

Université du Québec

Thèse

présentée à

l'Université du Québec à Trois-Rivières

comme exigence partielle

du Doctorat ès arts (Philosophie)

par

Pierre Plante

Maître ès arts (Philosophie)

Le Déredec, logiciel pour le traitement linguistique  
et l'analyse du contenu des textes

Juillet 1979

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

## INTRODUCTION

Le Déredec est un logiciel consacré au traitement linguistique et à l'analyse de contenu des textes. Il s'agit d'un langage de programmation dont la maîtrise n'exige de la part de son usager que très peu de prérequis en informatique. L'un des objectifs visés pour l'édification de ce système incluait qu'un chercheur en sciences humaines (philosophie, linguistique, psychologie...) puisse rapidement mettre au point et appliquer ses propres hypothèses de description de texte (description syntaxique, sémantique, logique...), ses propres idées sur l'exploration des descriptions construites et l'analyse du contenu des résultats, ainsi qu'éventuellement son propre système questions/réponses (le logiciel permet d'associer automatiquement un module questions/réponses à tout ensemble d'hypothèses sur la description des textes).

Ces possibilités de vérifier des hypothèses d'analyse de texte, de les faire varier et de comparer systématiquement et rapidement les résultats exigeraient normalement du chercheur un temps énorme consacré à la programmation des instruments, de même que la constitution d'une équipe interdisciplinaire (incluant des informaticiens professionnels). Il faut noter aussi qu'on découvre souvent trop tard, dans la majorité des expériences de ce type, le caractère "ad hoc" des outils informatiques mis au point pour la vérification d'un groupe d'hypothèses; et donc leur non-transportabilité pour le traitement d'hypothèses inscrites à d'autres niveaux d'analyse.

Ainsi, le Déredec répond à la demande d'un logiciel général pour le traitement sophistiqué des langues naturelles, logiciel dont la souplesse d'écriture est telle, qu'un usager averti peut par ailleurs lui-même le transformer ou l'augmenter de procédures plus particulières.

### L'investigation contextuelle

C'est un lieu commun de souligner aujourd'hui l'insertion rapide et croissante de la technologie informatique dans les sciences humaines. Cette situation rend difficile la taxinomie des disciplines et sous-disciplines développées dans cette traînée. Il semble bien toutefois qu'une distinction entre des utilisations de l'ordinateur à des fins purement gestionnaires et d'autres utilisations où l'instrument permet la vérification d'hypothèses heuristiques puisse servir de première balise.

En sciences humaines, et de façon générale dans les disciplines où le traitement de texte a une place prépondérante, les utilisations de type gestionnaire ont occupé la scène pendant plusieurs années. C'est ainsi par exemple qu'on possède aujourd'hui des "index" et des "concordances" complètes obtenus par ordinateur pour plusieurs ouvrages de philosophie classique.

Pour mieux situer par ailleurs les travaux qui ne se rapportent pas à une simple gestion de la documentation textuelle, on doit différencier les analyses selon qu'elles mettent en jeu des calculs de type statistique ou de type symbolique.

Cette distinction présuppose qu'au moins deux orientations théoriques puissent être associées au concept d'"investigation contextuelle". Dans les recherches liées habituellement au calcul statistique, le contexte d'un mot se définit dans les termes de traits d'édition (par exemple la "ligne" ou la "page") ou de bornes numériques (un certain nombre de mots avant ou après le mot étudié). Par la suite l'analyse opère des regroupements sur les contextes définis puis exécute des opérations de comptage sur les groupes composés. Quelle que soit la sophistication des calculs statistiques effectués sur les regroupements, l'interprétation finale des résultats restera toujours liée à la pauvreté initiale de l'investigation contextuelle.

C'est cette situation qui a motivé les développements plus récents d'analyseurs basés sur la calculation symbolique. On appelle ainsi les investigations contextuelles définies dans les termes d'une composition syntagmatique et d'un jeu de relations de dépendance dont la complexité est arbitraire et dont le contenu peut être associé à la résolu-



tion de différents problèmes linguistiques. Dans cette orientation, le chemin le plus court qui relie deux mots (ou entités textuelles quelconques) traverse plus souvent un parcours particulier d'une arborescence construite sur le segment auquel les mots appartiennent, que la suite linéaire des autres mots du segment qui les séparent.

Alors que pour le calcul statistique des co-occurrences, toutes les relations entre les mots dans une phrase sont aplaties à la relation de co-voisinage (concaténation), le calcul symbolique permet, au moyen d'algorithmes d'indexation de symboles descripteurs, de privilégier certaines relations dans les phrases. Le logiciel Déredec, dont la conception, l'implantation, la mise au point et l'expérimentation furent l'objet de nos recherches, est orienté vers la résolution des problèmes de sémiotique textuelle par l'utilisation spécifique du calcul symbolique.

Il s'agit d'annoncer ici une première démarcation d'avec les systèmes de traitement de texte dont la tâche se résume à l'édition de texte, à la production d'index et de concordances, ainsi qu'à l'analyse des co-occurrences. Le Déredec remplit minimalement certaines de ces tâches, mais avec peu d'efficacité (si on le compare à des logiciels comme JEUEMO (Ouellette 1972), ou SATO (Meunier, Rolland et Daoust 1976)). Il va permettre par ailleurs la programmation d'algorithmes où la notion de contexte peut se définir autrement que par des traits d'édition, ou par un simple décompte opéré sur la suite des mots du texte. En fait, la structure des expressions admissibles dans laquelle l'utilisateur sera amené à définir ses unités d'investigation est suffisamment sophistiquée pour traiter plusieurs problèmes qui caractérisent la simulation des phénomènes de compréhension des langues naturelles: divers types de désambiguation syntaxico-sémantique, les simulations de dialogues et de processus cognitifs, la production de paraphrases... etc.

Par là, le Déredec se rapproche des systèmes développés en intelligence artificielle. Toutefois, et c'est là une seconde démarcation tout aussi importante que la première, le Déredec, contrairement à la majorité de ces travaux de linguistique théorique va rendre possible le traitement d'unités langagières d'un ordre de grandeur élevé, permettant ainsi l'implantation de grammaires textuelles et la mise en relation des programmes de l'utilisateur à des objectifs d'analyse de contenu.

### Formalisme et analyse sémiotique

La sémiotique computationnelle est concernée par la simulation et l'analyse de contenu des divers phénomènes de signification ou de communication. L'étiquette "computationnelle" garantit de prime abord une écriture formelle des modèles qui soit telle que les expériences de simulation ou d'analyse puissent être produites par un ordinateur; nous verrons aussi qu'en plus d'imposer cette contrainte sur le type de formalisme employé, elle laisse soupçonner l'existence de caractéristiques d'écriture plus singulières telles les mécanismes de récursivité, de sensibilisation au contexte... etc.

On sait que les discussions sur l'idéologie, les courants de la philosophie analytique et les diverses sémiotiques modernes ont de part et d'autre été à l'origine de l'élévation des pratiques discursives au rang d'objet privilégié de la recherche philosophique.

L'insertion de l'ordinateur dans ces recherches se justifie premièrement de ce que cette élévation implique prioritairement une prise de possession du donné empirique que constituent les discours, et aussi de ce que l'analyse computationnelle, mettant en relief le caractère reproductible de chacune de ses opérations et de chacun de ses résultats, observe ainsi une condition nécessaire au traitement scientifique des textes.

### Quelques traits formels généraux du Déredec

Le Déredec est un langage de programmation offrant à ses usagers des fonctions dites descriptives de texte (il s'agit de procédures définies par l'utilisateur pour l'assignation aux séquences du texte de structures sémantiques, logiques, syntaxiques ou autres), des fonctions d'exploration de texte (consacrées au dépistage du contenu des descriptions indexées), ainsi qu'un certain nombre de fonctions de régie.

Le texte analysé (tel qu'il se donne à son entrée sur support magnétique), les descriptions de texte obtenues par la programmation des fonctions descriptives, de même que les sous-ensembles de descriptions de texte dépistés par les fonctions exploratrices, sont des entités qui, du point de vue informatique, possèdent la même syntaxe de représenta-

tion.

Cette structure d'information, commune à tous les objets textuels manipulés en Déredec, va permettre la plus haute composition entre les fonctions descriptives et exploratrices du système; les unes étant applicables sur les résultats des autres et vice-versa. Ici se trouve indiquée une première caractéristique formelle du logiciel.

Une seconde est à l'effet que toute EXpression ou suite d'expression de Forme ADmissible à cette syntaxe d'écriture (EXFAD), peut automatiquement devenir l'une des contraintes d'un quelconque patron de fouille, futur argument d'une fonction exploratrice. Nous y reviendrons.

Un troisième trait formel tient de ce qu'au moment de la construction d'une description de texte, la réalisation positive d'une fouille (une exploration) peut être une condition (parmi d'autres) à la poursuite éventuelle de la description. C'est dire ainsi que les fonctions d'exploration peuvent être appelées de l'intérieur des fonctions descriptives et y jouer le rôle de contraintes.

	Entrée:	Sortie:	Programmes:
(FD	EXFAD ----->	EXFAD	Automates )
(FE	EXFAD ----->	EXFAD	Modèles d'exploration )
(FT	EXFAD -----		)

Le tableau représente les relations exprimées par les trois caractéristiques formelles présentées. On y souligne d'une part que les fonctions descriptives (FD) et les fonctions exploratrices (FE) prennent toujours à l'entrée et fournissent toujours à la sortie des EXFAD, c'est-à-dire des objets textuels ayant même syntaxe, d'autre part qu'une fonction de traduction (FT) peut transformer toute EXFAD en une contrainte dans un modèle d'exploration, et enfin, qu'un tel modèle d'exploration peut lui-même agir comme contrainte dans l'exercice d'un automate à états finis qui, comme on le verra, est la forme générale des programmes de description

de texte.

Les relations de contraintes régissent la circulation formelle entre les objets textuels et les programmes (on notera que cette circulation est non-déterministe en ce sens que les nouvelles EXFAD produites à la suite d'une traduction EXFAD-->programmes-->EXFAD ne conservent pas nécessairement la marque de cette traduction).

Cette circularité entre les objets textuels et les programmes, associée à la souplesse de composition entre les fonctions et au pouvoir récursif général du système, permettra l'écriture en Déredec de programmes ayant pour tâche la construction automatique et contextualisée (en utilisant des variables fournies par le contexte) de d'autres programmes associés eux à des tâches d'exploration ou de description de texte.

La programmation automatique sensible au contexte (PASC) est une idée maîtresse du logiciel Déredec et plusieurs fonctions pré-programmées offertes à l'utilisateur y font appel (PRODEC, SIMULECART, LEXIDEC, APLEC...); ce dernier a par ailleurs accès aux fonctions de base de la méthode, et peut ainsi lui-même programmer ses propres procédures PASC.

Une autre idée est aussi, croyons-nous, une pierre angulaire du Déredec; elle a trait au caractère proprement textuel des entreprises de description et d'exploration permises par le système.

Il est évidemment possible d'utiliser le Déredec pour associer à chacune des séquences d'un texte des structures grammaticales de type phrastique, mais il sera aussi possible d'exiger de la description d'un événement textuel qu'elle tienne compte de l'emplacement particulier de cet événement dans l'ensemble du texte, différenciant de plus la partie du texte qui précède l'événement étudié de celle qui le suit. La sensibilité au contexte qui préside aux décisions de description se trouve ainsi fortement augmentée; elle permet ce qu'on appellera la mise au point de grammaires récursives descriptives de texte.

## Les Grammaires Descriptives de Texte

Plusieurs scénarios d'utilisation du Déredec sont imaginables. Nous en avons privilégié un pour les fins de présentation de ce travail, et nous aimerions qu'il loge dans l'esprit du lecteur plutôt à la façon d'un exemple qu'à celle d'une limite.

Ce scénario se laisse structurer en deux actes. Le Déredec pourrait ainsi servir:

a) à la mise au point et à l'application de Grammaires Descriptives de Textes (GDT);

b) à l'exploration ou à l'analyse de contenu d'un texte décrit par une GDT; le logiciel offre entre autres, la possibilité d'associer automatiquement un système questions/réponses à toute GDT soumise par l'utilisateur.

Une GDT est un ensemble de procédures reproductibles, susceptibles de décrire un texte à l'aide d'une batterie de catégories, de relations de dépendance contextuelle et de modèles d'exploration.

Dans cette définition, la mention reproductibilité renvoie à la notion de programmation: les GDT sont des programmes (écrits en Déredec) auxquels on peut soumettre des textes différents; elle renvoie aussi à la notion de contrôle: les expériences de description seront comparables entre elles grâce à une série de paramètres manipulés par le système.

-----

Le chapitre 1 fournit une description générale des différentes fonctions du logiciel (les fonctions descriptives ou automates Déredec, les fonctions d'exploration) ainsi qu'une description générale de la structure des expressions de forme admissible à ces fonctions. On trouvera aussi dans ce chapitre le cadre de validation des différentes expériences de programmation au Déredec. Le chapitre 2 donne pour ces mêmes objets une description détaillée et formelle. Le chapitre 3 présente une Grammaire Descriptive de Texte dont l'objectif est le dépistage de certaines structures de surface des phrases françaises. On y trouvera aussi des exercices d'exploration du contenu des structures textuelles indexées.

Le dernier chapitre présente une extension du logiciel consacrée à des techniques d'exploration de contenu basées sur des principes d'apprentissage et de programmation automatique sensible au contexte.

APLEC (APprenti-LECteur) permet en effet d'associer à toute GDT un système questions/réponses où les questions sont posées dans la langue (naturelle) du texte et où les réponses sont des segments du texte d'abord analysés par la GDT de l'utilisateur puis explorés par une série de modèles également fournis par ce dernier. Lorsqu'APLEC ne peut sur ces bases fournir une réponse adéquate, il dépiste quand même des éléments du texte qui occupent des positions structurales (eu égard à la GDT) analogues à celles de la question puis pourra à la suggestion de l'utilisateur "apprendre" des équivalences sémantiques qui lui permettront de dépister une réponse acceptable. Apprendre signifie ici tenir compte du contexte dans lequel se fait la proposition d'une équivalence, puis parachuter pour tous les segments du texte qui partagent ces contextes la même relation d'équivalence.

Ce système permet de distinguer formellement ce qui dans une entreprise sémiotique relève de la sémantique de la langue, de la sémantique d'un groupe de textes, de la sémantique d'un texte particulier ou encore de celle d'un usage particulier d'un texte donné. Il permet aussi de délimiter les bornes sémantiques d'une GDT donnée et de faciliter par là l'amélioration de cette dernière. On exposera dans ce chapitre deux utilisations différentes d'APLEC, l'une basée sur la grammaire des structures de surface, et l'autre basée sur des principes strictement sémantiques.

Certains lecteurs regretteront de ne pas trouver dans les pages qui suivent un survol général de la problématique qu'a fait sienne la linguistique computationnelle ou encore l'intelligence artificielle durant les dernières décennies. La raison de cet absence est simple; l'entreprise aurait doublé d'excellents travaux déjà publiés. Citons pour exemple le texte de Nick Cercone Representing Natural Language in Extended Semantic Networks (Cercone 1975), dont le premier chapitre est consacré à une revue générale du genre. On pourrait aussi renvoyer à l'ensemble de l'oeuvre de Yorick Wilks, qui, parmi les linguistes computationnels, est l'un des théoriciens les moins avares de mises en situation générales.

Nous avons quant à nous opté d'une part, pour discuter lorsqu'il y avait lieu de références directes aux problèmes posés (on trouvera par exemple une discussion sur la comparaison des grammaires Déredec aux ATN de Woods), et d'autre part pour la construction d'une bibliographie étendue sur l'ensemble des questions soulevées.

La présente version du Déredec a été mise au point dans le cadre d'un doctorat poursuivi au département de philosophie de l'UQTR (Université du Québec à Trois-Rivières). Cette recherche a été subventionnée par des bourses du Conseil des Arts du Canada et de la DGES du Québec (Direction Générale de l'Enseignement Supérieur); elle a de plus été rendue possible grâce à l'assistance matérielle du Service d'Informatique de l'UQAM; nous remercions particulièrement Monsieur le directeur général Richard Lacroix et le responsable des systèmes reliés à l'enseignement et à la recherche Monsieur Hubert Manseau.

Une première version avait été produite dans le cadre d'un projet plus vaste sur l'analyse des textes (J.C. Meunier, F.M. Denis, M. Paquette et F. Daoust; projet ALTP0 subventionné par le Conseil des Arts du Canada). Nous sommes particulièrement reconnaissant à Monsieur F. Daoust (du Service d'Informatique de l'UQAM) pour son assistance technique (programmation) lors de la mise au point de la première version.

Nous aimerions remercier Monsieur Jean Guy Meunier qui a tout au long de cette période suivi notre travail ainsi que Monsieur Claude Panaccio qui s'est joint plus récemment à cette tâche.

Nous sommes de plus reconnaissant à Madame S. Lefaiivre et à Messieurs Y. Wilks, F.M. Denis, J.N. Kaufman et E. Warot pour la lecture attentive qu'ils effectuèrent de notre manuscrit.



## CHAPITRE I

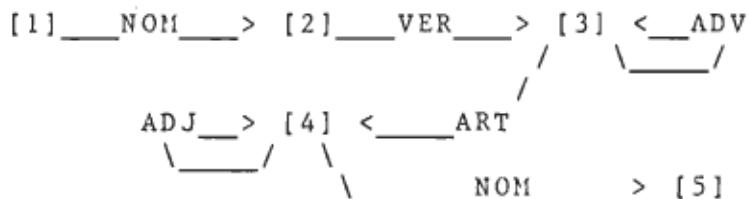
### Description générale du logiciel

#### 1.1 Les automates Déredec

Parmi les logiciels consacrés à la programmation de grammaires ayant vu le jour au cours des dernières quinze années (Mitre 1964, Petrick 1965, Kuno 1965, Earley 1968, Thorne, Bratley et Dewar 1968), les Augmented Transition Networks (ATN) de W.A. Woods (Woods 1970, 1972) ont su remporter la faveur de plusieurs chercheurs. Nous nous attarderons quelque peu à la description de ce système car il a d'une part récupéré les traits intéressants des systèmes qui l'ont précédé et il offre par ailleurs, à cause de sa grande diffusion une borne qualifiée à la comparaison et à la compréhension des automates Déredec.

Les ATN sont d'abord des grammaires à réseaux de transition. Une telle grammaire est un ensemble fini d'états (appelés aussi noeuds) reliés par des arcs sur lesquels se trouvent inscrites des conditions portant sur la composition des éléments de la séquence analysée, éléments dont la lecture est une condition à la transition d'un état à l'autre dans le réseau. On distingue habituellement dans ce modèle l'existence d'un état initial et celle d'un état final.

Ainsi la grammaire à réseaux de transition suivante:



peut rendre compte de phrases du type:

Jean mange.  
Jean mange rapidement.  
Jean mange la pomme.  
Jean mange la grosse pomme.  
Jean mange rapidement la grosse pomme.

Ces grammaires sont connues depuis déjà quelque temps. Elles avaient dans Syntactic Structures (Chomsky 1957) reçu à la fois audience et critique. Elles ont été renforcées dans la plupart des logiciels mentionnés plus haut par l'adjonction de mécanismes récurifs permettant le traitement des constituants emboîtés (pushdown store). Il s'agit de suspendre dans un état donné le travail d'un automate pour en appeler un autre, lui faire exécuter une routine d'analyse puis remettre le contrôle au premier automate à l'endroit dans celui-ci où l'opération d'appel s'est déroulée.

Ces grammaires augmentent le pouvoir récurif du modèle simple de grammaire à réseaux de transition présenté ci-haut, mais conserve un caractère trop déterministe, l'analyse de la phrase étant constituée par l'histoire ou la trace du parcours réel de l'unité de contrôle dans le réseau et les sous-réseaux de transition. Un automate "déterministe" émet comme analyse finale, la suite des catégories ou morphèmes rencontrés sur les arcs parcourus; il reste ainsi nécessairement lié au caractère linéaire de la phrase analysée.

Les ATN de Woods se distinguent de ces "Recursive Transition Networks" par l'adjonction de fonctions supplémentaires permettant à la suite d'une condition réalisée dans un état (cette condition ayant habituellement trait à la catégorie du mot lu sur la bande, mais pouvant par ailleurs être tout à fait arbitraire) d'emmagasiner de l'information dans des registres et cela, sous des symboles arbitrairement choisis. Ces registres peuvent par la suite être modifiés et faire l'objet de tests dans des états parcourus ultérieurement.

Les contenus des différents registres sont, à la fin du parcours de la grammaire et à la suite de la réalisation de certaines conditions, rassemblés dans une structure arborescente qui constitue en quelque sorte l'analyse désirée de la phrase étudiée.

Un trait important qui distingue les ATN des analyseurs précédents est la séparation entre le parcours réalisé effectivement dans l'ensemble des états (de l'automate principal et des sous-automates) et la structure construite par le jeu des manipulations (modifications plus tests) permises sur les registres. C'est cette séparation qui signe le caractère hautement non-déterministe des ATN.

La gamme des actions exécutées éventuellement à la suite d'une condition réalisée dans un état autorise l'implantation de grammaires sensibles au contexte pouvant poursuivre parallèlement à l'analyse de la structure de surface la construction de structures plus profondes, c'est-à-dire la manipulation de règles de transformation. On notera que cette possibilité peut se réaliser sans l'adjonction d'une composante séparée (composante transformationnelle), comme c'est le cas dans les grammaires chomskistes.

Les grammaires Déredec sont des grammaires récursives à réseaux de transition où dans un état donné, à la suite de conditions, différentes actions peuvent être effectuées. Plusieurs actions pré-programmées sont offertes par le logiciel et par ailleurs, toute fonction LISP peut être programmée en position d'action. Cette dernière caractéristique permet évidemment de programmer des actions de rétention d'information dans des registres (arbitrairement nommés) et de construire ainsi parallèlement à l'analyse de la chaîne d'entrée, des structures d'un niveau arbitraire de complexité (ceci à la suite de tests aussi arbitrairement complexes). Les automates Déredec étant par ailleurs récursifs (on peut appeler de l'intérieur d'un automate un ou une série d'automates), on confèrera ainsi aux grammaires Déredec la même puissance d'analyse (non-déterministe) qu'aux grammaires ATN lorsqu'elles sont appliquées à des tâches analogues.

Ce qui donne aux automates Déredec leur singularité et leur puissance originale vient de ce que plusieurs opérations Déredec (on appelle ainsi les actions offertes par le logiciel, programmables à la suite d'une condition satisfaite dans un état) ont trait à des procédures de description (modification physique) de la séquence analysée. La différence majeure entre les automates Déredec et les ATN tient à ce que les premiers sont ainsi orientés vers la construction et la modification d'une structure donnée pour la représentation de l'information textuelle et grammaticale (les EXFAD). On jugera de la puissance de programmation en Déredec par l'examen combiné des qualités structurales de réten-

tion d'information des EXFAD et des diverses opérations ré-  
cursives et sensibles au contexte (en Déredec la sensibilité  
 au contexte est directement reliée à la souplesse d'explora-  
 tion des EXFAD) qui autorisent la construction de ces EXFAD,  
leur modification et leur exploration.

Un premier groupe de ces opérations Déredec permet de catégoriser l'expression pointée, de la relier à d'autres expressions de la séquence par des relations de dépendance contextuelle orientées RDC (arbitrairement nommées), de rassembler différentes expressions d'une même séquence en syntagmes (arbitrairement nommés), un syntagme devenant par la suite une seule expression pouvant elle-même être catégorisée, reliée et composée dans un autre syntagme. Bref, des opérations qui permettent de composer avec un degré arbitraire de complexité des structures arborescentes sur les éléments de la séquence analysée.

Un second groupe d'opérations descriptives a pour fonction d'associer aux éléments initiaux de la séquence d'entrée (ce que nous avons appelé les "expressions atomiques") des réseaux d'information ayant la forme d'arborescences où le noeud initial est une expression atomique appartenant à la séquence analysée et les autres noeuds, d'autres expressions atomiques ayant avec la première (ou entre elles) des relations orientées représentées par des Modèles d'Exploration. Ces réseaux (que nous appelons EXFAL, Expression de Forme Admissible au Lexique) permettent de véhiculer de l'information de type paradigmatique: de l'information portant sur le comportement d'une expression atomique dans d'autres séquences que la séquence pointée, alors que le jeu des syntagmes construits, des catégories et des relations indexées informe plutôt des rapports qu'actualisent entre elles des expressions atomiques d'une même séquence (distinction saussurienne syntagme/paradigme).

Toutes les structures (syntagmatiques et paradigmatiques) indexées aux éléments de la séquence pourront être fouillées par ce que nous avons appelé les Modèles d'Exploration. Ces modèles sont des patrons de fouille et de dépistage (pattern-matching) arbitrairement complexes qui ont pour fonction de rassembler sous des registres (arbitrairement nommés) des éléments terminaux ou non-terminaux (des sections entières d'un arbre) de l'expression pointée dans la séquence. Les contenus ainsi rassemblés ont la même syntaxe de formation que les expressions arborescentes construites sur les séquences; ils peuvent donc faire l'objet des mêmes manipulations (modifications, tests...

etc.). De plus, une fonction du système (la fonction SOIT) permet de transformer automatiquement le contenu d'un registre construit par un modèle d'exploration en sous-partie d'un futur modèle d'exploration, c'est-à-dire de transformer automatiquement (en contexte) les résultats d'une fouille en un argument pour une fouille ultérieure.

La syntaxe des modèles d'exploration est très souple; elle permet de représenter élégamment les caractéristiques structurales des descriptions produites et de dépister de façon très sélective les éléments des structures indexées. On peut dès maintenant noter le double statut de ces modèles d'exploration; comme éléments entrant dans la composition des EXFAL, ils appartiennent aux structures descriptives indexées; comme arguments fournis aux fonctions exploratrices, ils feront partie de l'algorithme d'analyse. Ainsi, pour nous résumer quelque peu, disons que les opérations Déredec ont trait à des procédures de construction, de modification et de dépistage d'éléments structuraux associés aux expressions de la séquence analysée. Ces opérations sont exécutées dans les états des automates à la suite de la réalisation de conditions portant elles-mêmes éventuellement sur la composition de ces structures.

Voyons maintenant de façon un peu plus formelle la constitution des structures construites et le fonctionnement des automates Déredec avant de poursuivre une description un peu plus détaillée des diverses opérations qui y sont programmables.

Nous avons appelé EXFAD toute expression susceptible d'être construite, modifiée, fouillée ou simplement pointée par un automate Déredec, et Description de Texte (DDT) toute suite d'EXFAD.

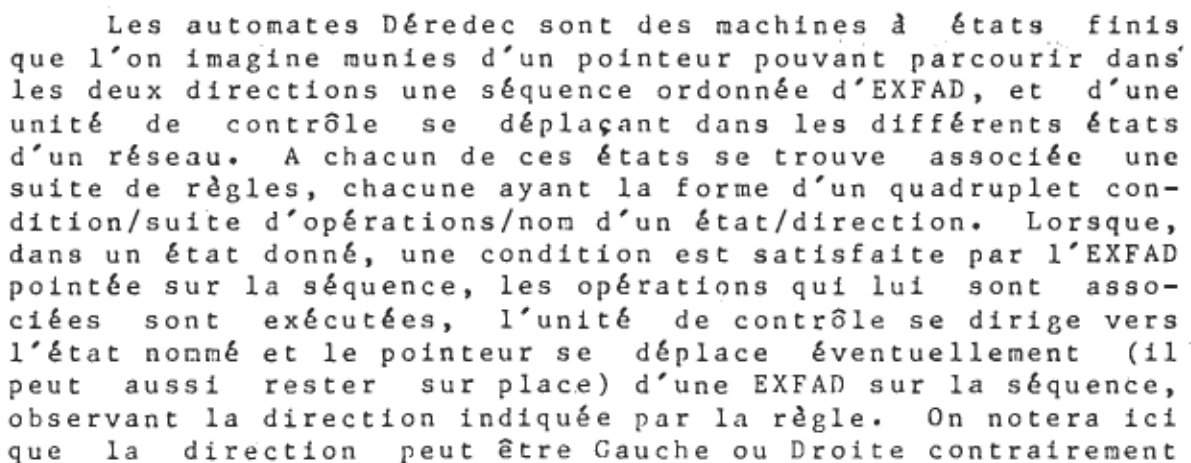
Au niveau le plus primitif, une EXFAD a l'aspect d'une chaîne de caractères placée entre guillemets. Exemple d'une DDT primitive:

"IL" "A" "FAIT" "UNE" "BONNE" "RECETTE" "."

Les contraintes d'entrée de texte sont minimales: seuls les guillemets sont nécessaires, les blancs et retours à la ligne sont facultatifs.

