;	Destination: CENTAINE_3;
Origine: CENTAINE_3; Composants: DIZAINE;	Destination: CENTAINE_4;
Origine: CENTAINE_3; Composants: ;	Destination: CENTAINE_4;

Sous-reseau: CHIF;

Noeud-d-entree: CHIF_1;
 Noeud-de-sortie: CHIF_5;

Description-des-branches:

Origine: CHIF_1; Composants: MILLION;	Destination: CHIF_2;
Origine: CHIF_1; Composants: ;	Destination: CHIF_2;
Origine: CHIF_2; Composants: MILLIER;	Destination: CHIF_3;
Origine: CHIF_2; Composants: ;	Destination: CHIF_3;

```

Origine: CHIF_3;
  Composants: CENTAINE;      Destination: CHIF_4;
Origine: CHIF_3;
  Composants: ;              Destination: CHIF_4;
Origine: CHIF_4;
  Composants: DIZAINE;      Destination: CHIF_5;
Origine: CHIF_4;
  Composants: ;              Destination: CHIF_5;

Sous-reseau: DATE;

Noeud-d-entree: DATE_1;
Noeud-de-sortie: DATE_6;

Description-des-branches:

Origine: DATE_1;
  Composants: det;          Destination: DATE_2;
Origine: DATE_1;
  Composants: ;             Destination: DATE_2;
Origine: DATE_2;
  Composants: jour;        Destination: DATE_3;
Origine: DATE_2;
  Composants: ;             Destination: DATE_3;
Origine: DATE_3;
  Composants: premier;     Destination: DATE_4;
Origine: DATE_3;
  Composants: DIZAINE;     Destination: DATE_4;
Origine: DATE_3;
  Composants: ;             Destination: DATE_5;
Origine: DATE_4;
  Composants: mois;        Destination: DATE_5;
Origine: DATE_5;
  Composants: MILLIER;     Destination: DATE_6;
Origine: DATE_5;
  Composants: ;             Destination: DATE_6;

```

...

3.2 Les résultats

Non-terminaux:

```

CENTAINE,
CHIF,
DATE,
DECLAR,
DIZAINE,
ENONCE,
GN,
GN_INTER,
GOI,
GP,
GR_ADJ,
GR_COMP,
GR_INTER,
GR_THEM,
GS,
GV,
INDIR,
INFINITIVE,
MILLIER,
MILLION,
PREST_1,
PREST_2,
PREST_INTER,

```

PROP,
 QUEST,
 RELAT,
 SGV,
 SUB_CIRC;

Terminaux:

adj,
 adj_indf,
 adj_inter,
 adv,
 adv_inter,
 aux,
 aux_inf,
 co_cor,
 co_sub,
 det,
 dizaine_1,
 dizaine_2,
 dizaine_bel,
 dizaine_fr,
 jour,
 mois,
 nom,
 nom_propre,
 prep,
 prep_obj,
 pr_inter,
 pron_obj1,
 pron_obj2,
 pron_suj,
 pr_relat,
 unite,
 v_flechi,
 v_indir_flechi,
 v_indir_n_flechi,
 v_n_flechi,

aussi,
 cent,
 ce_que,
 c_est,
 de,
 est_ce_que,
 est_ce_qui,
 et,
 et_un,
 le_moins,
 le_plus,
 mille,
 million,
 moins,
 ne,
 par,
 pas,
 plus,
 possible,
 premier,
 que,
 si,
 un,
 zero;

Description-des-sr:

Sous-reseau: CENTAINE;

Noeud-d-entree: CENTAINE_1;
Noeud-de-sortie: CENTAINE_4;

Description-des-branches:

Origine: CENTAINE_1; Composants: unite;	Destination: CENTAINE_2;
Origine: CENTAINE_1; Composants: dizaine_1;	Destination: CENTAINE_2;
Origine: CENTAINE_1; Composants: ;	Destination: CENTAINE_2;
Origine: CENTAINE_2; Composants: cent;	Destination: CENTAINE_3;
Origine: CENTAINE_3; Composants: DIZAINE;	Destination: CENTAINE_4;
Origine: CENTAINE_3; Composants: ;	Destination: CENTAINE_4;

Sous-reseau: CHIF;

Noeud-d-entree: CHIF_1;
Noeud-de-sortie: CHIF_5;

Description-des-branches:

Origine: CHIF_1; Composants: MILLION;	Destination: CHIF_2;
Origine: CHIF_1; Composants: ;	Destination: CHIF_2;
Origine: CHIF_2; Composants: MILLIER;	Destination: CHIF_3;
Origine: CHIF_2; Composants: ;	Destination: CHIF_3;
Origine: CHIF_3; Composants: CENTAINE;	Destination: CHIF_4;
Origine: CHIF_3; Composants: ;	Destination: CHIF_4;
Origine: CHIF_4; Composants: DIZAINE;	Destination: CHIF_5;
Origine: CHIF_4; Composants: ;	Destination: CHIF_5;

Sous-reseau: DATE;

Noeud-d-entree: DATE_1;
Noeud-de-sortie: DATE_6;

Description-des-branches:

Origine: DATE_1; Composants: det;	Destination: DATE_2;
Origine: DATE_1; Composants: ;	Destination: DATE_2;
Origine: DATE_2; Composants: jour;	Destination: DATE_3;
Origine: DATE_2; Composants: ;	Destination: DATE_3;
Origine: DATE_3; Composants: premier;	Destination: DATE_4;
Origine: DATE_3; Composants: DIZAINE;	Destination: DATE_4;
Origine: DATE_3; Composants: ;	Destination: DATE_5;
Origine: DATE_4; Composants: mois;	Destination: DATE_5;
Origine: DATE_5; Composants: MILLIER;	Destination: DATE_6;
Origine: DATE_5; Composants: ;	Destination: DATE_6;

...

Contenu du reseau a noeuds proceduraux: dem_rens

Liste des terminaux:

=====

adj
 adj_indf
 adj_inter
 adv
 adv_inter
 aux
 aux_inf
 co_cor
 co_sub
 det
 dizaine_1
 dizaine_2
 dizaine_bel
 dizaine_fr
 jour
 mois
 nom
 nom_propre
 prep
 prep_obj
 pr_inter
 pron_obj1
 pron_obj2
 pron_suj
 pr_relat
 unite
 v_flechi
 v_indir_flechi
 v_indir_n_flechi
 v_n_flechi
 aussi
 cent
 ce_que
 c_est
 de
 est_ce_que
 est_ce_qui
 et
 et_un
 le_moins
 le_plus
 mille
 million
 moins
 ne
 par
 pas
 plus
 possible
 premier
 que
 si
 un
 zero

Liste des non-terminaux:

=====

CENTAINE
 CHIF
 DATE
 DECLAR
 DIZAINE
 ENONCE
 GN
 GN_INTER
 GOI
 GP
 GR_ADJ
 GR_COMP
 GR_INTER
 GR_THEM
 GS
 GV
 INDIR
 INFINITIVE
 MILLIER
 MILLION
 PREST_1
 PREST_2
 PREST_INTER
 PROP
 QUEST
 RELAT
 SGV
 SUB_CIRC

La liste des sous-reseaux:
 =====

Le sous-reseau: CENTAINE

Noeud d'entree: CENTAINE_1
 Noeud de sortie: CENTAINE_4

Liste des branches:

Branche numero: 0
 Origine: CENTAINE_1
 Destination: CENTAINE_2
 Composants:

 unite

Branche numero: 1
 Origine: CENTAINE_1
 Destination: CENTAINE_2
 Composants:

 dizaine_1

Branche numero: 2
 Origine: CENTAINE_1
 Destination: CENTAINE_2
 Composants:

Branche numero: 3
 Origine: CENTAINE_2
 Destination: CENTAINE_3
 Composants:

 cent

Branche numero: 4
 Origine: CENTAINE_3
 Destination: CENTAINE_4
 Composants:

DIZAINE

Branche numero: 5
 Origine: CENTAINE_3
 Destination: CENTAINE_4
 Composants:

Liste des noeuds:

Noeud: CENTAINE_1
 Entrees:
 Sorties: 0 1 2
 Noeud: CENTAINE_2
 Entrees: 0 1 2
 Sorties: 3
 Noeud: CENTAINE_3
 Entrees: 3
 Sorties: 4 5
 Noeud: CENTAINE_4
 Entrees: 4 5
 Sorties:

Le sous-reseau: CHIF

Noeud d'entree: CHIF_1
 Noeud de sortie: CHIF_5

Liste des branches:

Branche numero: 6
 Origine: CHIF_1
 Destination: CHIF_2
 Composants:

MILLION

Branche numero: 7
 Origine: CHIF_1
 Destination: CHIF_2
 Composants:

Branche numero: 8
 Origine: CHIF_2
 Destination: CHIF_3
 Composants:

MILLIER

Branche numero: 9
 Origine: CHIF_2
 Destination: CHIF_3
 Composants:

Branche numero: 10
 Origine: CHIF_3
 Destination: CHIF_4
 Composants:

CENTAINE

Branche numero: 11

Origine: CHIF_3
 Destination: CHIF_4
 Composants:

Branche numero: 12
 Origine: CHIF_4
 Destination: CHIF_5
 Composants:

DIZAIN

Branche numero: 13
 Origine: CHIF_4
 Destination: CHIF_5
 Composants:

Liste des noeuds:

Noeud: CHIF_1
 Entrees:
 Sorties: 6 7
 Noeud: CHIF_2
 Entrees: 6 7
 Sorties: 8 9
 Noeud: CHIF_3
 Entrees: 8 9
 Sorties: 10 11
 Noeud: CHIF_4
 Entrees: 10 11
 Sorties: 12 13
 Noeud: CHIF_5
 Entrees: 12 13
 Sorties:

Le sous-reseau: DATE

Noeud d'entree: DATE_1
 Noeud de sortie: DATE_6

Liste des branches:

Branche numero: 14
 Origine: DATE_1
 Destination: DATE_2
 Composants:

det

Branche numero: 15
 Origine: DATE_1
 Destination: DATE_2
 Composants:

Branche numero: 16
 Origine: DATE_2
 Destination: DATE_3
 Composants:

jour

Branche numero: 17
 Origine: DATE_2
 Destination: DATE_3
 Composants:

Branche numero: 18
 Origine: DATE_3
 Destination: DATE_4
 Composants:

premier

Branche numero: 19
 Origine: DATE_3
 Destination: DATE_4
 Composants:

DIZAIN

Branche numero: 20
 Origine: DATE_3
 Destination: DATE_5
 Composants:

Branche numero: 21
 Origine: DATE_4
 Destination: DATE_5
 Composants:

mois

Branche numero: 22
 Origine: DATE_5
 Destination: DATE_6
 Composants:

MILLIER

Branche numero: 23
 Origine: DATE_5
 Destination: DATE_6
 Composants:

Liste des noeuds:

Noeud: DATE_1
 Entrees:
 Sorties: 14 15
 Noeud: DATE_2
 Entrees: 14 15
 Sorties: 16 17
 Noeud: DATE_3
 Entrees: 16 17
 Sorties: 18 19 20
 Noeud: DATE_4
 Entrees: 18 19
 Sorties: 21
 Noeud: DATE_5
 Entrees: 20 21
 Sorties: 22 23
 Noeud: DATE_6
 Entrees: 22 23
 Sorties:

...

La liste des occurrences des symboles:

=====

Liste des occurrences du symbole: unite
 Sous-reseau: CENTAINE, Branche numero 0

Sous-reseau: DIZAINE, Branche numero 35
Sous-reseau: DIZAINE, Branche numero 40

Liste des occurrences du symbole: dizaine_1
Sous-reseau: CENTAINE, Branche numero 1
Sous-reseau: DIZAINE, Branche numero 37

Liste des occurrences du symbole: cent
Sous-reseau: CENTAINE, Branche numero 3

Liste des occurrences du symbole: DIZAINE
Sous-reseau: CENTAINE, Branche numero 4
Sous-reseau: CHIF, Branche numero 12
Sous-reseau: DATE, Branche numero 19

Liste des occurrences du symbole: MILLION
Sous-reseau: CHIF, Branche numero 6

Liste des occurrences du symbole: MILLIER
Sous-reseau: CHIF, Branche numero 8
Sous-reseau: DATE, Branche numero 22

Liste des occurrences du symbole: CENTAINE
Sous-reseau: CHIF, Branche numero 10
Sous-reseau: MILLIER, Branche numero 171
Sous-reseau: MILLION, Branche numero 174

Liste des occurrences du symbole: det
Sous-reseau: DATE, Branche numero 14
Sous-reseau: GN, Branche numero 49

Liste des occurrences du symbole: jour
Sous-reseau: DATE, Branche numero 16

Liste des occurrences du symbole: premier
Sous-reseau: DATE, Branche numero 18

Liste des occurrences du symbole: mois
Sous-reseau: DATE, Branche numero 21

Liste des occurrences du symbole: GR_THEM
Sous-reseau: DECLAR, Branche numero 24
Sous-reseau: QUEST, Branche numero 229

Liste des occurrences du symbole: PREST_1
Sous-reseau: DECLAR, Branche numero 26
Sous-reseau: QUEST, Branche numero 233

Liste des occurrences du symbole: PREST_2
Sous-reseau: DECLAR, Branche numero 27
Sous-reseau: QUEST, Branche numero 237

Liste des occurrences du symbole: PROP
Sous-reseau: DECLAR, Branche numero 28
Sous-reseau: GR_ADJ, Branche numero 88
Sous-reseau: GR_ADJ, Branche numero 95
Sous-reseau: INDIR, Branche numero 154
Sous-reseau: INFINITIVE, Branche numero 170
Sous-reseau: PREST_1, Branche numero 180
Sous-reseau: PREST_2, Branche numero 192
Sous-reseau: PREST_2, Branche numero 195
Sous-reseau: PREST_2, Branche numero 198
Sous-reseau: PREST_INTER, Branche numero 215
Sous-reseau: QUEST, Branche numero 238
Sous-reseau: SGV, Branche numero 266
Sous-reseau: SUB_CIRC, Branche numero 267

Liste des occurrences du symbole: dizaine_fr
Sous-reseau: DIZAINE, Branche numero 29
Sous-reseau: DIZAINE, Branche numero 32

Liste des occurrences du symbole: dizaine_2
Sous-reseau: DIZAIN, Branche numero 31

Liste des occurrences du symbole: dizaine_bel
Sous-reseau: DIZAIN, Branche numero 33

Liste des occurrences du symbole: zero
Sous-reseau: DIZAIN, Branche numero 34

Liste des occurrences du symbole: un
Sous-reseau: DIZAIN, Branche numero 36

Liste des occurrences du symbole: et_un
Sous-reseau: DIZAIN, Branche numero 38

Liste des occurrences du symbole: SUB_CIRC
Sous-reseau: ENONCE, Branche numero 42
Sous-reseau: ENONCE, Branche numero 45
Sous-reseau: PREST_2, Branche numero 190

Liste des occurrences du symbole: DECLAR
Sous-reseau: ENONCE, Branche numero 43

Liste des occurrences du symbole: QUEST
Sous-reseau: ENONCE, Branche numero 44

Liste des occurrences du symbole: nom_propre
Sous-reseau: GN, Branche numero 48

Liste des occurrences du symbole: CHIF
Sous-reseau: GN, Branche numero 51

Liste des occurrences du symbole: GR_ADJ
Sous-reseau: GN, Branche numero 52
Sous-reseau: GN, Branche numero 55
Sous-reseau: GN_INTER, Branche numero 62
Sous-reseau: GN_INTER, Branche numero 64
Sous-reseau: PREST_2, Branche numero 189
Sous-reseau: SGV, Branche numero 260

Liste des occurrences du symbole: adj_indf
Sous-reseau: GN, Branche numero 53

Liste des occurrences du symbole: nom
Sous-reseau: GN, Branche numero 54
Sous-reseau: GN_INTER, Branche numero 63

Liste des occurrences du symbole: GP
Sous-reseau: GN, Branche numero 57
Sous-reseau: GN_INTER, Branche numero 65
Sous-reseau: SGV, Branche numero 261

Liste des occurrences du symbole: RELAT
Sous-reseau: GN, Branche numero 58
Sous-reseau: PREST_2, Branche numero 193

Liste des occurrences du symbole: co_cor
Sous-reseau: GN, Branche numero 59

Liste des occurrences du symbole: adj_inter
Sous-reseau: GN_INTER, Branche numero 61

Liste des occurrences du symbole: prep_obj
Sous-reseau: GOI, Branche numero 67
Sous-reseau: INFINITIVE, Branche numero 155

Liste des occurrences du symbole: GN
Sous-reseau: GOI, Branche numero 68
Sous-reseau: GP, Branche numero 70

Sous-reseau: GS, Branche numero 71
 Sous-reseau: GR_ADJ, Branche numero 97
 Sous-reseau: GR_COMP, Branche numero 102
 Sous-reseau: GR_COMP, Branche numero 104
 Sous-reseau: GR_THEM, Branche numero 114
 Sous-reseau: PREST_1, Branche numero 175
 Sous-reseau: PREST_1, Branche numero 176
 Sous-reseau: PREST_1, Branche numero 177
 Sous-reseau: PREST_2, Branche numero 187
 Sous-reseau: PREST_2, Branche numero 194
 Sous-reseau: PREST_2, Branche numero 196
 Sous-reseau: PREST_INTER, Branche numero 201
 Sous-reseau: QUEST, Branche numero 239
 Sous-reseau: SGV, Branche numero 255
 Sous-reseau: SGV, Branche numero 256
 Sous-reseau: SGV, Branche numero 265

Liste des occurrences du symbole: pron_obj2
 Sous-reseau: GOI, Branche numero 69
 Sous-reseau: GR_ADJ, Branche numero 96
 Sous-reseau: GR_THEM, Branche numero 115
 Sous-reseau: PREST_1, Branche numero 175
 Sous-reseau: PREST_1, Branche numero 176
 Sous-reseau: PREST_2, Branche numero 186