A un niveau plus sophistiqué de description, les EXFAD forment des structures arborescentes où les noeuds des arbres se trouvent indexés de catégories descriptives éventuellement reliées entre elles par des relations de dépendance contextuelle (RDC), et où les éléments terminaux, c'est-à-dire les feuilles des arbres, sont les expressions atomiques du texte auxquelles peuvent aussi s'associer des expressions de forme admissible au lexique (EXFAL). Les expressions atomiques définissent les unités primitives du texte, identifiées comme telles par l'utilisateur; il peut s'agir de morphèmes, de lexèmes, de phrases ou de toute autre entité textuelle.

Le tableau ci-bas représente une EXFAD où les symboles PRON, VER, CONNEC, ART, NOM, SN et SV sont des catégories descriptives, où DET, COMP et TP sont des RDC (relations de détermination, de complémentarité et de thème/propos), et où DETER\* est un modèle d'exploration relatif à la détermination nominale ("cuisine" a quelque part dans une DDT déterminé le nominal "recette"...).



aux ATN qui obligent une analyse prédictive unidirectionnelle Gauche ---> Droite.

De façon générale, dans une règle, les opérations exécutées à la suite de la réalisation d'une condition se rapportent à des procédures:

a) de catégorisation des EXFAD; lorsque l'EXFAD pointée est réduite à une expression atomique, le système permet de projeter la catégorie proposée pour toutes les expressions atomiques de la suite du texte qui ont même composition (pour tous les "tokens" du même "type"); opérations: CAT et PROCAT;

b) de regroupement syntagmatique opéré sur les EXFAD d'une même séquence; opérations: O et F; la procédure permet d'ouvrir ou de fermer des parenthèses arbitrairement à gauche ou à droite de l'expression pointée;

c) de dépistage de RDC entre des EXFAD d'une même séquence; opération R;

d) d'indexation de catégories à la suite de fouilles spéciales (par exemple la combinaison d'une fouille réalisée et d'une autre non réalisée) sur l'EXFAD pointée; opérations FOUDEC1 et FOUDEC2;

e) de modification de la structure environnante des expressions atomiques de l'EXFAD pointée, soit en associant des EXFAD à ces dernières, soit en les catégorisant par la base (c'est-à-dire malgré l'existence de structures syntagmatiques déjà composées); opérations LEXIDEC et TRANEXA;

f) de fouilles arbitrairement complexes sur les EXFAD pointées; opération FOUIL;

g) de suppression d'une EXFAD de la séquence; opération SUPPRIMER;



h) de transformation automatique d'une EXFAD ou d'une suite d'EXFAD (résultat d'une fouille ou d'une description de texte) en un argument pour une fouille à venir; opération SOIT;

i) de transition entre les automates; à cet égard la procédure est très souple, elle permet de suspendre le travail d'un automate et de faire exécuter un ou une série d'automates auxiliaires; on doit noter que les opérations d'ALLER et de RETOUR peuvent s'effectuer dans n'importe quel état d'un automate. Lorsqu'une opération RETOUR est programmée, l'unité de contrôle vient se loger dans l'automate appelant, puis y exécute les opérations restantes. Si l'automate appelant est le premier automate de la série, l'analyse de la séquence est terminée. Ainsi il n'y a pas dans les automates Déredec d'"état final", il n'y a pas non plus d'"état initial" sauf pour le premier état du premier automate;

j) de communication interactive automate/usager; l'utilisateur peut programmer des arrêts de la computation. Lors d'un tel arrêt, la procédure permet de visionner la séquence et l'EXFAD pointées, de connaître l'automate et l'état où se trouve l'unité de contrôle, de modifier des automates ou des registres, et enfin de faire exécuter directement l'une ou l'autre des opérations ou fonctions Déredec; opérations ARRET, ARRET2, VOIRP, VOIRD, ALLER, CONTINUER, SUSPENDRE. Des arrêts non-programmés peuvent aussi survenir, l'automate ne pouvant compléter la description d'une séquence; la machine Déredec fournit alors un diagnostic de l'erreur (aucune condition n'est satisfaite dans un état, l'unité de contrôle boucle indéfiniment... etc.) et une description de son contexte d'apparition.

k) d'analyse contextuelle hors-séquence; opérations PRODEC, PONDEC et ECART; la bande sur laquelle sont inscrites les EXFAD et sur laquelle se déplace le pointeur ne peut être, pour des questions de gestion mémorielle, de longueur indéfinie, et c'est séquence par séquence qu'une Description de Texte sera construite. La limite de ces séquences est déterminée par l'utilisateur selon des critères numériques ou grammaticaux. A ce type séquentiel d'analyse contextuelle s'ajoute une possibilité d'analyse contextuelle élargie sur soit l'ensemble de la description de texte, soit le segment de la description de texte qui précède la séquence pointée (depuis le début de la DDT), soit le segment de la description qui suit la séquence pointée (jusqu'à la fin de la DDT), ou soit encore une description de texte totalement

différente de celle où s'inscrit l'EXFAD pointée (un autre texte).

Les résultats de ces analyses hors-séquence s'indexeront de deux manières à l'EXFAD pointée. D'une première façon en associant une EXFAL à une expression atomique de l'EXFAD; EXFAL composée automatiquement par une exploration du contexte élargi en question (opération LEXIDEC). D'une autre façon, des catégories descriptives seront composées et indexées à l'EXFAD pointée (opérations PRODEC, ECART).

Dans ce dernier cas, lorsque l'analyse porte sur le segment de la description de texte qui suit la séquence courante, l'automate génère une catégorie descriptive de l'ATTENTE qu'exerce l'EXFAD pointée eu égard au phénomène que veut décrire la grammaire de l'utilisateur; et d'autre part, lorsque l'analyse porte sur le segment de la description qui précède la séquence courante, le traitement est associé à la RESOLUTION du phénomène étudié. Ces catégories peuvent être pondérées (opération PONDEC), et de plus, à la suite d'une mesure programmée (opération ECART) de l'écart entre l'ATTENTE et la RESOLUTION qu'exerce pour un phénomène donné une même EXFAD, l'automate indexera cette dernière de l'une des deux catégories PREMISSE ou DEVOUEMENT selon que dans cette EXFAD l'ATTENTE prédomine sur la RESOLUTION (PREMISSE), ou que la RESOLUTION prédomine sur l'ATTENTE (DEVOUEMENT).

Ainsi, de même que les analyses contextuelles séquentielles permettent de programmer des grammaires récursives de type phrastique, les analyses contextuelles élargies vont permettre de programmer des grammaires récursives d'un type nouveau, grammaires hautement sensibles au contexte, qu'il est permis de qualifier de proprement textuelles.

Cette possibilité d'analyser chaque événement du texte, en tenant compte de la structure narrative dans laquelle il se situe, c'est-à-dire en marquant à chaque fois l'ATTENTE spécifique qu'il crée et la RESOLUTION qu'il exerce, offre à l'utilisateur l'occasion d'effectuer sur les descriptions de texte qu'il obtient des analyses de contenu où la cohésion textuelle n'est plus représentée comme une résultante unique, à l'image d'un texte qui ne posséderait qu'un seul point de gravitation, mais comme une courbe où s'exprime la polarisation active et en quelque sorte changeante des forces d'attraction dans le texte.

On rappellera que l'utilisateur peut en fait programmer toute fonction LISP en position d'opération Déredec, ce qui lui permet:

a) tout emmagasinage d'information dans des registres arbitrairement nommés, toute modification ou tout test sur le contenu de ces registres;

b) et règle générale, l'emboîtement des opérations dans des tests arbitrairement complexes.

## Les fonctions Déredec

## 1.2 Les fonctions Déredec

Les fonctions Déredec se distinguent des opérations Déredec en ce qu'elles sont directement programmables au moniteur Déredec, pouvant par ailleurs comme toute fonction être programmées de l'intérieur d'un automate. Ces fonctions concernent:

a) l'application des automates sur des DDT (fonctions: DESCRIP, SIMULECART, USAGER, TRADES, IMPRI);

b) la construction (automatique ou assistée) et la manipulation des lexiques (fonctions: LEXIQUE, TRAFIC, PREPARE);

c) l'exploration des DDT. Les fonctions exploratrices analysent le contenu des informations rassemblées dans les descriptions de texte; elles permettent d'observer la composition des structures dépistées, de construire des EXFAL, c'est-à-dire à la limite d'obtenir un lexique du texte décrit où se trouve associé à chacune des entrées un schéma de son comportement dans la description de texte, et enfin, de construire pour une description de texte donnée, divers profils distributionnels des catégories descriptives indexées. Fonctions exploratrices: PROJEC, LEXIFORME, EXPAF, EXPA, PROFIL.

d) la traduction d'une EXFAD ou d'une DDT en un argument d'exploration (fonction SOIT);

e) des manipulations diverses sur les fichiers utilisés (fonctions: LIRFIC, COPIER, EMBELLIR).

### 1.3 Cadre de validation

La validation des systèmes d'analyse des langues naturelles à l'ordinateur est une démarche qui varie selon que les systèmes implantés ont l'aspect de "simulateurs" ou d'"explorateurs".

On oppose ici des exercices dits de simulation des phénomènes de communication caractéristiques des langues naturelles aux exercices d'exploration ou d'analyse de contenu des productions d'une langue naturelle. Tous les exercices de simulation ont en commun la définition préalable d'un objectif précis, c'est-à-dire une procédure caractéristique d'un procès de communication, la plupart du temps une aptitude qu'ont tous les sujets parlants de la langue naturelle utilisée (traduction d'une séquence, simulation d'un dialogue, production d'une paraphrase...), procédure que devra simuler le robot implanté. La validation d'une entreprise de simulation oblige la comparaison de la performance du robot à celle qu'aurait pour le même procès de communication les locuteurs de la langue (fameux test de Turing (Turing 1950)). Alors que pour ces exercices, l'on connaît à l'avance la solution idéale que devrait fournir l'automate à un problème donné, la validation des explorations de contenu est fondée sur l'idée inverse: on jugera l'algorithme à ses possibilités de dépister des éléments d'information sur le contenu, éléments que des moyens traditionnels d'investigation (par exemple les concordances et les analyses de co-occurrences) n'auraient pu mettre à jour.

Les sémioticiens computationnels se sont longtemps cantonnés dans la mise au point d'algorithmes d'exploration de contenu basés sur des principes de calcul strictement statistique. Ces essais centrés sur des problèmes de détermination stylistique (entre autres les questions d'"authorship") utilisaient les techniques de production d'index, de concordance et d'analyse des co-occurrences.

Alors que les chercheurs préoccupés de "simulations" se sont dès le début emparés des techniques du calcul symbolique, peu d'essais d'explorations de contenu, sinon de très récents (Zarri 1977), ont employé cette voie d'investigation.

On notera que les travaux présentés au chapitre 3 sont à inscrire à cette orientation, alors que ceux du chapitre 4 figurent à la fois sous la rubrique "simulations" et "explorations".

Alors que cette distinction entre les exercices de simulation et ceux d'exploration est relativement bien connue, une autre, tout aussi importante au regard d'une théorie de la validation des sémiotiques computationnelles, semble avoir été beaucoup moins débattue. Dans tous les systèmes faisant usage du calcul symbolique, un certain nombre de manipulations sont effectuées de façon ad hoc, c'est-à-dire manuellement. On fait référence par exemple à l'indexation de catégories grammaticales ou de représentations plus complexes (les formulas de Wilks, les représentations conceptuelles de Schank... etc.), indexations effectuées manuellement à certains éléments du corpus. Les manipulations peuvent s'appliquer sur les "tokens" des éléments étudiés du corpus ou encore sur les "types"; dans ce dernier cas, la procédure utilise un dictionnaire où se trouve corrélés les éléments que l'on pense pouvoir retrouver dans le corpus (par exemple des lexèmes) et des représentations définies dans les termes de la batterie sémantique utilisée. Ces manipulations ad hoc, ces heuristiques en quelque sorte, constituent la conception du monde que doit nécessairement recevoir le "robot" pour qu'il puisse fonctionner.

Nous aimerions souligner la nécessité de prendre en considération le rapport existant dans chaque système entre le nombre de manipulations ad hoc et le nombre total de manipulations exécutées. La mesure de ce rapport est primordiale pour la comparaison et pour l'évaluation des systèmes différents.

Le logiciel Déredec autorise différentes mesures relatives à ce rapport, en émettant pour chaque expérience programmée une série de paramètres communs qui aideront l'utilisateur à comparer et par là à valider ses expériences de description grammaticale. Ces paramètres ont trait:

a) au nombre d'EXFAD lues durant la description;

b) au nombre de règles exécutées;

## Cadre de validation

c) au nombre total de catégories indexées (automatiquement et manuellement);

d) au nombre total de RDC indexées (automatiquement et manuellement);

e) au nombre d'arrêts programmés;

f) au nombre total d'EXFAL indexées (automatiquement et manuellement);

Pour obtenir l'ensemble complet des paramètres pertinents à l'évaluation d'une description, on ajoutera:

g) le nombre de règles programmées dans les automates, information que l'on obtiendra à l'évaluation de la fonction TRADUIRE;

h) le nombre d'entrées aux listes arguments de TRANEXA; ces entrées correspondent à autant d'indexations manuelles de catégories ou d'EXFAL. TRANEXA est la fonction utilisée pour indexer des catégories ou des EXFAL à des expressions atomiques d'un texte; il s'agit en quelque sorte de la fonction dictionnaire du Déredec. On peut considérer que chaque entrée d'un fichier argumentée (si on nous permet ce néologisme) à TRANEXA est similaire à une indexation manuelle comme si cette dernière avait été produite lors d'un arrêt programmé.

Ces paramètres permettent donc plusieurs comparaisons intéressantes. L'utilisateur pourra plus particulièrement relever les rapports suivants:

a) le nombre de règles exécutées par rapport au nombre de règles programmées; une mesure de l'efficacité algorithmique;

b) le nombre d'opérations exécutées de façon ad hoc, c'est-à-dire le nombre d'heuristiques par rapport au nombre d'opérations effectuées automatiquement; ou encore plus spécifiquement:

## Cadre de validation

1) le nombre de catégories indexées manuellement par rapport au nombre total de catégories indexées dans la DDT;

2) le nombre de RDC indexées manuellement par rapport au nombre total de RDC indexées;

3) le nombre d'EXFAL indexées manuellement par rapport au nombre total d'EXFAL indexées.

On obtient le nombre d'opérations manuelles par le décompte du nombre d'arrêts programmés, et/ou par celui du nombre d'entrées aux listes argumentées à TRANEXA.

Le cadre de validation présenté ne garantit pas la pertinence des grammaires implantées en Déredec. On peut toujours imaginer une procédure entièrement automatisée ne produisant que des résultats triviaux; il n'en demeure pas moins que nos propositions de mesure viendront combler un vide que plusieurs ont remarqué dans la littérature sur la question. Il nous semble que l'actuelle prolifération des systèmes de sémiotique computationnelle rend de plus en plus caduque la philosophie du "il suffit que ça fonctionne" qui a jusqu'ici, et à de très rares exceptions près, prévalu pour unique mode de validation.



## CHAPITRE 2

### Description formelle du logiciel

#### 2.1 Le méta-langage BNF

Le Déredec est un logiciel, c'est-à-dire un mini-langage évolué dont les séquences sont programmées par l'utilisateur pour être par la suite traduites en UCI LISP 1.6 (Quam and Diffie 1973). Les séquences LISP peuvent être elles-mêmes traduites (compilées) en un langage plus primitif (le macro-assembleur de la DEC-10); l'utilisateur a ainsi la possibilité de manipuler une version plus rapide (facteur de 20) de ses programmes Déredec.

La difficulté liée à la description des langages formels est bien connue: il s'agit de trouver un mode de présentation qui soit apte à différencier ce qui dans la description appartient au langage (que l'on veut décrire) et ce qui appartient au méta-langage. Suivant en cela la manière la plus courante dont les informaticiens usent pour décrire les langages de programmation, nous nous sommes servis du langage BNF (Backus 1959) pour présenter la syntaxe d'écriture des fonctions, opérations, automates, modèles d'exploration et expressions de forme admissible au Déredec.

Le BNF se construit des symboles suivants:

< >...encadrent les métasymboles du langage décrit;  
:=...associe la définition d'une séquence du langage décrit à un métasymbole;  
/.....indique le "ou" alternatif, la simple concaténation des éléments d'une séquence indiquant par ailleurs le "et";  
et nous rajoutons:  
...les trois points qui précèdent les commentaires plus informels.

## Description du logiciel

Exemple: description BNF de la syntaxe des nombres.

```

<NOMBRE> ::= <ENTIER> / <FRACTION>
<ENTIER> ::= <VIDE> <SUITE DE CHIFFRES> / - <SUITE DE CHIFFRES>
<VIDE> ::=
...par opposition au signe moins (-);
<SUITE DE CHIFFRES> ::= <CHIFFRE> / <CHIFFRE> <SUITE DE CHIFFRES>
<CHIFFRE> ::= 0/1/2/3/4/5/6/7/8/9
<FRACTION> ::= <ENTIER> . <SUITE DE CHIFFRES>

```

Exemples de nombres: 42 -14 2.91 -34.12 00.14

Mises en garde:

1) Afin d'illustrer les possibilités offertes par le logiciel, j'ai puisé la majorité des exemples du présent chapitre à une grammaire descriptive des structures de surface du français moderne; or, on se souviendra que la syntaxe du Déredec reste indifférente à ces distinctions entre la syntaxe, la sémantique ou la logique des langues naturelles.

2) Ce chapitre ne suit pas une méthode programmée d'apprentissage. Nous suggérons une première lecture rapide et complète même s'il s'en dégage une compréhension partielle. Au moment de la mise au point d'un premier projet de programmation, l'utilisateur devra le relire à la manière d'un manuel de référence, c'est-à-dire dans un jeu de renvois multiples et non comme une suite linéaire de commandes indépendantes les unes des autres.

## 2.2 Syntaxe des EXFAD

```

<DESCRIPTION DE TEXTE> ::= <SUITE D'EXFAD>
<SUITE D'EXFAD> ::= <EXFAD> / <EXFAD> <SUITE D'EXFAD>
<EXFAD> ::= <FEUILLE> / <BRANCHE>
<BRANCHE> ::= (<CATEGORIE> <RELATIONS> <BRANCHE>) /
(<CATEGORIE> <RELATIONS> <FEUILLE>)
<RELATIONS> ::= (<SUITE DE RDC>) / NIL
<FEUILLE> ::= <EXPRESSION ATOMIQUE> / <EXFAL>
<EXPRESSION ATOMIQUE> ::= ...toute chaîne de caractères
encadrée de guillemets ("");
<EXFAL> ::= (<EXPRESSION ATOMIQUE> <SUITE DE RESEAUX>)
<SUITE DE RESEAUX> ::= <RESEAU> / <RESEAU> <SUITE DE RE-
SEAUX>
<RESEAU> ::= (<NMOD> <EXFAL>) / <VIDE>
<NMOD> ::= <IDENTIFICATEUR>*
...nom d'un modèle d'exploration; identificateur dont le
dernier caractère doit obligatoirement être un astérisque;
<IDENTIFICATEUR> ::= ...symbole choisi par l'utilisateur;
il s'agit d'une suite de caractères (chiffres et/ou lettres)
dont le premier élément est une lettre, et où sont exclus
les connecteurs (sauf indication contraire) et les blancs.
<VIDE> ::=
<CATEGORIE> ::= <IDENTIFICATEUR>
...nom de la catégorie;
<SUITE DE RDC> ::= <RDC> / <RDC> <SUITE DE RDC>
<RDC> ::= (<NRDC> <ENTIER> <+>) / (<NRDC> <ENTIER> <->)
<NRDC> ::= <IDENTIFICATEUR>
...nom de la RDC;
<ENTIER> ::= ...les nombres entiers indiquent ici la distance
séparant les EXFAD d'une même suite reliées entre elles par
une RDC. Les entiers négatifs (exemple: -4) signifient un
comptage vers le début de la suite et les entiers positifs
(exemple: 4) un comptage vers la fin de la suite;
<+> ::= +
...marque l'EXFAD où la RDC prend son point de départ;
<-> ::= -
...marque l'EXFAD qui reçoit la RDC;

```

## Exemples d'EXFAD:

- A) "JARDIN"
- B) ("JARDIN" (DAREX\* ("OCCUPATIONS")))
- C) (NOM NIL "JARDIN")
- D) (SN NIL (SN ((DET 2 -))
 (ART ((DET 1 +)) "LES")
 (NOM ((DET -1 -)) "FONDEMENTS))
 (PREP NIL "DE")
 (SN ((DET -2 +))
 (ART ((DET 1 +)) "SA")
 (NOM ((DET -1 -)) "METAPHYSIQUE"))))

## Description du logiciel

```

E) (SN NIL (ART ((DET 1 +)) "UNE")
      (SN ((DET -1 -))
            (ADJ ((DET 1 +)) "BONNE")
            (NOM ((DET -1 -))
                  ("RECETTE" (DETER* ("CUISINE"))
                              (DETER* ("TRAVAIL"
                                         (DETER*
                                          ("MAGASIN"))))))))

```

Dans ces exemples, SN, ART, NOM, ADJ, PREP sont des catégories, et DETER\* DAREX\* des noms de modèles d'exploration. DET symbolise une RDC, la relation de détermination.

### 2.3 Syntaxe des modèles d'exploration

```

<MODELE> ::= (<CATEGORIE> <SUITE DE PATRONS>)
<SUITE DE PATRONS> ::= (<PATRON>) <SUITE DE PATRONS> /
<NPATRON> / <VIDE>
<PATRON> ::= <MODELE> / <MODELE> <RELATION> <MODELE> /
(<FEUILLE>)
<RELATION> ::= (<NRDC> +) / (<NRDC> -) / (<NRDC>)
...dans un patron, une relation est toujours située entre
deux modèles. Lorsque la relation renferme un +, on exigera
que le modèle gauche indique le point de départ de la rela-
tion, le signe - indique le sens inverse, et l'absence de
signe signifie l'indifférence quant au sens de la relation.
<NPATRON> ::= :<IDENTIFICATEUR>
...variable dont le contenu est une suite de patrons préa-
lablement liée par l'utilisateur ou par une opération pro-
grammée. Le premier caractère de cet identificateur doit
obligatoirement être un deux-points (:).

```

Remarques sur les modèles d'exploration.

a) Dans un modèle d'exploration, le signe X simule toute catégorie, toute feuille ou toute expression atomique située dans une feuille (le cas d'une expression atomique dans une EXFAL). Par ailleurs, le signe X\* renvoie à tout nom de modèle d'exploration situé dans une EXFAL.

b) Le signe = lorsque constitué comme premier caractère d'une catégorie, ou encore lorsque composé avec le signe X, indique la partie (noeud, feuille ou exp. atom. dans une feuille) de l'EXFAD qui sera dépistée par le modèle.

c) Les contraintes représentées par les patrons dans les suites de patrons peuvent être disjonctives, c'est-à-dire que la réalisation d'un seul patron de la suite suffira à évaluer positivement le modèle, ou conjonctives, tous les patrons devant être réalisés pour que le modèle soit évalué positivement. L'utilisateur indique son choix en donnant une valeur à la variable CONJUNCTION: les contraintes seront conjointes si CONJUNCTION est lié à T, et disjointes si cette variable a la valeur NIL. Exemple d'un modèle susceptible de dépister des EXFAD différentes selon la valeur accordée à CONJUNCTION:



## Description du logiciel

(=X("raison"))..... dépistera toutes les EXFAD (quelle que soit leur catégorie dominante) contenant minimalement l'expression atomique "raison".

(SN((NOM((=X)))))......dépistera toute expression atomique entrée d'une EXFAL elle-même catégorisée NOM puis SN.

(=SN((SN)(D -)(SN)).....dépistera tous les SN contenant un SN déterminé par un autre SN.

(SN((SN)(D -)(SN((=NOM)))))......dépistera tous les NOM situés dans un SN déterminant un autre SN, les deux SN étant situés dans un troisième SN.

(SN((NOM((=X(DETER\*("raison")))))))...... .dépistera toute expression atomique située dans une EXFAL catégorisée NOM et par la suite SN, et ayant dans cette EXFAL une relation DETER\* avec "raison".

(SN((NOM(((X(DETER\*(=X))))))))...... dépistera toute expression atomique reliée par DETER\* à une expression atomique quelconque, et ceci, à l'intérieur d'une EXFAL catégorisée NOM, cette dernière catégorie étant elle-même catégorisée SN.

## 2.4 Syntaxe des automates

<AUTOMATE> ::= (<NAUT> <SUITE D'ETATS>)  
 <NAUT> ::= <IDENTIFICATEUR>  
 ...nom de l'automate;  
 <SUITE D'ETATS> ::= <ETAT> / <ETAT> <SUITE D'ETATS>  
 <ETAT> ::= (<NETAT> <SUITE DE REGLES>)  
 <NETAT> ::= ...nom de l'état; lettre S suivie d'un nombre entier; exemples: S4, S7; Si doit être le nom du premier état;  
 <SUITE DE REGLES> ::= <REGLE> / <REGLE> <SUITE DE REGLES>  
 <REGLE> ::= (<CONDITION> <SUITE D'OPERATIONS> <NETAT> <DIRECTION>)  
 <CONDITION> ::= <CATEGORIE> / <SUITE DE CATEGORIES> / <NMOD> / X / PHI  
 ...la condition sera réalisée si <CATEGORIE>, ou si l'une des catégories de <SUITE DE CATEGORIES> est identique à la catégorie dominante de l'EXFAD pointée, ou si encore un modèle d'exploration, celui lié à NMOD est réalisé sur l'EXFAD pointée. X est une condition par défaut, et PHI renvoie à une catégorie postiche composée au début de chaque séquence. Les catégories descriptives de la grammaire de l'utilisateur doivent être liées à une valeur (voir plus bas la section sur la liaison des variables). Ces valeurs prennent la forme d'arborescences, et lorsqu'une condition est ramenée au dépistage d'une catégorie, cette procédure peut prendre l'aspect d'une fouille plus ou moins complète sur cet arbre. Ainsi par exemple, la catégorie C2, qui a pour valeur la liste (C 2 2), sera dépistée sur une EXFAD catégorisée C25 (C 2 5), puisque ce C25 est avant tout un C2.  
 <SUITE DE CATEGORIES> ::= <CATEGORIE> / <CATEGORIE> <SUITE DE CATEGORIES>  
 <SUITE D'OPERATIONS> ::= <OPERATION> / <OPERATION> <SUITE D'OPERATIONS>  
 <DIRECTION> ::= G / D / <VIDE>  
 ...le pointeur poursuivra sa lecture sur la séquence vers la Gauche si "G", vers la droite si "D", et restera sur place par défaut <VIDE>.

<OPERATION> ::= <VIDE> / <APPEL A UN AUTOMATE> / <RETOUR A UN AUTOMATE> / <SUPPRIMER> / <OUVRIR UNE PARENTHESE> / <FERMER UNE PARENTHESE> / <CAT> / <PROCAT> / <DEPISTER UNE RDC> / <TRANEXA> / <LEXIDEC> / <FOUDEC1> / <FOUDEC2> / <PRODEC> / <PONDEC> / <ECART> / <ARRET> / <VOIRD> / <VOIRP> / <ALLER> / <LIRE> / <ECRIRE> / <CONTINUER> / <RETOUR> / <FOUIL> / <ARRET2> / <APRES> / <AVANT> / <REEMPLACER> / <SUSPENDRE>



## Description du logiciel

...les opérations sont des "fonctions" Déredec ne pouvant être appelées que de l'intérieur d'un automate, à la suite d'une condition réalisée. Par ailleurs, toutes les fonctions Déredec peuvent être programmées en position d'opération (en fait toute fonction LISP peut être programmée en position d'opération; cette conception permet à l'utilisateur avancé de programmer lui-même les opérations Déredec dont il aurait besoin).

<APPEL A UN AUTOMATE> ::= (E <NAUT> <NETAT> <DIRECTION>)

<RETOUR A UN AUTOMATE> ::= (RETURN)

...lorsqu'un appel est logé, l'unité de contrôle se déplace vers l'automate nommé (NAUT), dans l'état nommé (NETAT), s'orientant vers la direction indiquée. Lorsqu'un retour est commandé, l'unité de contrôle retourne à l'automate précédent, celui qui avait logé l'appel, et se positionne immédiatement après l'opération d'appel. La procédure est très souple, elle permet de revenir à un automate passé, de suspendre le travail d'un automate pour en appeler d'autres etc. Lorsqu'un retour est logé au premier automate, une fonction interne compose une autre séquence et lui applique ce premier automate.

<SUPPRIMER> ::= (SUPPRIMER <CATEGORIE>)

...opération dont le résultat est de supprimer une EXFAD de la description de texte, celle qui a pour catégorie dominante l'argument programmé.

<OUVRIR UNE PARENTHÈSE> ::= (O <DIRECTION2> <CATEGORIE>) / (O <DIRECTION2>)

<FERMER UNE PARENTHÈSE> ::= (F <DIRECTION2> <CATEGORIE>) / (F <DIRECTION2>)

<DIRECTION2> ::= G / D

...FERMER ou OUVRIR une parenthèse sont des opérations qui prennent pour premier argument une des deux directions gauche ou droite selon que l'utilisateur désire fermer ou ouvrir une parenthèse à la gauche ou à la droite de l'EXFAD pointée. Le regroupement syntagmatique se conclut lorsqu'un couple de parenthèses se trouve balancé. On peut d'abord ouvrir une parenthèse puis lire vers la droite et fermer à une ou plusieurs EXFAD de l'EXFAD pointée; mais on peut aussi faire la démarche inverse, c'est-à-dire d'abord fermer, puis lire vers la gauche et alors ouvrir une parenthèse. L'une ou l'autre des deux opérations devra être argumentée du nom de la catégorie syntagmatique composée.

## Description du logiciel

<CAT> ::= (CAT <CATEGORIE>)

...cette opération est équivalente à la combinaison des opérations (O G) (F D <CATEGORIE>); elle peut toujours être utilisée pour catégoriser une EXFAD pointée, mais elle est toutefois obligatoire lorsque cette dernière se ramène à une seule feuille.

<PROCAT> ::= (PROCAT <CATEGORIE>)

...PROCAT est aussi utilisé pour catégoriser directement une expression atomique, sauf que la catégorie proposée sera de plus indexée aux expressions atomiques des séquences restantes du texte qui ont même contenu que l'expression atomique pointée. Ces catégories peuvent de plus être projetées sur une DDT différente ou lors d'une séance ultérieure, à la condition que soit sauvée sur disque une liste créée par le système et nommée LISTCA. Le fichier ainsi formé (par exemple LISTCA.DEF) devra être réinitialisé en mémoire au début des séances concernées (voir la section sur les accès disques).

<DEPISTER UNE RDC> ::= (R <NRDC> <CONDITION> <DIRECTION2>)

...la RDC nommée reliera l'EXFAD pointée à la première EXFAD de la séquence (à gauche ou à droite de l'EXFAD pointée selon l'indication fournie en <DIRECTION2>) qui réalise la condition (deuxième argument); la relation origine toujours de l'EXFAD pointée;

<TRANEXA> ::= (TRANEXA <NLISTE>)

<NLISTE> ::= <IDENTIFICATEUR>

...nom donné à une liste construite par l'utilisateur et renfermant soit une <SUITE D'EXFAL>, soit une <SUITE DE COUPLES <CATEGORIES> <EXP. ATOM.>>

<SUITE DE COUPLES

<CATEGORIE> <EXP. ATOM.>> ::= <CATEGORIE> <EXP. ATOM.> /  
<CATEGORIE><EXP.ATOM.><SUITE DE COUPLES <CATEGORIE><EXP. ATOM.>>

TRANEXA est utilisé pour transformer la structure environnante d'une feuille de l'EXFAD pointée et ceci, en lui associant une EXFAL ou une catégorie descriptive.

D'une première façon, TRANEXA permet d'associer une EXFAL à une expression atomique de l'EXFAD, à la condition que cette expression atomique occupe une position d'entrée dans l'une des EXFAL de la liste proposée comme argument; exemple:

## Description du logiciel

```
LISTE: (("SENS" (DETER* ("RAISON")))  
        ("BON" (DAREX* ("CLIENT"))))
```

EXFAD avant le passage de l'automate contenant (TRANEXA LISTE)

```
(SN NIL (ART NIL "LE")  
         (ADJ NIL "BON")  
         (NOM NIL "SENS"))
```

EXFAD après TRANEXA:

```
(SN NIL (ART NIL "LE")  
         (ADJ NIL ("BON" (DAREX* ("CLIENT"))))  
         (NOM NIL ("SENS" (DETER* ("RAISON")))))
```

D'une seconde façon, TRANEXA permet de catégoriser une expression atomique de l'EXFAD pointée; l'opération doit alors être argumentée d'une liste de couples catégorie/expression atomique; si l'une des expressions atomiques de ces couples est présente dans l'EXFAD pointée, alors la catégorie associée à cette expression atomique dans le couple lui sera indexée; exemple:

```
LISTE: ((HUMAIN "HOMME")  
        (VER "ALLER")  
        (ADJ "JOYEUSE"))
```

EXFAD avant le passage de (TRANEXA LISTE):

```
(SN NIL (SN NIL (ART ((D 1 +)) "L")  
              (NOM ((D -1 -)) "HOMME")))
```

EXFAD après TRANEXA:

```
(SN NIL (SN NIL (ART ((D 1 +)) "L")  
              (NOM ((D -1 -)) (HUMAIN NIL "HOMME"))))
```

TRANEXA servira par exemple à lemmatiser les expressions atomiques du texte en associant une EXFAL du type:

```
("SERAIENT" (LEMME* ("ETRE")))
```

Notes:

## Description du logiciel

a) Dans les cas où TRANEXA est argumenté d'une suite d'EXFAL, et qu'une expression atomique de l'EXFAD pointée apparaît en position d'entrée dans l'une de ces EXFAL, et qu'en plus cette expression atomique de l'EXFAD se trouve déjà associée à une EXFAL, alors les deux EXFAL en formeront une nouvelle et cette dernière sera indexée à l'expression atomique en question; exemple:

```
LISTE: (("EAU" (DETER* ("VIE"))
        ("JARDIN" (DETER* ("SALADES"))))
```

```
EXFAD avant (TRANEXA LISTE):
(SN NIL (NOM NIL ("EAU" (DETER* ("SOURCE"))
                        (CAUSE* ("PLANTES")))))
```

```
EXFAD après TRANEXA:
(SN NIL (NOM NIL ("EAU" (DETER* ("VIE"))
                        (DETER* ("SOURCE"))
                        (CAUSE* ("PLANTES")))))
```

b) La liste argumentée à TRANEXA doit subir un traitement spécial de compilation préalablement à l'application de l'automate où se trouve programmé TRANEXA. Voir plus bas la fonction <PREPARE>....dans notre exemple on aurait (PREPARE LISTE);

c) Une autre fonction nommée <TRAFFIC> est associée de près à TRANEXA; elle rend possible la construction semi-automatique de la liste contenant la suite d'EXFAL ou de couples catégories/ expressions atomiques.

```
<LEXIDEC> ::= (LEXIDEC <ARG1> (<SUITE DE NMOD>) <ARG3>
<CHAIN?>)
<ARG1> ::= (<SUITE D'EXP. ATOM.>) / <NMOD>
<SUITE D'EXP. ATOM.> ::= <EXP. ATOM> / <EXP. ATOM.> <SUITE D'EXP. ATOM.>
<SUITE DE NMOD> ::= <NMOD> / <NMOD> <SUITE DE NMOD>
<ARG3> ::= AVANT / APRES / TOT
<CHAIN?> ::= OUI / NON
...LEXIDEC construit des EXFAL et les associe s'il y a lieu
aux expressions atomiques constitutives de l'EXFAD pointée.
Les EXFAL sont composées sur les expressions atomiques fournies soit directement par la suite proposée au premier argument, soit encore à la suite d'une fouille sur l'EXFAD pointée, fouille dont le modèle tient lieu de premier argument.
Le deuxième argument identifie la liste des modèles qui se-
```

## Description du logiciel

ront constitutifs des réseaux des EXFAL construites. Le troisième argument indique le segment de la description de texte où sera réalisée l'exploration; à cet égard, TOT renvoie à l'ensemble de la description de texte, AVANT à la partie de la description de texte qui précède la séquence contenant l'EXFAD pointée, et APRES, à celle qui va de cette séquence jusqu'à la fin de la description de texte. Le dernier argument informe si, dans la construction des EXFAL, les explorations seront réappliquées en chaîne. Il s'agit d'une procédure récursive qui remplace chacune des expressions atomiques dépistées par les modèles en position de contrainte dans ces mêmes modèles afin de dépister de nouvelles expressions atomiques qui elles-mêmes recevront le même traitement et ainsi de suite; la chaîne s'arrête lorsque les explorations ne dépistent plus rien, lorsqu'une expression atomique déjà rencontrée apparaît de nouveau, ou lorsque l'analyse atteint un certain niveau de profondeur dans la construction de l'EXFAL. L'utilisateur décide du niveau de profondeur en donnant à la variable CHAÎNE une valeur exprimée en nombre entier positif;

exemples (texte fictif, comme pour tous les exemples du chapitre):

a)

```
(LEXIDEC ("RAISON" "HOMME" "VIANDE")
      (DAREX* DETER*) AVANT NON)
EXFAD avant le passage de LEXIDEC:
      (SN NIL (NOM NIL "HOMME"))
EXFAD après LEXIDEC:
      (SN NIL (NOM NIL ("HOMME" (DAREX* ("OCCUPATIONS"))
                                (DETER* ("RUE")))))
```

b)

```
(LEXIDEC ("RAISON" "HOMME" "VIANDE")
      (DAREX* DETER*) AVANT OUI)
EXFAD avant LEXIDEC:
      (SN NIL (NOM NIL "HOMME"))
EXFAD après LEXIDEC:
      (SN NIL (NOM NIL
        ("HOMME"
          (DAREX* ("OCCUPATIONS"
            (DAREX* ("INSTANTS"
              (DETER*
                ("REPIT")))))
          (DETER* ("RUE"
            (DAREX* ("BATAILLE"))
            (DETER* ("QUARTIER"
              (DETER*
                ("MONTREAL")))))
```

## Description du logiciel

))))  
 où DETER\* et DAREX\* sont deux modèles d'exploration respectivement liés au contenu suivant (où D symbolise la relation de détermination nominale):

```

DETER*... (SN((SN :PATRON)(D -)(SN((NOM((=X))))))
DAREX*... (SN((SN :PATRON)(D +)(SN((NOM((=X))))))

```

Dans l'exemple b, CHAINE fut lié à 3 avant l'exécution de LEXIDEC.

Les modèles d'exploration utilisés dans LEXIDEC doivent contenir la variable :PATRON et le signe =X. La variable :PATRON sera liée à l'expression atomique que l'on veut décrire, et le signe =X doit se trouver à l'endroit du modèle où l'on désire dépister des expressions atomiques.

Ainsi DETER\*...(SN((SN :PATRON)(D -)(SN((NOM((=X)))))) deviera au moment de son évaluation:  
 (SN((SN("HOMME")))(D -)(SN((NOM((=X)))))).

Dans le cas des exemples présentés plus haut, ce modèle pourra alors dépister l'expression atomique "RUE" s'il est appliqué sur une description de texte contenant une EXFAD comme:

```

(SN NIL (SN ((D 2 -))
              (ART ((D 1 +)) "L")
              (NOM ((D -1 -)) "HOMME")
              (PREP NIL "DE")
              (SN ((D -2 +))
                  (ART ((D 1 +)) "LA")
                  (NOM ((D -1 -)) "RUE"))))

```

<FOUDECI> ::= (FOUDECI <NMOD1> <NMOD2> <CATEGORIE> <NBEXFAD>)

<NBEXFAD> ::= ...nombre entier;

...l'opération FOUDECI indexe une catégorie descriptive à l'EXFAD pointée à la suite d'une fouille sur cette dernière. La catégorie indexée se compose de l'identificateur proposé comme troisième argument auquel s'adjoint un chiffre associé à l'adresse relative de l'EXFAD pointée dans la description de texte. Cette adresse est calculée d'après le nombre d'EXFAD contenues dans l'ensemble de la description de texte. Cette valeur peut être fournie approximativement ou à la suite de l'évaluation d'une fonction du système (IMPRI). FOUDECI indexe la catégorie à la condition que le modèle

## Description du logiciel

d'exploration dont le nom apparaît au premier argument soit réalisé dans l'EXFAD pointée, et que le modèle dont le nom est lié au deuxième argument ne le soit pas. Exemple d'EXFAD indexée par FOUDEC1:

```
(SN43 NIL (NOM NIL ("RAISON" (ETER* ("EVIDENCE")))))
```

...ce SN apparaît au 43% de la description de texte.

```
<FOUDEC2> ::= (FOUDEC2 (<SUITE DE NMOD>) (<SUITE DE CATEGORIES>) <NBEXFAD>)
```

...FOUDEC2 indexe aussi une catégorie descriptive à la suite d'une fouille sur l'EXFAD pointée. Les modèles proposés au premier argument seront successivement testés sur cette EXFAD. A chacun de ces modèles correspond une catégorie de la seconde suite (deuxième argument). Les deux listes doivent être de la même longueur de telle sorte que si un modèle se trouve réalisé dans l'EXFAD pointée, l'identificateur qui lui est vis-a-vis dans la pile de catégories sera composé en première partie de la catégorie indexée par FOUDEC2; la seconde partie identifie l'adresse relative de l'EXFAD. Les modèles sont testés dans l'ordre où ils apparaissent dans la suite proposée par l'utilisateur au premier argument.

Exemple:

```
(FOUDEC2 (ETER* CAUSE* DAREX*) (SN VER ADJ))
```

EXFAD construite par FOUDEC2: (VER56 NIL "ROUPILLAIT")

```
<PRODEC> ::= (PRODEC <NMOD1> <NMOD2> ATTENT) /  
(PRODEC <NMOD1> <NMOD2> RESOLU) /  
(PRODEC <NMOD1> <NMOD2> TOT)
```

...PRODEC construit et indexe des catégories descriptives. Ces catégories ont une valeur numérique qui indique le nombre de fois où un modèle d'exploration constitué à partir d'une fouille particulière de l'EXFAD pointée se réalise dans l'ensemble de la description de texte ou dans l'une des deux parties qui environnent l'EXFAD dans cette description, soit celle qui va du début de la description jusqu'à la séquence de l'EXFAD ou celle qui court de cette dernière jusqu'à la fin de la description de texte.

## Description du logiciel

PRODEC applique d'abord sur l'EXFAD pointée le modèle d'exploration qui lui sert de premier argument; les éléments dépistés par ce modèle sont ensuite liés automatiquement à la variable :PATRON qui elle-même sera composée dans le second modèle fourni comme deuxième argument. Ce dernier modèle est alors projeté sur l'un des trois segments de la description de texte et cela selon l'indication donnée au troisième argument. A cet égard, le symbole TOT renvoie à la totalité de la description, ATTENT au segment qui suit l'EXFAD, et RESOLU à celui qui la précède. Les explorations effectuées sur ces deux dernières parties traitent respectivement de l'ATTENTE et de la RESOLUTION du phénomène grammatical étudié par le jeu des deux modèles programmés.

Une catégorie créée par PRODEC sera composée des trois arguments de cette opération suivis de l'évaluation numérique de la dernière exploration (le nombre de fois où le modèle s'est trouvé réalisé). Exemples:

(PRODEC NOM\* NOMEX\* TOT) pourrait créer (au moment de son application sur une EXFAD d'une DDT...) la catégorie: NOM\*NOMEX\*TOT44..... où NOM\* est lié au modèle: (=NOM), et où NOMEX\* est lié au modèle: (SN :PATRON). Cette catégorie informe que les noms de l'EXFAD pointée font 44 fois partie des SN de l'ensemble de la description de texte.

(PRODEC NOM\* NOMEX\* ATTENT) aurait créé la catégorie: NOM\*NOMEX\*ATTENT26... indiquant que les noms de l'EXFAD pointée se retrouvent 26 fois dans les SN des séquences qui suivent l'apparition de l'EXFAD dans la description de texte.

L'utilisateur peut aussi utiliser un modèle d'exploration vide: NIL\*, et le placer en position de premier argument à PRODEC. La catégorie indexée fournira alors des indications sur l'environnement de l'EXFAD, indications qui ne soient pas liées à sa constitution.

```
<PONDEC> ::= (PONDEC <NCATPON> (<PONDERATEUR> <PONDERE>))
<NCATPON> ::= (<NMOD1> <NMOD2> ATTENT*REL) /
(<NMOD1> <NMOD2> RESOLU*REL) /
(<NMOD1> <NMOD2> TOT*REL)
<PONDERATEUR> ::= ...nom d'une catégorie PRODEC;
```



## Description du logiciel

<PONDERE> ::= ...nom d'une catégorie PRODEC;  
 ...l'opération PONDEC pondère la valeur des catégories textuelles déjà indexées aux EXFAD par PRODEC. Le second argument de PONDEC rassemble deux noms de catégories construites par PRODEC; le premier renvoie à la catégorie dont la valeur servira de pondérateur, et le second a trait à la catégorie que l'on veut pondérer. PONDEC fouille l'EXFAD à la recherche des valeurs numériques de ces deux catégories, exécute le calcul puis construit et indexe une nouvelle catégorie composée des informations rassemblées au premier argument suivies de la valeur calculée. Exemple:

```
(PONDEC (NOM* NOMEX* ATTENT*REL)
        (NIL*NOM*ATTENT NOM*NOMEX*ATTENT))
```

EXFAD catégorisée:

```
(NOM*NOMEX*ATTENT*REL30 NIL
 (NOM*NOMEX*ATTENT14 NIL
  (NILNOM*ATTENT464 NIL
   (SN (ADJ ((D 1 +)) "VERTE")
    (NOM ((D -1 -)) "CAMPAGNE")))))
```

Les valeurs relatives sont signifiées en millième. Dans cet exemple, les noms de l'EXFAD pointée ("campagne") font 14 fois partie des 464 noms du restant du texte, c'est-à-dire des 464 noms qui suivent l'occurrence étudiée de "campagne". Cette fraction représente une ATTENTE relative de 30/1000.

<ECART> ::= (<ECARTSIGNE> <NPCAT> <NCATR> <NBEXFAD>)  
 <ECARTSIGNE> ::= ...nombre entier;  
 <NPCAT> ::= <IDENTIFICATEUR>  
 ...nom d'une partie de la catégorie construite par ECART;  
 <NCATA> ::= ...nom d'une catégorie PRODEC ou PONDEC portant sur l'ATTENTE;  
 <NCATR> ::= ...nom d'une catégorie PRODEC ou PONDEC portant sur la RESOLUTION;  
 ...l'opération ECART calcule l'importance du rapport entre l'ATTENTE et la RESOLUTION d'un phénomène grammatical dans une EXFAD donnée. Le premier argument d'ECART est un nombre entier choisi par l'utilisateur pour indiquer quel rapport sera retenu comme significatif. Les deuxième et troisième arguments identifient les catégories (construites préalablement par PRODEC et éventuellement pondérées par PONDEC) dont on calculera l'écart des valeurs numériques. Deux types de catégories seront construites par ECART; lorsque dans l'EXFAD pointée le rapport ATTENTE/RESOLUTION est égal ou supérieur à l'écartsigne, ECART construit une catégorie PRE-

## Description du logiciel

MISSE; dans le cas inverse (RESOLUTION/ATTENTE égal ou supérieur à l'écartsigne) ECART construit une catégorie DE-NOUEMENT. Ainsi, dans une EXFAD donnée, une ATTENTE forte associée à une RESOLUTION faible constitue une PREMISSE du développement du phénomène grammatical (analysé par le jeu des catégories PRODEC/PONDEC indexées), alors qu'une RESOLUTION forte associée à une ATTENTE faible en constitue un DE-NOUEMENT.

La nouvelle catégorie est composée de l'un des symboles PREMIS- ou DENOUE- suivi de l'argument NPCAT et d'un nombre entre 1 et 100, indiquant l'adresse relative de l'EXFAD dans la description. Exemple:

```
(ECART 3 NOMI* NOM*NOMEX*ATTENT*REL
      NOM*NOMEX*RESOLU*REL 80)
```

EXFAD pointée après le passage d'ECART:

```
(DENOUE-NOMI*13 NIL
 (NOM*NOMEX*RESOLU*REL73 NIL
 (NOM*NOMEX*ATTENT*REL30 NIL
 (NOM*NOMEX*RESOLU9 NIL
 (NOM*NOMEX*ATTENT14 NIL
 (NIL*NOM*RESOLU52 NIL
 (NIL*NOM*ATTENT*464 NIL
 (SN (ADJ ((D 1 +)) "VERTE")
 (NOM ((D -1 -)) "CAMPAGNE"))))))))
```

Ici, l'EXFAD pointée constitue pour le phénomène grammatical étudié (la nominalisation) un DENOUEMENT, et cet événement se produit au 13% de la description de texte.

<ARRET> ::= (ARRET)

...cette opération permet de programmer un arrêt de la computation et une communication automate/usager. Au moment de l'arrêt, le Déredéc imprime au terminal la séquence analysée en soulignant l'emplacement du pointeur.

Lors d'un arrêt, l'utilisateur peut faire exécuter directement n'importe quelle opération et/ou fonction Déredéc. A la fin de la communication l'utilisateur remettra le contrôle à l'automate par la commande OK.

Des arrêts non-programmés (attribuables à des erreurs de programmation de l'utilisateur) peuvent survenir durant la computation d'un automate. La machine Déredéc fournit alors un diagnostic de l'erreur, imprime au terminal la séquence, le pointeur ainsi que les noms de l'automate et de l'état où

## Description du logiciel

s'est produit l'arrêt. Diverses opérations sont spécialement utilisées lors des arrêts. Nous en donnons ci-bas la description.

<ARRET2> ::= (ARRET2)

...idem à ARRET sauf que le système n'imprime pas au terminal la séquence pointée.

<VOIRD> ::= (VOIRD POINTEUR)

...cette opération imprime au terminal la séquence courante jusqu'à l'EXFAD pointée.

<VOIRP> ::= (VOIRP POINTEUR)

...celle-ci imprime au terminal la séquence courante de l'EXFAD pointée jusqu'à la fin de la séquence. On peut aussi obtenir l'EXFAD pointée en faisant (CAR POINTEUR).

<ALLER> ::= (ALLER <DIRECTION2>)

...ALLER déplace le pointeur d'une EXFAD sur la séquence, et cela en observant la direction indiquée.

<RETOUR> ::= (RETOUR)

...RETOUR est une procédure de sortie rapide de l'automate. RETOUR écrit la séquence courante sur le fichier de sortie de la description de texte, ferme ce fichier dans l'état où il est et termine ainsi la description.

<SUSPENDRE> ::= (SUSPENDRE)

...fait le même travail que <RETOUR>, mais ferme toutefois le fichier de sortie après y avoir déposé toutes les séquences restantes du fichier d'entrée.

<FOUIL> ::= (FOUIL <NMOD>) / (FOUIL <NMOD> <NEXFAD>)

<NEXFAD> ::= <IDENTIFICATEUR>

...nom d'une EXFAD;

...FOUIL applique sur une EXFAD un modèle d'exploration et dépose le résultat sous la variable EXPLOR. Lorsqu'il n'y a pas de nom d'EXFAD proposé, la fouille s'exécute sur l'EXFAD pointée. Ainsi par exemple (FOUIL MOD\*) serait équivalent à (SETQ COCO (CAR POINTEUR)) (FOUIL MOD\* COCO). FOUIL s'évalue au nombre de fois où le modèle se trouve réalisé.

## Description du logiciel

<AVANT> ::= (AVANT)

...AVANT construit sur un fichier parallèle la partie de la description de texte qui va du début de la description jusqu'à la séquence pointée.

<APRES> ::= (APRES)

...tout comme AVANT, sauf que le fichier construit se rapporte à la section qui suit la séquence courante. Le fichier construit par <AVANT> ou <APRES> se nomme EXPLOR.TMP.

<LIRE> ::= (LIRE)

<ECRIRE> ::= (ECRIRE)

...LIRE est la fonction qui construit la séquence courante, alors que ECRIRE est celle qui imprime cette séquence sur le fichier de sortie de la description de texte. Ces fonctions peuvent être quelquefois utilisées lorsque l'utilisateur est surpris par un arrêt non programmé, ne peut corriger adéquatement ses automates et préfère plutôt sauter la séquence problématique. ECRIRE écrira la séquence courante sur le fichier de sortie, LIRE construira la prochaine séquence et CONTINUER fera poursuivre la computation sur la nouvelle séquence. Note: LIRE et ECRIRE n'ont pas à être programmés en dehors de ces cas exceptionnels d'arrêts.

<CONTINUER> :: (CONTINUER)

...CONTINUER est utilisé de façon générale pour sortir d'un arrêt non programmé (alors que OK sert à sortir d'un arrêt programmé). CONTINUER reconstruit la séquence courante dans l'état où elle était avant l'arrêt, et lui réapplique l'automate défaillant que l'utilisateur aura pris soin auparavant de corriger.

Règle générale, les choses lors d'un arrêt non programmé se produisent de la façon suivante:

a) l'unité de contrôle établit la communication en fournissant un diagnostic, la séquence et l'EXFAD pointées ainsi que les noms de l'automate et de l'état défaillant;

b) l'utilisateur visionne le contexte, fait éditer l'automate problématique, le corrige et refait traduire la liste des automates (voir la fonction TRADUIRE);

## Description du logiciel

c) il redonne alors le contrôle au Déredec par l'opération CONTINUER.

Note: au moment d'un arrêt (programmé ou pas), l'utilisateur peut se remémorer les noms de l'automate et de l'état où se trouve l'unité de contrôle en faisant évaluer la variable: OU.

<REEMPLACER> ::= (REEMPLACER <EXFAD>)

... Cette opération remplace l'EXFAD pointée par l'EXFAD fournie en position d'argument.

## 2.5 Les fonctions Déredec

Règle générale, les fonctions Déredec manipulent des fichiers ou des listes. <NFE> renvoie au nom du fichier d'entrée; <NFS> à celui du fichier de sortie. Lorsque la gestion de la mémoire le permet, l'utilisateur a intérêt à déposer ses descriptions de texte sur des listes plutôt que sur des fichiers. Une liste d'EXFAD est une suite d'EXFAD mise entre parenthèses. Exemple: (<EXFAD><EXFAD><EXFAD>). <NLE> renverra au nom de la liste d'entrée et <NLS> à celui de la liste de sortie.

## 2.6 La fonction LEXIQUE

(LEXIQUE (<NFE>) (<NFS>))

LEXIQUE prend à l'entrée la suite alphabétiquement triée des expressions atomiques constitutives du texte que désire analyser l'utilisateur. Habituellement, ce dernier a d'abord entré son texte à l'aide d'un éditeur de texte, puis en a fait trier une copie. Trier signifie ici que tous les mots du texte sont en ordre alphabétique. La plupart des services informatiques offrent ce type de programme, et nous aurions trouvé redondant de l'inclure dans le Déredec. Sur la DEC-10 l'utilisateur peut appeler le programme SORT (COBOL). Exemple:

```
.R SORT
*TEXTE.TRI=TEXTE.TXT/R40/K1.39
```

...où TEXTE.TRI sera le nom du fichier trié et où TEXTE.TXT est celui du texte brut. Le commutateur R précède la longueur maximale des mots, et K, les caractères sur lesquels se fera le tri.

Note: pour l'utilisation que nous en faisons, SORT oblige à ce que chaque mot (record) du texte d'entrée (par ex.: TEXTE.TXT) soit suivi d'un retour à la ligne (clé RETURN du terminal).

LEXIQUE produira sur le fichier trié le lexique du texte, c'est-à-dire une suite alphabétique des expressions atomiques du texte et de leur fréquence absolue dans ce dernier. LEXIQUE imprime au terminal à la fin de son évalua-

## Les fonctions Dérédec

tion, la somme des occurrences et le nombre total de mots différents. Le lexique construit est déposé sur <NFS>; exemple:

```
(LEXIQUE (TEXTE.TRI) (TEXTE.LEX))
```

## 2.7 La fonction TRAFIC

L'une des premières tâches que l'utilisateur désire accomplir sur son texte concerne habituellement l'indexation de catégories descriptives ou encore celle d'EXFAL aux expressions atomiques.

On imagine par exemple un utilisateur voulant marquer les catégories fonctionnelles ou encore les lemmes des expressions de son texte. Il aura à utiliser pour ce faire l'opération TRANEXA (dans un automate). TRANEXA doit s'argumenter d'un fichier contenant les EXFAL ou les couples catégories/expressions atomiques.