Liste des occurrences du symbole: prep
 Sous-reseau: GP, Branche numero 70
 Sous-reseau: GR_COMP, Branche numero 98
 Sous-reseau: GR_COMP, Branche numero 104
 Sous-reseau: GR_INTER, Branche numero 105
 Sous-reseau: PREST_INTER, Branche numero 202
 Sous-reseau: PREST_INTER, Branche numero 209
 Sous-reseau: QUEST, Branche numero 241
 Sous-reseau: RELAT, Branche numero 246
 Sous-reseau: SUB_CIRC, Branche numero 268

Liste des occurrences du symbole: pron_suj
 Sous-reseau: GS, Branche numero 72
 Sous-reseau: GV, Branche numero 121
 Sous-reseau: GV, Branche numero 123
 Sous-reseau: INDIR, Branche numero 134
 Sous-reseau: INDIR, Branche numero 135

Liste des occurrences du symbole: le_plus
 Sous-reseau: GR_ADJ, Branche numero 73

Liste des occurrences du symbole: le_moins
 Sous-reseau: GR_ADJ, Branche numero 75

Liste des occurrences du symbole: aussi
 Sous-reseau: GR_ADJ, Branche numero 76

Liste des occurrences du symbole: moins
 Sous-reseau: GR_ADJ, Branche numero 77

Liste des occurrences du symbole: plus
 Sous-reseau: GR_ADJ, Branche numero 79

Liste des occurrences du symbole: adv
 Sous-reseau: GR_ADJ, Branche numero 80
 Sous-reseau: GR_ADJ, Branche numero 82
 Sous-reseau: GR_COMP, Branche numero 100

Liste des occurrences du symbole: adj
 Sous-reseau: GR_ADJ, Branche numero 84
 Sous-reseau: GR_ADJ, Branche numero 86

Liste des occurrences du symbole: que
 Sous-reseau: GR_ADJ, Branche numero 88
 Sous-reseau: GR_ADJ, Branche numero 92

Sous-reseau: PREST_2, Branche numero 192
Sous-reseau: PREST_2, Branche numero 198
Sous-reseau: PREST_INTER, Branche numero 215
Sous-reseau: SGV, Branche numero 266

Liste des occurrences du symbole: possible
Sous-reseau: GR_ADJ, Branche numero 90

Liste des occurrences du symbole: de
Sous-reseau: GR_ADJ, Branche numero 91

Liste des occurrences du symbole: DATE
Sous-reseau: GR_COMP, Branche numero 101

Liste des occurrences du symbole: est_ce_que
Sous-reseau: GR_INTER, Branche numero 107
Sous-reseau: GR_INTER, Branche numero 111
Sous-reseau: INDIR, Branche numero 130
Sous-reseau: PREST_INTER, Branche numero 211
Sous-reseau: QUEST, Branche numero 232

Liste des occurrences du symbole: GN_INTER
Sous-reseau: GR_INTER, Branche numero 108
Sous-reseau: PREST_INTER, Branche numero 206
Sous-reseau: PREST_INTER, Branche numero 212
Sous-reseau: QUEST, Branche numero 243

Liste des occurrences du symbole: pr_inter
Sous-reseau: GR_INTER, Branche numero 109
Sous-reseau: PREST_INTER, Branche numero 207
Sous-reseau: PREST_INTER, Branche numero 213
Sous-reseau: QUEST, Branche numero 244

Liste des occurrences du symbole: adv_inter
Sous-reseau: GR_INTER, Branche numero 110
Sous-reseau: PREST_INTER, Branche numero 208
Sous-reseau: PREST_INTER, Branche numero 214
Sous-reseau: QUEST, Branche numero 245

Liste des occurrences du symbole: est_ce_qui
Sous-reseau: GR_INTER, Branche numero 112

Liste des occurrences du symbole: ne
Sous-reseau: GV, Branche numero 116
Sous-reseau: INDIR, Branche numero 131
Sous-reseau: INDIR, Branche numero 139
Sous-reseau: INFINITIVE, Branche numero 157

Liste des occurrences du symbole: pron_obj1
Sous-reseau: GV, Branche numero 118
Sous-reseau: GV, Branche numero 125
Sous-reseau: INDIR, Branche numero 146
Sous-reseau: INDIR, Branche numero 150
Sous-reseau: INFINITIVE, Branche numero 161

Liste des occurrences du symbole: v_flechi
Sous-reseau: GV, Branche numero 119

Liste des occurrences du symbole: aux
Sous-reseau: GV, Branche numero 120
Sous-reseau: INDIR, Branche numero 135
Sous-reseau: INDIR, Branche numero 140

Liste des occurrences du symbole: v_n_flechi
Sous-reseau: GV, Branche numero 126
Sous-reseau: INFINITIVE, Branche numero 162

Liste des occurrences du symbole: pas
Sous-reseau: GV, Branche numero 127
Sous-reseau: INDIR, Branche numero 137

Sous-reseau: INDIR, Branche numero 142
 Sous-reseau: INDIR, Branche numero 144
 Sous-reseau: INDIR, Branche numero 149
 Sous-reseau: INFINITIVE, Branche numero 157
 Sous-reseau: PREST_1, Branche numero 175
 Sous-reseau: PREST_1, Branche numero 181
 Sous-reseau: PREST_2, Branche numero 184
 Sous-reseau: PREST_INTER, Branche numero 205

Liste des occurrences du symbole: GS

Sous-reseau: INDIR, Branche numero 133
 Sous-reseau: PROP, Branche numero 219
 Sous-reseau: PROP, Branche numero 221
 Sous-reseau: QUEST, Branche numero 236
 Sous-reseau: RELAT, Branche numero 250

Liste des occurrences du symbole: v_indir_flechi

Sous-reseau: INDIR, Branche numero 134
 Sous-reseau: INDIR, Branche numero 141

Liste des occurrences du symbole: v_indir_n_flechi

Sous-reseau: INDIR, Branche numero 147
 Sous-reseau: INDIR, Branche numero 151
 Sous-reseau: INFINITIVE, Branche numero 163

Liste des occurrences du symbole: si

Sous-reseau: INDIR, Branche numero 152
 Sous-reseau: INFINITIVE, Branche numero 167

Liste des occurrences du symbole: GR_INTER

Sous-reseau: INDIR, Branche numero 153
 Sous-reseau: INFINITIVE, Branche numero 166
 Sous-reseau: QUEST, Branche numero 234

Liste des occurrences du symbole: aux_inf

Sous-reseau: INFINITIVE, Branche numero 159

Liste des occurrences du symbole: SGV

Sous-reseau: INFINITIVE, Branche numero 165
 Sous-reseau: PREST_1, Branche numero 181
 Sous-reseau: PREST_1, Branche numero 182
 Sous-reseau: PROP, Branche numero 223
 Sous-reseau: RELAT, Branche numero 251

Liste des occurrences du symbole: GR_COMP

Sous-reseau: INFINITIVE, Branche numero 168
 Sous-reseau: PREST_2, Branche numero 191
 Sous-reseau: PROP, Branche numero 218
 Sous-reseau: PROP, Branche numero 225
 Sous-reseau: RELAT, Branche numero 253

Liste des occurrences du symbole: mille

Sous-reseau: MILLIER, Branche numero 173

Liste des occurrences du symbole: million

Sous-reseau: MILLION, Branche numero 174

Liste des occurrences du symbole: c_est

Sous-reseau: PREST_1, Branche numero 175
 Sous-reseau: PREST_1, Branche numero 176
 Sous-reseau: PREST_1, Branche numero 181
 Sous-reseau: PREST_1, Branche numero 182
 Sous-reseau: PREST_2, Branche numero 183
 Sous-reseau: PREST_2, Branche numero 184
 Sous-reseau: PREST_INTER, Branche numero 204
 Sous-reseau: PREST_INTER, Branche numero 205
 Sous-reseau: PREST_INTER, Branche numero 211

Liste des occurrences du symbole: INFINITIVE

Sous-reseau: PREST_1, Branche numero 178

Sous-reseau: PREST_2, Branche numero 185
Sous-reseau: PREST_2, Branche numero 197
Sous-reseau: SGV, Branche numero 264
Sous-reseau: SUB_CIRC, Branche numero 268

Liste des occurrences du symbole: ce_que
Sous-reseau: PREST_1, Branche numero 180
Sous-reseau: PREST_2, Branche numero 195

Liste des occurrences du symbole: GOI
Sous-reseau: PREST_2, Branche numero 188
Sous-reseau: SGV, Branche numero 259
Sous-reseau: SGV, Branche numero 262

Liste des occurrences du symbole: GV
Sous-reseau: PROP, Branche numero 221
Sous-reseau: PROP, Branche numero 222
Sous-reseau: QUEST, Branche numero 236
Sous-reseau: RELAT, Branche numero 249
Sous-reseau: RELAT, Branche numero 250

Liste des occurrences du symbole: PREST_INTER
Sous-reseau: QUEST, Branche numero 227

Liste des occurrences du symbole: INDIR
Sous-reseau: QUEST, Branche numero 228

Liste des occurrences du symbole: pr_relat
Sous-reseau: RELAT, Branche numero 248

Liste des occurrences du symbole: par
Sous-reseau: SGV, Branche numero 255

Liste des occurrences du symbole: co_sub
Sous-reseau: SUB_CIRC, Branche numero 267

ANNEXE E : Modélisation sémantique du sous-langage

L'annexe E présente la sémantique du sous-langage de l'application test « demande de renseignements » telle qu'elle a été définie par G. Deville dans sa thèse de doctorat (Deville, 1989). Nous nous sommes limités ici à la présentation du modèle fonctionnel. Nous renvoyons le lecteur à la thèse de G. Deville pour une discussion plus détaillée du sujet.

1 Symboles utilisés dans le modèle fonctionnel

1.1 Les traits sémantiques des primitives

CTRL	:	contrôlé
DYN	:	dynamique
ATTR	:	attributif
SPAT	:	spatial
COGN	:	cognitif
TRS	:	transitif
DITRS	:	ditransitif

1.2 Les primitives

1.2.1 Les actes

ACT_1	:	prédict à un argument dénotant une activité contrôlée à l'exclusion des actes demouvement, cognitifs ou attributifs p. ex. : pointer
ACT_2	:	prédict à deux arguments dénotant une activité contrôlée à l'exclusion des actes de mouvement, cognitifs ou attributifs p. ex. : éditer
ACT_3	:	prédict à trois arguments dénotant une activité contrôlée à l'exclusion des actes de mouvement, cognitifs ou attributifs p. ex. : associer
ECHPROD_3	:	prédict à trois arguments dénotant un transfert contrôlé de possession p. ex. : acquérir
MVMT_1	:	prédict à un argument dénotant dénotant un mouvement contrôlé d'une entité sans point de référence explicite dans l'espace p. ex. : se balader
MVMT_2	:	prédict à deux arguments dénotant un mouvement contrôlé d'une entité par rapport à un point de référence explicite dans l'espace p. ex. : aller
MVMT_3	:	prédict à trois arguments dénotant un mouvement contrôlé par une entité d'une autre entité par rapport à un point de référence explicite dans l'espace p. ex. : mettre
COGNT_1	:	prédict à un argument dénotant un acte contrôlé cognitif p. ex. : réfléchir
COGNT_2	:	prédict à deux arguments dénotant un acte contrôlé cognitif p. ex. : regarder
COGNT_3	:	prédict à trois arguments dénotant un acte contrôlé cognitif p. ex. : dire

1.2.2 Les états

STATUT_1	:	précat à un argument dénotant un état incontrôlé à l'exclusion des états attributifs, spatiaux ou cognitifs p. ex. : être mineur
STATUT_2	:	précat à deux arguments dénotant un état incontrôlé à l'exclusion des états attributifs, spatiaux ou cognitifs p. ex. : ressembler
POSSESS_2	:	précat à deux arguments dénotant un état attributif incontrôlé p. ex. : posséder
LOCATION_2	:	précat à deux arguments dénotant un état locatif incontrôlé p. ex. : se trouver
LOCATION_3	:	précat à trois arguments dénotant un état locatif incontrôlé p. ex. : tenir
ST_COGN_2	:	précat à deux arguments dénotant un état cognitif incontrôlé p. ex. : savoir

1.2.3 Les processus

PROCESS_1	:	précat à un argument dénotant un processus incontrôlé à l'exclusion des processus spatiaux ou cognitifs p. ex. : fonctionner
PROCESS_2	:	précat à deux arguments dénotant un processus incontrôlé à l'exclusion des processus spatiaux ou cognitifs p. ex. : abîmer
ECHOBT_2	:	précat à deux arguments dénotant un processus attributif p. ex. : recevoir
PR_MVMT_1	:	précat à un argument dénotant un mouvement incontrôlé d'une entité sans point de référence explicite dans l'espace p. ex. : tomber
PR_MVMT_2	:	précat à deux arguments dénotant un mouvement incontrôlé d'une entité par rapport à un point de référence explicite dans l'espace p. ex. : aboutir
PR_MVMT_3	:	précat à trois arguments dénotant un processus spatial

PR_COGNT_2 : p. ex. : déplacer
 prédicat à deux arguments
 dénotant un processus cognitif
 p. ex. : voir

2 Le modèle fonctionnel de notre sous-langage

2.1 La typologie des primitives

Types des primitives	Noms des primitives	Traits sémantiques des primitives						
		CTRL	DYN	ATTR	SPAT	COGN	TRS	DITRS
ACTES	ACT_1	+	+	-	-	-	-	-
	ACT_2	+	+	-	-	-	+	-
	ACT_3	+	+	-	-	-	-	+
	ECHPROD_3	+	+	+	-	-	-	+
	MVMT_1	+	+	-	+	-	-	-
	MVMT_2	+	+	-	+	-	+	-
	MVMT_3	+	+	-	+	-	-	+
	COGNT_1	+	+	-	-	+	-	-
	COGNT_2	+	+	-	-	+	+	-
	COGNT_3	+	+	-	-	+	-	+
ETATS	STATUT_1	-	-	-	-	-	-	-
	STATUT_2	-	-	-	-	-	+	-
	POSSESS_2	-	-	+	-	-	+	-
	LOCATION_2	-	-	-	+	-	+	-
	LOCATION_3	+	-	-	+	-	-	+
	ST_COGN_2	-	-	-	-	+	+	-
PROCESSUS	PROCESS_1	-	+	-	-	-	-	-
	PROCESS_2	-	+	-	-	-	+	-
	ECHOBT_2	-	+	+	-	-	+	-
	PR_MVMT_1	-	+	-	+	-	-	-
	PR_MVMT_2	-	+	-	+	-	+	-
	PR_MVMT_3	-	+	-	+	-	-	+
	PR_COGNT_2	-	+	-	-	+	+	-

fig. C.1 : Les traits sémantiques de primitives

2.2 Les schémas de cas associés aux primitives

Noms des primitives	Configurations de cas
ACT_1 ACT_2 ACT_3 ECHPROD_3 MVMT_1 MVMT_2 MVMT_3 COGNT_1 COGNT_2 COGNT_3	Agent Agent + Theme Agent + Theme + Referent Agent + Theme + Benefactive Agent Agent + (Source Direction Path) Agent + Theme + (Source Direction Path) Agent Agent + Theme Agent + Theme + Cognizant
STATUT_1 STATUT_2 POSSESS_2 LOCATION_2 LOCATION_3 ST_COGN_2	Theme Theme + Referent Benefactive + Theme Theme + Locative Positioner + Theme + Locative Cognizant + Theme
PROCESS_1 PROCESS_2 ECHOBT_2 PR_MVMT_1 PR_MVMT_2 PR_MVMT_3 PR_COGNT_2	Processed Effector + Theme Benefactive + Theme Processed Processed + (Source Direction Path) Effector + Theme + (Source Direction Path) Cognizant + Theme

fig. C.2 : Les configurations de cas

ANNEXE F : Le programme CRESEM

L'annexe F fournit un complément d'informations au sujet du programme de création des connaissances sémantiques CRESEM. Outre un bref rappel des objectifs de ce programme, elle contient la syntaxe complète d'un texte traité par CRESEM. Elle présente également un exemple complet d'utilisation de ce programme.

1 Objectif du programme CRESEM

Le programme de création des connaissances sémantiques a pour objectif de permettre un passage aisé d'une description externe de la sémantique d'un langage vers sa représentation interne. La description externe est fournie par l'utilisateur sous forme d'un texte respectant une certaine présentation (définie dans le paragraphe suivant) qui est alors traité par le programme CRESEM.

L'utilisateur active le programme CRESEM et lui fournit certains paramètres qui peuvent être des options et/ou un fichier. Les deux options possibles sont '-l' si l'utilisateur désire un listing source et '-d' s'il désire une version lisible de la représentation interne construite. Le fichier contient le texte à traiter. Si aucun fichier n'est spécifié, CRESEM attend les données sur le fichier standard en entrée.

Si le fichier spécifié n'existe pas, CRESEM termine son exécution de façon anormale. Si le fichier existe mais que le texte ne respecte pas la syntaxe acceptée par CRESEM, les erreurs sont signalées et aucune représentation interne n'est construite. Sinon, la représentation interne est créée sur un fichier de même nom que le fichier en entrée suffixé par '.cgr'.

Une version ancienne de ce programme est décrite en détail dans (Mousel, 1988). Si ce rapport reflète encore assez bien la philosophie qui nous a guidée dans la conception de ce programme, la syntaxe du texte traité par

CRESEM ainsi que la représentation interne ont quand-même fortement évoluées comme on pourra le constater dans les paragraphes suivants.

2 Syntaxe du texte traité par CRESEM

Pour décrire la syntaxe du texte destiné à être traité par le programme CRESEM, nous avons utilisé les conventions suivantes :

- les non-terminaux sont entre crochets '<' et '>';
- les terminaux sont en caractères gras ou entre simples quotes ;
- la production est indiquée par '::=' ;
- les alternatives de production sont indiquées par la barre verticale '|';
- la fin d'une règle de production est indiquée par le point-virgule ';';
- le symbole vide est indiqué par l'accent circonflexe '^'.

La syntaxe est la suivante :

Les terminaux peuvent s'écrire indépendamment en majuscules ou en minuscules. Certains sont des mots clés, il s'agit de :

Enfin, un **Identificateur** est composé d'une lettre suivie de lettres, chiffres ou de caractères '_' (souligné). Il faut encore remarquer que deux identificateurs, ne se distinguant que par l'écriture majuscule ou minuscule de certains des caractères les composant, sont considérés comme différents.

3 Exemple d'utilisation

L'exemple suivant illustre l'utilisation du programme CRESEM. Le texte fourni en entrée décrit la grammaire de cas présentée en Annexe E, définissant ainsi la sémantique de notre application test. Ce texte est rédigé à l'aide d'un éditeur de texte quelconque. Un listing en est fourni dans le paragraphe 'Les données'. Traité par le programme CRESEM avec les options '-l' (listing source) et '-d' (démonstration), il donne lieu à deux types de résultat, un fichier sur disque (suffixé par '.cgr') contenant la représentation interne de la sémantique et un texte affiché à l'écran (sortie standard redirigeable) composé d'un listing source et d'une version lisible de la représentation interne. Un listing du texte affiché à l'écran est fourni dans le paragraphe 'Les résultats'.

3.1 Les données

```
Primitives:    act_1,
               act_2,
               act_3,
               echprod_3,
               mvmt_1,
               mvmt_2,
               mvmt_3,
               cognt_1,
               cognt_2,
               cognt_3,
               statut_1,
               statut_2,
               possess_2,
               location_2,
               location_3,
               st_cognt_2,
               process_1,
               process_2,
               echobt_2,
               pr_mvmt_1,
               pr_mvmt_2,
               pr_mvmt_3,
               pr_cognt_2;
```

Cas-centraux:

cas	trait-semantique
AGENT	animate,
THEME	aucun,
REFERENT	aucun,
BENEFACTIVE	animate,
SOURCE	space,
DIRECTION	space,
PATH_1	space,
COGNIZANT	human,
LOCATIVE	space,
POSITIONER	human,
PROCESSED	aucun,
EFFECTOR	aucun;

Cas-peripheriques:

cas	trait-semantique
MEDIUM	aucun,
MANNER	aucun,
CLIENT	animate,
COMITATIVE	animate,
PATH_2	space,
DURATION	time,
FREQUENCY	time,
LOCATION	space,
TIME	time,
CAUSE	aucun,
CONDITION	aucun,
PURPOSE	aucun;

Regles:

```
act_1 (AGENT),
act_2 (AGENT,THEME),
act_3 (AGENT,THEME,REFERENT),
echprod_3 (AGENT,THEME,BENEFACTIVE),
mvmt_1 (AGENT),
mvmt_2 (AGENT,SOURCE),
```

```

mvmt_2 (AGENT, DIRECTION) ,
mvmt_2 (AGENT, PATH_1) ,
mvmt_3 (AGENT, THEME, SOURCE) ,
mvmt_3 (AGENT, THEME, DIRECTION) ,
mvmt_3 (AGENT, THEME, PATH_1) ,
cognt_1 (AGENT) ,
cognt_2 (AGENT, THEME) ,
cognt_3 (AGENT, THEME, COGNIZANT) ,

statut_1 (THEME) ,
statut_2 (THEME, REFERENT) ,
possess_2 (BENEFACTIVE, THEME) ,
location_2 (THEME, LOCATIVE) ,
location_3 (POSITIONER, THEME, LOCATIVE) ,
st_cognt_2 (COGNIZANT, THEME) ,

process_1 (PROCESSED) ,
process_2 (EFFECTOR, THEME) ,
echobt_2 (BENEFACTIVE, THEME) ,
pr_mvmt_1 (PROCESSED) ,
pr_mvmt_2 (PROCESSED, SOURCE) ,
pr_mvmt_2 (PROCESSED, DIRECTION) ,
pr_mvmt_2 (PROCESSED, PATH_1) ,
pr_mvmt_3 (PROCESSED, THEME, SOURCE) ,
pr_mvmt_3 (PROCESSED, THEME, DIRECTION) ,
pr_mvmt_3 (PROCESSED, THEME, PATH_1) ,
pr_cognt_2 (COGNIZANT, THEME) ;

```

3.2 Les résultats

```

Primitives:      act_1,
                  act_2,
                  act_3,
                  echprod_3,
                  mvmt_1,
                  mvmt_2,
                  mvmt_3,
                  cognt_1,
                  cognt_2,
                  cognt_3,
                  statut_1,
                  statut_2,
                  possess_2,
                  location_2,
                  location_3,
                  st_cognt_2,
                  process_1,
                  process_2,
                  echobt_2,
                  pr_mvmt_1,
                  pr_mvmt_2,
                  pr_mvmt_3,
                  pr_cognt_2;

```

Cas-centraux:

cas	trait-sémantique
AGENT	animate,
THEME	aucun,
REFERENT	aucun,
BENEFACTIVE	animate,
SOURCE	space,
DIRECTION	space,
PATH_1	space,

```

COGNIZANT human,
LOCATIVE space,
POSITIONER human,
PROCESSED aucun,
EFFECTOR aucun;

```

Cas-peripheriques:

```

cas trait-semantic

MEDIUM aucun,
MANNER aucun,
CLIENT animate,
COMITATIVE animate,
PATH_2 space,
DURATION time,
FREQUENCY time,
LOCATION space,
TIME time,
CAUSE aucun,
CONDITION aucun,
PURPOSE aucun;

```

Regles:

```

act_1 (AGENT),
act_2 (AGENT, THEME),
act_3 (AGENT, THEME, REFERENT),
echprod_3 (AGENT, THEME, BENEFACTIVE),
mvmt_1 (AGENT),
mvmt_2 (AGENT, SOURCE),
mvmt_2 (AGENT, DIRECTION),
mvmt_2 (AGENT, PATH_1),
mvmt_3 (AGENT, THEME, SOURCE),
mvmt_3 (AGENT, THEME, DIRECTION),
mvmt_3 (AGENT, THEME, PATH_1),
cognt_1 (AGENT),
cognt_2 (AGENT, THEME),
cognt_3 (AGENT, THEME, COGNIZANT),

statut_1 (THEME),
statut_2 (THEME, REFERENT),
possess_2 (BENEFACTIVE, THEME),
location_2 (THEME, LOCATIVE),
location_3 (POSITIONER, THEME, LOCATIVE),
st_cognt_2 (COGNIZANT, THEME),

process_1 (PROCESSED),
process_2 (EFFECTOR, THEME),
echobt_2 (BENEFACTIVE, THEME),
pr_mvmt_1 (PROCESSED),
pr_mvmt_2 (PROCESSED, SOURCE),
pr_mvmt_2 (PROCESSED, DIRECTION),
pr_mvmt_2 (PROCESSED, PATH_1),
pr_mvmt_3 (PROCESSED, THEME, SOURCE),
pr_mvmt_3 (PROCESSED, THEME, DIRECTION),
pr_mvmt_3 (PROCESSED, THEME, PATH_1),
pr_cognt_2 (COGNIZANT, THEME);

```

Contenu de la grammaire de cas: dem_rens

Liste des primitives:

=====

```

act_1
act_2

```

act_3
 echprod_3
 mvmt_1
 mvmt_2
 mvmt_3
 cognt_1
 cognt_2
 cognt_3
 statut_1
 statut_2
 possess_2
 location_2
 location_3
 st_cognt_2
 process_1
 process_2
 echobt_2
 pr_mvmt_1
 pr_mvmt_2
 pr_mvmt_3
 pr_cognt_2

Liste des cas centraux:
 =====

animate (AGENT)
 aucun (THEME)
 aucun (REFERENT)
 animate (BENEFACTIVE)
 space (SOURCE)
 space (DIRECTION)
 space (PATH_1)
 human (COGNIZANT)
 space (LOCATIVE)
 human (POSITIONER)
 aucun (PROCESSED)
 aucun (EFFECTOR)

Liste des cas peripheriques:
 =====

aucun (MEDIUM)
 aucun (MANNER)
 animate (CLIENT)
 animate (COMITATIVE)
 space (PATH_2)
 time (DURATION)
 time (FREQUENCY)
 space (LOCATION)
 time (TIME)
 aucun (CAUSE)
 aucun (CONDITION)
 aucun (PURPOSE)

La liste des regles semantiques:
 =====

act_1 (AGENT)
 act_2 (AGENT, THEME)
 act_3 (AGENT, THEME, REFERENT)
 echprod_3 (AGENT, THEME, BENEFACTIVE)
 mvmt_1 (AGENT)
 mvmt_2 (AGENT, SOURCE)
 mvmt_2 (AGENT, DIRECTION)

```

mvmt_2 (AGENT, PATH_1)
mvmt_3 (AGENT, THEME, SOURCE)
mvmt_3 (AGENT, THEME, DIRECTION)
mvmt_3 (AGENT, THEME, PATH_1)
cognt_1 (AGENT)
cognt_2 (AGENT, THEME)
cognt_3 (AGENT, THEME, COGNIZANT)
statut_1 (THEME)
statut_2 (THEME, REFERENT)
possess_2 (BENEFACTIVE, THEME)
location_2 (THEME, LOCATIVE)
location_3 (POSITIONER, THEME, LOCATIVE)
st_cognt_2 (COGNIZANT, THEME)
process_1 (PROCESSED)
process_2 (EFFECTOR, THEME)
echobt_2 (BENEFACTIVE, THEME)
pr_mvmt_1 (PROCESSED)
pr_mvmt_2 (PROCESSED, SOURCE)
pr_mvmt_2 (PROCESSED, DIRECTION)
pr_mvmt_2 (PROCESSED, PATH_1)
pr_mvmt_3 (PROCESSED, THEME, SOURCE)
pr_mvmt_3 (PROCESSED, THEME, DIRECTION)
pr_mvmt_3 (PROCESSED, THEME, PATH_1)
pr_cognt_2 (COGNIZANT, THEME)

```

La liste des primitives par cas:

=====

```

AGENT (act_1, act_2, act_3, echprod_3, mvmt_1, mvmt_2, mvmt_2,
      mvmt_2, mvmt_3, mvmt_3, mvmt_3, cognt_1, cognt_2, cognt_3)
THEME (act_2, act_3, echprod_3, mvmt_3, mvmt_3, mvmt_3, cognt_2,
      cognt_3, statut_1, statut_2, possess_2, location_2,
      location_3, st_cognt_2, process_2, echobt_2, pr_mvmt_3,
      pr_mvmt_3, pr_mvmt_3, pr_cognt_2)
REFERENT (act_3, statut_2)
BENEFACTIVE (echprod_3, possess_2, echobt_2)
SOURCE (mvmt_2, mvmt_3, pr_mvmt_2, pr_mvmt_3)
DIRECTION (mvmt_2, mvmt_3, pr_mvmt_2, pr_mvmt_3)
PATH_1 (mvmt_2, mvmt_3, pr_mvmt_2, pr_mvmt_3)
COGNIZANT (cognt_3, st_cognt_2, pr_cognt_2)
LOCATIVE (location_2, location_3)
POSITIONER (location_3)
PROCESSED (process_1, pr_mvmt_1, pr_mvmt_2, pr_mvmt_2, pr_mvmt_2,
          pr_mvmt_3, pr_mvmt_3, pr_mvmt_3)
EFFECTOR (process_2)

```


ANNEXE G : La génération automatique de nœuds procéduraux

L'annexe G fournit un complément d'informations au sujet des programmes de génération des nœuds procéduraux NPR et DEFNPR. Outre un bref rappel des objectifs de ces programmes, elle décrit la façon de les utiliser et présente un exemple complet.

1 Objectif du programme NPR

Un des problèmes auxquels nous avons été confrontés lors du développement des analyseurs syntaxico-sémantiques, est un problème lié aux nœuds procéduraux des RNP. Comme leur nom l'indique, il s'agit de procédures qui sont à exécuter à des instants précis du parcours du RNP. Ayant choisi le langage C, un langage de programmation compilé, pour implanter les analyseurs syntaxico-sémantiques du système DIAL, le code source des nœuds procéduraux devait être disponible à la compilation et à l'édition de liens des analyseurs. Or, un des objectifs des RNP était une nette séparation entre connaissances et traitement de ces connaissances. Ceci signifie que, lors de la conception des analyseurs, on ignore tout des RNP et en particulier des nœuds procéduraux. Cependant, si au moment de la compilation et de l'édition de liens, le lien entre l'analyseur et les nœuds procéduraux doit se faire sans conflits, il est nécessaire d'avoir une interface bien définie entre les deux. Ce rôle d'interface revient à une fonction particulière, la fonction « activer ».

L'analyseur syntaxico-sémantique est conçu indépendamment d'un RNP particulier, ce n'est que lors du parcours d'un RNP que l'analyseur saura quels sont les nœuds procéduraux à activer. L'activation d'un nœud ne peut donc être réalisée par un simple appel de la fonction correspondant à ce nœud. Pour activer un nœud procédural, l'analyseur syntaxico-sémantique fait toujours appel à la fonction « activer » en lui fournissant le nom du nœud procédural à activer. Cette dernière se charge ensuite de faire appel à la

procédure correspondante. La fonction « activer » est par conséquent une fonction qui dépend d'un RNP particulier et elle doit être réécrite pour chaque RNP. Toutefois son code est assez simple et se limite à la recherche de la fonction correspondant au nom du nœud procédural que l'analyseur syntactico-sémantique lui a fourni, à l'appel de cette fonction et, après retour dans « activer », au passage du résultat du nœud procédural à l'analyseur appelant. Qui plus est, ce code peut être construit mécaniquement à partir d'un RNP particulier. D'où l'idée d'écrire un générateur de code pour cette fonction « activer », le programme NPR.

2 Objectif du programme DEFNPR

La fonction « activer » n'est pas la seule à devoir être réécrite pour chaque nouvel RNP, il en va de même pour toutes les procédures associées aux nœuds procéduraux. Ici encore, il s'agit de fonctions à écrire en langage C et qui devraient permettre de prendre en compte des phénomènes contextuels. Un nœud procédural doit par conséquent pouvoir prendre connaissance d'un contexte et limiter les productions possibles sur base de ce contexte, ce qui, en termes de RNP, signifie limiter les sorties possibles de ce nœud. Dans le cas le plus simple, un RNP génère un langage hors-contexte lorsque toutes les sorties d'un nœud sont toujours possibles, indépendamment du contexte. On peut écrire ces fonctions dans un premier temps et les raffiner par la suite en y ajoutant des tests sur le contexte et les limitations appropriées. Si cette dernière tâche semble difficilement automatisable, il n'en va pas de même de la première. En effet, l'écriture des fonctions qui font que toutes les sorties d'un nœud sont toujours possibles est tout-à-fait mécanique. D'où l'idée d'écrire un générateur de code pour ces fonctions associées aux nœuds procéduraux, le programme DEFNPR.

3 Utilisation des générateurs

Les deux générateurs de code s'utilisent de façon similiaire. A partir du fichier contenant la représentation interne des RNP construite par CRE-RES, ils génèrent le code de la fonction « activer » respectivement des fonctions associées aux nœuds procéduraux. Les deux générateurs envoient leurs résultats sur le fichier standard de sortie, il suffit de rediriger ce dernier pour avoir le code sur un fichier disque. Le générateur de la fonction « activer » a besoin, en plus de la représentation interne du RNP, du nom du fichier qui va contenir les fonctions associées aux nœuds procéduraux. La figure G.1 montre l'utilisation des deux générateurs de code.