TRAFIC a pour objectif la construction semi-automatique de ce fichier. TRAFIC prend pour fichier d'entrée le lexique du texte:

```
(TRAFIC (<NFE>) (<NFS>)) /  
(TRAFIC (<NFE>) (<NFS>) <NMOD> )
```

Dans le premier cas TRAFIC sert à associer une catégorie à une expression atomique et dans le second cas à construire une EXFAL. Exemple:

```
.RUN DEREDEC  
* (TRAFIC (TEXTE.LEX) (CATEGO.TRA))  
CATEGORISEZ-LES-EXPRESSIONS-SUIVANTES:  
"BLEU"*ADJ  
"BRILLE"*VER  
"CIEL"*NOM ...ETC
```

..après l'astérisque(\*), l'utilisateur donne la catégorie désirée suivie d'un retour de chariot. S'il ne veut pas fournir, il écrit: NON ; si la catégorie à fournir est la même que celle fournie juste avant il écrit: ID ;

```
* (TRAFIC (TEXTE.LEX) (LEMME.TRA) LEMME*)  
VOULEZ-VOUS-EXFALISER-LES-EXPRESSIONS-SUIVANTES:  
"BLEU"*"BLEU"  
"BRILLE"*"BRILLER"
```

## Les fonctions Déredec

```
"CIEL"*"CIEL"
"EST"*"ETRE"      ...ETC
  ..ici aussi l'usager peut utiliser les commandes NON et
ID;
```

On aura eu ainsi:

```
CATEGO.TRA :
(ADJ "BLEU")
(VER "BRILLE")
(NOM "CIEL")
```

```
LEMNES.TRA :
("BLEU" (LEMME* ("BLEU")))
("BRILLE" (LEMME* ("BRILLER")))
("CIEL" (LEMME* ("CIEL")))
("EST" (LEMME* ("ETRE")))
```

## 2.8 La fonction DESCRIP

```
(DESCRIP <NAUT> (<NFE>) (<NFS>) (<NFA>))
(DESCRIP <NAUT> (<NFE>) (<NFS>)) /
(DESCRIP <NAUT> <NLE> <NLS> (<NFA>)) /
(DESCRIP <NAUT> <NLE> <NLS>)
```

..lorsque DESCRIP est argumenté d'une liste d'EXFAD <NLE>, une seule séquence sera construite, celle contenant toutes les EXFAD de cette liste. Attention: la longueur de cette séquence peut être problématique pour la gestion de la mémoire;

<NFA> ::= ...nom d'un fichier auxiliaire devant contenir la description de texte sur laquelle seront exécutées les analyses hors séquences (PRODEC, LEXIDEC, AVANT, APRES...). <NFA> est une copie de <NFE> dans les cas où l'analyse est appliquée sur les segments "avant" ou "après" de la séquence courante; dans les cas où le contexte d'analyse est élargi à l'ensemble d'une description de texte, <NFA> peut bien sur être aussi une copie de <NFE>, mais peut aussi contenir une toute autre description de texte, c'est-à-dire la description d'un texte différent de celui qui renferme l'EXFAD analysée.



## Les fonctions Déredec

Lorsqu'une section de <NFA> est construite pour permettre les analyses "avant" et "après", on retrouvera cette section dans un fichier nommé: EXPLOR.TMP.

Avant l'exécution de DESCRIP, l'utilisateur aura soin d'associer à la variable LIMITE une valeur définie soit dans des termes numériques (valeur associée à un nombre entier), soit dans des termes grammaticaux (liée au nom d'une catégorie). LIMITE fixe la limite des séquences construites.

Exemples:

```
(DESCRIP FD3 (TEXTIN.DES) (TEXTOU.DES) (TEXTIN.DES))
(DESCRIP FD2 (TEXTIN.DES) (TEXTI.2ES) (TEXTI.2ES))
(DESCRIP FD7 (TEXTI7.DES) (TEST7.DES))
(DESCRIP AUTO6 LISEN LISOU )
```

## 2.9 La fonction IMPRI

(IMPRI)

Lorsqu'une description de texte vient d'être construite, l'utilisateur peut être intéressé à connaître différentes statistiques liées au travail de description:

- le nombre d'EXFAD lues;
- le nombre de règles appliquées;
- le nombre de catégories indexées;
- le nombre de RDC indexées;
- le nombre d'EXFAL indexées;
- le nombre d'arrêts programmés effectués;
- les définitions des nouvelles catégories créées par PRODEC, PONDEC etc.... ces définitions se retrouvent sous la variable DEFINITIONS.

IMPRI imprime au terminal ces résultats. Toutefois certaines informations ne seront ramassées que si les automates ont été compilés dans certaines conditions (voir la fonction TRADUIRE).

## Les fonctions Déredec

## 2.10 La fonction EXPAF

```
(EXPAF (<NF>))
```

EXPAF est une fonction d'impression. Elle imprime au terminal toutes et seulement les expressions atomiques des EXFAD déposées sur le fichier argumenté.

## 2.11 La fonction EXPA

```
(EXPA <NEXFAD>)
```

EXPA remplit la même fonction qu'EXPAF mais s'applique sur une seule EXFAD. De l'intérieur d'un automate on pourrait ainsi faire:

```
(SETQ COCO (CAR POINTEUR))  
(EXPA COCO)
```

...afin d'obtenir au terminal l'impression des expressions atomiques constitutives de l'EXFAD pointée.

## 2.12 La fonction TRADES

```
(TRADES (<NFE>) (<NFS>) <NLISTE>)
```

TRADES est une fonction descriptive. Elle équivaut à argumenter DESCRIP d'un automate contenant TRANEXA auquel serait proposée la liste <NLISTE>. <NFE> contient la DDT d'entrée, et <NFS> le fichier sur lequel la DDT ira se déposer une fois ses expressions atomiques décrites. Un paramètre concerne les descriptions produites par TRADES: EXPATOM. Lorsque cette variable a la valeur T, seules les EXFAD de la DDT qui sont réduites à des expressions atomiques seront fouillées par TRANEXA. Lorsqu'elle a la valeur NIL toutes les EXFAD le seront. Ce paramètre permet d'augmenter beaucoup la rapidité d'exécution; il sera principalement utilisé dans les cas de catégorisation de base intensive.

## Les fonctions Déreded

La liste argumentée à TRADES doit auparavant être préparée par la fonction PREPARE (voir plus bas).

TRADES permet une gestion mémorielle beaucoup plus efficace que son équivalent DESCRIP argumenté de l'automate nécessaire. Elle doit lui être préférée dans les cas où <NLISTE> est quelque peu volumineux.

### 2.13 La fonction SIMULECART

L'opération ECART (programmée dans un automate) construit les catégories DEVOUEMENT et PREMISSE, en calculant l'écart entre l'ATTENTE et la RESOLUTION d'un phénomène donné dans une EXFAD donnée. Le premier argument d'ECART indique le rapport minimum devant exister entre l'ATTENTE et la RESOLUTION pour qu'une catégorie soit créée. Plus ce chiffre est élevé, moins il y aura de catégories indexées par ECART (relativement au phénomène grammatical et à la description de texte étudiés). Inversement, plus l'écartsigne est bas, plus la liste des catégories construites sera allongée.

Il existera toutefois toujours pour un phénomène donné, une liste maximale et une liste minimale de ces catégories; à la liste maximale correspond un écartsigne à peine supérieur à 1, alors qu'à la liste minimale correspond un écartsigne dont la valeur serait très élevée (la valeur 1000. lorsque le Déreded est chargé en mémoire).

SIMULECART permet de construire des descriptions de texte où l'écart entre les catégories puisse être maximal, minimal, ou moyen. L'écart moyen correspond à la moyenne de tous les écarts réels observés dans toutes les EXFAD de la description de texte concernée.

```
(SIMULECART <E> <NPCAT> <NCATA> <NCATR>
                    <NBEXFAD> (<NFE>) (<NFS>))
<E> ::= MIN / MAX / MOY
```

...SIMULECART prend donc les mêmes arguments que l'opération ECART sauf pour le premier et les deux derniers. Cette fonction s'applique dans les mêmes conditions que DESCRIP (variables, fichiers d'entrée et de sortie).

## 2.14 La fonction USAGER

USAGER prend quatre arguments. Le premier est une suite d'EXFAD, le second le nom d'un automate, et les troisième et quatrième des noms de fichiers relatifs au fichier de sortie et à un fichier auxiliaire contenant une DDT parallèle. Cette fonction argumente automatiquement DESCRIP de l'automate nommé et l'applique sur la suite d'EXFAD. USAGER est habituellement lié à un automate où a été programmé l'opération SPRINT de telle sorte que l'utilisateur peut définir par ce biais un système questions/réponses. USAGER peut aussi servir plus simplement à la description de très courts segments de texte.

```
(USAGER (<SUITE D'EXFAD>) <NAUT> (<NFS>) (<NFA>))
```

Cette fonction ne nécessite pas la définition d'un fichier d'entrée: <NFE>... puisque la suite d'EXFAD en tient lieu; elle produira toutefois un fichier de sortie: <NFS>; elle n'exige pas non plus que la variable LIMITE soit liée; une seule séquence sera construite: la suite d'EXFAD. L'utilisateur peut par ailleurs utiliser le fichier parallèle: <NFA>, qui dès lors ne pourra servir à l'édification des segments qui environnent la séquence courante, mais pourra occasionner des projections entre la suite d'EXFAD questionnée et l'ensemble d'une description de texte donnée.

L'utilisateur peut, s'il prévoit se servir plusieurs fois de la fonction USAGER au cours d'une même séance, lier la variable QUESFIC à une liste contenant le nom de l'automate et celui du fichier auxiliaire. Cette procédure lui évitera de répéter ces informations; toutefois la description construite ira se déposer sur un fichier de sortie bison (\*.TMP). Exemple:

```
(SETQ QUESFIC @(AUTOQ (TEXTE.AUX)))  
(USAGER ("CECI" "N" "EST" "IL" "PAS" "UNE" "QUESTION" "?"))
```

Lorsque l'utilisateur se sert d'un fichier auxiliaire, il peut profiter d'une caractéristique supplémentaire de USAGER permettant de déposer la description construite sur la séquence d'entrée à la fin du fichier auxiliaire plutôt que sur le fichier de sortie. Il devra pour ce faire lier préalablement la variable MEMO à T (SETQ MEMO T); lorsque le Déredec est chargé en mémoire cette dernière est liée à NIL. La procédure permet d'édifier un système questions/réponses où les questions peuvent être mémorisées et par la suite explorées puisqu'elles sont déposées sur un fichier

accessible de l'intérieur des automates...

## 2.15 La fonction SOIT

SOIT traduit automatiquement une suite d'EXFAD en une suite de patrons. Les suites de patrons sont des éléments des modèles d'exploration.

```
(SOIT <NPATRON> (<NFE>)) / (SOIT <NPATRON> <NLE>)  
<NPATRON> ::= <IDENTIFICATEUR>
```

...variable dont le contenu est une suite de patrons. Le premier caractère de cet identificateur doit obligatoirement être un deux-points (:).

La suite d'EXFAD est inscrite dans un fichier ou une liste. Si la variable SOITRDC est liée à NIL, la traduction opère une réduction sur ces EXFAD: elle élimine toutes les contraintes relatives aux RDC. Supposons, pour donner un exemple, que le fichier TEXT0.EXP contienne les EXFAD:

```
(SN NIL (ART ((D 1 +)) "L")  
          (NOM ((D -1 -)) "HOMME" ))  
(SN NIL (NOM NIL "DOCTEUR" ))
```

```
(SOIT :PATEXP (TEXT0.EXP))  
...associera à :PATEXP l'expression:
```

```
(( (SN((ART(("L"))))  
    ((NOM(("HOMME"))))  
    ((SN((NOM(("DOCTEUR"))))))))
```

Par la suite tout modèle contenant :PATEXP contiendra cette suite de patrons; ainsi un modèle comme (=SP :PATEXP) dépistera tous les syntagmes propositionnels qui contiennent l'un des patrons en question (ou tous, dépendant de la valeur donnée à CONJUNCTION).

Si la variable SOITRDC est par ailleurs liée à T, alors les contraintes relatives aux RDC sont insérées dans la suite de patrons. Lorsque le Déredec est chargé en mémoire, SOITRDC est lié à NIL.

## 2.16 La fonction TRADUIRE

Une fois ses automates définis, l'utilisateur devra les faire traduire du Déredec vers le UCI LISP 1.6. TRADUIRE prend un argument:

```
(TRADUIRE <NLA>)
```

```
<NLA> ::= <IDENTIFICATEUR>
```

```
...nom de la liste des automates. Exemple:
```

```
(TRADUIRE AUTLIS)
```

...où AUTLIS est lié à (AUT1 AUT2 AUT3). TRADUIRE va créer deux fichiers: <NLA>.DEC et <NLA>.LSP correspondant respectivement aux versions Déredec et LISP des automates. Ainsi dans notre exemple. TRADUIRE aura créé AUTLIS.DEC et AUTLIS.LSP. A la fin de son travail, TRADUIRE imprime au terminal le nombre de règles programmées dans l'ensemble des automates.

Lorsque l'usager désire pratiquer une comptabilisation sophistiquée des différentes opérations programmées dans ses automates (le nombre de RDC, de catégories, d'EXFAL... etc. indexées automatiquement), il doit lier préalablement à l'exécution de TRADUIRE la variable MESSAGES à T (variable liée à NIL lorsque le Déredec est chargé en mémoire). Il faut par ailleurs noter que cette procédure augmente la mémoire nécessaire au rangement des automates et qu'elle augmente aussi le temps machine consacré à l'exécution des automates. Les différentes informations compilées sont fournies à l'évaluation de la fonction IMPRI.

## 2.17 La fonction PREPARE

```
(PREPARE <NLISTE>) / (PREPARE (<NF>))
```

Cette fonction prépare la liste argumentée à TRANEXA. Elle prend à l'entrée soit une liste comprenant une suite d'EXFAL, soit une liste constituée d'une suite de couples catégories/exp. atomiques.

Elle peut aussi accepter le nom d'un fichier sur lequel aurait été déposée une telle liste. Fonctions associées: TRAFIC, TRANEXA, TRADES.

## 2.18 La fonction PROJEC

La fonction PROJEC permet d'observer dans les descriptions de texte l'organisation des structures arborescentes. L'utilisateur peut rassembler sur fichier (ou liste) les EXFAD d'une description de texte (ou d'une sous-partie d'une description de texte) qui satisfont un modèle d'exploration donné. PROJEC permet par là de tester rapidement et sur de larges corpus le comportement de telle ou telle structure grammaticale. PROJEC peut aussi être utilisé pour dépister sur une description de texte divers schémas de récurrence affectant le jeu des indications catégorielles, celui des réseaux de RDC, ou encore celui des EXFAL. Ces schémas peuvent s'associer à des expressions atomiques du texte mais aussi à des structures plus abstraites. Ils seront de manière générale exploités à des fins d'analyse de contenu. PROJEC prend trois arguments:

```
(PROJEC (<NFE>) (<NFS>) <NMOD>) /
(PROJEC <NLE> <NLS> <NMOD>) /
(PROJEC (<NFE>) <NLS> <NMOD>)
```

Les explorations obtenues sont des suite d'EXFAD (des descriptions de texte), elles pourront donc être à leur tour manipulées par des fonctions descriptives ou exploratrices. Enfin, il est à noter que PROJEC s'évalue au nombre d'EXFAD rassemblées par le modèle. PROJEC peut aussi s'écrire:

```
(PROJEC (<NFE>) (<NFS>) (QUOTE <MODELE>)) /
(PROJEC <NLE> <NLS> (QUOTE <MODELE>)) /
(PROJEC (<NFE>) <NLS> (QUOTE <MODELE>)))
```

c'est-à-dire que l'on peut remplacer le nom du modèle par le modèle lui-même précédé d'un quote. (QUOTE MODELE) peut aussi s'écrire @MODELE.

## 2.19 La fonction PROFIL

La fonction PROFIL fournit sous la forme d'une liste, les valeurs numériques des catégories descriptives indexées lors d'une description (et cela, quelque soit le niveau des catégories dans les EXFAD).

```
(PROFIL (<NFE>) (<NFS>) <NCAT>)
```

<NFE> ::= ...nom du fichier d'entrée; ce fichier est composé de la liste des définitions des catégories construites par des opérations programmées dans des automates. Cette liste est fournie à l'évaluation des fonctions descriptives (DESCRIP SIMULECART USAGER).

<NFS> ::= ...nom du fichier contenant la liste construite.

<NCAT> ::= ...nom de la catégorie à valeur numérique dont le profil distributionnel est étudié.

On se souviendra que les valeurs associées aux catégories numériques construites par PRODEC où PONDEC sont liées au nombre d'EXFAD rassemblées dans une exploration ou au résultat d'un calcul de pondération; alors que celles indexées par ECART ou FOUDEC sont les adresses relatives (%) des EXFAD catégorisées dans la description de texte. Exemples:

(PROFIL (DEFINI.CAT) (PROFIL.ATT) NOM\*NOMEX\*ATTENT)

produira:

(4 2 1 2 6 5 1 2 1 3 9 6 3 2 2 4 2 8 2 1 1 1 1 4 3 2 3 1 1 8  
1 5 6 1 1 6 1 1 2 3 5 1 2 1 1 2 1 3 1 1 1 ) c'est-à-dire une  
liste des attentes réelles (non-pondérées); alors que:

(PROFIL (DEFIN2.CAT) (PROFIL.DRE) DENOUE-REFE\*)

produira:

(37 41 63 67 70 71 72 74 80 83 90 91 93 94 95 96 97 98 99),  
une liste où chaque valeur indique l'adresse d'une EXFAD catégorisée d'un DENOUEMENT nominal.

## 2.20 La fonction LEXIFORME

LEXIFORME construit des EXFAL. Elle est pour cela argumentée de deux noms de fichier (entrée et sortie), d'une suite d'expressions atomiques pour lesquelles LEXIFORME générera les EXFAL, de la suite des modèles d'exploration qui serviront à dépister les expressions atomiques reliées à celles de la suite argumentée, et enfin d'une indication pour ordonner ou pas la réapplication en chaîne des modèles.

(LEXIFORME (<NFE>) (<NFS>) (<SUITE D'EXP. ATOM.>)  
(<SUITE DE NMOD>) <CHAIN?>)

<CHAIN?> ::= OUI / NON

...ce dernier argument informe si, dans la construction des EXFAL, les explorations seront réappliquées en chaîne. Cette procédure a déjà été décrite lors de la présentation de l'opération LEXIDEC. Avant l'application de LEXIFORME, l'utilisateur doit lier la variable CHAINE qui on le sait établit



la limite de récursion dans la construction en chaîne des EXFAL. Par défaut de liaison de cette variable, la récursion ne s'arrêtera que si une même expression atomique est dépistée une seconde fois, ou si les explorations s'avèrent vides.

## La manipulation des fichiers

Nous présentons ici certaines fonctions permettant la manipulation des fichiers de l'intérieur du Déredec.

### 2.21 La fonction LIRFIC

LIRFIC imprime le contenu d'un fichier au terminal.

(LIRFIC (<NFE>))

### 2.22 La fonction COPIER

(COPIER (<NFI>) (<NF2>))

<NF1> ::= ...nom du fichier dont on veut obtenir une copie;

<NF2> ::= ...nom du fichier copié.

### 2.23 La fonction EMBELLIR

L'embellissement des descriptions de texte est relatif à une réécriture des EXFAD qui met en relief leur structure arborescente.

(EMBELLIR (<NF1>) (<NF2>))

<NF1> ::= ...nom du fichier que l'on désire embellir;

<NF2> ::= ...nom du fichier embelli.

## 2.24 Fonctions LISP couramment utilisées

## La fonction SPRINT

SPRINT est une fonction générale pour l'impression. Elle prend deux arguments. Le premier se rapporte à la variable dont on veut voir s'imprimer le contenu au terminal (SPRINT évalue ses arguments; pour empêcher cette évaluation on doit mettre @ (quote)), et le second indique la nième colonne sur laquelle on désire voir l'impression commencer; exemples:

```
(SPRINT (CAR POINTEUR) 2)
...imprimera l'EXFAD courante. SPRINT comme toute
fonction Déredec ou LISP peut être programmée à l'intérieur
d'un automate.

(SPRINT NIL 3)
(SPRINT @JE-SUIS-DANS-L'ETAT-S4-ET-TOUT-VA-BIEN' 2)
(SPRINT NBEXFAD 1)
```

## La fonction DSKIN

DSKIN charge en mémoire les fichiers utilisés au cours d'une séance Déredec. Ces fichiers concernent les programmes Déredec, les automates, les définitions des catégories, les définitions des modèles... etc.

```
(DSKIN <SUITE DE NF>)
<SUITE DE NF> ::= (<NF>) / (<NF>) <SUITE DE NF>
```

Exemple:

```
(DSKIN (EXPLOR.DEC) (AUTLIS.DEC) (AUTLIS.LSP) (MODELE.DEC))
```

## La fonction DSKOUT

DSKOUT permet d'envoyer sur espace-disque le contenu d'une variable (liste de catégories ou de modèles... etc.) sous la forme d'un fichier.

```
(DSKOUT (<NF>) <NL> )
```

Exemples:

## Les fonctions Déredec

```
(DSKOUT(DEFINI.CAT)DEFINITIONS)
(DSKOUT(MODELE.DEF)MODELES)
(DSKOUT(COCO.DEC)COCO)
(DSKOUT(COCO2.DEC)LISON)
```

## La liaison des variables

## La fonction SETQ

```
(SETQ <NARIABLE> @<VALEUR>)
<NARIABLE> ::= <IDENTIFICATEUR>
...nom de la variable que l'on veut lier;
<VALEUR> ...valeur que l'on veut donner à la variable;
...le signe @ (qui se lit quote) indique que l'élé-
ment qu'il précède doit être lié tel quel; si on omet ce
signe, LISP tentera de lier le nom de la variable à la va-
leur de la valeur proposée; il existe trois exceptions à
cette règle: NIL T et les chiffres qui sont toujours liés à
eux-mêmes;
```

```
(SETQ T T) = (SETQ T @T)
```

La fonction SETQ associe une valeur à une variable. On distingue deux types de variables: les variables particulières de l'utilisateur et les variables générales du système.

Les variables particulières de l'usager se rapportent à:

a) la liste des automates; cette variable sera liée à une suite de valeurs: la suite de chacun des noms des automates; exemple:

```
(SETQ AUTLIS @(AUT1 AUT2 AUT3 AUT4 AUT5))
```

b) le contenu de chacun des automates; exemple:

```
(SETQ AUT4 @(AUT4
  (SI (PHI S2 D)
    (X S1 G))
  (S2 (CONNEC (RETURN))
    ((NOM SH) S3 G))
```

## Les fonctions Déredéc

```

      (X S2 D))
(S3 (ADJ (R D (NOM SN) D) (O G) S4 D)
      (X S5 D))
(S4 ((NOM SN) (F D SN) S2 D))
(S5 ((NOM SN) S2 D)))

```

c) la liste des modèles d'exploration; exemple:

```
(SETQ MODLIS @(MOD1 MOD2 MOD3 MOD4))
```

d) le contenu de chacun des modèles; exemple:

```

(SETQ MOD1 @ (=SN(("RAISON"))))
(SETQ MOD2 @(SN((SN)(D -)(SN((=NOM)))))

```

e) la liste des catégories descriptives; exemple:

```
(SETQ LISCAT @(NOM ART VER1 VER2 VER3 ADJ CONNEC))
```

f) le contenu de chacune des catégories descriptives; chaque catégorie descriptive peut être liée à une valeur ou à une suite de valeurs. Dans ce dernier cas, il s'agit de ramener la procédure d'identification d'une catégorie à une fouille arborescente sur une suite de valeurs. Par exemple les liaisons suivantes:

```

(SETQ VER @(VER))
(SETQ VER1 @(VER 1))
(SETQ VER2 @(VER 2))
(SETQ VER3 @(VER 3))
(SETQ VER31 @(VER 3 1))
(SETQ VER32 @(VER 3 2))

```

permettraient de manipuler VER2 comme un VER tout aussi bien qu'un VER2, où de manipuler VER32 comme un VER, un VER3 ou un VER32.

g) la liste des relations de dépendance contextuelle (RDC); exemple:

```
(SETQ RDCLIS @(DET INTER COM1 COM2))
```

## Les fonctions Déredéc

h) le contenu de chacune des RDC; les RDC sont liées à elles-mêmes, exemple:

```
(SETQ DET @DET)
```

Les différentes "listes" sont pratiques pour "sauver" sur espace-disque les éléments qu'elles renferment (plutôt que de sauver ceux-ci un à un).

Les variables générales du système reçoivent une valeur lorsque le Déredéc est chargé en mémoire. L'utilisateur peut changer la valeur de ces variables tout au long d'une séance au Déredéc, même de l'intérieur d'un automate; la valeur courante étant toujours la dernière attribuée. Remarques:

LIMITE doit être lié avant un appel à DESCRIP; cette variable peut avoir une valeur numérique ou grammaticale; dans ce dernier cas, elle est liée au nom d'une catégorie; exemple: (SETQ LIMITE 5) ou (SETQ LIMITE @CONNEC);

CONJUNCTION, HAUTEUR, EXPRES et LEMMA sont des variables affectant les explorations; elles sont liées à NIL lorsque le Déredéc est chargé en mémoire; leur autre valeur est T; exemple: (SETQ HAUTEUR T) (SETQ HAUTEUR NIL);

CHAINE prend pour valeur un nombre entier. Lorsque le système est chargé en mémoire sa valeur est de 1000. Comme on le sait la valeur de CHAINE affecte les procédures LEXIFORME et LEXIDEC; exemple: (SETQ CHAINE 3);

DEFINITIONS est lié à la liste des catégories créées par certaines opérations (PRODEC, PONDEC, ECART...) programmées dans les automates. On obtient DEFINITIONS à la fin de l'évaluation de DESCRIP.

CONTEXTE est associé à l'utilisation d'APLEC (voir le chapitre sur APLEC).

NBEXFAD est lié à un nombre entier: le nombre d'EXFAD lues dans une description de texte.

## Les fonctions Déredéc

NBARRET est lié au nombre d'arrêts programmés dans l'élaboration d'une description de texte.

NBREGLES est lié au nombre de règles appliquées.

NBCATE au nombre de catégories indexées.

NBRDC au nombre de RDC indexées.

NBEXFAL au nombre d'EXFAL indexées par TRANEXA.

NBRT est lié au nombre de séquences construites par DESCRIP.

LISTCA est lié à la liste des couples catégorie/expression atomique produite par PROCAT. Cette liste est pratique pour poursuivre en plus d'une séance une catégorisation automatique des expressions atomiques.

MEMO est lié à T ou NIL (voir la fonction USAGER).

EXPATOM est lié à T ou NIL (voir la fonction TRADES).

MESSAGES est lié à T ou NIL (voir la fonction TRADUIRE).

EXPLOR est lié au contenu de la dernière EXFAD dépistée par une fonction de fouille.

SOITRDC est une variable liée au fonctionnement de SOIT.

## Les erreurs de programmation

## 2.25 Les erreurs de programmation

Les messages d'erreurs qui s'inscrivent au terminal lors d'une séance Déredec sont de deux types. Ils peuvent parvenir soit du système Déredec, soit du système LISP 1.6, langage avec lequel l'utilisateur est toujours en contact direct. Certaines erreurs entraînent une rupture de l'évaluation, c'est-à-dire que l'évaluation d'une fonction est stoppée et que le contrôle passe à ce que les usagers du LISP 1.6 nomment le "break package"; exemple:

```
.RUN DEREDEC
*(SETQ COCO POIN)
POIN
UNBOUND VARIABLE-EVAL
(POIN BROKEN)
1:(SETQ POIN @POIN)
1: OK
```

...l'utilisateur veut lier COCO à la valeur de POIN mais cette dernière variable n'a pas de valeur; l'utilisateur doit d'abord lui en donner une;

...à l'intérieur d'un "break" (toujours identifié par la présence d'un chiffre ex: 3; et précédé de la liste constituée du nom de la fonction où se produit l'arrêt et du terme "BROKEN"), l'utilisateur peut faire exécuter toute fonction LISP ou toute fonction Déredec. Le chiffre qui remplace l'habituel astérisque (\*) signale le niveau du "break", ceci parce que plusieurs "breaks" peuvent s'empiler les uns à l'intérieur des autres; c'est-à-dire que l'évaluation d'une fonction, commandée de l'intérieur d'un arrêt peut elle aussi être sujette à un arrêt... On peut remonter d'un niveau par la commande ^ (accent circonflexe ou flèche vers le haut, selon le terminal), et remonter d'un coup jusqu'au plus haut niveau par la double commande: ^^ . Il y a donc deux façons de quitter le "break package":

a) par la commande OK, la computation continue;

b) par la commande ^, on revient au niveau immédiatement supérieur.

## Les messages d'erreurs du Déredéc

## 2.26 Les messages d'erreurs du Déredéc

## \*\*\*ERREUR:LIMITE-N'A-PAS-RECU-DE-VALEUR\*\*\*

...erreur survenant lors de l'évaluation de DESCRIP; l'utilisateur a oublié de lier LIMITE à une valeur. Il peut lier LIMITE de l'intérieur de l'arrêt, puis poursuivre la computation par OK.

## \*\*\*LA-VERSION-TRADUITE-DE-VOS-AUTOMATES-N'EST-PAS-CHARGEE-EN-MEMOIRE\*\*\*

...l'erreur peut venir de ce que les automates de l'utilisateur ne sont pas traduits, qu'ils n'ont pas été chargés en mémoire ou qu'ils n'ont pas encore été définis. Selon le cas, l'utilisateur devra faire TRADUIRE, DSKIN, ou SETQ puis remettre le contrôle par OK.

## Les erreurs de programmation des automates.

Ces erreurs surviennent durant l'évaluation de DESCRIP ou d'une autre fonction descriptive (SIMULECART USAGER).