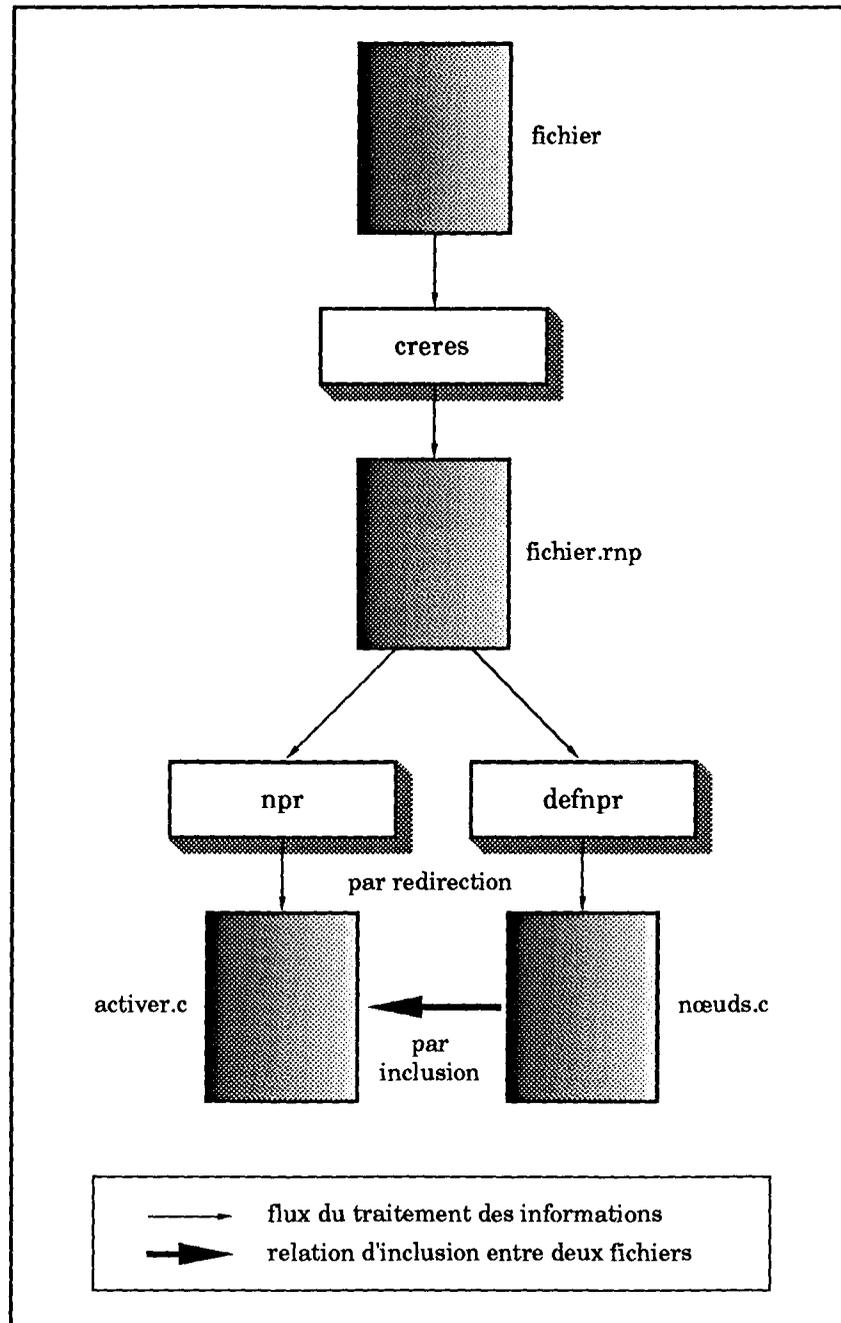


fig. G.1 : Utilisation de NPR et de DEFNPR

4 Exemple d'utilisation

Soit un fichier « fichier.rnp » contenant la représentation interne d'un RNP. Les programmes DEFNPR et NPR pourraient s'utiliser respectivement de la façon suivante :

```

$defnpr fichier.rnp > noeuds.c
$npr noeuds.c fichier.rnp > activer.c
$
  
```

Si le fichier « fichier.rnp » contenait la représentation interne du RNP présenté dans l'annexe C et construite par le programme CRERES présenté en annexe D, les résultats respectifs des deux générateurs de code seraient les suivants :

4.1 Résultat de DEFNPR

Afin de pouvoir interpréter les résultats, quelques remarques d'ordre technique sont nécessaires. Elles ne sont cependant pas indispensables à la bonne compréhension de la suite, le lecteur non intéressé par ces détails peut donc les ignorer sans problème.

Le générateur de code DEFNPR crée pour chaque nœud procédural une fonction C. Chaque fonction reçoit comme paramètre un nœud (le paramètre nds de type NDS de chaque fonction) d'un arbre syntaxique¹, qui correspond en principe au dernier symbole de la branche qui a provoqué l'activation du nœud procédural et qui lui permet de consulter le contexte de l'analyse, et fournit comme résultat la liste des sorties possibles du nœud procédural (toutes les fonctions sont de type LENT). Comme le générateur ne construit que des nœuds procéduraux élémentaires où toutes les sorties des nœuds sont possibles quelque soit le contexte de l'analyse, la tâche des différentes fonctions se limite à la création d'une liste de sorties vide, à l'ajout de toutes les sorties du nœud dans cette liste et au passage de cette liste vers la fonction appelante. Pour des raisons de place, nous nous sommes limités aux trois premiers RNP, CENTAINE, CHIF et DATE.

```
# include "global.h"

LENT CENTAINE_1 (nds)

NDS nds;

{
  LENT liste_sorties;
  liste_sorties = LENTINIT;
  creer_lent(&liste_sorties);
  ajout_lent(&liste_sorties,0);
  ajout_lent(&liste_sorties,1);
  ajout_lent(&liste_sorties,2);
  return(liste_sorties);
}

LENT CENTAINE_2 (nds)

NDS nds;

{
  LENT liste_sorties;
  liste_sorties = LENTINIT;
  creer_lent(&liste_sorties);
```

¹ La structure syntaxique de l'énoncé est construite par l'analyseur syntaxico-sémantique sur base du RNP au fur et à mesure de l'avancement de l'analyse. Elle est représentée sous forme d'un arbre syntaxique. L'analyseur syntaxico-sémantique sera décrit plus en détail dans un des chapitres suivants. Dans ce chapitre, il importe de ne pas confondre « nœud procédural » d'un RNP et « nœud » dans un arbre syntaxique.

```
    ajout_lent(&liste_sorties,0);
    return(liste_sorties);
}

LENT CENTAINE_3 (nds)

NDS nds;

{
    LENT liste_sorties;
    liste_sorties = LENTINIT;
    creer_lent(&liste_sorties);
    ajout_lent(&liste_sorties,0);
    ajout_lent(&liste_sorties,1);
    return(liste_sorties);
}

LENT CENTAINE_4 (nds)

NDS nds;

{
    LENT liste_sorties;
    liste_sorties = LENTINIT;
    creer_lent(&liste_sorties);
    return(liste_sorties);
}

LENT CHIF_1 (nds)

NDS nds;

{
    LENT liste_sorties;
    liste_sorties = LENTINIT;
    creer_lent(&liste_sorties);
    ajout_lent(&liste_sorties,0);
    ajout_lent(&liste_sorties,1);
    return(liste_sorties);
}

LENT CHIF_2 (nds)

NDS nds;

{
    LENT liste_sorties;
    liste_sorties = LENTINIT;
    creer_lent(&liste_sorties);
    ajout_lent(&liste_sorties,0);
    ajout_lent(&liste_sorties,1);
    return(liste_sorties);
}

LENT CHIF_3 (nds)

NDS nds;

{
    LENT liste_sorties;
    liste_sorties = LENTINIT;
    creer_lent(&liste_sorties);
    ajout_lent(&liste_sorties,0);
    ajout_lent(&liste_sorties,1);
    return(liste_sorties);
}

LENT CHIF_4 (nds)

NDS nds;
```

```
{
  LENT liste_sorties;
  liste_sorties = LENTINIT;
  creer_lent(&liste_sorties);
  ajout_lent(&liste_sorties,0);
  ajout_lent(&liste_sorties,1);
  return(liste_sorties);
}
```

LENT CHIF_5 (nds)

NDS nds;

```
{
  LENT liste_sorties;
  liste_sorties = LENTINIT;
  creer_lent(&liste_sorties);
  return(liste_sorties);
}
```

LENT DATE_1 (nds)

NDS nds;

```
{
  LENT liste_sorties;
  liste_sorties = LENTINIT;
  creer_lent(&liste_sorties);
  ajout_lent(&liste_sorties,0);
  ajout_lent(&liste_sorties,1);
  return(liste_sorties);
}
```

LENT DATE_2 (nds)

NDS nds;

```
{
  LENT liste_sorties;
  liste_sorties = LENTINIT;
  creer_lent(&liste_sorties);
  ajout_lent(&liste_sorties,0);
  ajout_lent(&liste_sorties,1);
  return(liste_sorties);
}
```

LENT DATE_3 (nds)

NDS nds;

```
{
  LENT liste_sorties;
  liste_sorties = LENTINIT;
  creer_lent(&liste_sorties);
  ajout_lent(&liste_sorties,0);
  ajout_lent(&liste_sorties,1);
  ajout_lent(&liste_sorties,2);
  return(liste_sorties);
}
```

LENT DATE_4 (nds)

NDS nds;

```
{
  LENT liste_sorties;
  liste_sorties = LENTINIT;
  creer_lent(&liste_sorties);
  ajout_lent(&liste_sorties,0);
}
```

```

return(liste_sorties);
}

LENT DATE_5 (nds)

NDS nds;

{
LENT liste_sorties;
liste_sorties = LENTINIT;
creer_lent(&liste_sorties);
ajout_lent(&liste_sorties,0);
ajout_lent(&liste_sorties,1);
return(liste_sorties);
}

LENT DATE_6 (nds)

NDS nds;

{
LENT liste_sorties;
liste_sorties = LENTINIT;
creer_lent(&liste_sorties);
return(liste_sorties);
}

...

```

4.2 Post-traitement des résultats de DEFNPR

Les nœuds procéduraux ainsi générés peuvent ensuite être modifiés à l'aide d'un éditeur quelconque. C'est ce que nous avons fait pour les nœuds CHIF_4, et DATE_5¹, pour lesquels nous avons voulu introduire des limitations contextuelles, d'où le fichier avec les nœuds procéduraux suivants (pour des raisons de place ne sont repris ci-dessous que les nœuds procéduraux modifiés) :

```

# include "global.h"

...

LENT CHIF_4 (nds)

NDS nds;

{
LENT liste_sorties;
NDS pere;
int i;
liste_sorties = LENTINIT;
creer_lent(&liste_sorties);
ajout_lent(&liste_sorties,0);
pere = ndpere_nds(nds);
for (i=0;
      (i<long_lnds(listef_nds(pere))) &&
      (long_chcar(symb_nds(elen_lnds(listef_nds(pere),i))) != 0);
      i++);
if (i<long_lnds(listef_nds(pere))) ajout_lent(&liste_sorties,1);

```

¹ Bien qu'ils n'apparaissent pas ici pour des raisons de place, les nœuds GR_ADJ_6, GR_ADJ_8 et SGV_2 ont également été modifiés dans le même sens.

```

return(liste_sorties);
}

...

LENT DATE_5 (nds)

NDS nds;

{
LENT liste_sorties;
NDS pere;
int i;
liste_sorties = LENTINIT;
creer_lent(&liste_sorties);
ajout_lent(&liste_sorties,0);
pere = ndpere_nds(nds);
for (i=0;
      (i<long_lnds(listef_nds(pere))) &&
      (long_chcar(symb_nds(elen_lnds(listef_nds(pere),i))) == 0);
      i++);
if (i<long_lnds(listef_nds(pere))) ajout_lent(&liste_sorties,1);
return(liste_sorties);
}

...

```

Les limitations contextuelles que nous avons introduites dans un premier temps sont les mêmes pour tous les nœuds procéduraux modifiés, il s'agit d'interdire la réécriture de certains RNP en symboles vides uniquement.

4.3 Résultat de NPR

Afin de pouvoir interpréter les résultats, quelques remarques d'ordre technique sont nécessaires. Elles ne sont cependant pas indispensables à la bonne compréhension de la suite, le lecteur non intéressé par ces détails peut donc les ignorer sans problème.

Le générateur de code NPR crée pour chaque nœud procédural une entrée dans une table (de nom « table_fonction ») constituée d'une chaîne de caractères correspondant au nom du nœud procédural et d'un pointeur de fonction correspondant à la fonction associée au nœud procédural. La fonction « activer » générée par NPR reçoit comme paramètres le nom du nœud procédural à activer (le paramètre *procedure* de type CHCAR) ainsi qu'un nœud (le paramètre *nds* de type NDS de chaque fonction) d'un arbre syntaxique, qui correspond en principe au dernier symbole de la branche qui a provoqué l'activation du nœud procédural, et fournit comme résultat la liste des sorties possibles du nœud procédural à activer (la fonction est de type LENT). La tâche de la fonction « activer » se résume à la recherche de l'entrée dans la table « table_fonction » dont le nom correspond au nom de nœud procédural qu'elle a reçu en paramètre, à l'appel de la fonction correspondant à cette entrée avec le nœud d'arbre syntaxique qu'elle a elle-même reçu en paramètre et, finalement, au passage du résultat de cette fonction vers la fonction appelante.

```
# include "global.h"
```

```
# include "noeuds.c"

struct str_table_fonction
{
    char *nom;
    LENT (*fonction) ();
} table_fonction [] = {
    "CENTAINE_1", CENTAINE_1,
    "CENTAINE_2", CENTAINE_2,
    "CENTAINE_3", CENTAINE_3,
    "CENTAINE_4", CENTAINE_4,
    "CHIF_1", CHIF_1,
    "CHIF_2", CHIF_2,
    "CHIF_3", CHIF_3,
    "CHIF_4", CHIF_4,
    "CHIF_5", CHIF_5,
    "DATE_1", DATE_1,
    "DATE_2", DATE_2,
    "DATE_3", DATE_3,
    "DATE_4", DATE_4,
    "DATE_5", DATE_5,
    "DATE_6", DATE_6,
    "DECLAR_1", DECLAR_1,
    "DECLAR_2", DECLAR_2,
    "DECLAR_3", DECLAR_3,
    "DIZAINE_1", DIZAINE_1,
    "DIZAINE_2", DIZAINE_2,
    "DIZAINE_3", DIZAINE_3,
    "DIZAINE_4", DIZAINE_4,
    "ENONCE_1", ENONCE_1,
    "ENONCE_2", ENONCE_2,
    "ENONCE_3", ENONCE_3,
    "ENONCE_4", ENONCE_4,
    "GN_1", GN_1,
    "GN_2", GN_2,
    "GN_5", GN_5,
    "GN_3", GN_3,
    "GN_4", GN_4,
    "GN_6", GN_6,
    "GN_INTER_1", GN_INTER_1,
    "GN_INTER_2", GN_INTER_2,
    "GN_INTER_3", GN_INTER_3,
    "GN_INTER_4", GN_INTER_4,
    "GOI_1", GOI_1,
    "GOI_2", GOI_2,
    "GOI_3", GOI_3,
    "GP_1", GP_1,
    "GP_2", GP_2,
    "GS_1", GS_1,
    "GS_2", GS_2,
    "GR_ADJ_1", GR_ADJ_1,
    "GR_ADJ_2", GR_ADJ_2,
    "GR_ADJ_3", GR_ADJ_3,
    "GR_ADJ_4", GR_ADJ_4,
    "GR_ADJ_5", GR_ADJ_5,
    "GR_ADJ_6", GR_ADJ_6,
    "GR_ADJ_8", GR_ADJ_8,
    "GR_ADJ_10", GR_ADJ_10,
    "GR_ADJ_9", GR_ADJ_9,
    "GR_ADJ_7", GR_ADJ_7,
    "GR_COMP_1", GR_COMP_1,
    "GR_COMP_2", GR_COMP_2,
    "GR_COMP_4", GR_COMP_4,
    "GR_COMP_3", GR_COMP_3,
    "GR_INTER_1", GR_INTER_1,
    "GR_INTER_2", GR_INTER_2,
    "GR_INTER_4", GR_INTER_4,
    "GR_INTER_3", GR_INTER_3,
    "GR_THEM_1", GR_THEM_1,
    "GR_THEM_2", GR_THEM_2,
```

"GV_1", GV_1,
"GV_2", GV_2,
"GV_4", GV_4,
"GV_3", GV_3,
"GV_5", GV_5,
"GV_6", GV_6,
"GV_7", GV_7,
"INDIR_1", INDIR_1,
"INDIR_2", INDIR_2,
"INDIR_3", INDIR_3,
"INDIR_4", INDIR_4,
"INDIR_7", INDIR_7,
"INDIR_8", INDIR_8,
"INDIR_5", INDIR_5,
"INDIR_9", INDIR_9,
"INDIR_6", INDIR_6,
"INDIR_10", INDIR_10,
"INDIR_11", INDIR_11,
"INDIR_12", INDIR_12,
"INDIR_13", INDIR_13,
"INDIR_14", INDIR_14,
"INFINITIVE_1", INFINITIVE_1,
"INFINITIVE_2", INFINITIVE_2,
"INFINITIVE_3", INFINITIVE_3,
"INFINITIVE_4", INFINITIVE_4,
"INFINITIVE_5", INFINITIVE_5,
"INFINITIVE_6", INFINITIVE_6,
"INFINITIVE_7", INFINITIVE_7,
"INFINITIVE_8", INFINITIVE_8,
"INFINITIVE_9", INFINITIVE_9,
"MILLIER_1", MILLIER_1,
"MILLIER_2", MILLIER_2,
"MILLIER_3", MILLIER_3,
"MILLION_1", MILLION_1,
"MILLION_2", MILLION_2,
"PREST_1_1", PREST_1_1,
"PREST_1_3", PREST_1_3,
"PREST_1_2", PREST_1_2,
"PREST_2_1", PREST_2_1,
"PREST_2_2", PREST_2_2,
"PREST_2_3", PREST_2_3,
"PREST_2_4", PREST_2_4,
"PREST_2_5", PREST_2_5,
"PREST_2_6", PREST_2_6,
"PREST_INTER_1", PREST_INTER_1,
"PREST_INTER_2", PREST_INTER_2,
"PREST_INTER_3", PREST_INTER_3,
"PREST_INTER_4", PREST_INTER_4,
"PREST_INTER_5", PREST_INTER_5,
"PREST_INTER_6", PREST_INTER_6,
"PREST_INTER_7", PREST_INTER_7,
"PREST_INTER_8", PREST_INTER_8,
"PROP_1", PROP_1,
"PROP_2", PROP_2,
"PROP_3", PROP_3,
"PROP_4", PROP_4,
"PROP_5", PROP_5,
"PROP_6", PROP_6,
"QUEST_1", QUEST_1,
"QUEST_8", QUEST_8,
"QUEST_2", QUEST_2,
"QUEST_3", QUEST_3,
"QUEST_4", QUEST_4,
"QUEST_5", QUEST_5,
"QUEST_6", QUEST_6,
"QUEST_7", QUEST_7,
"RELAT_1", RELAT_1,
"RELAT_2", RELAT_2,
"RELAT_3", RELAT_3,
"RELAT_4", RELAT_4,

```
    "RELAT_5", RELAT_5,  
    "RELAT_6", RELAT_6,  
    "SGV_1", SGV_1,  
    "SGV_4", SGV_4,  
    "SGV_2", SGV_2,  
    "SGV_3", SGV_3,  
    "SUB_CIRC_1", SUB_CIRC_1,  
    "SUB_CIRC_2", SUB_CIRC_2  
    };  
  
LENT activer(procedure, nds)  
  
CHCAR procedure;  
NDS nds;  
  
{  
    char *procstr;  
    int i;  
    LENT liste_sorties;  
    liste_sorties = LENTINIT;  
    if (procedure == CHCARINIT) return(liste_sorties);  
    if (vide_chcar(procedure)) return(liste_sorties);  
    if (procstr = (char *) malloc(long_chcar(procedure)+1))  
    {  
        chcartostr(procstr, procedure);  
        for (i=0;  
            strcmp(procstr, table_fonction[i].nom);  
            i++;  
            liste_sorties = table_fonction[i].fonction (nds);  
            return(liste_sorties);  
        }  
    }  
}
```


ANNEXE H : Résultats de la composante SYNSEM

L'annexe H fournit un complément d'informations au sujet du prototype d'évaluation de la composante SYNSEM. Pour des raisons de place, nous avons renoncé à une description détaillée du prototype. Nous avons préféré présenter en détail deux exemples d'analyses d'énoncés qui mettent bien en évidence les possibilités du prototype.

1 Objectif du prototype d'évaluation

La composante SYNSEM a été conçue, dans un premier temps, comme un prototype d'évaluation. Ceci signifie qu'elle possède de nombreuses fonctions qui n'auront plus de sens dans une version opérationnelle du système DIAL mais qui, dans un contexte de développement, sont d'une utilité non négligeable. Ces caractéristiques sont :

- la possibilité de paramétrer la composante par les connaissances ;
- la possibilité de paramétrer la composante par les stratégies ;
- la possibilité d'interrompre une analyse et de la reprendre ultérieurement ;
- la possibilité d'avoir des statistiques sur l'évolution de l'espace des théories dans le temps ;
- la possibilité d'avoir une trace chronologique des événements ;
- la possibilité de sauvegarder régulièrement, en cours d'analyse, l'espace des théories ;
- la possibilité de garder l'espace des théories complet ou non jusqu'à la fin de l'analyse.

Le but des facilités offertes par le prototype d'évaluation est de rendre plus aisé le dépouillement des résultats de SYNSEM. Nous essaierons dans les paragraphes suivants de d'en illustrer l'utilisation et de montrer quelques résultats ainsi que les conclusions qu'on peut en tirer. Il faut encore signaler que dans les deux exemples, nous n'avons pas pris en compte les phénomènes d'erreurs au niveau du décodage acoustico-phonétique. Nous avons fait ce choix pour des raisons de clarté de l'exposé et afin de pouvoir présenter des résultats sur support papier.

2 Un exemple d'analyse

2.1 Les conditions de l'analyse

Dans l'exemple ci-dessous, la composante SYNSEM a analysé un énoncé extrait du corpus enregistré au Ministère de l'Emploi et du Travail (au cours de la transaction 1.9) : « est-ce que vous avez une documentation ». L'analyse a été effectuée en appliquant une stratégie de sélection totale. La stratégie de propagation consistait à affaiblir le score de l'arbre syntaxique lors de l'ajout d'un nœud non terminal en pondérant le score de l'arbre par le nombre de nœuds, à ne pas affaiblir le score de l'arbre lors de l'ajout d'un nœud terminal et à propager le score des mots validés en pondérant le score de l'arbre par le nombre de nœuds correspondant à des mots validés (les détails au sujet des stratégies sont exposés au chapitre 9). L'analyse a eu recours à toutes les facilités, une trace chronologique des événements et des statistiques sur l'évolution de l'espace des théories ont été produites, l'espace des théories a été sauvegardé après chaque phase d'analyse et l'espace des théories n'a pas été nettoyé.

2.2 L'espace des théories construit lors de l'analyse

Pour des raisons de place, nous n'avons repris dans cet exemple d'espace des théories que quelques théories parmi celles dont la sélection a conduit à une théorie complète. Les relations de filiation entre les théories qui ont conduit à une analyse complète sont présentées sur l'arbre de la figure H.1. Les feuilles de cet arbre correspondent aux théories complètes dans l'espace des théories. La présentation de l'espace des théories sous forme d'un texte lisible a été assurée par le programme LESP.

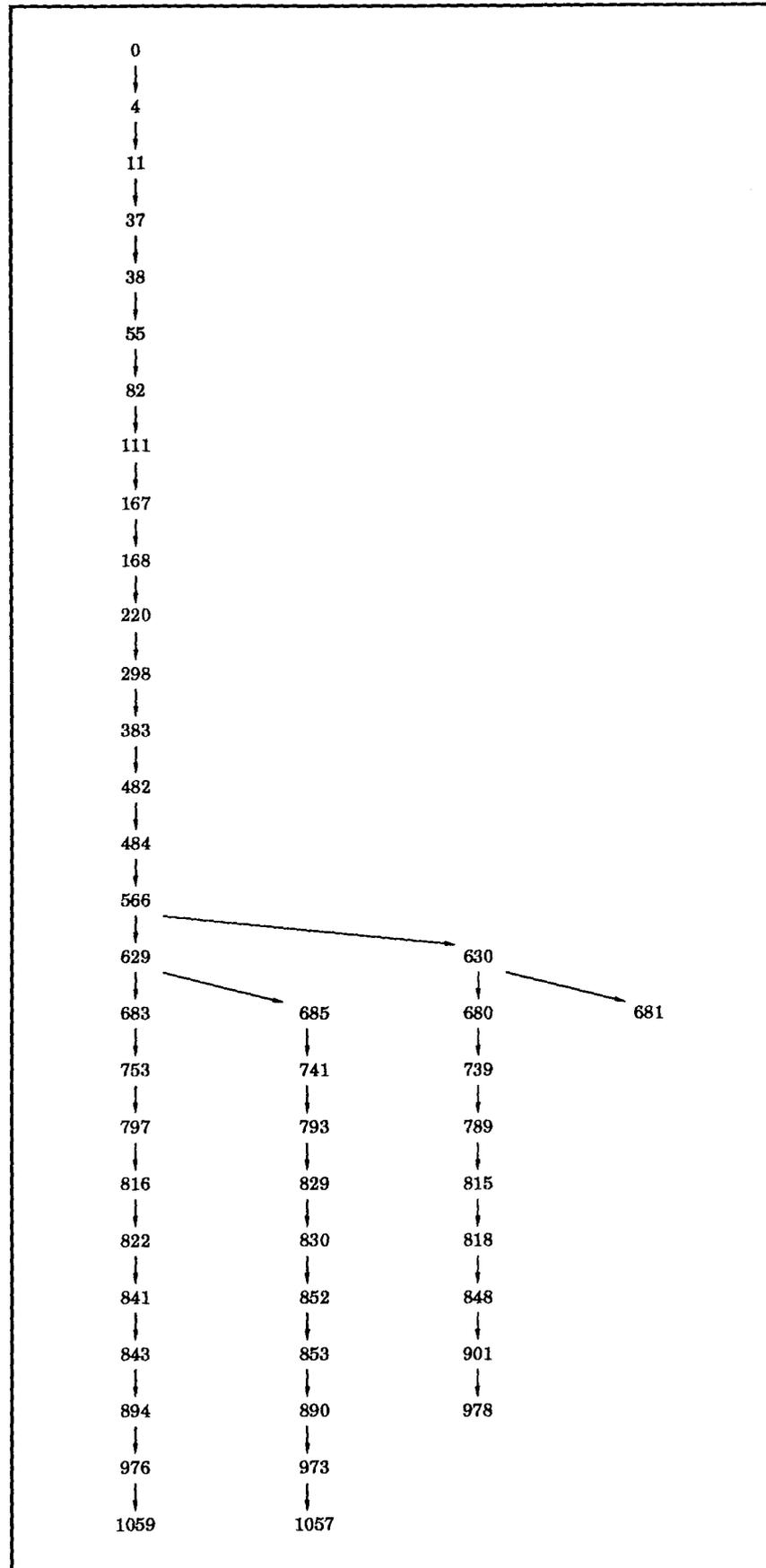


fig. H.1 : Les relations de filiation entre les théories

Strategies

```

debut de l'annonce..... : 0.000000
fin de l'annonce..... : 100.000000
score minimum..... : 0.000000
score maximum..... : 100.000000
intervalle acceptable..... : 100.000000
selection maximum..... : 2147483647
affaiblissement NDSNTERM..... : oui
propagation NDSNTERM..... : en fonction des noeuds
affaiblissement NDSTERM..... : non
propagation NDSREAL..... : en fonction des noeuds NDSREAL

```

Facilites

```

fichier contenant le rnp..... :
  /usr/users3/rfia/dialogue/mousel/OUTILS/ASYN/EXEMPLE/dem_rens.rnp
fichier contenant la cgr..... :
  /usr/users3/rfia/dialogue/mousel/OUTILS/ASYN/EXEMPLE/dem_rens.cgr
fichier contenant la trace..... :
  /usr/users3/rfia/dialogue/mousel/OUTILS/ASYN/EXEMPLE/demo.trace
fichier contenant les statistiques..... :
  /usr/users3/rfia/dialogue/mousel/OUTILS/ASYN/EXEMPLE/demo.stat
fichier contenant les sauvetages automatiques :
  /usr/users3/rfia/dialogue/mousel/OUTILS/ASYN/EXEMPLE/demo.the
numero derniere phase d'analyse..... : 23
dernier numero de theorie..... : 1251
frequence des sauvetages..... : 1
nettoyage automatique de l'espace..... : non

```

Contenu de l'espace des theories
*****Theorie numero: 0
=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 100.000000
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: 1 2 3 4 5 6 7

```

```

Arbre syntaxique de la theorie:
-----

```

```

Score de l'arbre syntaxique: 100.000000

```

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:

```

```

Arbre semantique de la theorie:
-----

```

```

Indetermine

```

```

...

```

Theorie numero: 4
=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 100.000000
Merel de la theorie : 0
Mere2 de la theorie : aucune
Filles de la theorie: 11 12 13

```

```

Arbre syntaxique de la theorie:
-----

```

```

Score de l'arbre syntaxique: 100.000000

```

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00

```

```

Arbre semantique de la theorie:
-----

```

Indetermine

...

Theorie numero: 11
 =====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 66.666664
 Merel de la theorie : 4
 Mere2 de la theorie : aucune
 Filles de la theorie: 35 36 37

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 66.666664

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

...

Theorie numero: 37
 =====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie en attente
 Score de la theorie : 66.666664
 Merel de la theorie : 11
 Mere2 de la theorie : aucune
 Filles de la theorie: 38

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 66.666664

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

...

Theorie numero: 630
 =====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 85.732323
 Merel de la theorie : 566
 Mere2 de la theorie : aucune
 Filles de la theorie: 680 681

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 85.732323

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00

```

term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00

```

Arbre sémantique de la théorie:

Indetermine

...

Theorie numero: 681

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie complete
Score de la theorie : 85.732323
Merel de la theorie : 630
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 85.732323

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00

```

Arbre sémantique de la théorie:

Indetermine

...

Theorie numero: 901

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 82.530899
Merel de la theorie : 848
Mere2 de la theorie : aucune
Filles de la theorie: 977 978

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 82.530899

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00

```

```

term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR COMP num: - sem: - deb: 45.00 fin: 56.00 fils:
nterm.: DATE num: - sem: - deb: 45.00 fin: 56.00 fils:
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00

```

Arbre sémantique de la théorie:

Indetermine

...

Theorie numero: 973

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 85.865463
Merel de la theorie : 890
Mere2 de la theorie : aucune
Filles de la theorie: 1056 1057

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 85.865463

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
term.: nom num: - sem: - deb: 56.00 fin: 100.00
real.: documentation num: - sem: - deb: 56.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indetermine

...

Theorie numero: 976

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 85.723907
Merel de la theorie : 894
Mere2 de la theorie : aucune
Filles de la theorie: 1058 1059

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 85.723907

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00

```

```

    real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
term.: nom num: - sem: - deb: 56.00 fin: 100.00
real.: documentation num: - sem: - deb: 56.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indetermine

...

Theorie numero: 978

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie complete
Score de la theorie : 82.530899
Merel de la theorie : 901
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 82.530899

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 56.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 56.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 56.00 fils:
nterm.: DATE num: - sem: - deb: 45.00 fin: 56.00 fils:
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00

```

Arbre sémantique de la théorie:

Indetermine

...

Theorie numero: 1057

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie complete
Score de la theorie : 85.865463
Merel de la theorie : 973

```

Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 85.865463

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 0.00 fin: 0.00
  nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
    term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
      real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
    nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
      term.: vide num: - sem: - deb: 18.00 fin: 18.00
      nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
        term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
          real.: vous num: - sem: - deb: 18.00 fin: 27.00
        nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
          term.: vide num: - sem: - deb: 27.00 fin: 27.00
          term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
            real.: avez num: - sem: - deb: 27.00 fin: 45.00
          term.: vide num: - sem: - deb: 45.00 fin: 45.00
          term.: vide num: - sem: - deb: 45.00 fin: 45.00
        nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
          term.: vide num: - sem: - deb: 45.00 fin: 45.00
          nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
            term.: vide num: - sem: - deb: 45.00 fin: 45.00
            term.: det num: - sem: - deb: 45.00 fin: 56.00
              real.: une num: - sem: - deb: 45.00 fin: 56.00
            term.: nom num: - sem: - deb: 56.00 fin: 100.00
              real.: documentation num: - sem: - deb: 56.00 fin: 100.00
            term.: vide num: - sem: - deb: 100.00 fin: 100.00
            term.: vide num: - sem: - deb: 100.00 fin: 100.00
          term.: vide num: - sem: - deb: 100.00 fin: 100.00
  term.: vide num: - sem: - deb: 100.00 fin: 100.00

```

Arbre semantique de la theorie:

Indetermine

..

Theorie numero: 1059

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie complete
 Score de la theorie : 85.723907
 Merel de la theorie : 976
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 85.723907

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 0.00 fin: 0.00
  nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
    term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
      real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
    nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
      term.: vide num: - sem: - deb: 18.00 fin: 18.00
      nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
        term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
          real.: vous num: - sem: - deb: 18.00 fin: 27.00
        nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
          term.: vide num: - sem: - deb: 27.00 fin: 27.00
          term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
            real.: avez num: - sem: - deb: 27.00 fin: 45.00
          term.: vide num: - sem: - deb: 45.00 fin: 45.00
          term.: vide num: - sem: - deb: 45.00 fin: 45.00
        nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
          nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
            term.: vide num: - sem: - deb: 45.00 fin: 45.00
            term.: det num: - sem: - deb: 45.00 fin: 56.00
              real.: une num: - sem: - deb: 45.00 fin: 56.00
            term.: nom num: - sem: - deb: 56.00 fin: 100.00
              real.: documentation num: - sem: - deb: 56.00 fin: 100.00
            term.: vide num: - sem: - deb: 100.00 fin: 100.00
            term.: vide num: - sem: - deb: 100.00 fin: 100.00
          term.: vide num: - sem: - deb: 100.00 fin: 100.00
          term.: vide num: - sem: - deb: 100.00 fin: 100.00
        term.: vide num: - sem: - deb: 100.00 fin: 100.00
  term.: vide num: - sem: - deb: 100.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indetermine

2.3 L'évolution de l'espace des théories

```

*** Phase d'analyse numero..... : 1
  Nombre de theories completes. : 0
    - recouvrant tout l'annonce. : 0
    - recouvrant une partie.... : 0
  Nombre de theories bloquées.. : 0
  Nombre de theories en attente : 0
  Nombre de theories pretes:... : 0
    - a etre developpees..... : 0
    - deja developpees..... : 0
  -----
  Nombre total de theories..... : 0

*** Phase d'analyse numero..... : 2
  Nombre de theories completes. : 0
    - recouvrant tout l'annonce. : 0
    - recouvrant une partie.... : 0
  Nombre de theories bloquées.. : 0
  Nombre de theories en attente : 1
  Nombre de theories pretes:... : 7
    - a etre developpees..... : 6
    - deja developpees..... : 1
  -----
  Nombre total de theories..... : 8

*** Phase d'analyse numero..... : 3
  Nombre de theories completes. : 0
    - recouvrant tout l'annonce. : 0
    - recouvrant une partie.... : 0
  Nombre de theories bloquées.. : 0
  Nombre de theories en attente : 7
  Nombre de theories pretes:... : 22
    - a etre developpees..... : 14
    - deja developpees..... : 8
  -----
  Nombre total de theories..... : 29

*** Phase d'analyse numero..... : 4
  Nombre de theories completes. : 0
    - recouvrant tout l'annonce. : 0
    - recouvrant une partie.... : 0
  Nombre de theories bloquées.. : 5
  Nombre de theories en attente : 17
  Nombre de theories pretes:... : 33
    - a etre developpees..... : 11
    - deja developpees..... : 22
  -----
  Nombre total de theories..... : 55

*** Phase d'analyse numero..... : 5
  Nombre de theories completes. : 0
    - recouvrant tout l'annonce. : 0
    - recouvrant une partie.... : 0
  Nombre de theories bloquées.. : 21
  Nombre de theories en attente : 14
  Nombre de theories pretes:... : 44
    - a etre developpees..... : 11
    - deja developpees..... : 33
  -----
  Nombre total de theories..... : 79

*** Phase d'analyse numero..... : 6
  Nombre de theories completes. : 0
    - recouvrant tout l'annonce. : 0
    - recouvrant une partie.... : 0
  Nombre de theories bloquées.. : 33
  Nombre de theories en attente : 14
  Nombre de theories pretes:... : 61

```

```

- a etre developpees..... : 16
- deja developpees..... : 45
-----
Nombre total de theories..... : 108

*** Phase d'analyse numero..... : 7

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie... : 0
Nombre de theories bloquées.. : 50
Nombre de theories en attente : 27
Nombre de theories pretes:... : 86
- a etre developpees..... : 25
- deja developpees..... : 61
-----
Nombre total de theories..... : 163

*** Phase d'analyse numero..... : 8

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie... : 0
Nombre de theories bloquées.. : 90
Nombre de theories en attente : 11
Nombre de theories pretes:... : 119
- a etre developpees..... : 33
- deja developpees..... : 86
-----
Nombre total de theories..... : 220