## \*\*\*ERREUR:BOUCLE:AUTO-EMBOITEMENT-TROP-DE-PARENTHESES\*\*\*

...il s'agit d'une boucle logique dépistée grâce à un décompte des parenthèses ouvrantes et fermantes indexées aux EXFAD. L'automate est programmé de sorte qu'il ferme beaucoup trop de parenthèses pour celles qu'il ouvre ou vice-versa.

## \*\*\*ERREUR:BOUCLE:TROP-DE-RDC-DEPISTEE-ENTRE-LES-MEMES-EXFAD\*\*\*

...ici aussi l'erreur est dépistée à l'aide d'un décompte; trop de RDC relie les mêmes deux EXFAD, de telle sorte que l'existence d'une boucle logique dans le parcours de l'unité de contrôle est soupçonnée.

## \*\*\*ERREUR:LECTURE-IMPOSSIBLE-SUR-LA-SEQUENCE:HORS-LIMITE\*\*\*

...de la façon dont il est programmé, le pointeur devrait lire à la gauche de PHI, ou à la droite de la dernière EXFAD de la séquence, or le Déredéc ne le lui permet pas.

## \*\*\*ERREUR:AUCUNE-CONDITION-SATISFAITE-DANS-L'ETAT\*\*\*

...dans l'état où se trouve l'unité de contrôle, aucune condition n'est satisfaite par l'EXFAD pointée; l'unité de contrôle ne peut donc pas poursuivre sa route dans l'automate.



## Les messages d'erreurs du Déredec

Au moment de ces erreurs, l'utilisateur devra éditer l'automate défaillant, corriger l'erreur de programmation, faire traduire la liste des automates (celle qui contient l'automate défaillant), puis redonner le contrôle par la commande (CONTINUER); sur tout ceci voir les opérations CONTINUER, RETOUR, ECRIRE, et LIRE.

Note: la correction des automates ou de toute autre valeur se fait à l'aide de l'éditeur LISP appelé par la fonction EDITV (documentation: Quam and Diffie 1973); exemple: (EDITV AUTO1).

### Les messages du LISP 1.6.

<NOM D'UNE VARIABLE>  
UNBOUND VARIABLE-EVAL

...Le premier élément du message est le nom d'une variable; le message informe que cette variable n'a pas reçu de valeur. Si l'on se trouve dans un break, on devra lier la variable puis faire poursuivre la computation par OK.

<NOM D'UN FICHER>  
CAN'T FIND FILE-INPUT

...le message signifie l'impossibilité qu'il y a de trouver ce fichier sur espace-disque. Vérifier l'existence du fichier.

Note: tous les autres messages LISP signalent des erreurs plus importantes. Il s'agira dans tous les cas de revoir le caractère bien formé des expressions sur lesquelles on travaille (variables, fonctions...), de revenir au niveau le plus élevé (marqué par l'astérisque), puis de re-commander l'exécution de la fonction problématique.

## 2.27 Séance de programmation

Nous aimerions montrer comment il est possible de programmer des automates de telle sorte qu'ils puissent:

## Séance de programmation

a) aider l'utilisateur à catégoriser à la main les expressions atomiques de son texte; on utilisera dans l'exemple des catégories élémentaires (NOM ADJ ART POIN);

b) construire des syntagmes élémentaires (syntagmes nominaux SN);

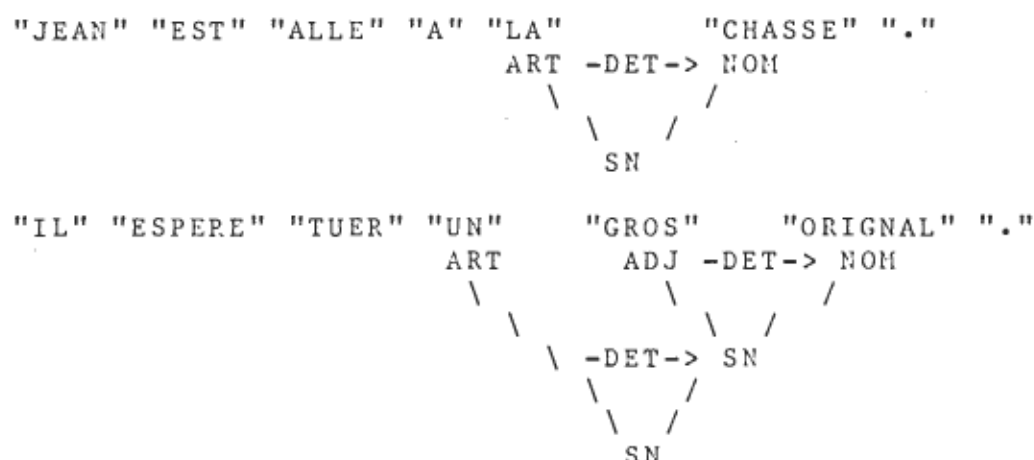
c) explorer la description de texte construite.

L'utilisateur a d'abord construit sa DDT d'entrée (avec un éditeur de texte, par exemple TECO sur l'ordinateur DEC-10). Cette DDT a été déposée sur le fichier CHASSE.TXT:

.TYPE CHASSE.TXT

"JEAN" "EST" "ALLE" "A" "LA" "CHASSE" "."  
 "IL" "ESPERE" "TUER" "UN" "GROS" "ORIGINAL" "."

Les automates proposés permettront d'obtenir la description suivante:



...Dans ce qui suit, SETQ servira à lier les catégories RDC, automates ainsi que les listes où sont rassemblés ces éléments.

.RUN DEREDEC

```

*(SETQ CATEL1 @(NOM ADJ ART POIN SN))
*(SETQ NOM @NOM)
*(SETQ ADJ @ADJ)
  
```

## Séance de programmation

```

*(SETQ ART @ART)
*(SETQ POIN @POIN)
*(SETQ SN @SN)

*(SETQ MODEL1 @(CONNEC* DETER*))

*(SETQ DETER* @(SN ((X) (DET +) (NOM ((=X))))))

*(SETQ CONNEC* @ (=X (".")))

*(SETQ RDCL1 @(DET))

*(SETQ DET @DET)

*(SETQ AUTO1 @(AUTO1 (S1 (PHI S2 D) (X S1 G))
  (S2 (CONNEC* (CAT POIN) (RETURN) )
    (X (RETURN) ))))

*(EDITV AUTO1)
#PP
(AUTO1 (S1 (PHI S2 D) (X S1 G))
  (S2 (CONNEC* (CAT POIN) (RETURN) )
    (X (RETURN) )))
#OK

```

...vérification de la bonne formation de l'automate à l'aide de l'éditeur LISP.

...L'automate AUTO1 fouille le texte à la recherche des "."; lorsqu'il est appliqué sur la DDT d'entrée, LIMITE est lié au nombre un (1), ainsi une première séquence est construite renfermant la première EXFAD; AUTO1 teste si cette EXFAD est un ".", la catégorise POIN dans l'affirmative, puis s'applique sur la séquence suivante. Cet automate a donc pour unique fonction de catégoriser les "." de la DDT d'entrée. Une fois ces indexations effectuées, l'automate suivant pourra s'appliquer sur des séquences élargies aux segments compris entre deux POIN.

```

*(SETQ AUTO2 @(AUTO2 (S1 (PHI S2 D) (X S1 G))
  (S2 ((NOM ADJ ART) S2 D) (POIN (E AUTO3 S1 G)
    (RETURN) ) (X (ARRET) S2 D))))

*(SETQ AUTO3 @(AUTO3 (S1 (PHI S2 D) (X S1 G))
  (S2 ((NOM SN) S3 G) (POIN (RETURN) ) (X S2 D))
  (S3 ((ART ADJ) (R DET (NOM SN) D) (O G) S4 D)
    (X S5 D)) (S4 ((NOM SN) (F D SN) S2 G)) (S5 ((NOM
    SN) S2 D))))

```

## Séance de programmation

```

*(EDITV AUTO2)
#PP
(AUTO2 (S1 (PHI S2 D) (X S1 G))
      (S2 ((NOM ADJ ART) S2 D)
          (POIN (E AUTO3 S1 G) (RETURN) )
          (X (ARRET) S2 D)))
#OK

*(EDITV AUTO3)
#PP
(AUTO3 (S1 (PHI S2 D) (X S1 G))
      (S2 ((NOM SN) S3 G) (POIN (RETURN) ) (X S2 D))
      (S3 ((ART ADJ) (R DET (NOM SN) D) (O G) S4 D)
          (X S5 D))
      (S4 ((NOM SN) (F D SN) S2 G))
      (S5 ((NOM SN) S2 D)))
#OK

```

L'automate AUTO2 permet lorsque le pointeur vise une expression non catégorisée d'arrêter la procédure et de passer le contrôle à l'utilisateur qui peut alors utiliser l'une des opérations Déredec (dans notre exemple <CAT>) pour catégoriser l'expression pointée.

Lorsque le pointeur atteint la fin de la séquence, l'automate AUTO3 est appelé et l'unité de contrôle s'y loge à l'état S1, le pointeur s'apprêtant à lire vers la gauche. Il tournera d'ailleurs dans cette direction jusqu'à la rencontre de la catégorie PHI. Cette catégorie domine une expression postiche située au début de toute séquence, indiquant la limite gauche de celle-ci. En S2, le pointeur fouillera chacune des expressions à la recherche d'un nom ou d'un SN. S'il en découvre, il y cherchera à la gauche la présence d'un article ou d'un adjectif afin de compléter la construction d'un SN. O et F ouvre et ferme des parenthèses; ces opérations sont argumentées d'une indication (gauche ou droite) permettant de savoir si l'utilisateur désire ou pas inclure l'expression pointée dans le syntagme construit. En S2, à la fin de la séquence, l'unité de contrôle retourne à l'automate quitté précédemment, et se positionne immédiatement à la droite de l'opération E (E pour exécuter). L'opération suivante lui commande encore un retour. Cet automate (AUTO2) étant le premier de la série, elle s'y loge au premier état, attendant qu'une nouvelle séquence soit construite et que la séquence précédente soit envoyée sur le fichier de sortie.

## Séance de programmation

```

*(TRADUIRE AUTOL1)
NOMBRE-DE-REGLES-PROGRAMMEES:
22

*(SETQ LIMITE 1)

*(DESCRIP AUTO1 (CHASSE.TXT) (CHASSE.POI))

*(LIRFIC (CHASSE.POI))
"JEAN"
"EST"
"ALLE"
"A"
"LA"
"CHASSE"
(POIN NIL ".")
"IL"
"ESPERE"
"TUER"
"UN"
"GROS"
"ORIGINAL"
(POIN NIL ".")

*(SETQ LIMITE @POIN)

*(DESCRIP AUTO2 (CHASSE.POI) (CHASSE.DES))

-ARRET-PROGRAMME-SEQUENCE:
(PHI NIL NIL)
***POINTEUR:*****
"JEAN"
"EST"
"ALLE"
"A"
"LA"
"CHASSE"
(POIN NIL ".")
((AUTO2 . S2) BROKEN)
1:(CAT NOM)
1:OK

-ARRET-PROGRAMME-SEQUENCE:
(PHI NIL NIL)
(NOM NIL "JEAN")
***POINTEUR:*****
"EST"

```

## Séance de programmation

```

"ALLE"
"A"
"LA"
"CHASSE"
(POIN NIL ".")
((AUTO2 . S2) BROKEN)
1:OK

```

-ARRET-PROGRAMME-SEQUENCE:

```

(PHI NIL NIL)
(NOM NIL "JEAN")
"EST"
***POINTEUR:*****
"ALLE"
"A"
"LA"
"CHASSE"
(POIN NIL ".")
((AUTO2 . S2) BROKEN)
1:OK

```

-ARRET-PROGRAMME-SEQUENCE:

```

(PHI NIL NIL)
(NOM NIL "JEAN")
"EST"
"ALLE"
***POINTEUR:*****
"A"
"LA"
"CHASSE"
(POIN NIL ".")
((AUTO2 . S2) BROKEN)
1:OK

```

-ARRET-PROGRAMME-SEQUENCE:

```

(PHI NIL NIL)
(NOM NIL "JEAN")
"EST"
"ALLE"
"A"
***POINTEUR:*****
"LA"
"CHASSE"
(POIN NIL ".")
((AUTO2 . S2) BROKEN)
1:(CAT ART)
1:OK

```

## Séance de programmation

-ARRET-PROGRAMME-SEQUENCE:

```
(PHI NIL NIL)
(NOM NIL "JEAN")
"EST"
"ALLE"
"A"
(ART NIL "LA")
***POINTEUR:*****
"CHASSE"
(POIN NIL ".")
((AUTO2 . S2) BROKEN)
1:(CAT NOM)
1:OK
```

-ARRET-PROGRAMME-SEQUENCE:

```
(PHI NIL NIL)
***POINTEUR:*****
"IL"
"ESPERE"
"TUER"
"UN"
"GROS"
"ORIGNAL"
(POIN NIL ".")
((AUTO2 . S2) BROKEN)
1:OK
```

-ARRET-PROGRAMME-SEQUENCE:

```
(PHI NIL NIL)
"IL"
***POINTEUR:*****
"ESPERE"
"TUER"
"UN"
"GROS"
"ORIGNAL"
(POIN NIL ".")
((AUTO2 . S2) BROKEN)
1:OK
```

-ARRET-PROGRAMME-SEQUENCE:

```
(PHI NIL NIL)
"IL"
"ESPERE"
***POINTEUR:*****
"TUER"
"UN"
```

```
"GROS"  
"ORIGNAL"  
(POIN NIL ".")  
((AUTO2 . S2) BROKEN)  
1:OK
```

```
-ARRET-PROGRAMME-SEQUENCE:  
(PHI NIL NIL)  
"IL"  
"ESPERE"  
"TUER"  
***POINTEUR:*****  
"UN"  
"GROS"  
"ORIGNAL"  
(POIN NIL ".")  
((AUTO2 . S2) BROKEN)  
1:(CAT ART)  
1:OK
```

```
-ARRET-PROGRAMME-SEQUENCE:  
(PHI NIL NIL)  
"IL"  
"ESPERE"  
"TUER"  
(ART NIL "UN")  
***POINTEUR:*****  
"GROS"  
"ORIGNAL"  
(POIN NIL ".")  
((AUTO2 . S2) BROKEN)  
1:(CAT ADJ)  
1:OK
```

```
-ARRET-PROGRAMME-SEQUENCE:  
(PHI NIL NIL)  
"IL"  
"ESPERE"  
"TUER"  
(ART NIL "UN")  
(ADJ NIL "GROS")  
***POINTEUR:*****  
"ORIGNAL"  
(POIN NIL ".")  
((AUTO2 . S2) BROKEN)  
1:(CAT NOM)  
1:OK
```



```

*(LIRFIC (CHASSE.DES))
(NOM NIL "JEAN")
"EST"
"ALLE"
"A"
(SN NIL (ART ((DET 1 +)) "LA")
          (NOM ((DET -1 -)) "CHASSE"))
(POIN NIL ".")
"IL"
"ESPERE"
"TUER"
(SN NIL
  (ART ((DET 1 +)) "UN")
  (SN ((DET -1 -))
      (ADJ ((DET 1 +)) "GROS")
      (NOM ((DET -1 -)) "ORIGINAL")))
(POIN NIL ".")

*DETER*
(SN ((X) (DET +) (NOM ((=X)))))

*(PROJEC (CHASSE.DES) (EXPLOR.CHA) DETER*)
2

*(LIRFIC (EXPLOR.CHA))
"CHASSE"
"ORIGINAL"

*(DSKOUT (MODEL1.DEF) MODEL1)
*(DSKOUT (CATEL1.DEF) CATEL1)
*(DSKOUT (RDCL1.DEF) RDCL1)

^C ^C
.

.CLOSE

```

..cette commande au moniteur (de la DEC-10) procure plus de sécurité concernant la création des fichiers produits durant une séance.

## CHAPITRE 3

### Une Grammaire de Texte

#### 3.1 Généralités

La première grammaire descriptive de texte que nous ayons programmée à l'aide du logiciel Déredéc s'inscrit dans le cadre d'expériences de type "bottom-up", c'est-à-dire des expériences où se réalise un minimum d'investissement hors-contexte pour une batterie impressionnante de manipulations. Il s'agissait en bref de réussir l'indexation de structures grammaticales de surface à chacune des phrases d'un texte, pour permettre par la suite l'observation du comportement de certains lexèmes dans la distribution des structures indexées.

Le modèle le plus général qui soustend l'articulation des structures dépistées par cette grammaire décrit des interactions entre deux entités de la phrase: les verbes et les nominaux.

Trois relations de dépendance contextuelle codifient les différentes formations des groupes verbaux. Dans une proposition, le thème est ce dont on parle et le propos ce qu'on en dit. Le thème peut être un groupe nominal, un groupe propositionnel ou un groupe verbal, mais le propos ne peut être qu'un groupe propositionnel ou qu'un groupe verbal. On distingue habituellement cette relation de surface de la relation profonde sujet/verbe par un exemple du type:

"La police a arrêté les bandits."

"Les bandits ont été arrêtés par la police."

où, dans la première phrase, "la police" est à la fois sujet et thème, et où dans la seconde phrase "les bandits" est le thème, et "la police" demeure le sujet.

Deux autres RDC se rapportent aux développements du propos. Les premières rassemblent entre autres choses l'ensemble des objets directs alors que les secondes regroupent de façon incomplète les développements circonstantiels.

Ces deux relations sont plus "faibles" que la relation thème/propos, en ce sens que les éléments qu'elles dépistent se juxtaposent dans certains cas difficilement identifiables. Nous les avons quand même présenté à titre indicatif.

Les nominaux de la phrase sont dans cette grammaire des entités dont la mise en place ou l'existence est en quelque sorte présumée à la construction des groupes verbaux. La mise à jour des groupes nominaux se fait le plus souvent à l'occasion du dépistage de relations de détermination.

La version en "français contemporain" du "Discours de la méthode" de René Descartes nous a servi de corpus. Ce texte fut retenu pour deux raisons; d'une part, il se donne dans une syntaxe exemplaire et il offre à la question du dépistage des syntagmes propositionnels emboîtés un niveau très élevé de difficulté; par ailleurs il est en quelque sorte saturé du point de vue de l'interprétation de son contenu facilitant par là toute nouvelle approche de type analytique. Sur ce texte, les lexèmes et connecteurs furent déclarés "expressions atomiques", alors que les "séquences" furent ramenées aux phrases ou proto-phrases, c'est-à-dire à tous les segments du texte compris entre deux ponctuations fortes.

Nous avons différencié trois types de catégories descriptives: les catégories de base, les catégories temporaires qui servent de relais à l'indexation des catégories de base, et les catégories syntagmatiques qui, comme leur nom l'indique, représentent des regroupements sur les catégories.

L'ensemble de tout le traitement est divisé en deux étapes; dans la première, des automates indexent les catégories de base aux lexèmes, alors que, dans la seconde, d'autres automates composent les groupes syntagmatiques, dé-

pistant parallèlement les relations de dépendance contextuelle.

### 3.2 L'indexation des catégories de base

Les catégories de base sont les primitifs de la grammaire implantée; c'est-à-dire qu'une fois leur indexation terminée, toutes les manipulations concernant la composition syntagmatique et le dépistage des RDC sont entièrement automatisées. Le tableau ci-bas rassemble les catégories de base et les catégories temporaires; ces dernières se définissent comme des alternatives entre différentes catégories de base.

#### Les Catégories de Base:

N1....Noms propres et communs;  
 N211..Pronoms pers. sujets (je tu ...);  
 N212..Pronoms pers. compléments (te me..);  
 N22...Autres pronoms;  
 N23...Pronoms relatifs;  
 D11...Déterminants verbaux (ne pas...);  
 D12...Déterminants nominaux (mes ton...);  
 D2....Déterminants connecteurs (que comme...);  
 V1....Autres verbes;  
 V21...Infinitifs;  
 V22...Participes présents;  
 V23...Participes passés et adjectifs;  
 C1....Conjonctions de coordination;  
 C21...Prépositions faibles (de d);  
 C22...Prépositions fortes (avec sans...);  
 C31...Ponctuation faible ( , );  
 C32...Ponctuations fortes ( . ; : ? ...);

#### Les Catégories Temporaires:

T11....N212/D12/N22 (le les leur...);  
 T12....N212/C22 (en...);  
 T131...D12/N1 (son...);  
 T132...D12/N22 (un une aucun...);  
 T133...D12/N22/V23 (certain plusieurs...);  
 T134...N22/V23 (seuls seule...);  
 T141...N212/D2 (s);  
 T142...D11/D2 (si);  
 T151...C22/N1 (contre pour...);  
 T1521..D11/N22 (plus moins...);  
 T1522..D11/N1 (abord...);

## Une Grammaire de Texte

T1531..D11/N22/V23 (même...);  
 T1532..D11/D12 (toutes...);  
 T1533..D11/N22/D12 (tout);  
 T16....C22/V23 (pourvu vu...);  
 T17....D11/C22 (après avant);  
 T18....N211/N212 (nous vous).

L'automate FD-1 est responsable de l'indexation des catégories de base et des catégories temporaires. Au moment de l'application de cet automate, la limite des séquences est de 1; c'est-à-dire qu'une seule expression est composée par séquence.

```

(FD-1 (S1 (PH1 S2 D) (X S1 G))
      (S2 (X (TRANEXA CATEGO) S3))
      (S3 ((D1 D2 N V C T1) (RETURN))
           (X (ARRET) (RETURN) )))
  
```

On teste en S2 si cette expression n'apparaît pas sur CATEGO, entité construite par TRAFIC et renfermant une liste des formes fonctionnelles (déposée en annexe) du français; on appelle formes fonctionnelles des lexèmes qui, quoique peu nombreux, sont très fréquents dans les textes (le la les qui dont où que pour vers...). Si ce test est positif, une catégorie est indexée, l'unité de contrôle s'en rendra compte en S3 et une autre séquence sera composée; dans le cas contraire, la computation est arrêtée et deux possibilités s'offrent à l'utilisateur pour indexer l'expression pointée:

a) catégoriser le seul lexème pointé;

b) catégoriser le lexème pointé ainsi que tous les autres lexèmes qui ont même morphologie dans la suite du texte. Dans ce dernier cas, la catégorie proposée (il peut s'agir d'une catégorie de base ou d'une catégorie temporaire) doit être valable quel que soit le contexte des lexèmes concernés. Les catégories temporaires posent des restrictions sur le nombre de catégories que peut recevoir la forme lexicale pointée, ces catégories seront donc plus facilement désambiguées par la suite. Par exemple la forme lexicale "en" recevra hors contexte une catégorie temporaire T12 qui deviendra par la suite un N212 ou un C22 selon le contexte. La série des automates FDO (déposée en annexe) transformera automatiquement les catégories temporaires en catégories de

base.

Sur le texte du "Discours de la méthode", l'indexation des catégories de base a été automatisée à 85%. Ce chiffre est obtenu en comparant le nombre total de lexèmes du texte au nombre d'arrêts programmés survenus. Ce pourcentage est retenu comme un paramètre de première importance pour penser la validation des descriptions finales obtenues. Il est à remarquer que, dans les cas où on appliquerait le même algorithme d'indexation à des textes différents, le pourcentage d'automatisation varierait alors selon les rapports suivants:

a) nombre de formes lexicales / nombre de réalisations des formes (rapport "types"/"tokens");

b) nombre de formes fonctionnelles / nombre de formes pleines;

Exemple d'une phrase à la sortie de FDO (dans cet exemple et dans d'autres qui suivent, nous avons omis d'entourer les expressions atomiques de leurs guillemets, ceci, pour des raisons d'édition):

L' idée que Jean proposait devait vite tomber dans l'oubli  
 T11 N1 D2 N1 V1 V1 D11 V21 C22 T11 N1  
 D12 D12

La seconde étape du traitement est subdivisée en trois temps. Des automates vont d'abord composer sur les catégories de base indexées aux lexèmes, les molécules les plus fortement liées (certains syntagmes déterminatifs et nominaux), puis d'autres automates vont assurer le dépistage des connecteurs propositionnels dans la phrase, et enfin un dernier groupe d'automates va construire l'ensemble des syntagmes emboîtés. Ainsi, pour reprendre l'exemple déjà présenté à la section précédente, nous obtiendrons, après le passage des automates des séries FD1 FD2 et FD3-4-5, respectivement chargés des trois phases décrites ci-haut (la description de ces automates est déposée en annexe), la séquence:

### Les Catégories Syntagmatiques:

- ```
GN....Groupe nominal;
CV1...Groupe verbal construit sur un V1;
GV21..Groupe verbal construit sur un V21;
GV22..Groupe verbal construit sur un V22;
GV23..Groupe verbal construit sur un V23;
GD11..Groupe déterminant construit sur un D11;
GP....Groupe propositionnel;
CP....Connecteur de propositions.
```



Quatre RDC codifient les rapports possibles entre les catégories du système:

--I--> l'interdépendance ou la relation thème/propos  
 --P1--> première relation de développement du propos;  
 --P2--> seconde relation de développement du propos;  
 --D--> la relation de détermination nominale.

a) La relation d'interdépendance ou thème/propos; exemples:

(Courageux jusqu'au bout), (Luc (ne cédait pas)).  
 GN--I--> GV1  
 GV23-----I-----> GV1

(Celui qui a inventé cela) (n'est pas revenu).  
 GN-----I-----> GV1

(Jean (ne parle pas souvent)), (c' (est cela qui...))  
 GN---I-----> GV1 GN---I---> GV1  
 GP-----I-----> GP

b) Les relations de développement du propos:

Luc mange (le canard). Luc a mangé (ce matin).  
 GV1 <-P1--GN GV23 <-P1--GN

Il voulait jouer pour gagner. Il mange avec appétit.  
 GV21 <-P2--GV21 GV1 <-P2--GN

c) Les relations de détermination nominale:

L'autobus des écoliers. Celui (que vous voyez là-bas).  
 GN <--D-----GN GN <--D---GP

L'ensemble de notre grammaire (les automates des séries FD-1 FD0 FD1 FD2 FD3-4-5) totalise une soixantaine d'automates où se répartissent quelque 1,500 règles. Sur le "Discours de la méthode", l'indexation complète des structures à toutes les séquences aura nécessité 2,237,600 applications de ces règles. 46,030 catégories et 7058 RDC auront alors

été dépistées. Le nombre total d'heuristiques, c'est-à-dire le nombre de manipulations ad hoc, représente seulement 8% du nombre total d'indexations accomplies; on obtient ce chiffre en comparant le nombre de catégorisations manuelles plus le nombre d'entrées au fichier des formes fonctionnelles au nombre total de catégories et de RDC dépistées.

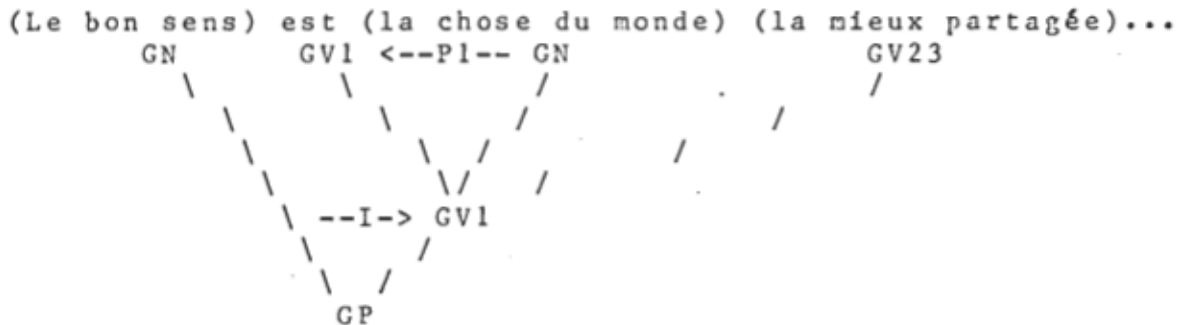
L'algorithme de structuration de surface a fait l'objet d'une première proposition (Plante 1975), qui fut longuement commentée du point de vue linguistique. Nous avons préféré pour cette nouvelle version une pédagogie d'exposition différente. Nous croyons que le lecteur intéressé au contenu de l'algorithme pourra en prendre rapidement connaissance en l'appliquant "manuellement" et étape par étape sur les exemples qui sont pour cette fin déposés en annexe à ce chapitre.