*** Phase d'analyse numero..... : 9

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie... : 0
Nombre de theories bloquées.. : 110
Nombre de theories en attente : 26
Nombre de theories pretes:... : 161
- a etre developpees..... : 42
- deja developpees..... : 119
-----
Nombre total de theories..... : 297

*** Phase d'analyse numero..... : 10

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie... : 0
Nombre de theories bloquées.. : 161
Nombre de theories en attente : 25
Nombre de theories pretes:... : 196
- a etre developpees..... : 35
- deja developpees..... : 161
-----
Nombre total de theories..... : 382

*** Phase d'analyse numero..... : 11

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie... : 0
Nombre de theories bloquées.. : 217
Nombre de theories en attente : 36
Nombre de theories pretes:... : 229
- a etre developpees..... : 33
- deja developpees..... : 196
-----
Nombre total de theories..... : 482

*** Phase d'analyse numero..... : 12

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie... : 0
Nombre de theories bloquées.. : 289
Nombre de theories en attente : 19
Nombre de theories pretes:... : 257
- a etre developpees..... : 27
- deja developpees..... : 230
-----
Nombre total de theories..... : 565

*** Phase d'analyse numero..... : 13

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0

```

```

- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 338
Nombre de theories en attente : 7
Nombre de theories pretes:... : 284
- a etre developpees..... : 27
- deja developpees..... : 257
-----
Nombre total de theories..... : 629

*** Phase d'analyse numero..... : 14

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 365
Nombre de theories en attente : 6
Nombre de theories pretes:... : 309
- a etre developpees..... : 25
- deja developpees..... : 284
-----
Nombre total de theories..... : 680

*** Phase d'analyse numero..... : 15

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 399
Nombre de theories en attente : 7
Nombre de theories pretes:... : 330
- a etre developpees..... : 21
- deja developpees..... : 309
-----
Nombre total de theories..... : 736

*** Phase d'analyse numero..... : 16

Nombre de theories completes. : 1
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 1
Nombre de theories bloquées.. : 429
Nombre de theories en attente : 15
Nombre de theories pretes:... : 343
- a etre developpees..... : 14
- deja developpees..... : 329
-----
Nombre total de theories..... : 788

*** Phase d'analyse numero..... : 17

Nombre de theories completes. : 1
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 1
Nombre de theories bloquées.. : 454
Nombre de theories en attente : 9
Nombre de theories pretes:... : 351
- a etre developpees..... : 8
- deja developpees..... : 343
-----
Nombre total de theories..... : 815

*** Phase d'analyse numero..... : 18

Nombre de theories completes. : 1
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 1
Nombre de theories bloquées.. : 457
Nombre de theories en attente : 18
Nombre de theories pretes:... : 365
- a etre developpees..... : 12
- deja developpees..... : 353
-----
Nombre total de theories..... : 841

*** Phase d'analyse numero..... : 19

Nombre de theories completes. : 1
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 1
Nombre de theories bloquées.. : 476
Nombre de theories en attente : 22
Nombre de theories pretes:... : 390
- a etre developpees..... : 24
- deja developpees..... : 366
-----
Nombre total de theories..... : 889

```

```

*** Phase d'analyse numero..... : 20

Nombre de theories completes. : 1
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 1
Nombre de theories bloquee.. : 509
Nombre de theories en attente : 32
Nombre de theories pretes:... : 426
- a etre developpees..... : 36
- deja developpees..... : 390
-----
Nombre total de theories..... : 968

*** Phase d'analyse numero..... : 21

Nombre de theories completes. : 1
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 1
Nombre de theories bloquee.. : 557
Nombre de theories en attente : 19
Nombre de theories pretes:... : 477
- a etre developpees..... : 51
- deja developpees..... : 426
-----
Nombre total de theories..... : 1054

*** Phase d'analyse numero..... : 22

Nombre de theories completes. : 2
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 2
Nombre de theories bloquee.. : 607
Nombre de theories en attente : 24
Nombre de theories pretes:... : 522
- a etre developpees..... : 46
- deja developpees..... : 476
-----
Nombre total de theories..... : 1155

```

2.4 La chronologie des événements

Ici encore, pour des raisons de place, nous ne présenterons pas tout le fichier trace produit mais uniquement les événements qui ont eu lieu durant les trois premières phases de l'analyse.

```

*** Phase d'analyse n.: 1

Creation theorie n.: 0
Traitement theorie n.: 0
Creation theorie n.: 1
Creation theorie n.: 2
Creation theorie n.: 3
Creation theorie n.: 4
Creation theorie n.: 5
Creation theorie n.: 6
Emission hypothese theorie n.: 6
Creation theorie n.: 7

*** Phase d'analyse n.: 2

Validation theorie n.: 6
Creation theorie n.: 8
Traitement theorie n.: 8
Creation theorie n.: 9
Traitement theorie n.: 5
Creation theorie n.: 10
Traitement theorie n.: 4
Creation theorie n.: 11
Creation theorie n.: 12
Creation theorie n.: 13
Traitement theorie n.: 7
Creation theorie n.: 14
Emission hypothese theorie n.: 14
Creation theorie n.: 15
Emission hypothese theorie n.: 15
Creation theorie n.: 16
Creation theorie n.: 17

```

Creation theorie n.: 18
 Creation theorie n.: 19
 Emission hypothese theorie n.: 19
 Traitement theorie n.: 3
 Creation theorie n.: 20
 Creation theorie n.: 21
 Emission hypothese theorie n.: 21
 Traitement theorie n.: 2
 Creation theorie n.: 22
 Creation theorie n.: 23
 Emission hypothese theorie n.: 23
 Validation theorie n.: 23
 Creation theorie n.: 24
 Creation theorie n.: 25
 Emission hypothese theorie n.: 25
 Creation theorie n.: 26
 Traitement theorie n.: 1
 Creation theorie n.: 27
 Creation theorie n.: 28

*** Phase d'analyse n.: 3

Rejet theorie n.: 14
 Rejet theorie n.: 15
 Rejet theorie n.: 21
 Rejet theorie n.: 19
 Rejet theorie n.: 25
 Traitement theorie n.: 24
 Creation theorie n.: 29
 Traitement theorie n.: 12
 Creation theorie n.: 30
 Traitement theorie n.: 9
 Creation theorie n.: 31
 Emission hypothese theorie n.: 31
 Creation theorie n.: 32
 Emission hypothese theorie n.: 32
 Traitement theorie n.: 13
 Creation theorie n.: 33
 Creation theorie n.: 34
 Emission hypothese theorie n.: 34
 Traitement theorie n.: 11
 Creation theorie n.: 35
 Emission hypothese theorie n.: 35
 Creation theorie n.: 36
 Creation theorie n.: 37
 Emission hypothese theorie n.: 37
 Validation theorie n.: 37
 Creation theorie n.: 38
 Traitement theorie n.: 10
 Creation theorie n.: 39
 Emission hypothese theorie n.: 39
 Creation theorie n.: 40
 Emission hypothese theorie n.: 40
 Traitement theorie n.: 27
 Creation theorie n.: 41
 Emission hypothese theorie n.: 41
 Creation theorie n.: 42
 Creation theorie n.: 43
 Emission hypothese theorie n.: 43
 Creation theorie n.: 44
 Emission hypothese theorie n.: 44
 Traitement theorie n.: 26
 Creation theorie n.: 45
 Emission hypothese theorie n.: 45
 Creation theorie n.: 46
 Emission hypothese theorie n.: 46
 Traitement theorie n.: 22
 Creation theorie n.: 47
 Traitement theorie n.: 18
 Creation theorie n.: 48
 Emission hypothese theorie n.: 48
 Creation theorie n.: 49
 Emission hypothese theorie n.: 49
 Traitement theorie n.: 28
 Creation theorie n.: 50
 Traitement theorie n.: 20
 Creation theorie n.: 51
 Traitement theorie n.: 17
 Creation theorie n.: 52
 Emission hypothese theorie n.: 52
 Creation theorie n.: 53
 Traitement theorie n.: 16
 Creation theorie n.: 54

...

3 Un autre exemple d'analyse

3.1 Les conditions de l'analyse

Dans cet exemple, la composante SYNSEM a analysé le même énoncé : « est-ce que vous avez une documentation ». L'analyse a été effectuée en appliquant une stratégie de sélection des n meilleures d'abord avec n variable et où l'intervalle acceptable a été fixé à 20. La stratégie de propagation consistait à affaiblir le score de l'arbre syntaxique lors de l'ajout d'un nœud non terminal en pondérant le score de l'arbre par le nombre de nœuds, à ne pas affaiblir le score de l'arbre lors de l'ajout d'un nœud terminal et à propager le score des mots validés en pondérant le score de l'arbre par le nombre de nœuds correspondant à des mots validés. L'analyse a eu recours à toutes les facilités, une trace chronologique des événements et des statistiques sur l'évolution de l'espace des théories ont été produites, l'espace des théories a été sauvegardé après chaque phase d'analyse mais cette fois-ci, l'espace des théories a été nettoyé.

3.2 L'espace des théories construit

Comme l'espace des théories a été nettoyé automatiquement à chaque phase d'analyse, nous n'avons pas eu de problèmes pour présenter l'espace des théories complet à la fin de l'analyse de l'énoncé. Cependant, les théories intermédiaires étant régulièrement effacées, il n'est plus possible de retracer jusqu'à la théorie initiale le développement d'une théorie complète.

Strategies

```
debut de l'annonce..... : 0.000000
fin de l'annonce..... : 100.000000
score minimum..... : 0.000000
score maximum..... : 100.000000
intervalle acceptable..... : 20.000000
selection maximum..... : 2147483647
affaiblissement NDSNTERM..... : oui
propagation NDSNTERM..... : en fonction des noeuds
affaiblissement NDSTERM..... : non
propagation NDSREAL..... : en fonction des noeuds NDSREAL
```

Facilites

```
fichier contenant le rnp..... :
/usr/users3/rfia/dialogue/mousel/OUTILS/ASYN/EXEMPLE2/dem_rens.rnp
fichier contenant la cgr..... :
/usr/users3/rfia/dialogue/mousel/OUTILS/ASYN/EXEMPLE2/dem_rens.cgr
fichier contenant la trace..... :
/usr/users3/rfia/dialogue/mousel/OUTILS/ASYN/EXEMPLE2/demo.trace
fichier contenant les statistiques..... :
/usr/users3/rfia/dialogue/mousel/OUTILS/ASYN/EXEMPLE2/demo.stat
fichier contenant les sauvetages automatiques :
/usr/users3/rfia/dialogue/mousel/OUTILS/ASYN/EXEMPLE2/demo.the
numero dernière phase d'analyse..... : 24
dernier numero de theorie..... : 589
frequence des sauvetages..... : 1
nettoyage automatique de l'espace..... : oui
```

Contenu de l'espace des theories

Theorie numero: 1

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 50.000000
 Mere1 de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 50.000000

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 nterm.: PREST_INTER num: - sem: - deb: 0.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 2

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 50.000000
 Mere1 de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 50.000000

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 nterm.: INDIR num: - sem: - deb: 0.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 3

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 50.000000
 Mere1 de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 50.000000

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 nterm.: GR_THEM num: - sem: - deb: 0.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 6

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie en attente
 Score de la theorie : 100.000000
 Mere1 de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 100.000000

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 7
=====

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 50.000000
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 50.000000

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
nterm.: PREST_1 num: - sem: - deb: 0.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 18
=====

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 50.000000
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 50.000000

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GS num: - sem: - deb: 0.00 fin: 100.00 fils:
nterm.: GN num: - sem: - deb: 0.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 22
=====

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie en attente
Score de la theorie : 66.666664
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 66.666664

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 32

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 53.333332
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 53.333332

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GN_INTER num: - sem: - deb: 0.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 44

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie en attente
 Score de la theorie : 72.916672
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 72.916672

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 73

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 53.571430
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 53.571430

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: PROP num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GS num: - sem: - deb: 0.00 fin: 100.00 fils:
 nterm.: GN num: - sem: - deb: 0.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 78

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 53.571430
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 53.571430

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: PROP num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_COMP num: - sem: - deb: 0.00 fin: 100.00 fils:
 nterm.: DATE num: - sem: - deb: 0.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 80

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie en attente
 Score de la theorie : 78.598488
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 78.598488

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechl num: - sem: - deb: 27.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 90

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 54.687500
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 54.687500

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: PROP num: - sem: - deb: 0.00 fin: 100.00 fils:

```

term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_COMP num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GN num: - sem: - deb: 0.00 fin: 100.00 fils:

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 102

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie complete
Score de la theorie : 85.732323
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 85.732323

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 128

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie en attente
Score de la theorie : 76.707863
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 76.707863

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
nterm.: DATE num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: det num: - sem: - deb: 45.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 136
=====

Type de la théorie : Theorie synt-sem. descendante
Etat de la théorie : Theorie en attente
Score de la théorie : 76.206512
Merel de la théorie : aucune
Mere2 de la théorie : aucune
Filles de la théorie: aucune

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 76.206512

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 100.00

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 142
=====

Type de la théorie : Theorie synt-sem. descendante
Etat de la théorie : Theorie prete
Score de la théorie : 60.576931
Merel de la théorie : aucune
Mere2 de la théorie : aucune
Filles de la théorie: aucune

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 60.576931

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GR_COMP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GN num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: CHIF num: - sem: - deb: 18.00 fin: 100.00 fils:

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 143
=====

Type de la théorie : Theorie synt-sem. descendante

Etat de la theorie : Theorie prete
 Score de la theorie : 60.576931
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 60.576931

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GR_COMP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GN num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GR_ADJ num: - sem: - deb: 18.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 147
 =====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 59.413586
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 59.413586

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GR_COMP num: - sem: - deb: 18.00 fin: 100.00 fils:
 nterm.: DATE num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: DIZAIN num: - sem: - deb: 18.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 149
 =====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 59.413586
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 59.413586

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00

```

nterm.: GS num: - sem: - deb: 18.00 fin: 100.00 fils:
  term.: GN num: - sem: - deb: 18.00 fin: 100.00 fils:
    term.: vide num: - sem: - deb: 18.00 fin: 18.00
  term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: CHIF num: - sem: - deb: 18.00 fin: 100.00 fils:

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 150

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 59.413586
Mere1 de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 59.413586

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 0.00 fin: 0.00
  nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
    term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
    real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
  nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
    term.: vide num: - sem: - deb: 18.00 fin: 18.00
  nterm.: GS num: - sem: - deb: 18.00 fin: 100.00 fils:
    nterm.: GN num: - sem: - deb: 18.00 fin: 100.00 fils:
      term.: vide num: - sem: - deb: 18.00 fin: 18.00
    term.: vide num: - sem: - deb: 18.00 fin: 18.00
  nterm.: GR_ADJ num: - sem: - deb: 18.00 fin: 100.00 fils:

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 160

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie en attente
Score de la theorie : 82.154884
Mere1 de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 82.154884

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 0.00 fin: 0.00
  nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
    term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
    real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
  nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
    term.: vide num: - sem: - deb: 18.00 fin: 18.00
  nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
    term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
    real.: vous num: - sem: - deb: 18.00 fin: 27.00
  nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
    term.: vide num: - sem: - deb: 27.00 fin: 27.00
    term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
    real.: avez num: - sem: - deb: 27.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
    nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
      term.: vide num: - sem: - deb: 45.00 fin: 45.00
    term.: det num: - sem: - deb: 45.00 fin: 56.00
    real.: une num: - sem: - deb: 45.00 fin: 56.00
  term.: nom num: - sem: - deb: 56.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 167

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie en attente
 Score de la theorie : 76.442444
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 76.442444

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 178

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 59.829063
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 59.829063

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GR_COMP num: - sem: - deb: 18.00 fin: 100.00 fils:
 nterm.: DATE num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: MILLIER num: - sem: - deb: 18.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 190

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie en attente
 Score de la theorie : 82.331833

Mere1 de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 82.331833

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechl num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00
 term.: nom num: - sem: - deb: 56.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 202
 =====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie en attente
 Score de la theorie : 76.920944
 Mere1 de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 76.920944

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechl num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 236
 =====

Type de la theorie : Theorie synt-sem. descendante

Etat de la theorie : Theorie en attente
 Score de la theorie : 82.690704
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 82.690704

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00
 term.: nom num: - sem: - deb: 56.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 316

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie complete
 Score de la theorie : 82.530899
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 82.530899

nterm.: QUEST num: - sem: - deb: 0.00 fin: 56.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 56.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 56.00 fils:
 nterm.: DATE num: - sem: - deb: 45.00 fin: 56.00 fils:
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00
 term.: vide num: - sem: - deb: 56.00 fin: 56.00
 term.: vide num: - sem: - deb: 56.00 fin: 56.00
 term.: vide num: - sem: - deb: 56.00 fin: 56.00
 term.: vide num: - sem: - deb: 56.00 fin: 56.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 395

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie complete
 Score de la theorie : 85,865463
 Mere1 de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 85.865463

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00
 term.: nom num: - sem: - deb: 56.00 fin: 100.00
 real.: documentation num: - sem: - deb: 56.00 fin: 100.00
 term.: vide num: - sem: - deb: 100.00 fin: 100.00
 term.: vide num: - sem: - deb: 100.00 fin: 100.00
 term.: vide num: - sem: - deb: 100.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 397

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie complete
 Score de la theorie : 85,723907
 Mere1 de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 85.723907

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00

```

term.: nom num: - sem: - deb: 56.00 fin: 100.00
      real.: documentation num: - sem: - deb: 56.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 493

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 86.152565
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 86.152565

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
      real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
      real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
      real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 56.00
      real.: une num: - sem: - deb: 45.00 fin: 56.00
term.: nom num: - sem: - deb: 56.00 fin: 100.00
      real.: documentation num: - sem: - deb: 56.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 496

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 82.839005
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 82.839005

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
      real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00

```

```

    real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
  term.: vide num: - sem: - deb: 27.00 fin: 27.00
  term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
    real.: avez num: - sem: - deb: 27.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: det num: - sem: - deb: 45.00 fin: 56.00
    real.: une num: - sem: - deb: 45.00 fin: 56.00
  term.: nom num: - sem: - deb: 56.00 fin: 100.00
    real.: documentation num: - sem: - deb: 56.00 fin: 100.00
nterm.: GR_ADJ num: - sem: - deb: 100.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 100.00 fin: 100.00
  term.: vide num: - sem: - deb: 100.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 498

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 82.839005
Mere1 de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 82.839005

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
  term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
    real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
  term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
    real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
  term.: vide num: - sem: - deb: 27.00 fin: 27.00
  term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
    real.: avez num: - sem: - deb: 27.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: det num: - sem: - deb: 45.00 fin: 56.00
    real.: une num: - sem: - deb: 45.00 fin: 56.00
  term.: nom num: - sem: - deb: 56.00 fin: 100.00
    real.: documentation num: - sem: - deb: 56.00 fin: 100.00
nterm.: GR_ADJ num: - sem: - deb: 100.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 100.00 fin: 100.00
  term.: vide num: - sem: - deb: 100.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 501

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 82.685265
Mere1 de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

 Score de l'arbre syntaxique: 82.685265

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00
 term.: nom num: - sem: - deb: 56.00 fin: 100.00
 real.: documentation num: - sem: - deb: 56.00 fin: 100.00
 term.: vide num: - sem: - deb: 100.00 fin: 100.00
 term.: vide num: - sem: - deb: 100.00 fin: 100.00
 nterm.: GR_COMP num: - sem: - deb: 100.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 100.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 503
 =====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 79.732216
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 79.732216

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00
 term.: nom num: - sem: - deb: 56.00 fin: 100.00
 real.: documentation num: - sem: - deb: 56.00 fin: 100.00
 term.: vide num: - sem: - deb: 100.00 fin: 100.00
 term.: vide num: - sem: - deb: 100.00 fin: 100.00
 nterm.: GR_COMP num: - sem: - deb: 100.00 fin: 100.00 fils:
 nterm.: DATE num: - sem: - deb: 100.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 505

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 82.548950
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 82.548950

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00
 term.: nom num: - sem: - deb: 56.00 fin: 100.00
 real.: documentation num: - sem: - deb: 56.00 fin: 100.00
 term.: vide num: - sem: - deb: 100.00 fin: 100.00
 term.: vide num: - sem: - deb: 100.00 fin: 100.00
 term.: vide num: - sem: - deb: 100.00 fin: 100.00
 nterm.: GR_COMP num: - sem: - deb: 100.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 100.00 fin: 100.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 507

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 79.600777
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 79.600777

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 56.00

```

real.: une num: - sem: - deb: 45.00 fin: 56.00
term.: nom num: - sem: - deb: 56.00 fin: 100.00
real.: documentation num: - sem: - deb: 56.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
nterm.: GR_COMP num: - sem: - deb: 100.00 fin: 100.00 fils:
nterm.: DATE num: - sem: - deb: 100.00 fin: 100.00 fils:

```

Arbre sémantique de la théorie:

Indetermine

Théorie numero: 509

=====

```

Type de la théorie : Théorie synt-sem. descendante
Etat de la théorie : Théorie prête
Score de la théorie : 82.430847
Mère1 de la théorie : aucune
Mère2 de la théorie : aucune
Filles de la théorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 82.430847

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
term.: nom num: - sem: - deb: 56.00 fin: 100.00
real.: documentation num: - sem: - deb: 56.00 fin: 100.00
nterm.: GR_ADJ num: - sem: - deb: 100.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indetermine

Théorie numero: 511

=====

```

Type de la théorie : Théorie synt-sem. descendante
Etat de la théorie : Théorie prête
Score de la théorie : 82.430847
Mère1 de la théorie : aucune
Mère2 de la théorie : aucune
Filles de la théorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 82.430847

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00

```

```

nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
      real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
      real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 56.00
      real.: une num: - sem: - deb: 45.00 fin: 56.00
term.: nom num: - sem: - deb: 56.00 fin: 100.00
      real.: documentation num: - sem: - deb: 56.00 fin: 100.00
nterm.: GR_ADJ num: - sem: - deb: 100.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00
term.: vide num: - sem: - deb: 100.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 517

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 79.245255
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 79.245255

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
      real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
      real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
      real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 56.00
      real.: une num: - sem: - deb: 45.00 fin: 56.00
nterm.: GR_ADJ num: - sem: - deb: 56.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 519

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 79.245255
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 79.245255

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
nterm.: GR_ADJ num: - sem: - deb: 56.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00

```

Arbre semantique de la theorie:

Indetermine

Theorie numero: 520

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 76.310242
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 76.310242

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
nterm.: CHIF num: - sem: - deb: 56.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
nterm.: CENTAINE num: - sem: - deb: 56.00 fin: 100.00 fils:

```

Arbre semantique de la theorie:

Indetermine

Theorie numero: 521

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 79.245255
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 79.245255

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00
 nterm.: CHIF num: - sem: - deb: 56.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 56.00 fin: 56.00
 term.: vide num: - sem: - deb: 56.00 fin: 56.00
 term.: vide num: - sem: - deb: 56.00 fin: 56.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 522

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 76.295227
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 76.295227

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 56.00 fils:
 nterm.: DATE num: - sem: - deb: 45.00 fin: 56.00 fils:
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00
 term.: vide num: - sem: - deb: 56.00 fin: 56.00

```

term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
nterm.: GR_COMP num: - sem: - deb: 56.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 56.00 fin: 56.00
nterm.: GN num: - sem: - deb: 56.00 fin: 100.00 fils:

```

Arbre semantique de la theorie:

Indetermine

Theorie numero: 523

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 75.835442
Mere1 de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 75.835442

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
nterm.: CHIF num: - sem: - deb: 56.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
nterm.: DIZAINE num: - sem: - deb: 56.00 fin: 100.00 fils:

```

Arbre semantique de la theorie:

Indetermine

Theorie numero: 528

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 73.375237
Mere1 de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 73.375237

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:

```

```

term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
  real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: det num: - sem: - deb: 45.00 fin: 56.00
    real.: une num: - sem: - deb: 45.00 fin: 56.00
nterm.: CHIF num: - sem: - deb: 56.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 56.00 fin: 56.00
  nterm.: MILLIER num: - sem: - deb: 56.00 fin: 100.00 fils:
    nterm.: CENTAINE num: - sem: - deb: 56.00 fin: 100.00 fils:

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 529

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 76.197357
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 76.197357

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 0.00 fin: 0.00
  nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
    term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
    real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
  nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
    term.: vide num: - sem: - deb: 18.00 fin: 18.00
    nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
      term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
      real.: vous num: - sem: - deb: 18.00 fin: 27.00
    nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
      term.: vide num: - sem: - deb: 27.00 fin: 27.00
      term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
        real.: avez num: - sem: - deb: 27.00 fin: 45.00
      term.: vide num: - sem: - deb: 45.00 fin: 45.00
      term.: vide num: - sem: - deb: 45.00 fin: 45.00
      term.: vide num: - sem: - deb: 45.00 fin: 45.00
    nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
      term.: vide num: - sem: - deb: 45.00 fin: 45.00
      nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
        term.: vide num: - sem: - deb: 45.00 fin: 45.00
        term.: det num: - sem: - deb: 45.00 fin: 56.00
          real.: une num: - sem: - deb: 45.00 fin: 56.00
        nterm.: CHIF num: - sem: - deb: 56.00 fin: 100.00 fils:
          term.: vide num: - sem: - deb: 56.00 fin: 56.00
          nterm.: MILLIER num: - sem: - deb: 56.00 fin: 100.00 fils:
            term.: vide num: - sem: - deb: 56.00 fin: 56.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 530

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie en attente
Score de la theorie : 76.182365
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 76.182365

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 56.00 fils:
nterm.: DATE num: - sem: - deb: 45.00 fin: 56.00 fils:
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
nterm.: GR_COMP num: - sem: - deb: 56.00 fin: 100.00 fils:
nterm.: DATE num: - sem: - deb: 56.00 fin: 100.00 fils:
term.: det num: - sem: - deb: 56.00 fin: 100.00

```

Arbre sémantique de la théorie:

Indéterminé

Théorie numéro: 531

=====

```

Type de la théorie : Théorie synt-sem. descendante
Etat de la théorie : Théorie prête
Score de la théorie : 76.182365
Mère1 de la théorie : aucune
Mère2 de la théorie : aucune
Filles de la théorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 76.182365

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 56.00 fils:
nterm.: DATE num: - sem: - deb: 45.00 fin: 56.00 fils:
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
term.: vide num: - sem: - deb: 56.00 fin: 56.00
nterm.: GR_COMP num: - sem: - deb: 56.00 fin: 100.00 fils:
nterm.: DATE num: - sem: - deb: 56.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 56.00 fin: 56.00

```

Arbre sémantique de la théorie:

Indéterminé

Théorie numéro: 535

=====

```

Type de la théorie : Théorie synt-sem. descendante

```

Etat de la theorie : Theorie prete
 Score de la theorie : 75.723259
 Mere1 de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 75.723259

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00
 nterm.: CHIF num: - sem: - deb: 56.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 56.00 fin: 56.00
 term.: vide num: - sem: - deb: 56.00 fin: 56.00
 nterm.: CENTAINE num: - sem: - deb: 56.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 56.00 fin: 56.00

Arbre semantique de la theorie:

Indetermine

Theorie numero: 538
 =====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 75.404396
 Mere1 de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 75.404396

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: det num: - sem: - deb: 45.00 fin: 56.00
 real.: une num: - sem: - deb: 45.00 fin: 56.00
 nterm.: CHIF num: - sem: - deb: 56.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 56.00 fin: 56.00
 term.: vide num: - sem: - deb: 56.00 fin: 56.00
 term.: vide num: - sem: - deb: 56.00 fin: 56.00
 nterm.: DIZAINE num: - sem: - deb: 56.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 56.00 fin: 56.00

Arbre sémantique de la théorie:

Indetermine

Théorie numero: 547
=====

Type de la théorie : Théorie synt-sem. descendante
Etat de la théorie : Théorie prête
Score de la théorie : 73.576553
Mère1 de la théorie : aucune
Mère2 de la théorie : aucune
Filles de la théorie: aucune

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 73.576553

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_ADJ num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00

Arbre sémantique de la théorie:

Indetermine

Théorie numero: 549
=====

Type de la théorie : Théorie synt-sem. descendante
Etat de la théorie : Théorie prête
Score de la théorie : 73.576553
Mère1 de la théorie : aucune
Mère2 de la théorie : aucune
Filles de la théorie: aucune

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 73.576553

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00

```

nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_ADJ num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 550

```

=====
Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 70.746689
Mere1 de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 70.746689

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: CHIF num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: CENTAINE num: - sem: - deb: 45.00 fin: 100.00 fils:

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 551

```

=====
Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 73.576553
Mere1 de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 73.576553

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:

```

```

term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: CHIF num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 554

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 73.149467
Mere1 de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 73.149467

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
nterm.: CHIF num: - sem: - deb: 56.00 fin: 100.00 fils:
nterm.: MILLION num: - sem: - deb: 56.00 fin: 100.00 fils:
nterm.: CENTAINE num: - sem: - deb: 56.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 56.00 fin: 56.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 555

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 70.161339
Mere1 de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 70.161339

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: CHIF num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: DIZAIN num: - sem: - deb: 45.00 fin: 100.00 fils:

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 562

```

=====
Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 72.694328
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la théorie:

Score de l'arbre syntaxique: 72.694328

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 0.00 fin: 0.00
nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 18.00 fin: 18.00
nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: det num: - sem: - deb: 45.00 fin: 56.00
real.: une num: - sem: - deb: 45.00 fin: 56.00
nterm.: CHIF num: - sem: - deb: 56.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 56.00 fin: 56.00
nterm.: MILLIER num: - sem: - deb: 56.00 fin: 100.00 fils:
nterm.: CENTAINE num: - sem: - deb: 56.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 56.00 fin: 56.00

```

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 565

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 67.916817
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 67.916817

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: CHIF num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: MILLIER num: - sem: - deb: 45.00 fin: 100.00 fils:
 nterm.: CENTAINE num: - sem: - deb: 45.00 fin: 100.00 fils:

Arbre semantique de la theorie:

Indetermine

Theorie numero: 566
 =====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 70.633492
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 70.633492

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: CHIF num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: MILLIER num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 570

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 70.049080
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 70.049080

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: CHIF num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: CENTAINE num: - sem: - deb: 45.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 45.00 fin: 45.00

Arbre sémantique de la théorie:

Indetermine

Theorie numero: 573

=====

Type de la theorie : Theorie synt-sem. descendante
 Etat de la theorie : Theorie prete
 Score de la theorie : 69.674522
 Merel de la theorie : aucune
 Mere2 de la theorie : aucune
 Filles de la theorie: aucune

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 69.674522

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 0.00 fin: 0.00
 nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
 term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
 nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
 term.: vide num: - sem: - deb: 18.00 fin: 18.00
 nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
 term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
 real.: vous num: - sem: - deb: 18.00 fin: 27.00
 nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
 term.: vide num: - sem: - deb: 27.00 fin: 27.00
 term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
 real.: avez num: - sem: - deb: 27.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 term.: vide num: - sem: - deb: 45.00 fin: 45.00
 nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
 nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:

```

term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: CHIF num: - sem: - deb: 45.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
  term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: DIZAINE num: - sem: - deb: 45.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 45.00 fin: 45.00

```

Arbre semantique de la theorie:

Indetermine

Theorie numero: 583

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 67.690430
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 67.690430

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 0.00 fin: 0.00
  nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
    term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
    real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
  nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
    term.: vide num: - sem: - deb: 18.00 fin: 18.00
    nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
      term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00
      real.: vous num: - sem: - deb: 18.00 fin: 27.00
    nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
      term.: vide num: - sem: - deb: 27.00 fin: 27.00
      term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
      real.: avez num: - sem: - deb: 27.00 fin: 45.00
      term.: vide num: - sem: - deb: 45.00 fin: 45.00
      term.: vide num: - sem: - deb: 45.00 fin: 45.00
      term.: vide num: - sem: - deb: 45.00 fin: 45.00
    nterm.: GR_COMP num: - sem: - deb: 45.00 fin: 100.00 fils:
      term.: vide num: - sem: - deb: 45.00 fin: 45.00
      nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
        term.: vide num: - sem: - deb: 45.00 fin: 45.00
        term.: vide num: - sem: - deb: 45.00 fin: 45.00
      nterm.: CHIF num: - sem: - deb: 45.00 fin: 100.00 fils:
        nterm.: MILLION num: - sem: - deb: 45.00 fin: 100.00 fils:
          nterm.: CENTAINE num: - sem: - deb: 45.00 fin: 100.00 fils:
            term.: vide num: - sem: - deb: 45.00 fin: 45.00

```

Arbre semantique de la theorie:

Indetermine

Theorie numero: 586

=====

```

Type de la theorie : Theorie synt-sem. descendante
Etat de la theorie : Theorie prete
Score de la theorie : 67.130371
Merel de la theorie : aucune
Mere2 de la theorie : aucune
Filles de la theorie: aucune

```

Arbre syntaxique de la theorie:

Score de l'arbre syntaxique: 67.130371

```

nterm.: QUEST num: - sem: - deb: 0.00 fin: 100.00 fils:
  term.: vide num: - sem: - deb: 0.00 fin: 0.00
  nterm.: GR_INTER num: - sem: - deb: 0.00 fin: 18.00 fils:
    term.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
    real.: est_ce_que num: - sem: - deb: 0.00 fin: 18.00
  nterm.: PROP num: - sem: - deb: 18.00 fin: 100.00 fils:
    term.: vide num: - sem: - deb: 18.00 fin: 18.00
    nterm.: GS num: - sem: - deb: 18.00 fin: 27.00 fils:
      term.: pron_suj num: - sem: - deb: 18.00 fin: 27.00

```

```

real.: vous num: - sem: - deb: 18.00 fin: 27.00
nterm.: GV num: - sem: - deb: 27.00 fin: 45.00 fils:
term.: vide num: - sem: - deb: 27.00 fin: 27.00
term.: v_flechi num: - sem: - deb: 27.00 fin: 45.00
real.: avez num: - sem: - deb: 27.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: SGV num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: GN num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: CHIF num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00
nterm.: MILLIER num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: CENTAINE num: - sem: - deb: 45.00 fin: 100.00 fils:
term.: vide num: - sem: - deb: 45.00 fin: 45.00

```

Arbre sémantique de la théorie:

Indetermine

3.3 L'évolution de l'espace des théories

```

*** Phase d'analyse numero..... : 1
Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 0
Nombre de theories pretes:... : 0
- a etre developpees..... : 0
- deja developpees..... : 0
-----
Nombre total de theories..... : 0

*** Phase d'analyse numero..... : 2
Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 1
Nombre de theories pretes:... : 6
- a etre developpees..... : 6
- deja developpees..... : 0
-----
Nombre total de theories..... : 7

*** Phase d'analyse numero..... : 3
Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 1
Nombre de theories pretes:... : 9
- a etre developpees..... : 9
- deja developpees..... : 0
-----
Nombre total de theories..... : 10

*** Phase d'analyse numero..... : 4
Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 1
Nombre de theories pretes:... : 9
- a etre developpees..... : 9
- deja developpees..... : 0
-----
Nombre total de theories..... : 10

*** Phase d'analyse numero..... : 5
Nombre de theories completes. : 0