### 3.4 Le traitement des ambiguïtés

Certains segments du texte peuvent recevoir un traitement ambigu eu égard à la grammaire implantée. C'est-à-dire qu'à certaines occasions la machine est inadéquate pour décrire correctement la structure du segment pointé. On a distingué deux types d'inadéquations:

#### a) les inadéquations faibles:

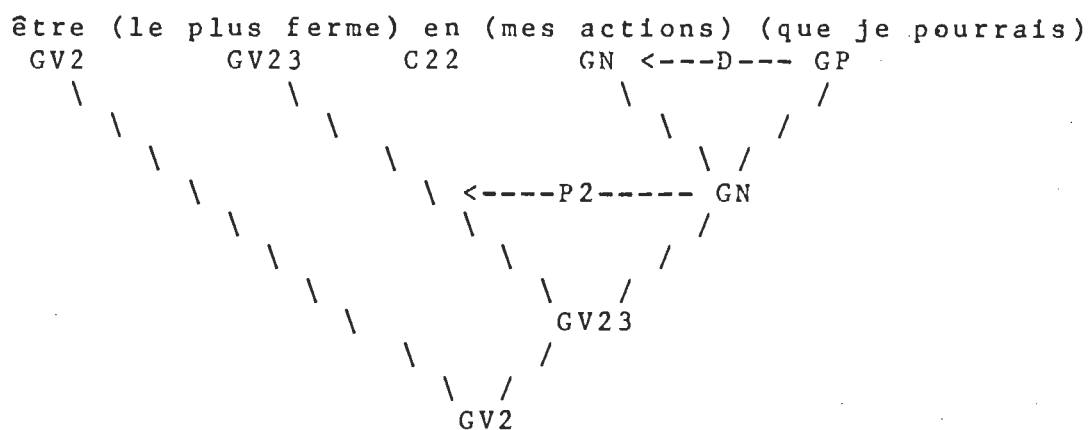
Chaque fois que l'analyse du contexte le lui permet, l'automate marque les EXFAD des RDC qui les caractérisent. Toutefois, dans certains cas, il percevra une construction comme ambiguë, n'assignera pas de RDC aux catégories mais proposera quand même un regroupement syntagmatique. Exemple:



Le GV23 (la mieux partagée) n'a reçu aucune RDC mais s'est trouvé rattaché au GV1, plutôt qu'à d'autres éléments de la suite de la phrase. La description obtenue ne va pas à l'encontre du modèle mais en souligne certaines limitations, le GV23 aurait dû se retrouver relié par une relation de détermination au GN (la chose du monde).

#### b) Les inadéquations fortes;

Les automates ont complété une description de la phrase dans les termes de la composition syntagmatique et du dépistage des RDC. Toutefois, la description obtenue va à l'encontre de l'intuition grammaticale. Exemple:



Le GP (que je pourrais) devrait déterminer le GV22 (être le plus ferme), plutôt que le GN (mes actions). Les inadéquations fortes s'inscrivent à différents registres de la grammaire. Aussi, envisageons différentes possibilités de solution au problème.

D'aucuns diraient que c'est l'amalgame des propriétés sémantiques du verbe pouvoir avec celles du GN (mes actions) et du GV2 (être le plus ferme) qui permet à Descartes d'omettre la virgule entre "actions" et "que". On aurait ainsi:

amalgame permis: "je pourrais être ferme"  
 amalgame défendu: "je pourrais des actions"

L'implantation de cette première solution nécessiterait toutefois une très grande sophistication du système des catégories de base (une sémantisation), augmentant de beaucoup le caractère "ad hoc" de ces informations primitives, et réduisant par là à presque rien l'automatisation de la procédure de leur indexation.

Une seconde solution impliquerait deux transformations au système. Il faudrait d'une part distinguer les pronoms démonstratifs des autres pronoms, retenant qu'ils ont la propriété d'impliquer une détermination, une référence au monde déjà connue; d'autre part, les syntagmes composés devraient garder trace du nombre de relations de détermination qui y sont constituées. L'hypothèse sous-jacente à

tout ceci est que plus un syntagme est déterminé, plus il devient facilement l'objet de nouvelles déterminations. Il aurait ainsi, dans les cas ambigus, la priorité d'attribution. Considérons les exemples:

- 1) être (le mieux) en moi-même (que je pouvais...);
- 2) être (mieux) en (ceci) (que je pouvais...);
- 3) être (le mieux) en (ceci) (que je pouvais...);
- 4) être (le mieux) en (tout ce) (que je pouvais...);

Aux nos.1 et 2, le GP (que je pouvais ...) s'attribue plus naturellement au syntagme déjà déterminé ((le mieux) au no.1, et (ceci) au no.2)). Au no.3, l'ambiguïté demeure, les deux syntagmes sont déterminés au même degré. Au no.4, le GP s'attribue au syntagme le plus déterminé (tout ce).

Alors que la première solution impliquait des transformations importantes au premier niveau d'intervention (l'indexation des catégories de base), celle-ci impliquerait des modifications majeures au niveau de la construction syntagmatique: il faudrait marquer dans la nomenclature des syntagmes la trace des relations de détermination qui y sont dépistées.

Une autre solution serait d'incorporer au système une règle qui défende l'attribution d'une proposition subordonnée au GN qui la précède lorsque ce GN est lui-même précédé d'une préposition (C22), motivant par là que tout GN précédé d'une préposition se rapporte tout d'abord aux éléments de sa gauche séparés de lui par cette préposition. Ainsi,

1) être le plus ferme en mes actions que je pourrais... serait bien analysé par cette règle, alors que:

2) être le plus ferme en toutes les actions que j'accomplissais... lui donnerait contre-exemple.

Par ailleurs, les cas du type 2 sont plus fréquents que ceux du type 1 dans le corpus étudié, et c'est finalement sur ce critère empirique que la question fut tranchée.

Cette solution, quoique moins complète (nous croyons qu'elle laisse passer plus d'erreurs que la première sinon que la deuxième), laisse toutefois intacte l'économie désirée des rapports d'efficacité.

On voit ici, à l'occasion de cet exemple, comment une grammaire Déredec permet de ramener sur le plan de la mesure et de l'empirie la question de la décidabilité grammaticale: toute proposition relative à l'évaluation d'une description produite par une grammaire pourra être accompagnée de données portant sur:

a) l'automatisation de l'indexation des catégories et RDC d'une grammaire; il s'agit d'une mesure du caractère "ad hoc" d'un système;

b) l'importance numérique des règles de manipulation; une mesure de la machinerie automatique;

c) les caractéristiques distributionnelles de la description étudiée; par exemple: la fréquence d'une solution relativement au nombre d'occurrences d'une structure problématique.

Ces trois niveaux d'information organisent le cadre empirique dans lequel peuvent se comparer différentes hypothèses sur le dépistage des structures grammaticales.

De façon générale, ces dispositions éviteront le rejet trop rapide des hypothèses: alors que l'on peut toujours assez facilement imaginer un contre-exemple à une règle donnée, on ne peut toutefois avec la même facilité construire un contre-corpus ou substituer une règle qui améliorerait la description dans les mêmes conditions économiques d'application.

## L'exploration de contenu

## 3.5 L'exploration du contenu des structures indexées

## A) L'organisation lexicale

Nous présentons ici les résultats d'un premier exercice d'exploration du contenu des descriptions indexées aux phrases du "Discours de la méthode" par la GDT décrite dans les sections précédentes.

L'exercice est divisé en deux temps. Il s'agissait en premier lieu de rassembler les noms (N1) qui ont reçu dans le texte des relations de détermination. Cette contrainte peut se représenter par le modèle d'exploration:

(GN ((X) (D +) (GN ((N1 ((=X)))))))

Ainsi, le lexème "connaissance" apparaît-il dix fois en position N1 dans cette structure, le lexème "dieu" neuf fois, etc... Il nous a paru intéressant de comparer ce nombre de déterminations à la fréquence dans le texte des lexèmes déterminés afin de pondérer les résultats obtenus. Le calcul a été effectué pour les noms ayant une fréquence absolue supérieure ou égale à dix.

L'ensemble dépisté est un ensemble flou (fuzzy set) où chaque élément est accompagné d'un indice d'appartenance calculé par la formule:

nombre de déterminations / fréquence absolue d'occurrences d'une structure problématique.

## L'exploration de contenu

Tableau no.1

|              | nombre de<br>déterminations | fréquence<br>absolue | indice de<br>détermination |
|--------------|-----------------------------|----------------------|----------------------------|
| choses       | 58                          | 66                   | .878                       |
| opinions     | 21                          | 27                   | .777                       |
| raisons      | 14                          | 23                   | .608                       |
| pensées      | 13                          | 22                   | .590                       |
| corps        | 22                          | 38                   | .578                       |
| sang         | 20                          | 36                   | .555                       |
| chose        | 27                          | 51                   | .529                       |
| hommes       | 17                          | 33                   | .515                       |
| sens         | 11                          | 22                   | .500                       |
| connaissance | 10                          | 20                   | .500                       |
| dessein      | 10                          | 20                   | .500                       |
| moyen        | 7                           | 23                   | .304                       |
| vérité       | 8                           | 27                   | .296                       |
| nature       | 10                          | 35                   | .285                       |
| dieu         | 9                           | 34                   | .264                       |
| raison       | 10                          | 38                   | .263                       |
| monde        | 7                           | 28                   | .250                       |
| esprit       | 4                           | 28                   | .142                       |
| coeur        | 3                           | 47                   | .063                       |

Un trait important de la structuration de surface concerne le dépistage des relations de détermination nominale. Chaque détermination nominale souligne dans une phrase l'existence d'une présupposition portant sur la connaissance du nominal déterminé. Ainsi dans:

L'hypothèse de Davidson semblait fructueuse.

GN <---D--- GN

la connaissance du type d'"hypothèse" dont on parle quand on la détermine par "Davidson" est présupposée à la pleine compréhension de la phrase. Ces faits sont en général bien connus des logiciens et linguistes (voir sur la question Ducrot O. 1972).

Contrairement à la relation de détermination nominale, la RDC thème/propos relie des éléments dont on peut dire qu'ils sont "posés" par la phrase plutôt que présupposés à celle-ci.

(L'hypothèse de Davidson) (semblait fructueuse).

GN -----I-----> GV1



## L'exploration de contenu

Il me semblait dès lors intéressant de comparer l'univers des nominaux déterminés à celui des nominaux thématiques afin d'observer si cette relation théorique de la complémentarité entre les deux RDC étudiées se décalquait empiriquement dans un texte donné. L'ensemble flou des nominaux thématiques fut construit sur les mêmes critères (hormis celui du modèle d'exploration) que ceux décrits ci-haut pour le dépistage des nominaux déterminés.

Modèle d'exploration: (GP ((X) (I -) (GN ((N1 (( =X))))))

Tableau no.2

|              | indice de<br>thématisation |
|--------------|----------------------------|
| dieu         | .441                       |
| raison       | .236                       |
| nature       | .228                       |
| sang         | .222                       |
| raisons      | .217                       |
| esprit       | .214                       |
| sens         | .181                       |
| dessein      | .150                       |
| hommes       | .121                       |
| corps        | .105                       |
| connaissance | .100                       |
| opinions     | .074                       |
| monde        | .071                       |
| coeur        | .063                       |
| pensées      | .045                       |
| choses       | .045                       |

Il existe effectivement entre les deux ensembles une complémentarité remarquable. Sauf quelques exceptions, les éléments les plus déterminés sont les moins thématiques et vice-versa. En fait, seulement deux exceptions sont frappantes: le lexème "coeur" qui à la fois est faiblement déterminé et faiblement thématique, et le lexème "sang" qui à la fois est fortement déterminé et fortement thématique. Or il est à remarquer que ces deux lexèmes sont presque exclusivement utilisés dans un chapitre très périphérique de la problématique du "Discours de la méthode", celui réservé aux descriptions physiologiques (cinquième partie). On peut croire qu'il paraît donc soutenable de débiter la régularité observée au compte d'une théorie de la cohésion textuelle. C'est de plus l'interprétation qu'est venue renforcer une seconde expérience de description similaire quant à la procédure suivie mais appliquée cette fois à un corpus constitué d'une vingtaine de lettres différentes mais écrites sur

un sujet commun (ce corpus est présenté au chapitre suivant). Ce corpus a un double avantage. La redondance y est suffisamment élevée pour permettre des études comparatives (entre les nominaux déterminés et thématisés) sans qu'il constitue pour autant ce qu'on appelle habituellement un texte. Or, contrairement au "Discours de la méthode", ce n'est qu'à de rares exceptions près que les nominaux les plus déterminés ne se trouvaient pas être en même temps les plus thématisés.

Plusieurs questions restent toutefois ouvertes. Quoique les régularités qui ressortent de notre examen semblent s'inscrire au niveau d'une structure de texte, nous ne pouvons trancher si elles sont communes à tous les textes, ou spécifiques à un groupe de textes qui partageraient avec le "Discours" certains traits communs (les textes philosophiques, les textes "cartésiens"...).

Cette impression que les éléments fortement déterminés ("choses", "opinions", "raisons", "pensées", "chose") sont des concepts quelque peu "vides" et que la détermination serait en quelque sorte une opération première devant être assumée pour que le lecteur puisse acquérir (par le texte) des concepts plus riches, est-elle spécifique au texte étudié ?

Des éléments de réponse à ces questions ne sauront être apportés que par des expérimentations comparatives multipliées.

## B) L'organisation textuelle

Une première partie de cette seconde expérience d'exploration du contenu des descriptions indexées au "Discours de la méthode" consiste à associer aux GP (groupes propositionnels) dominants de chacune des séquences du texte, des catégories descriptives de la participation caractéristique de chacun de ces GP à certains phénomènes grammaticaux. Nous avons traité plus spécifiquement des phénomènes de la "nominalisation", de la "distribution des présuppositions" et de l'"organisation thématique". Dans les lignes qui suivent, les catégories relatives à ces études sont appelées "textuelles" pour les distinguer de celles déjà introduites.

Nous avons cherché à connaître, pour les trois phénomènes, l'organisation textuelle de leurs Prémises, Dénouements et Autonomies. Les Prémises et Dénouements sont construits à la suite d'un calcul sur les rapports qu'entretiennent entre elles les Attentes et Résolutions d'un phénomène dans une EXFAD donnée. Dans les deux cas, pour effectuer cette description, nous nous sommes servi de la fonction SIMULECART à laquelle nous avons donné comme premier argument la commande MOY, signifiant par là que l'écart significatif nécessaire à l'indexation d'une catégorie devait être égal ou supérieur à l'écart moyen de tous les écarts observés; par exemple, une catégorie Dénouement aurait été indexée à un GP si le nombre de Résolutions s'écartait d'au moins un écart moyen du nombre d'attentes du phénomène grammatical étudié dans ce GP.

La catégorie Autonomie indexe les GP pour lesquels il n'existe d'une part ni attente ni résolution, mais qui sont par ailleurs eux-mêmes des réalisations du phénomène étudié. Ces catégories ont été construites par FOUDEC1. On trouvera en fin de chapitre l'ensemble des fonctions SIMULECART programmées ainsi qu'une description de l'automate qui a effectué l'indexation des catégories Attente et Résolution.

1) La récurrence nominale

La récurrence nominale est un facteur prédominant de la cohésion textuelle; les catégories textuelles relatives à cette étude concernent:

## L'exploration de contenu

a) le nombre de fois où des nominaux du GP pointé apparaissent dans la partie de la description de texte qui précède l'apparition du GP: NOM\*NOMEX\*RESOLU; cette catégorie est construite des noms de deux modèles d'exploration NOM\* et NOMEX\* respectivement liés à (GN ((N1 ((=X)))) et (GN :PATRON). La procédure d'indexation des catégories reliées aux analyses hors-séquence est décrite en détail sous la rubrique des opérations PRODEC, PONDEC et ECART dans le chapitre réservé à la description du logiciel. Rappelons en bref que:

1) le premier modèle (ici NOM\*) est projeté sur l'EXFAD pointée (au niveau de description où l'automate est appliqué, ces EXFAD sont des GP pouvant dominer des séquences entières du texte), ce modèle dépiste les nominaux (N1);

2) puis le second modèle (où :PATRON est lié au résultat dépisté en 1) est projeté sur l'ensemble de la description de texte qui précède la séquence pointée;

3) une catégorie sera construite et indexée à l'EXFAD pointée. Le nom de cette catégorie est constitué de quatre parties (par exemple: NOM\*NOMEX\*RESOLU44), la dernière renseignant sur le nombre de fois où le dernier modèle s'est trouvé réalisé sur le segment de la DDT précédant la séquence pointée.

b) Le nombre de fois où des nominaux du GP apparaissent dans la partie de la DDT qui suit l'apparition du GP: NOM\*NOMEX\*ATTENT.

On a de plus observé les effets causés par la pondération de ces catégories. Le calcul a été effectué en comparant NOM\*NOMEX\*RESOLU à NIL\*NOM\*RESOLU pour pondérer la résolution nominale, et en comparant NOM\*NOMEX\*ATTENT à NIL\*NOM\*ATTENT pour pondérer l'attente nominale (comme on le sait, NIL\* est un modèle vide). Dans les deux cas, on compare le nombre de fois où les nominaux du GP apparaissent dans la partie du texte (avant ou après) au nombre total de nominaux de cette même partie. Cette pondération permet de contrecarrer la courbe naturelle des attentes et résolutions

## L'exploration de contenu

dans un texte, l'attente étant plus forte au début et la résolution plus forte à la fin. Les valeurs pondérées informent des attentes et résolutions nominales comme si la distance qui les séparait de la fin du texte était toujours uniforme.

Observons plutôt les listes (produites par PROFIL) des adresses des dénouements et prémisses des deux premières parties du "Discours de la méthode" (pour lesquelles l'expérience a été tentée de façon séparée pour des fins de comparaison intertextuelle) selon que les catégories sont pondérées ou pas. Dans ces listes, les adresses sont relatives, elles sont représentées par un chiffre de 1 à 100 indiquant le pourcentage de texte déjà lu au moment de l'indexation de la catégorie textuelle.

## Attentes et Résolutions pondérées:

## Première partie du "Discours de la méthode":

DENOUEMENTS: (2 5 11 13 15 20 22 24 27 28 29 30 33 37 41 63 67 70 71 72 74 80 83 93 94 95 96 98 99)

PREMISSSES: (1 6 7 9 10 17 32 35 39 43 46 48 49 51 52 56 59 85 87 89)

## Seconde partie:

DENOUEMENTS: (6 9 10 23 24 29 30 35 40 41 46 49 50 51 62 67 70 73 78 79 80 83 85 87 88 90 91 93 98 99)

PREMISSSES: (2 4 5 7 11 13 15 16 17 20 21 32 38 43 48 52 56 74 89)

## Attentes et Résolutions non-pondérées:

## Première partie:

DENOUEMENTS: (37 41 63 70 72 74 80 83 88 89 90 91 94 96 98 99)

## L'exploration de contenu

PREMISSSES: (1 2 5 6 7 9 10 11 16 17 18 32 35 39 43 46 48 51  
52 56 59 85 87 89)

Seconde partie:

DENOUEMENTS: (23 24 29 30 33 35 40 41 44 46 49 50 51 62 65  
67 70 73 78 79 80 83 85 87 88 90 91 93 96 98 99)

PREMISSSES: (2 4 5 6 7 11 13 15 16 17 21 32 52 74)

Une régularité ressort de l'examen: la pondération n'augmente pas nécessairement la liste des catégories mais elle accentue toujours les contrastes, et ceci en étalant la liste vers l'extrémité gauche (début du texte) pour les dénouements, et vers l'extrémité droite (fin du texte) pour les prémisses. Une telle procédure permet de dépister des événements grammaticaux qui autrement resteraient cachés.

## 2) La distribution des présuppositions

Les déterminations nominales reçoivent dans notre grammaire une attention particulière. Nous avons déjà vu qu'elles soulignent un phénomène important de la structuration de surface: ces relations marquent la présence d'éléments présupposés à l'élaboration des groupes verbaux.

Il sera du plus haut niveau d'intérêt pour la description textuelle de savoir si les déterminants des nominaux d'un GP sont présents ailleurs dans le texte, de telle sorte que ces présuppositions du GP soient textuellement résolues, ou si au contraire on déduira de leur absence que certaines présuppositions du GP appartiennent à l'univers de la sémantique extérieure au texte, plutôt qu'à celle du texte décrit. Ces questions trouveront réponse par l'indexation des catégories textuelles suivantes:

a) DARN\*NOMEX\*RESOLU, où DARN\* est lié à (GP ((GN) (D -) (GN ((N1 ((=X))))))), indique le nombre de fois où des déterminants nominaux apparaissent dans la partie de la description située avant l'apparition du GP pointé;

b) DARN\*NOMEX\*ATTENT indique le nombre de fois où des déterminants nominaux apparaissent dans la partie de la DDT située après l'apparition du GP pointé.

### 3) L'organisation thématique

La thématisation textuelle est sans doute la problématique la plus répandue dans l'ensemble des travaux de linguistique de texte. La thématique du texte demeure ce qui intuitivement semble le plus relier les phrases-les unes aux autres. Nous tenterons de cerner comment se repolarise à chacun des segments de la DDT le jeu des appartenances à l'univers thématique.

a) NOT\*NOMEX\*RESOLU, où NOT\* est lié à (GP ((X) (I -) (GN ((N1 ((=X))))))), indique dans quelle mesure les nominaux du thème apparaissent dans l'avant texte;

b) NOT\*NOMEX\*ATTENT... dans quelle mesure les nominaux du thème se retrouvent dans l'après texte.

Ces catégories ont été pondérées par la fréquence relative des nominaux du thème dans l'ensemble de la DDT (NOT\*NOMEX\*TOT). Les valeurs relatives rendent compte des résolutions et attentes thématiques quelle que soit la fréquence des nominaux appartenant au thème. Cela permettra de dépister les nominaux qui, quoique peu nombreux, sont fortement liés à la structure thématique du texte.

La seconde partie de l'expérience est consacrée à l'observation dans le texte de la distribution des nouvelles catégories textuelles indexées.

Le tableau no.1 ci-bas représente les profils conjoints des trois catégories (dénouements (D), prémisses (P), autonomies (A)) reliées au phénomène de la nominalisation. Chacune des rangées de ce tableau renvoie à une tranche de 10% du texte (la première rangée informant du premier 10% du texte, la seconde du second 10%... etc.). Les lettres indiquent le nombre de catégories dépistées dans chacune de ces tranches et les tirets signifient l'absence de données dans une rangée. Ce sont des résultats relatifs à la pre-

## L'exploration de contenu

mière partie du "Discours de la méthode".

Ce tableau montre bien la diminution graduelle des prémisses et l'augmentation des dénouements. On remarquera aussi qu'à la troisième rangée, la plus forte concentration d'autonomies est associée à une absence de prémisses et de dénouements.

(Nominalisation, première partie du texte)

|       |          |    |
|-------|----------|----|
| PPPPP | -        | -  |
| PPPPP | -        | -  |
| -     | -        | AA |
| PPP   | -        | A  |
| PPPP  | D        | -  |
| PPP   | -        | -  |
| -     | DD       | -  |
| -     | DDDD     | A  |
| PP    | DDD      | A  |
| -     | DDDDDDDD | -  |

Tableau no.1

Il peut être intéressant de comparer pour des textes différents et pour des phénomènes différents des tableaux du type présenté ci-haut. Ainsi, les trois catégories (P D A) pour la seconde partie du "Discours de la méthode", étude de la nominalisation:

|       |        |    |
|-------|--------|----|
| PPPPP | -      | A  |
| PPPPP | -      | -  |
| P     | DDD    | AA |
| P     | DDD    | AA |
| -     | DDDD   | A  |
| P     | DD     | A  |
| -     | DDD    | AA |
| P     | DDDD   | AA |
| -     | DDDDD  | A  |
| -     | DDDDDD | A  |

Tableau no.2

Quelle que soit le phénomène étudié, il semble bien que l'on doive s'attendre à ce que les prémisses soient plus nombreuses au début du texte, et les dénouements plus nombreux à la fin. Toutefois la ligne d'inversion, c'est-à-dire l'endroit du texte où les dénouements prennent le pas sur



## L'exploration de contenu

les prémisses, peut être bien différente d'un texte à l'autre ou pour des phénomènes grammaticaux différents. Le tableau no.2 est un bel exemple d'inversion rapide et net, alors que le no.5 (plus bas) offre une ligne lente, et que le no.4 présente une ligne d'inversion plutôt diffuse.

On observera aussi dans chacun des tableaux le rapport entre le nombre des dénouements et le nombre des prémisses, ainsi que celui entre le nombre des autonomies et la somme des trois catégories.

La première partie du "Discours de la méthode" est consacrée à une courte biographie de l'auteur, alors que la seconde partie a l'allure d'un classique traité de philosophie où Descartes présente les principales règles de sa méthode. L'étude des lignes d'inversion ne va pas à l'encontre des intuitions que l'on peut avoir sur l'opposition entre une biographie et un traité de philosophie. Alors que le tableau no.1 présente une ligne d'inversion plutôt tardive, le second tableau se caractérise par une ligne d'inversion plus rapide (survenant près du début du texte). La prédominance du nombre des dénouements sur celui des prémisses dans le second texte renforce cette impression d'un discours systématique ayant l'allure d'un long théorème; s'il y a peu de prémisses, c'est que les attentes nominales sont concentrées dans les mêmes phrases, et qu'inversement beaucoup plus de phrases servent à la diffusion des résolutions nominales.

Passons maintenant à l'étude comparative (toujours dans les deux premières parties du texte) des catégories textuelles relatives au phénomène de la présupposition.

Organisation textuelle des présuppositions:  
première partie:                      seconde partie:

|     |      |     |
|-----|------|-----|
| PPP | -    | -   |
| PP  | -    | A   |
| -   | -    | AA  |
| -   | -    | A   |
| PPP | D    | -   |
| P   | -    | -   |
| P   | -    | A   |
| -   | DD   | AAA |
| P   | D    | AA  |
| -   | DDDD | -   |

Tableau no.3

|    |      |     |
|----|------|-----|
| PP | D    | AAA |
| -  | -    | A   |
| P  | D    | AAA |
| -  | D    | -   |
| -  | DD   | AA  |
| P  | D    | AA  |
| -  | D    | A   |
| -  | D    | -   |
| -  | DDD  | -   |
| -  | DDDD | A   |

Tableau no.4

On interprétera dès lors la prédominance du nombre de dénouements sur celui des prémisses comme une marque de rigueur discursive: l'auteur ne se permettant pas d'introduire des présuppositions sans les avoir préalablement discutées. On soulignera ici la très forte prédominance des dénouements sur les prémisses dans le cas du texte philosophique (seconde partie), contrairement à la prédominance inverse dans le texte biographique (première partie).

Organisation textuelle de l'univers thématique:  
première partie:                      seconde partie:

Tableau no.5Tableau no.6

Ce qui ressort surtout de l'examen des tableaux no.5 et no.6 a trait à la différence entre les quantités de catégories Autonomies indexées. La seconde partie se caractérise par une forte prédominance à cet égard. Cela signifie que les nominaux des thèmes sont des éléments qui n'apparaissent pas ailleurs que dans la phrase où ils sont composés. On doit donc déduire que l'enchaînement thématique s'effectue par le biais des propos plutôt que par celui des thèmes.

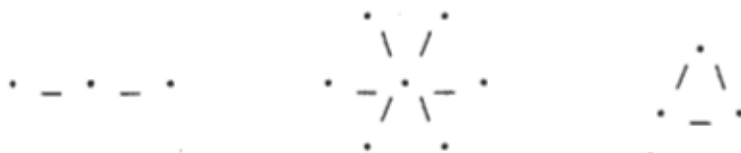
Pour les trois phénomènes grammaticaux (nominalisation, présupposition et thématique), une différence persiste quant à l'importance de l'indexation totale des catégories. La somme des P, D et A de chacun des tableaux est très semblable pour un phénomène donné, quelle que soit la partie du texte, et elle est par ailleurs très différente d'un phénomène à l'autre. Cette régularité semble donc se rapporter à une structure commune aux textes.

Ici encore, nous ne pouvons présenter de conclusions définitives, et nous proposons plutôt de classer les résultats de ces exercices comme des schémas de recherches ultérieures, recherches plus empiriques en ce sens que de nombreux textes seraient alors soumis au même canevas expérimental.

### 3.6 Le dépistage automatique des paradigmes

La fonction LEXIFORME permet la construction automatique des EXFAL. On sait que cette structure d'information fut élaborée pour faciliter la rétention des réseaux paradigmatiques pouvant être dépistés sur les textes décrits par les automates Déredec. Les EXFAL lorsqu'elles sont construites par LEXIFORME sont des structures de graphe comprenant une expression atomique comme sommet, puis d'autres expressions atomiques reliées à celle-ci par un ou plusieurs modèles d'exploration, et encore d'autres expressions atomiques reliées aux dernières, et ainsi de suite... la construction en chaîne ne s'arrêtant que lorsqu'une expression atomique déjà dépistée réapparaît de nouveau, ou lorsqu'un seuil de récursion préalablement fixé a été atteint.

Des modes récursifs de construction de graphes ont déjà été proposés dans le cadre d'expériences en recherche documentaire (cf. Spark Jones 1971). Denis F.M. (1975) rapporte trois de ces modes systématiques d'exploration: la chaîne, l'étoile et le cliqué.