```

```

- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 8
Nombre de theories pretes:... : 8
- a etre developpees..... : 8
- deja developpees..... : 0
-----
Nombre total de theories..... : 16

*** Phase d'analyse numero..... : 6

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 2
Nombre de theories pretes:... : 8
- a etre developpees..... : 8
- deja developpees..... : 0
-----
Nombre total de theories..... : 10

*** Phase d'analyse numero..... : 7

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 4
Nombre de theories pretes:... : 11
- a etre developpees..... : 11
- deja developpees..... : 0
-----
Nombre total de theories..... : 15

*** Phase d'analyse numero..... : 8

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 2
Nombre de theories pretes:... : 14
- a etre developpees..... : 14
- deja developpees..... : 0
-----
Nombre total de theories..... : 16

*** Phase d'analyse numero..... : 9

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 7
Nombre de theories pretes:... : 15
- a etre developpees..... : 15
- deja developpees..... : 0
-----
Nombre total de theories..... : 22

*** Phase d'analyse numero..... : 10

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 6
Nombre de theories pretes:... : 15
- a etre developpees..... : 15
- deja developpees..... : 0
-----
Nombre total de theories..... : 21

*** Phase d'analyse numero..... : 11

Nombre de theories completes. : 0
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 0
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 9
Nombre de theories pretes:... : 14
- a etre developpees..... : 14
- deja developpees..... : 0
-----
Nombre total de theories..... : 23

```

```

*** Phase d'analyse numero..... : 12
    Nombre de theories completes. : 0
      - recouvrant tout l'annonce. : 0
      - recouvrant une partie.... : 0
    Nombre de theories bloquées.. : 0
    Nombre de theories en attente : 9
    Nombre de theories pretes:... : 12
      - a etre developpees..... : 12
      - deja developpees..... : 0
    -----
    Nombre total de theories..... : 21

*** Phase d'analyse numero..... : 13
    Nombre de theories completes. : 0
      - recouvrant tout l'annonce. : 0
      - recouvrant une partie.... : 0
    Nombre de theories bloquées.. : 0
    Nombre de theories en attente : 5
    Nombre de theories pretes:... : 13
      - a etre developpees..... : 13
      - deja developpees..... : 0
    -----
    Nombre total de theories..... : 18

*** Phase d'analyse numero..... : 14
    Nombre de theories completes. : 0
      - recouvrant tout l'annonce. : 0
      - recouvrant une partie.... : 0
    Nombre de theories bloquées.. : 0
    Nombre de theories en attente : 5
    Nombre de theories pretes:... : 13
      - a etre developpees..... : 13
      - deja developpees..... : 0
    -----
    Nombre total de theories..... : 18

*** Phase d'analyse numero..... : 15
    Nombre de theories completes. : 0
      - recouvrant tout l'annonce. : 0
      - recouvrant une partie.... : 0
    Nombre de theories bloquées.. : 0
    Nombre de theories en attente : 4
    Nombre de theories pretes:... : 14
      - a etre developpees..... : 14
      - deja developpees..... : 0
    -----
    Nombre total de theories..... : 18

*** Phase d'analyse numero..... : 16
    Nombre de theories completes. : 0
      - recouvrant tout l'annonce. : 0
      - recouvrant une partie.... : 0
    Nombre de theories bloquées.. : 0
    Nombre de theories en attente : 5
    Nombre de theories pretes:... : 19
      - a etre developpees..... : 19
      - deja developpees..... : 0
    -----
    Nombre total de theories..... : 24

*** Phase d'analyse numero..... : 17
    Nombre de theories completes. : 1
      - recouvrant tout l'annonce. : 0
      - recouvrant une partie.... : 1
    Nombre de theories bloquées.. : 0
    Nombre de theories en attente : 13
    Nombre de theories pretes:... : 21
      - a etre developpees..... : 21
      - deja developpees..... : 0
    -----
    Nombre total de theories..... : 35

*** Phase d'analyse numero..... : 18
    Nombre de theories completes. : 1
      - recouvrant tout l'annonce. : 0
      - recouvrant une partie.... : 1
    Nombre de theories bloquées.. : 0
    Nombre de theories en attente : 12
    Nombre de theories pretes:... : 22

```

```

- a etre developpees..... : 22
- deja developpees..... : 0
-----
Nombre total de theories..... : 35

*** Phase d'analyse numero..... : 19

Nombre de theories completes. : 1
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 1
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 16
Nombre de theories pretes:... : 27
- a etre developpees..... : 27
- deja developpees..... : 0
-----
Nombre total de theories..... : 44

*** Phase d'analyse numero..... : 20

Nombre de theories completes. : 1
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 1
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 20
Nombre de theories pretes:... : 39
- a etre developpees..... : 39
- deja developpees..... : 0
-----
Nombre total de theories..... : 60

*** Phase d'analyse numero..... : 21

Nombre de theories completes. : 1
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 1
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 30
Nombre de theories pretes:... : 51
- a etre developpees..... : 51
- deja developpees..... : 0
-----
Nombre total de theories..... : 82

*** Phase d'analyse numero..... : 22

Nombre de theories completes. : 1
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 1
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 17
Nombre de theories pretes:... : 66
- a etre developpees..... : 66
- deja developpees..... : 0
-----
Nombre total de theories..... : 84

*** Phase d'analyse numero..... : 23

Nombre de theories completes. : 2
- recouvrant tout l'annonce. : 0
- recouvrant une partie.... : 2
Nombre de theories bloquées.. : 0
Nombre de theories en attente : 22
Nombre de theories pretes:... : 61
- a etre developpees..... : 61
- deja developpees..... : 0
-----
Nombre total de theories..... : 85

```

3.4 La chronologie des événements

De nouveau, dans le but de limiter l'encombrement en place, nous ne présenterons que les événements des trois premières phases d'analyse.

```
*** Phase d'analyse n.: 1
```

```
Creation theorie n.: 0
```

```

Traitement theorie n.: 0
Creation theorie n.: 1
Creation theorie n.: 2
Creation theorie n.: 3
Creation theorie n.: 4
Creation theorie n.: 5
Creation theorie n.: 6
Emission hypothese theorie n.: 6
Creation theorie n.: 7
Destruction theorie n.: 0

*** Phase d'analyse n.: 2

Validation theorie n.: 6
Creation theorie n.: 8
Traitement theorie n.: 8
Creation theorie n.: 9
Traitement theorie n.: 5
Creation theorie n.: 10
Traitement theorie n.: 4
Creation theorie n.: 11
Creation theorie n.: 12
Creation theorie n.: 13
Destruction theorie n.: 4
Destruction theorie n.: 5
Destruction theorie n.: 8

*** Phase d'analyse n.: 3

Traitement theorie n.: 12
Creation theorie n.: 14
Destruction theorie n.: 12

```

4 Conclusions

En considérant ces deux exemples d'analyse, on peut tout d'abord constater que la notion de théorie telle que nous l'avons définie et mise en œuvre est une notion tout à fait viable. En choisissant, même grossièrement, des stratégies, on peut sensiblement réduire le nombre de théories développées (voir le numéro de la dernière théorie produite dans les deux cas, numéro qui est fourni au début de l'espace des théories), sans parler des gains en place mémoire que permet d'obtenir la facilité de nettoyage automatique de l'espace des théories (on pourra comparer les nombres de théories dans les deux espaces à la fin de l'analyse).

La figure H.2 illustre l'évolution de l'espace des théories complet avec une stratégie totale tandis que la figure H.3 montre cette évolution avec une stratégie des n meilleurs d'abord (n variable), l'intervalle acceptable ayant été fixé à 20.

Les figures H.4 et H.5 présentent ensuite l'évolution du nombre des théories utiles (théories complètes, théories en attente ainsi que théories prêtes mais non encore développées) respectivement pour la stratégie totale et pour la stratégie des n meilleurs d'abord. Ces deux graphiques nécessitent quelques explications. On pourrait penser, si on ne compare que l'évolution du nombre de théories utiles, que le gain est négligeable. Il ne faut cependant pas oublier de tenir compte également du nombre absolu de théories qui ont été développées jusqu'au succès de l'analyse. Ce nombre est connu par le numéro de la dernière théorie créée (1251 pour la stratégie totale, 589 pour la stratégie des n meilleurs d'abord). On peut donc aisément constater que le gain est significatif. De plus, ce gain se traduit par un nombre d'hypothèses mot réduit. Ceci peut être vérifié grâce à la chronologie des événements.

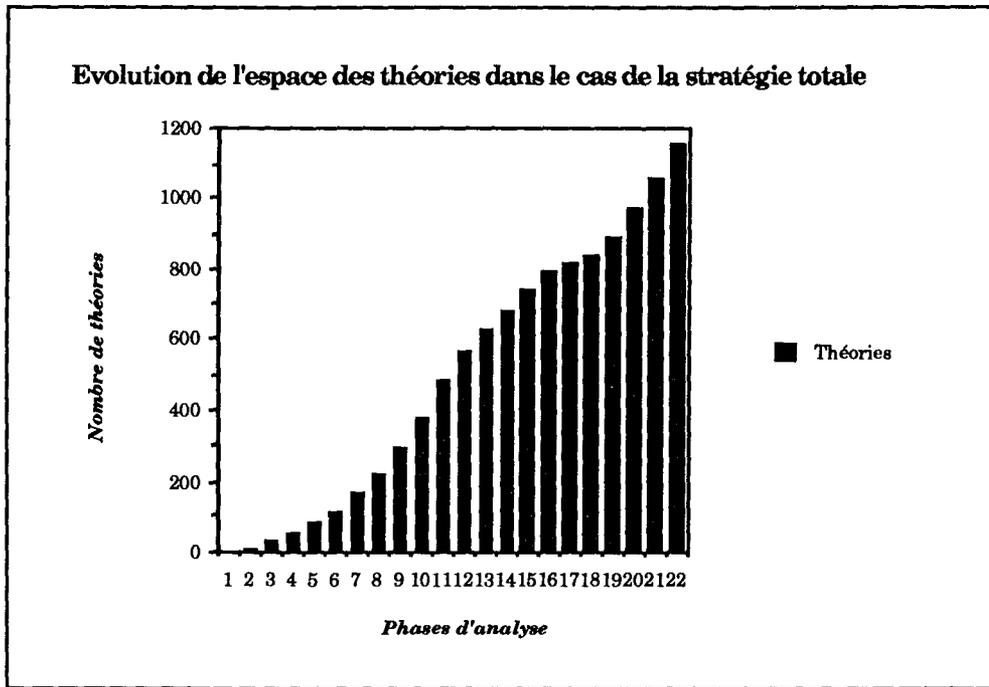


fig. H.2 : Evolution de l'espace des théories dans le cas de la stratégie totale

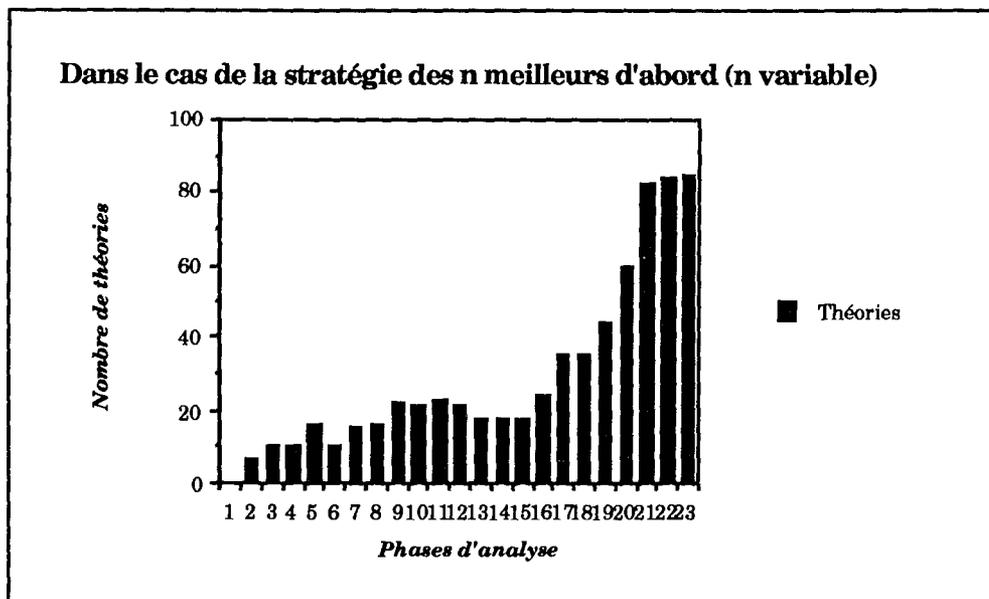


fig. H.3 : Evolution de l'espace des théories dans le cas de la stratégie des n meilleurs d'abord (n variable), l'intervalle étant fixé à 20

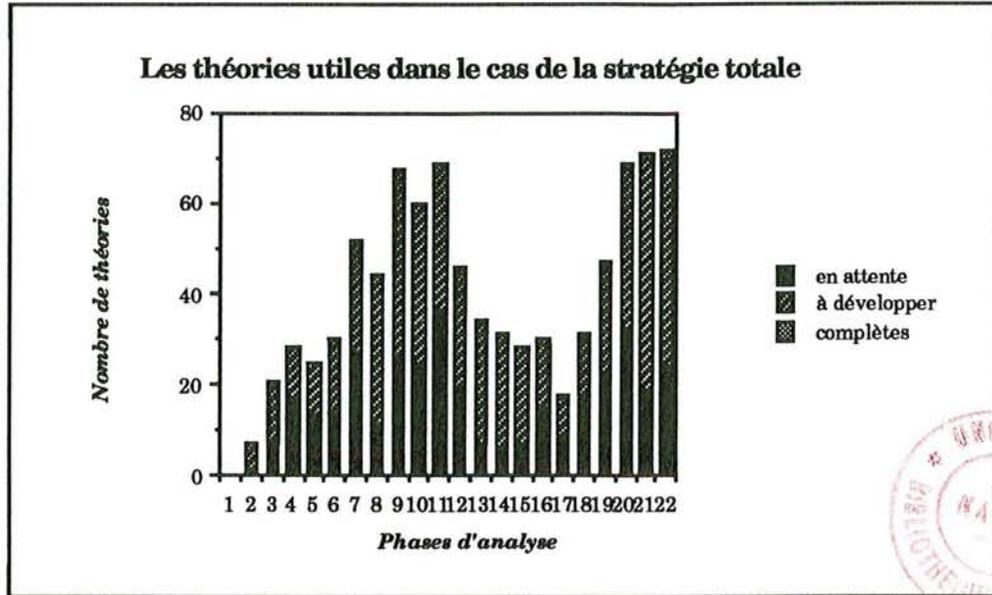


fig. H.4 : Evolution du nombre des théories utiles dans le cas de la stratégie totale

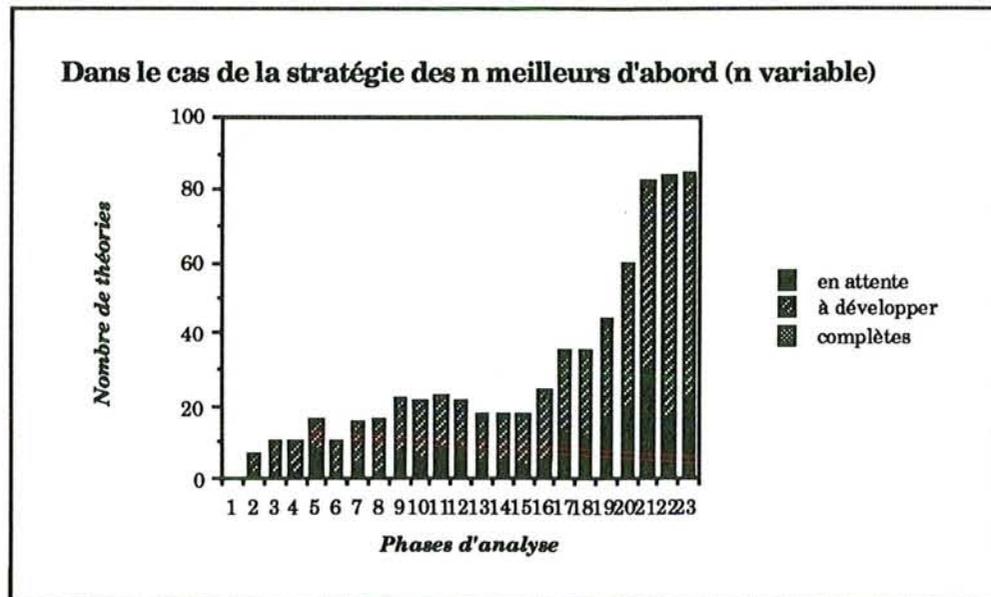


fig. H.5 : Evolution du nombre des théories utiles dans le cas de la stratégie des n meilleurs d'abord (n variable), l'intervalle étant fixé à 20

Ces résultats nous laissent optimistes quant à la validité de notre approche.

NOM DE L'ETUDIANT : MOUSEL Pierre

NATURE DE LA THESE : Doctorat de l'Université de NANCY I en Informatique

VU, APPROUVE ET PERMIS D'IMPRIMER

NANCY, le 23 MARS 1990 n°641

LE PRESIDENT DE L'UNIVERSITE DE NANCY I



M. BOULANGE

Résumé

Cette thèse se propose d'exposer la composante syntaxico-sémantique d'un système de dialogue oral homme-machine finalisé en langage naturel (le système DIAL) actuellement en cours de développement au CRIN. Dans un premier temps, nous essayons de mettre en évidence les liens étroits qui existent entre l'informatique et les sciences de la langue. Une deuxième partie nous permet d'exposer les théories linguistiques desquelles notre travail s'inspire. La troisième partie est alors consacrée à notre réalisation, la composante syntaxico-sémantique (la composante SYNSEM) du système DIAL. La quatrième et dernière partie nous permet de faire le bilan et de présenter les perspectives ouvertes par notre travail.

Mots-clefs

Dialogue homme-machine ; Langage naturel ; Parole ; Syntaxe ; Sémantique