La chaîne est construite à partir d'une forme lexicale donnée prise comme premier sommet, en la reliant par un arc à la forme lexicale avec laquelle elle est le plus corrélée; à partir de celle-là, prise comme deuxième sommet, on recommence la procédure et à son tour on la relie par un arc à la forme lexicale la plus corrélée, ainsi de suite. La procédure s'arrête lorsqu'une forme lexicale identifiée comme sommet potentiel apparaît antérieurement comme sommet de la chaîne...

## L'exploration de contenu

...L'étoile est construite en sélectionnant les h formes lexicales les plus corrélées avec une forme lexicale donnée et en les reliant tous par un arc avec cette forme lexicale...

...Le clique est un graphe où il n'y a aucun sommet qui ne soit relié à chacun des autres sommets. Dans un tel graphe apparaîtront les formes lexicales qui à la fois sont corrélées avec une forme lexicale donnée et entre elles deux à deux... (Denis F.M. 1975, pages 64 et 65).

On entend ici par corrélation entre deux lexèmes une affinité statistique donnée pour le partage de contextes communs c'est-à-dire une certaine mesure du "nombre de fois" où ces deux lexèmes se retrouvent dans les mêmes segments du texte. La principale différence entre ces voies d'exploration et celles permises par LEXIFORME survient de l'explosion de la notion de "proximité lexicale" comme pilier définitionnel du concept d'investigation contextuelle. Les contextes représentés par les arcs des graphes sont définis, dans le cas des EXFAL, comme autant de modèles différents d'exploration des structures descriptives indexées par les automates Déredec. Par ailleurs, on notera:

a) sur le plan de la comparaison des modes de récursion, les EXFAL LEXIFORME sont des "étoiles" où chaque noeud peut lui-même devenir le noeud central d'une nouvelle étoile; il s'agit en quelque sorte d'une "étoile en chaîne";

b) les arcs des EXFAL sont (si désiré) orientés; dans une EXFAL, un arc est un modèle d'exploration où deux positions sont différenciées, celle occupée par l'expression atomique qui constitue l'une des contraintes du modèle, et celle occupée par l'expression atomique que l'on désire voir être dépistée par le modèle; lorsque les syntagmes qui contiennent les deux positions sont reliés par une RDC, il apparaît extrêmement difficile d'exiger l'obtention de "cliques" (ces réseaux où toutes les expressions atomiques seraient reliées entre elles deux à deux); toutefois, de façon plus générale, si l'on désire obtenir des étoiles plus restrictives on peut évidemment appliquer LEXIFORME sur des

## L'exploration de contenu

explorations de texte préalablement constituées, explorations relatives aux contraintes souhaitées;

c) il peut y avoir dans une même EXFAL des arcs de nature différente; il y en aura en fait autant que de modèles d'exploration proposés comme arguments à LEXIFORME; on retrouve ces modèles à chaque nouvel embranchement.

Un paradigme de la détermination nominale

Dans l'exemple qui suit, LEXIFORME a été appliqué sur l'ensemble du "Discours de la Méthode" décrit par notre grammaire de surface. Une EXFAL a été contruite pour chacune des huit expressions atomiques les plus déterminées du texte. L'EXFAL renvoie à la chaîne des déterminations: déterminé <--- déterminant, c'est-à-dire qu'on cherche pour l'expression atomique de départ (:PATRON) les expressions atomiques qui dans le texte occupent la position =X du modèle:

```
DETER* (GN ((GN :PATRON)(D -)(GN ((N1 ((=X)))))))
```

Pour les résultats qui suivent, la récursion ne s'arrête que si une expression atomique déjà dépistée l'est de nouveau (dans un même graphe), ou si évidemment une expression atomique ne reçoit aucune détermination.

```
("sang" (DETER* ("veines" (DETER* ("corps")))))
("sens" (DETER* ("vue")))
("connaissance" (DETER* ("causes"))
  (DETER* ("âme" (DETER* ("bêtes"))))
  (DETER* ("dieu"))
  (DETER* ("vérité" (DETER* ("foi")))))
("moyen" (DETER* ("écoles"))(DETER* ("disputes")))
("vérité" (DETER* ("foi")))
("nature"
  (DETER* ("expérience" (DETER* ("chirurgiens"))))
  (DETER* ("sang" (DETER* ("veines" (DETER* ("corps")))))))
```

Ainsi par exemple "sang" fut déterminé par "veines" et ce dernier mot fut déterminé quelque part dans le texte par "corps" (le sang des veines... les veines du corps).

L'examen de ces paradigmes montre bien comment l'investissement des relations de dépendance contextuelle permet une plus grande précision dans l'interprétation des graphes que ne le font les études sur la distribution des co-occurrences. Que les mots "âme" et "bêtes" eussent été dépistés comme fortement co-occurents dans des mêmes segments d'un texte aurait déjà sûrement permis de dévoiler quelque peu la problématique de ce dernier; qu'une seule relation de détermination nominale (l'âme des bêtes) soit dépistée permet de donner beaucoup plus de précision à l'interprétation de





On remarquera dans ce tableau que les lexèmes qui sont reliés entre eux deux-à-deux soit directement (veine  $\Leftrightarrow$  artère, artère  $\Leftrightarrow$  coeur), soit indirectement ("sang" et "coeur" en passant par les relations sang  $\rightarrow$  artère et artère  $\rightarrow$  coeur), et qui en quelque sorte forment une figure rapprochée du clique, constituent pour le lexème sommet du graphe ("sang") un véritable résumé du traitement sémantique réservé à ce lexème dans le texte, à savoir la question de sa circulation dans le "coeur" via les "artères" et les "veines".

## Appendice au chapitre 3.

On présente ici l'automate FDT utilisé pour l'indexation des catégories textuelles. Cet automate construit et indexe les catégories Attente, Résolution et Autonomie.

```
(FDT
(S1 (PHI S2 D) (X S1 G))
(S2 (C32 (RETURN))
(X (PRODEC NIL* NOM* ATTENT)
  (PRODEC NIL* NOM* RESOLU)
  (PRODEC NOM* NOMEX* ATTENT)
  (PONDEC (NOM* NOMEX* ATTENT*REL)
    (NIL*NOM*ATTENT NOM*NOMEX*ATTENT))
  (PRODEC NOM* NOMEX* RESOLU)
  (PONDEC (NOM* NOMEX* RESOLU*REL)
    (NIL*NOM*RESOLU NOM*NOMEX*RESOLU))
  (FOUDECI NOM* NOM-CONTEX* AUTONO-NOM* 122)
  (PRODEC DARN* NOMEX* RESOLU)
  (PRODEC DARN* NOMEX* ATTENT)
  (FOUDECI DARN* PRE-CONTEX* AUTONO-PRE* 122)
  (PRODEC NOT* NOMEX* TOT)
  (PRODEC NOT* NOMEX* ATTENT)
  (PONDEC (NOT NOMEX* ATTENT*REL)
    (NOT*NOMEX*TOT NOT*NOMEX*ATTENT))
  (PRODEC NOT* NOMEX* RESOLU)
  (PONDEC (NOT* NOMEX* RESOLU*REL)
    (NOT*NOMEX*TOT NOT*NOMEX*RESOLU))
  (FOUDECI NOT* THE-CONTEX* AUTONO-THE* 122)
  S2 D)))
```

où les modèles sont liés aux valeurs:

```
NOM-CONTEX* (=X ((NOM*NOMEX*RESOLU)) ((NOM*NOMEX*ATTENT)))
PRE-CONTEX* (=X ((DARN*NOMEX*RESOLU)) ((DARN*NOMEX*ATTENT)))
THE-CONTEX* (=X ((NOT*NOMEX*RESOLU)) ((NOT*NOMEX*ATTENT)))
NOM*..... (=N1)
NOMEX*..... (=GN :PATRON)
DARN*.... (GN ((GN) (D -) (GN ((N1 ((=X))))))
NOT*..... (GP ((X) (I +) (GN ((N1 ((=X))))))
```

## L'exploration de contenu

Fonctions SIMULECART programmées pour l'indexation des catégories textuelles DENOUEMENT et PREMISSE:

```
(SIMULECART MOY NOMI* NOM*NOMEX*ATTENT
                                NOM*NOMEX*RESOLU (<NFE>) (<NFS>) 122)
(SIMULECART MOY P-NOMI* NOM*NOMEX*ATTENT*REL
                                NOM*NOMEX*RESOLU*REL
                                (<NFE>) (<NFS>) 122)
(SIMULECART MOY PRE* DARN*NOMEX*ATTENT
                                DARN*NOMEX*RESOLU (<NFE>) (<NFS>) 122)
(SIMULECART MOY THE* NOT*NOMEX*ATTENT
                                NOT*NOMEX*RESOLU (<NFE>) (<NFS>) 122)
(SIMULECART MOY P-THE* NOT*NOMEX*ATTENT*REL
                                NOT*NOMEX*RESOLU*REL
                                (<NFE>) (<NFS>) 122)
```

dans tous ces exercices, 122 est le nombre approximatif de phrases dans chacune des deux premières parties du texte (prises individuellement).

## CHAPITRE 4

### L'apprenti-lecteur (APLEC)

#### 4.1 Systèmes "top-down" VS systèmes "bottom-up"

Une forte tension secoue actuellement les milieux de la recherche en linguistique computationnelle. Elle origine d'une confrontation entre deux options.

Une première favorise le développement d'analyseurs syntaxiques et morphologiques pouvant être appliqués sur de larges corpus. Dans cette option, si l'on ne se satisfait pas d'une description incomplète, les problèmes de sémantique (qui ne manquent pas de se poser lorsque la question analysée est aussi complexe que la simulation d'une traduction, ou la mise en place d'un système questions/réponses) sont résolus soit de façon apriorique, par la construction préalable d'un sous-langage auquel doit obligatoirement appartenir le corpus étudié, soit par l'adjonction d'une composante interactive qui permet à un opérateur humain de résoudre de façon ad hoc les impasses rencontrées à certaines étapes de la description. Le projet TAUM (Traduction Automatique Université de Montréal) fonctionne sous la première modalité; TAUM-METEO et TAUM-AVIATION sont deux exemples de systèmes où la grande majorité des décisions sémantiques sont prises hors-algorithme au moment de la construction du sous-langage. Le projet de traduction assistée de l'université Brigham Young (Melby 1978) fonctionne sous la seconde modalité, l'algorithme peut traduire plusieurs types de corpus mais plusieurs décisions (sémantiques) sont prises par un opérateur humain à certains arrêts du programme.

Les tenants de la seconde option proposent plutôt d'aborder directement le problème de la représentation sémantique. Ici les différentes sous-options tiennent principalement au choix des structures d'information retenues pour l'organisation des données (les ensembles sémiques, les formules, les réseaux sémantiques, les représentations inter-

lingua, les frames...). On appelle habituellement "top-down" ces systèmes où se trouve parachutée une batterie sémantique, une conception du monde; alors que l'on réserve l'appellation "bottom-up" à ces autres systèmes où les décisions sémantiques sont totalement absentes ou prises en dehors de l'algorithme.

Il semble jusqu'à maintenant que la simulation de fonctions complexes des langues naturelles (LN) telles la traduction automatique complète, la mise au point de systèmes questions/réponses intelligents ou encore la production de paraphrases ou de déductions logiques, ait été accomplie par des systèmes "top-down", systèmes habituellement présentés sous l'étiquette "intelligence artificielle".

Toutefois, et il apparaît que c'est là le prix imposé à l'obtention de résultats qualifiés, les systèmes "top-down" obligent presque inévitablement une réduction draconienne du corpus. Celle-ci survient quelquefois pour des raisons purement pratiques; le système de Y. Wilks par exemple peut traiter théoriquement des corpus étendus, mais la représentation sémantique que doit recevoir hors contexte chacun des lexèmes est si importante qu'on ne peut penser manipuler plus de quelques paragraphes à la fois, la routine d'indexation manuelle exigeant beaucoup trop du temps de l'utilisateur. Dans d'autres systèmes "top-down", celui du célèbre Jeu de Blocs de Winograd par exemple, la réduction est en quelque sorte inhérente à la conception même de l'algorithme, elle résulte de l'osmose trafiquée entre la sémantique de l'univers textuel analysé et celle étalée dans la batterie descriptive du système; la description de chacun des lexèmes n'étant valide que pour le texte étudié et non pour l'ensemble de la langue, on ne peut penser manipuler un corpus d'envergure.

Ainsi il semble bien que le principal problème associé au traitement des LN demeure un problème de gestion. L'ensemble des relations signifiantes déposées entre les mots d'un texte représente une quantité si grande d'information que tout traitement computationnel de cet ensemble doit s'accommoder d'une perspective réductionniste. Cette réduction portera ou bien sur le corpus analysé (systèmes "top-down"), ou bien sur la batterie sémantique engagée dans l'algorithme (systèmes "bottom-up").

Un troisième type de système laisse entrevoir une fissure au dilemme réductionniste: il s'agit de machines basées sur des principes d'apprentissage. Au point de départ tout se passe dans ces systèmes comme dans certains systèmes "bottom-up", c'est-à-dire qu'aucune contrainte n'est appliquée sur le choix du corpus ou sur sa longueur, la description sémantique impliquée étant nulle ou minime. Par la suite l'automate tente la résolution d'une procédure, il peut s'agir de répondre à une question, de produire une paraphrase, de traduire une séquence... s'il échoue dans sa tentative, il communique avec l'utilisateur, lui diagnostique la difficulté et attend son aide; ce dernier peut alors fournir une catégorie, une relation de dépendance contextuelle, une association sémantique...etc. Cette information pourrait être valide pour toute la langue, mais sera le plus souvent pertinente dans le contexte singulier du texte analysé.

#### 4.2 APLEC

APLEC (apprenti-lecteur) est une extension du logiciel Déredec consacrée à l'exploration de contenu augmentée de procédures d'apprentissage. APLEC est un automate Déredec fortement inspiré des principes de programmation automatique sensible au contexte (la procédure PASC présentée en introduction).

Pour toute grammaire de texte (GDT) soumise à son attention par un usager, APLEC construit un système d'investigation questions/réponses.

La GDT que l'on veut soumettre à APLEC doit d'abord avoir été appliquée sur le texte dont on désire interroger le contenu. Une fois cette démarche effectuée, il ne reste plus pour l'usager qu'à fournir une série de modèles d'exploration qu'il juge représentatifs des structures indexées par sa grammaire.

Une séance avec APLEC a l'aspect d'une suite de questions posées par l'usager et auxquelles tente de répondre l'apprenti-lecteur. Les questions sont formulées dans la langue du texte. Sur chacune d'entre elles, la GDT de l'usager est appliquée et APLEC essaie par la suite de caractériser au mieux la description obtenue, ceci au moyen des modèles d'exploration préalablement proposés. Ces explorations servent à construire de nouveaux modèles qu'APLEC parachute sur le texte (procédure PASC). Les segments du texte qui réalisent ces modèles deviennent les réponses appropriées aux questions posées.

Si l'un des modèles soumis par l'usager ne peut servir à fournir directement une réponse, l'apprenti-lecteur cherche quand même à dépister les éléments du texte qui occupent une position analogue à l'élément problématique de la question, puis entre en communication avec l'utilisateur pour lui demander de confirmer si possible l'existence d'une équivalence sémantique entre cet élément et l'un ou l'autre des éléments dépistés sur le texte. Dans l'affirmative, les segments du texte qui contiennent les équivalents sémantiques dans les positions structurales exigées seront considérés comme appropriés et la suite des tests (modèles) leur sera appliquée.

## L'apprenti-lecteur

Les équivalents sémantiques peuvent être appris par APLEC (d'où son nom), c'est-à-dire qu'à la demande de l'utilisateur (voir plus bas la fonction MERCI) APLEC redécrit le texte en y inscrivant les équivalences sémantiques au moyen de la construction automatique d'EXFAL (exemple: ("serviette" (EQUI\* ("sac")))). Cet apprentissage peut être contextualisé ou pas, le contexte de l'équivalence sémantique étant défini par certains mots de la question (ceux dépistés par les modèles de l'utilisateur).

L'impression générale qui se dégage de ce système d'analyse de contenu est celle d'un "robot" personnalisé fonctionnant avec les schémas sémantiques (l'ensemble des équivalences sémantiques) de son utilisateur; schémas qui se construisent au fur et à mesure des conversations avec le texte. Ainsi, contrairement à ces systèmes de questions/réponses où se trouve déterminé à l'avance l'ensemble complet des relations sémantiques admises, et par le fait même le type d'univers textuel sur lequel il sera possible de le projeter, l'apprenti-lecteur permet de définir pour une GDT donnée un système de questions/réponses qui soit à même de construire les seules associations sémantiques dont il aura besoin pour améliorer son travail de fouille.

Il est à souligner de plus que l'étude de la liste des équivalents sémantiques renseignera l'utilisateur sur les limites sémantiques d'exploration de sa grammaire. APLEC constitue par là un moyen rapide de connaître ces dernières.

Notes:

1) S'il le préfère, l'utilisateur peut laisser à APLEC le soin de caractériser la description produite par la CDT sur la question. Cette caractéristique souligne un degré plus élevé d'automatisation, mais par ailleurs un extrémisme plus grand dans la détermination des contraintes. C'est qu'un seul modèle d'exploration sera construit à partir de l'étude de la question. Les caractéristiques structurales de celle-ci deviendront autant de PATRONS dans ce modèle, PATRONS dont la suite pourra être évaluée conjonctivement ou disjonctivement sur le texte, selon la valeur qu'aura auparavant donnée l'utilisateur à la variable CONJUNCTION (T ou NIL). Dans le premier cas (T), toutes les caractéristiques structurales doivent se retrouver dans un seul segment du texte pour que celui-ci soit considéré comme une réponse, alors que dans le second cas l'une ou l'autre de ces caractéristiques suffira à dépister une réponse.



2) Lorsque l'utilisateur décide plutôt de fournir lui-même la suite de modèles, il doit lier la variable MODUSA à la liste des noms des modèles d'exploration (SETQ MODUSA @(MOD1\* MOD2\* MOD3\*..)) et lier par ailleurs chacun de ces modèles à son contenu particulier. Ici une contrainte supplémentaire a été retenue, tous ces modèles ne doivent dépister que des expressions atomiques.

Les modèles seront testés dans un ordre prioritaire défini de gauche à droite. C'est-à-dire qu'un modèle dans la liste ne pourra toujours dépister qu'un sous-ensemble de la partie de texte jusqu'alors dépistée par le ou les modèles situés plus à gauche dans la liste. L'utilisateur peut par ailleurs exiger un minimum d'intersections entre les sous-ensembles construits, pour qu'il y ait réponse à ses questions, ceci, en donnant une valeur à la variable INTERSEC; le nombre d'intersections construites sera toujours au moins égal à la valeur donnée à INTERSEC. Ainsi si INTERSEC est lié à 2, au moins deux intersections devront avoir été dépistées pour qu'une réponse soit donnée (la première intersection est celle produite par le premier dépistage réussi sur le texte questionné). Si à la suite d'une fouille complète (impliquant tous les modèles de MODUSA) les intersections nécessaires n'ont pas pu être construites, la liste MODUSA est automatiquement transformée, le premier modèle étant déposé à la fin de la liste, le second modèle devenant le premier. Cette procédure sera réappliquée tant que le nombre désiré d'intersections n'aura pas été trouvé ou que la liste ne redeviendra pas ce qu'elle était au point de départ (auquel cas évidemment, aucune réponse n'est fournie).

INTERSEC se trouvera automatiquement réduit (de 1) à chaque fois qu'un modèle de la série proposée (MODUSA) ne se réalise pas sur la question posée. Ainsi par exemple si INTERSEC est lié à 3 et que seulement 3 des 4 modèles proposés s'appliquent sur la question, alors deux intersections seront exigées avant qu'une réponse soit fournie.

3) Chacune des questions doit se terminer par un point d'interrogation ("?"). Si l'utilisateur la termine par un point ("."), APLEC ira tout simplement déposer le contenu de cette question (une fois transformée par la GDT) à la fin du texte questionné. APLEC s'emploie de concert avec la fonction USAGER. Cette fonction permet d'appliquer une série d'automates (une GDT) sur une seule séquence à la fois. Au début d'une séance avec APLEC, l'utilisateur devra lier la variable QUESFIC (voir la description de la fonction USAGER) à la liste constituée du nom du premier automate de sa GDT, et du

## L'apprenti-lecteur

nom du fichier sur lequel est déposée la DDT qu'il désire analyser.

4) APLEC est en fait le nom d'un automate que devra faire appeler le dernier automate de la GDT de l'utilisateur, ceci juste avant la programmation du (RETURN) final. Il suffira d'y mettre la commande (E APLEC S1 G); exemple:  
(S8 (POIN (E APLEC S1 G) (RETURN))).

## 4.3 Premier exemple d'utilisation d'APLEC

La GDT utilisée a trait à la structuration de surface des phrases françaises (chapitre 3), augmentée d'une procédure de lemmatisation (il s'agit de ramener à une même forme canonique les items lexicaux qui dans un texte appartiennent à une même famille (radical), et ceci, selon des règles simples de troncature par la droite de ces items). Cet algorithme est appliqué automatiquement lorsque la variable LEMALGO a T pour valeur.

Voyons un peu l'aspect d'un dialogue constitué à propos de la lecture d'un texte court, décrit par cette grammaire, volontairement redondant afin de mieux exhiber les facultés de discernation de notre apprenti-lecteur.

.TYPE TEXTEQ.TXT

"Les" "enfants" "boivent" "du" "lait" "."  
 "Les" "hommes" "boivent" "du" "café" "."  
 "Marie" "se" "lève" "tard" "."  
 "Les" "enfants" "se" "lèvent" "tôt" "."

.RUN DEREDEC

\*(SETQ QUESFIC @(AUTOQ (TEXTEQ.DES)))

\*(LIRFIC (TEXTEQ.DES))

```
(GP NIL
  (GN ((I 1 +))
    (D12 NIL (T11 NIL "LES"))
    (N1 NIL "ENFANTS"))
  (GV1 ((I -1 -))
    (GV1 ((P1 2 -))
      (V1 NIL "BOIVENT"))
    (C21 NIL "DU")
    (GN ((P1 -2 +))
      (N1 NIL "LAIT"))))
(C32 NIL ".")
(GP NIL
  (GN ((I 1 +))
    (D12 NIL (T11 NIL "LES"))
    (N1 NIL "HOMMES"))
  (GV1 ((I -1 -))
    (GV1 ((P1 2 -))
      (V1 NIL "BOIVENT"))
    (C21 NIL "DU"))
```

## L'apprenti-lecteur

```

      (GN ((P1 -2 +))
        (N1 NIL "CAFE"))))
(C32 NIL ".")
(GP NIL
  (GN ((I 1 +))
    (N1 NIL "MARIE"))
  (GV1 ((I -1 -))
    (N212 NIL "SE")
    (GV1 NIL
      (V1 NIL "LEVE")
      (GD11 NIL (D11 NIL "TARD")))))
(C32 NIL ".")
(GP NIL
  (GN ((I 1 +))
    (D12 NIL (T11 NIL "LES"))
    (N1 NIL "ENFANTS"))
  (GV1 ((I -1 -))
    (N212 NIL "SE")
    (GV1 NIL
      (V1 NIL "LEVENT")
      (GD11 NIL (D11 NIL "TOT")))))
(C32 NIL ".")

```

```

*MODUSA
(PRO* THE*)

```

```

*PRO*
(GP ((X) (I +) (GV ((V1 ((=X)))))))

```

```

*THE*
(GP ((X) (I -) (GN ((N1 ((=X)))))))

```

```

*(USAGER ("Marie" "boit" "elle" "?"))
Pouvez-vous fournir un équivalent sémantique à l'expression:
"Marie" parmi les expressions: ("enfant" "homme") ?
*"enfant"
--"Les" "enfants" "boivent" "du" "lait"
Autre-question?

```

```

*(MERCI)

```

## L'apprenti-lecteur

```

*(USAGER ("Marie" "boit" "elle" "du" "lait" "?"))
--"Les" "enfants" "boivent" "du" "lait"
Autre-question?

```

```

*(USAGER ("Marie" "se" "lève" "t" "elle" "tôt" "?"))
--"Marie" "se" "lève" "tard"
Autre-question?

```

.....MERCİ est la fonction utilisée pour l'apprentissage des équivalents sémantiques. Cet apprentissage peut être contextualisé sur demande. C'est-à-dire que pour chaque équivalence sémantique proposée par l'utilisateur, APLEC mémorise le contexte dans lequel se fait cette proposition; ce dernier est constitué de certains mots de la question courante (ceux dépistés par les modèles de MODUSA; ceci a pour conséquence que la finesse d'apprentissage se définit directement dans les termes de la GDT et des modèles de l'utilisateur). Ainsi dans notre exemple, APLEC a appris pour le mot "enfant" l'équivalent "Marie"; il aura de plus appris que cette équivalence sémantique n'est valable que dans les phrases où l'on retrouve le mot "boire". Sans cette mesure, tous les "enfants" du texte auraient eu pour équivalent "Marie"; on voit à la dernière phrase du texte quel type d'erreur cela aurait entraîné.

L'exemple souligne de plus le caractère personnalisé du système: alors que des équivalences sémantiques sont vraisemblables pour des usagers donnés dans des contextes d'utilisation donnés, il serait aberrant de les déposer dans un dictionnaire apriorique de la langue.

L'utilisateur peut toutefois décider à sa guise du caractère contextualisé de l'apprentissage des équivalents sémantiques, il n'a pour cela qu'à donner une valeur (T ou NIL) à la variable CONTEXTE, ceci préalablement à l'évaluation de (MERCİ); lorsque le Déredéc est chargé en mémoire, CONTEXTE est lié à T (T implique que l'apprentissage soit contextualisé).

## 4.4 Second exemple d'utilisation d'APLEC

La GDT utilisée a été proposée par Claude Panaccio du département de philosophie de l'UQTR (Panaccio 1978).

Le corpus étudié est composé d'une vingtaine de lettres écrites à des journaux à l'occasion du débat sur la chasse aux bébés phoques dans l'estuaire du Saint-Laurent. L'hypothèse centrale de cette grammaire est que l'on peut par l'indexation de certaines catégories aux mots d'une lettre, et par l'application ultérieure sur ces catégories de règles de composition syntagmatique, réussir à déterminer si l'auteur de la lettre (ou l'auteur d'une question à APLEC) est pour ou contre la chasse aux bébés phoques. Les catégories permettent d'identifier les AGENTS et les ACTES relatifs à l'univers sémantique de la chasse aux phoques ainsi que deux prédicats LOUABLE et CONDAMNABLE que ces éléments ou d'autres lexèmes de la phrase peuvent recevoir. Les règles de composition syntagmatique quant à elles servent à transférer ces cotes axiologiques à des noyaux de plus en plus gros d'une séquence; ainsi, un AGENT qui commet un ACTE LOUABLE sera composé dans un syntagme AGENT-LOUABLE...

La composition syntagmatique se fait préférentiellement de droite à gauche; de plus, si un élément reçoit une cote dans une phrase, il conserve cette cote pour toute la lettre. Nous avons programmé cette grammaire en Dérédec et nous l'avons de plus augmenté de la procédure de lemmatisation ainsi que d'une procédure pour la saisie de certaines relations logiques entre les questions successives posées par l'utilisateur.

Exemple d'un dialogue avec APLEC sur cette grammaire:

.RUN DEREDEC

\*(BONJOUR)

.....cette commande initialise en mémoire les informations apprises par APLEC dans des séances antérieures;

## L'apprenti-lecteur

Bonjour--

Aimeriez-vous donner une opinion sur la chasse aux phoques ?

\*(DECO)

Donnez votre opinion, le paramètre ARGUCO étant lié à T, je m'opposerai à celle-ci!

.....la fonction DECO contrôle la mise à zéro des compteurs logiques associés à chaque conversation; elle doit être appliquée au début de chacune d'entre elles;

.....ARGUCO peut aussi être lié à NIL, auquel cas APLEC fournit pour réponse un argument d'appui. Dans ce cas-ci, il tentera de dépister sur le corpus la ou les séquences qui constituent les meilleurs contre-arguments.

\*(USAGER ("Brigitte" "Bardot" "cache" "des" "intérêts" "peu" "honorables" "?"))

Pouvez-vous fournir un équivalent sémantique à l'expression: "Bardot" parmi les expressions: ("défenseur") ?  
\*"défenseur"

--"Il" "me" "semble" "qu'il" "faudrait" "louer" "ce" "défenseur" "de" "la" "nature" "plutôt" "que" "de" "le" "dénigrer" "en" "prêtant" "à" "son" "action" "des" "mobiles" "purement" "d" "intérêts". Autre question ?

\*(USAGER ("Pourtant" "Bardot" "est" "une" "dame" "bien" "correcte" "?"))

Vous vous contredisez, vous étiez pour et là vous êtes contre ! Etes-vous pour cette chasse oui ou non ?

\*NON

## L'apprenti-lecteur

--"Brigitte" "Bardot" "vient" "de" "se" "couvrir" "de"" "ridicule" "par" "ses" "déclarations" "farfelues" "sur" "la" "chasse" "aux" "bébés" "phoques". Autre question ?

\*(SETQ ARGUCO NIL)

.....commande permettant de changer la valeur d'ARGUCO;

\*(DECO)

Donnez votre opinion, le paramètre ARGUCO étant lié à NIL, je vous donnerai mon appui!

\*(USAGER ("Weber" "agit" "pour" "protéger" "les" "beaux" "phoques" "?"))

Pouvez-vous catégoriser l'expression de l'une des catégories: AGENT ACTE LOUA CONDA FF FP FT ?  
"AGIT"

(APLEC BROKEN)

1: (PROCAT ACTE)

.....le mot "agit" n'apparaît pas dans le corpus et n'a par ailleurs jamais été mentionné dans une conversation antérieure; la catégorie proposée sera retenue et servira dans les séances ultérieures. La fonction CAT (utilisée à la place de PROCAT) aurait permis une catégorisation sans apprentissage, donc valide uniquement pour cette question. FF signifie Forme Fonctionnelle (ex.: me te se...); FP (pour Forme Pleine) catégorise les lexèmes chargés sémantiquement mais qui ne reçoivent pas de catégories spécifiques à l'univers concerné (c'est-à-dire la chasse aux phoques); FT renvoie aux fonctions de transfert des cotes, ainsi les particules de négations (ne pas...) qui lorsque rencontrées dans une séquence inversent la cote déjà attribuée. Trois doubles catégories ont de plus été définies: ACTELOUA, ACTECONDA, et ACTEFT ("massacrer" est par exemple un ACTECONDA).



## L'apprenti-lecteur

Pouvez-vous fournir un équivalent sémantique à l'expression:  
 "phoques" parmi les expressions: ("nanti" "merveille" "ter-  
 re" "maison") ?  
 "merveille"

Pouvez-vous fournir un équivalent sémantique à l'expression:  
 "Weber" parmi les expressions: ("célébrité") ?  
 "célébrité"

--"D'autres" "célébrités" "très" "nantis" "ne" "font"  
 "rien" "pour" "la" "protection" "des" "merveilles" "de"  
 "cette" "terre" "qui" "est" "notre" "maison" "à" "tous" "."  
 Autre question ?

\*(DECO)

\*(USAGER ("Brigitte" "Bardot" "vient" "défendre" "les"  
 "bons" "chasseurs" "?"))

Je crois discerner une contradiction dans le processus éva-  
 luatif de votre question. Je ne puis y répondre. Autre  
 question ?

Une dernière fonction appelée DIALOGUE ordonne à APLEC  
 de considérer la réponse qu'il dépiste comme une nouvelle  
 question. L'utilisateur voit ainsi se dérouler devant lui ques-  
 tions et réponses. Il n'aura qu'à programmer la première  
 question (DIALOGUE remplace alors USAGER) et à fournir les  
 équivalents sémantiques lorsqu'APLEC le lui demande. La  
 procédure ne s'arrête que si une question ne trouve pas ré-  
 ponse dans le corpus. On peut assister à un dialogue où les  
 intervenants sont d'avis opposés (ARGUCO = T) ou de même  
 avis (ARGUCO = NIL).

Dans l'exemple qui suit, lorsque la réponse contient  
 plus d'une phrase, seule la première deviendra la "question"  
 auquel APLEC tentera d'associer une nouvelle réponse.

(DIALOGUE ("Les" "chasseurs" "sont" "des" "sadiques" "assoiffés" "de" "sang" "et" "de" "violence" "?"))

--"Le" "contraste" "est" "frappant" ";" "les" "phoques" "sont" "réellement" "de" "belles" "bêtes" "durant" "les" "premières" "semaines" "de" "leur" "existence" "et" "la" "publicité" "sur" "cette" "chasse" "y" "met" "même" "de" "la" "couleur" "en" "présentant" "toujours" "les" "coulées" "de" "sang" "rouge" "sur" "fond" "blanc" ";" "si" "c" "était" "du" "sang" "de" "loup" ", " "une" "grande" "majorité" "réagirait" "différamment"

--"Mais" "tuer" "n" "est" "il" "pas" "par" "définition" "un" "acte" "barbare" "privant" "un" "être" "vivant" "de" "son" "droit" "le" "plus" "fondamental" ";" "est" "il" "possible" "de" "poser" "un" "tel" "geste" "sans" "violence"

-----contre-réponse:

--"Au" "contraire" "ce" "ne" "sont" "pas" "des" "personnes" "blasées" "de" "la" "vie" "qui" "essaient" "de" "prendre" "la" "part" "des" "phoques" ", " "mais" "bien" "des" "gens" "assez" "sensibles" "pour" "être" "capables" "de" "se" "mettre" "dans" "leur" "peau" "et" "pouvoir" "sensibiliser" "le" "monde" "à" "ce" "problème"

--"Quel" "carnage" ", " "quelle" "monstruosité" "envers" "ces" "petites" "bêtes" "sans" "défense" "qu" "on" "exécute" "à" "coups" "de" "bâton" "pour" "assouvir" "son" "besoin" "de" "tuer" "et" "contenter" "son" "souci" "financier"

--"Je" "m" "asseois" "devant" "la" "télévision" "à" "sept-heures" "pour" "écouter" "les" "nouvelles" ", " "après" "une" "journée" "bien" "remplie" ", " "je" "voulais" "relaxer" "un" "peu" "et" "soudain" "j" "aperçois" "ce" "massacre" ", " "et" "c" "est" "le" "seul" "mot" "possible" "ou" "impossible" ", " "on" "assomme" "ces" "merveilleux" "bébés" "phoques" "blancs" "à" "coups" "de" "masse" ", " "on" "les" "éventre" ", " "le" "sang" "coule" "et" "tout" "le" "monde" ", " "il" "est" "heureux" "il" "est" "content"

--"Personnellement" ", " "je" "suis" "contre" "ce" "massacre"  
", " "ce" "qui" "me" "place" "parmi" "la" "majorité" "cana-  
dienne"

-----

## L'apprenti-lecteur

Pour tous les exemples présentés ci-haut, le paramètre INTERSEC était lié à 3, et MODUSA était lié à (AG\* AC\* FP\*), ces modèles étant respectivement définis de la façon suivante:

(ARG? ((AGENT (((X (X\* (=X)))) ((=X))))))

(ARG? ((ACTE (((X (X\* (=X)))) ((=X)~))))

(ARG? ((FP (((X (X\* (=X)))) ((=X))))))

Les lignes qui suivent présentent un dernier exemple auquel on a ajouté des commentaires pour chacune des étapes de la procédure.

\*(DECO)

Donnez votre opinion, le paramètre ARGUCO étant lié à T, je m'opposerai à celle-ci!

\*(USAGER ("Brigitte" "Bardot" "défend" "de" "nobles" "principles" "?"))

Pouvez-vous fournir un équivalent sémantique à l'expression: "principes" parmi les expressions: ("sensibilité" "causes")?

\* "causes"

--"Alors" "madame" "Bardot" "et" "cie" ",," "mettez" "donc" "vos" "sous" "et" "votre" "sensibilité" "à" "défendre" "des" "causes" "plus" "pressantes". Autre question ?

Première étape: description de la question par la GDT:

## L'apprenti-lecteur

(ARGUCON NIL

(AGENTLOUA NIL

(AGENT NIL "Brigitte")

(AGENT NIL "Bardot")

(ACTELOUA NIL

(ACTE NIL "défend")

(FF NIL "de")

(LOUA NIL "nobles")

(FP NIL "principes"))))

"Bardot" est donc AGENTLOUA. Puisqu'APLEC sait que cet agent s'oppose à la chasse aux phoques (il possède des informations de ce type pour les principaux agents), il cherchera la réponse dans les lettres qu'il a dépistées auparavant comme étant favorables à celle-ci (ARGUCO est lié à T) d'où l'indexation de cette catégorie ARGUCON (ARGUCON pour argument contre).

Seconde étape: application du premier modèle sur la question (AGENT\*). On y dépiste les expressions atomiques "Brigitte" et "Bardot". Un modèle interne est construit autour de ce résultat et son application sur le corpus des lettres favorables à la chasse aux phoques donnera l'ensemble des phrases où apparaît l'expression "Bardot" (sous l'une des formes: "Brigitte", "Bardot", "madame" et certains pronoms "elle" qui ont été désambigués à la main).

Troisième étape: le second modèle (ACTE\*) dépiste l'expression "défend" sur la question; un modèle interne est construit sur ce résultat puis est appliqué sur la réponse courante (le sous-corpus dépisté en la seconde étape) éliminant du même coup les phrases de cette dernière qui ne renferment pas à la fois un AGENT/"Bardot" et un ACTE/"défend".

Quatrième étape: le dernier modèle (FP\*) dépiste sur la question l'expression "principes". Le modèle interne construit sur ce résultat et appliqué sur la réponse courante (étape précédente) n'y dépiste aucun élément. Le modèle FP\* sera alors directement appliqué sur la réponse courante, y dépistant les expressions "sensibilité" et "causes". Une réponse valable (le nombre désiré d'intersections étant atteint) sera fournie à cette étape si l'utilisateur, comme c'est le cas dans notre exemple, indique à APLEC l'existence d'une relation d'équivalence sémantique entre "principes" et "causes". Dans le cas contraire, puisque deux intersections auraient seulement été dépistées, la liste des modèles aurait

## L'apprenti-lecteur

été réorganisée (elle serait devenue (ACTE\* FP\* AGENT\*)) et la recherche d'une réponse aurait été amorcée sur ces nouvelles bases. Si aucun arrangement possible de MODUSA (dans notre exemple il en existe trois) n'avait permis de dépister le nombre désiré d'intersections, alors APLEC aurait annoncé ne pas trouver de réponse. Dans des situations comme celles-là, l'utilisateur peut a) fixer INTERSEC à une valeur moindre, b) changer l'ordre, le nombre ou le contenu des modèles de MODUSA, ou encore se montrer plus souple dans l'attribution des équivalences.

## Bibliographie

Nous avons regroupé dans cette bibliographie différents titres dont certains constituent pour notre recherche des références directes, mais dont la majorité se trouvent plutôt rassemblés pour permettre au lecteur de situer de façon plus générale notre travail au sein de la production en linguistique computationnelle et en intelligence artificielle.

## Liste des principales abréviations utilisées:

- AIJ -- Artificial Intelligence Journal; North Holland Publishing Co. Amsterdam.
- BBN -- Bolt Beranek and Newman Inc. Camb. Mass..
- CACM -- Communications of the Association for Computing Machinery.
- CHTL -- Computer Models of Thought and Language. Schanck and Colby (eds) Freeman W.H., San Francisco, 1973.
- COLING -- Computational Linguistic, Conférences Internationales (67, 69, 71, 73 (Pisa), 76 (Ottawa), 78 (Bergen).
- FJCC -- Fall Joint Computer Conference. Spartan, 1966.
- ICCH -- International Conference on Computing in the Humanities. Chicago (1975), Waterloo (1977).
- ICAL -- International Congress of Applied Linguistics. Stuttgart (1975), Montréal (1978).
- IJCAI -- International Joint Conference on Artificial Intelligence. Washington DC (1969), London (1971), Cambridge (1973, 1975, 1977).
- MI -- Machine Intelligence. Dale E. et Michie D. (eds), American Elsevier Publishing Company; New York.
- SIP -- Semantic Information Processing. Minsky (ed), MIT, Camb. Mass. 1968.
- SONL -- Semantics of Natural Language. Reidel D. Publishing Company, Boston Mass..
- TINLAP -- Theoretical Issues in Natural Language Processing. (An interdisciplinary workshop) Yale Univ. (1975), Univ. of Illinois at Urbana-Champaign (1978).

-----

- Abelson, R.P. Concepts for Representing Mundane Reality in Plans. in Representation and Understanding, Bobrow D. and Collins A. (eds), New-York: Academic Press, 1975.
- Abelson, R.P. Does a Story Understander Need a Point of View. TINLAP-75.
- Abelson, R.P. The Reasoner and the Inferencer don't Talk Much to Each Other. TINLAP-75.

- Aho A.D. and Ullman J.D. The Theory of Parsing Translation and Compiling. vol.1, Prentice-Hall, Englewood Cliff, N. J., 1972.
- Anderson, J.R. Language, Memory and Thought. Hillsdale N.J., Lawrence Erlbaum Associates, 1976.
- Anderson, J.R. The Processing of Referring Expressions within a Semantic Network. TINLAP-78.
- Allen, S. Text-Based Lexicography and Algorithmic Text Analyses. in COLING-76.
- Apostel, L. Philosophy and the Computer. in L.A.S.L.A., no.3, pp.121-141, 1971.
- Bach, E. and Harms, R. T. Nouns and Noun Phrases. in Universals in Linguistic Theory, Chicago: Holt, Rinehart and Winston Inc, 1968.
- Bar-Hillel, Y. The Present Status of Automatic Translation of Languages. in Advances in Computers, vol.1, pp.91-163, 1960.
- Bartsch, R. Syntax and Semantics of Relative Clauses. in Bartsch R., Groenendijk J. and Stokhof M.(eds) Amsterdam Papers in Formal Grammars, The Netherlands: Univ. of Amsterdam, 1976.
- Bates, M. Syntactic Analysis in a Speech Understanding System. in BBN Report no.3116, BBN Camb., Mass., 1975.
- Baudot J., Bourbeau L., Chevalier M., Dansereau J., Isabelle P., Lehrberger J., Poulin G. et Stewart G. Projet Aviation, TAUM. Rapport d'étape, Univ. de Montréal, Mai 1977.
- Becker, J.D. An Information Processing Model of Intermediate-Level Cognition. in Memo AI-119, Stanford Artificial Intelligence Project, Stanford Univ., May 1970.
- Becker, J.D. The Phrasal Lexicon. in TINLAP-75.
- Belnap, N.D. and Steel, T.B. The Logic of Questions and Answers. Yale Uni. Press., New Haven, Conn., 1976.
- Bely N., Borillo A., Siot-Decauville N. et Virbel J. Procédures d'analyse sémantique appliquées à la documentation scientifique. Paris, Gauthier-Villars.
- Berry-Rogghe G.L., Dilger W. et Zifonum G. A Cooperative Deductive Question-Answering System Incorporating User-Defined Heuristics. in COLING-78.
- Bever, T.C. and Ross J.R. Underlying Structures in Discourse. in Proc. of the Conference on Computer-Related Semantic Analysis, Las Vegas, 1965.
- Binnick, R. Generative Semantics and Automatic Translation. in COLING-69.
- Biss K., Chein R., Stahi F. and Weissman S. Semantic Modeling for Deductive Question-Answering. in Proc. of IJCAI-73, Stanford, pp.356-363, August 1973.
- Black, F. A Deductive Question-Answering System. in Semantic Information Processing, Minsky, M.(ed), MIT Press, Camb. Mass., 1968.



- Bledsoe W.W., Boyer R.S. and Henneman W.H. Computer Proofs of Limit Theorems. in Artificial Intelligence, no.3, pp.27-60, 1972.
- Bobrow, D. Natural Language Input for a Computer Problem-Solving System. in SIP, 1968.
- Bobrow, D.G. and Fraser, J.B. An Augmented State Transition Network Analysis Procedure. in IJCAI-69, 1969.
- Bobrow, D.G. A Note on Hash Lenking. in CACM, pp.413-415, 1975.
- Bobrow, D.G. and Collins, A. Representation and Understanding. Academic Press, N.J., 1975.
- Bobrow D.G., Murphy D., Teitelman W. and Hartley A. BBN-LISP System. Bolt Beranek and Newman Inc., Camb., Mass., July 1971.
- Bobrow, D.G. and Winograd, T. An Overview of KRL, a Knowledge Representation Language. in Cognitive Science, 1:1, pp.3-46, January 1977.
- Bobrow D.G., Winograd T. and KRL Research Group; Experience with KRL-0: One Cycle of a Knowledge Representation Language. in Proceedings of IJCAI-77.
- Boilet, C. et Chauché, J. Approches sémantiques pour les modèles d'analyse automatique de langues naturelles. in Colloque Metz 1974.
- Borillo, A. Problèmes d'analyse syntactico-sémantique de constructions interrogatives en français dans le cadre de traitement automatique de questions. in COLING-74.
- Brachman, R.J. What's in a Concept: Structural Foundations for Semantic Network. in COLING-76.
- Brachman, R.J. The Evolution of a Structural Paradigm for Representing Knowledge. in Ph.D. Dissertation, Harvard Univ., 1976.
- Brachman, R.J. A Structural Paradigm for Representing Knowledge. in Tech. Report, no.3605, BBN, 1978.
- Bratley, P. and Pakin, D. A Limited Dictionary for Syntactic Analysis. in MI, U2, pp.173-181, 1968.
- Bruce, B. A Model for Temporal References and its Application in a Question-Answering Program. in Artificial Intelligence Journal, U3, no.1, pp.1-26, 1972.
- Bruce, B.C. Belief Systems and Language Understanding. in BBN Report, no.2973, Camb., Mass., January 1975.
- Bruce, B.C. Case Systems for Natural Language. BBN Report 3010, 1975.
- Bruce, B.C. Generations as a Social Action. in TINLAP-78.
- Burton, R.R. and Woods, W.A. A Compiling System for Augmented Transition Networks. in COLING-76.
- Burton, R.R. Semantic Grammar: a Technique for Efficient Language Understanding in Limited Domains. in Doctoral Dissertation, Information and Computer Science Dept., Univ. of Calif., Calif., 1976.

- Cappelli A., Ferrari G., Moretti L., Prodanof I. and Stock O. An ATN Parser for Italian: Some Experiments. in COLING-78.
- Carbonell, J.R. and Collin, A.M. Natural Semantics in Artificial Intelligence. in IJCAI-73.
- Carbonell J.G., Cullingford R.E. and Gershman A. Knowledge-Based Machine Translation. in Computer Research Report, Yale Univ., New Haven, CT, 1978.
- Carbonell, J.G. Politics: Automated Ideological Reasoning. in Cognitive Science, vol.2, no.1, 1978.
- Carbonell J.G., Cullingford R.E. and Gershman A. Towards Knowledge-Based Machine Translation. in COLING-78.
- Celce-Murcia, M. Paradigms for Sentences Recognition. in UCLA, Dept. of Linguistics, 1972.
- Cerccone, N. Representing Natural Language in Extended Semantic Networks. in Techn. Report, TR 75-11, Dept. of Computing Science, Univ. of Alberta Canada, July 1975.
- Cerccone, N. and Schubert, L.K. Toward a State-Based Conceptual Representation. in Tech. Report, TR 74-19, Dept. of Computing Science, Univ. of Alberta Canada, 1974.
- Chafe, W.L. Creativity in Verbalization as Evidence for Analogic Knowledge. in TINLAP-75.
- Chafe, W.L. Discourse Structure and Human Knowledge. in Language Comprehension and the Acquisition of Knowledge, Carroll, J.B. and Freedle, R.O. (eds), Washington: Winston and Sons, 1972.
- Chafe, W.L. Some Thoughts on Schemata. in TINLAP-75, Univ. of Calif., Berkeley CA.
- Chafe, W.L. Meaning and Structure in Language. Univ. of Chicago Press, Chicago, Illinois, 1970.
- Chauché, J. Le Système SYGMART; un système pour le traitement automatique de texte. in COLING-78.
- Chassagne C., Courtin J., Grandjean E. et Mathieu B. Applications du traitement assisté des langues. in COLING-78.
- Chauché, J. Transducteurs et arborescences. Thèse, Grenoble, 1974.
- Chauché, J. Présentation du système CETA. Doc. CETA G.3100a, Grenoble 1975.
- Chauché, J. Vers un modèle algorithmique pour le traitement automatique des langues naturelles. in COLING-76.
- Charniak, E. Computer Solution of Calculus Word Problems. in IJCAI-69, pp.303-316, 1969.
- Charniak, E. Toward a Model of Children's Story Comprehension. MIT AI Lab, AI-TR-266, PhD Thesis, Camb., Mass., 1972.
- Charniak, E.A. Partial Taxonomy of Knowledge About Actions. in IJCAI-75, 1975.

- Charniak, E. Organization and Inference in a Frame-Like System of Common Sense Knowledge. in TINLAP-75, 1975.
- Charniak, E. A Brief on Case. Working Paper 22, Castagnola, Switzerland: Instituto Per Gli Studi Semantecie Cognitivi, 1975.
- Charniak, E. MALAPROP, an Language Comprehension Program. in IJCAI-77, 1977.
- Charniak, E. With a Spoon in Hand this Must Be the Eating Frame. in TINLAP-78, 1978.
- Chevalier M., Dansereau J. et Poulin G. TAUM-METEO: Description du système. in ICAL-5, TAUM, Univ. de Montréal, Janvier 1978.
- Chomsky, N. Syntactic Structures. Monton, The Hague, 1957.
- Chomsky, N. and Miller, G.A. L'analyse Formelle Des Langues Naturelles. Gauthier-Villars, Paris, 1968.
- Church, A. The Calculi of Lambda Conversion. Princeton NJ: Princeton Univ. Press, 1941.
- Clark, H.H. Bridging. in TINLAP-75, 1975.
- Clarke, J.D. The Resolution of Ambiguities by Computer. in ICCH-3, 1977.
- Clarke, J.D. The Resolution of Semantic and Syntactic Ambiguities Using an Interactive Computer Procedure. in ICAL-4, August 1975.
- Clippenger, J.H. Jr. Considerations for Computational Theories of Speaking: Seven Things Speakers Do. in TINLAP-75, 1975.
- Clippenger, J.H. Jr. Speaking with Many Tongues: Some Problems in Modeling Speakers of Actual Discourse. in TINLAP-78, 1975.
- Cohen, P.A. Prototype Natural Language Understanding System. in Tech. Report, no.64, Dept. of Computer Science, Univ. of Toronto, Ontario, 1974.
- Cohen, P. On Knowing What to Say: Planning Speech Acts. in Tech. Report, no.118, Dept. of Computer Science, Univ. of Toronto, 1978.
- Colby, K.M. Simulations of Belief Systems. in CMTL, 1973.
- Colby K.M., Tesler L. and Enea H. Experiments with a Search Algorithm for the Data Base of a Human Belief Structure. in IJCAI-69, 1970.
- Coles, L. Techniques for Information Retrieval Using an Inferential Question-Answering System with Natural Language Input. SRI Project 8696, Tech. Note 74, Stanf. Calif..
- Coles, L. An On-Line Question-Answering System with Natural Language and Pictorial Input. in Proc. ACM 23rd Natl. Conf., pp.157-167, 1968.
- Collins, A. and Ouilliam, M. How to Make a Language User. in Organisation of Memory, Trilving, E. and Donaldson, W. (eds), Academic Press, N.Y., 1972.

- Collens, A.M. and Warnock, E.H. Semantic Networks. in BBN Report no.2833 and in AI Report no.15, May 1977.
- Collins, A. The Trouble with Memory Distinctions. in TINLAP-75, 1975.
- Collins A., Brown J.S. and Larkin K. Inference in Text Understanding. Center for the Study of Reading, Univ. of Illinois and BBN, 1977.
- Collins, A. Fragments of a Theory of Human Plausible Reasoning. in TINLAP-78, 1978.
- Colmerauer, A. Les systèmes-Q ou un formalisme pour analyser et synthétiser des phrases sur ordinateurs. in Publication Interne, no.43, Dép. d'Informatique, Univ. de Montréal, Septembre 1970.
- Colmerauer, A. et al. TAUM 71. Projet TAUM, Univ. de Montréal, 1971.
- Colmerauer, A. Un Système de communication homme-machine en français. Rapport interne, VER Luniny, Univ. d'Aix-Marseille.
- Conway, M.E. Design of a Separable Transition Diagram Compiler. in Comm. ACM 6,7, pp.396-408, July 1963.
- Courtin, J. Un Analyseur syntaxique interactif pour la communication homme-machine. in COLING-73, 1973.
- Courtin, J. Utilisation des redondances pour l'analyse et le contrôle automatique d'énoncés en langue naturelle. COLING-76.
- Courtin J., Grandjean E. et Veillar G. PIAF: un Système de manipulations de données en langue naturelle. Colloque International de Terminologie, Juin 1976.
- Courtin, J. Algorithmes pour le traitement interactif des langues naturelles. Thèse d'état, Univ. de Grenoble, Oct. 1977.
- Cullingford, R.E. The Users of World Knowledge in Text Understanding. COLING-76.
- Cullingford, R. Organizing World Knowledge for Story Understanding by Computer. Thesis, Dept. of Engineering and Applied Science, Yale Univ., 1977.
- Cresswell, M. Logics and Language. Methuen and Co. Ltd, London, England, 1973.
- Cullingford, R.E. The Users of World Knowledge in Text Understanding. COLING-76.
- Cullingford, R. Organizing World Knowledge for Story Understanding by Computer. Thesis, Dept. of Engineering and Applied Science, Yale Univ., 1977.
- Damerau, F.J Automatic Language Processing. in Annual Review of Information Science and Technology, vol.2, 1976.
- Dascal M. and Maryalet A. A New "Revolution" in Linguistic? Text-Grammars VS Sentence-Grammars. in Theoretical Linguistics 1:195-231.

- Dascal, M. Conversational Relevance. in Journal of Pragmatics, vol.2, pp.309-328, 1977.
- Davey, The Formalisation of Discourse Production. Ph.D. Thesis, Edinburgh Univ., 1974.
- Davies, D.J.M. POPLER: a POP-2 Planner. in MIP-84, School of AI, Edinburgh Univ..
- Davies, D.J.M. Representing Negation in a Planner System. in Proc. of AISB Conference, Univ. of Sussex, July 1974.
- Dejong, G.E. Skimming Newspaper Stories by Computer. in Computer Science Research Report, no.104, Yale Univ., New Haven, CT, 1977.
- Dejong, G.E. Understanding Multi-Event News Issues with Script-Like Data Structures. in COLING-78.
- Denis, F.M. Proposition d'un cadre formel pour la reconnaissance de contenu de texte. Thèse de maîtrise, dépt. de Philosophie, Univ. du Québec à Montréal, Août 1975.
- Dennis J.A., Miller J.P. and Psathas G. An Introduction to the Inquirer II System of Context Analysis. in Computer Studies in the Humanities and Verbal Behavior, vol.3, no.1, 1970.
- Dewar H., Bratley P. and Thorne J. A Program for the Syntactic Analysis of English Sentences. in CACM, vol.12, no.8, pp.476-479, 1969.
- Dobree, N.J.S. Clam: a Computer Language Model. in AJCL Microfiche 49:2, 1976.
- Dresber B.E. and Hornstein N. On Some Supposed Contribution of Artificial Intelligence to the Scientific Study of Language. in Cognition, 4, pp.321-398, 1976.
- Dreyfus, H. What Computers Can't Do. NY Harper and Row, 1972.
- Ducrot, O. Dire et ne pas dire, Principes de sémantique linguistique. Hermann Paris; 1972.
- Earley, J. An Efficient Context-Free Parsing Algorithm. Ph.D. Thesis, Dept. Computer Science, Carnegie-Mellon Univ., Pittsburgh, 1968.
- Earley, J. An Efficient Context-Free Parsing Algorithm. in CACM, vol.13, pp.97-102, 1970.
- Fahlman, S.E. A Planning System for Robot Construction Tasks. in AIJ, vol.5, 1974.
- Fahlman, S.E. A System for Representing and Using Real-World Knowledge. Ph.D. Dissertation, Dept. of Electrical Engineering and Computer Science, MIT, 1977.
- Feigenbaum E.A. and Feldman J. Computer and Thought. Mc Graw Hill Book CO., NY, 1963.
- Feldman J.A., Lips J., Horning J.J. and Rider S. Grammatical Complexity and Inference. in Tech. Report CS125, Com. Science Dept., Stanford Univ., 1969.

- Feldman, J. Bad-Monthing Frames. TINLAP-75.
- Fillmore, C. The Case for Case. in *Universals in Linguistic Theory*, Bach and Harms (eds), Holt, Rinehart and Winston, 1963.
- Fillmore, C. Pragmatics and the Description of Discourse. in *Berkeley Studies in Syntax and Semantics*, vol.1, vol.21, Berkeley, 1974.
- Fillmore, C. The Future of Semantics. in *Berkeley Studies in Syntax and Semantics*, Dept. of Linguistics, Univ. of Calif. Berkeley, 1974.
- Fillmore, C. An Alternative to Checklist Theories of Meaning. in *Proc. of the First Annual Meeting of the Berkeley Linguistics Society*, Cogen et al (eds), Univ. of Calif. Berkeley, 1974.
- Fillmore, C. Scenes and Frames Semantics. Miméo, 1975.
- Findler N. and Chen D. On the Problems of Time, Retrieval of Temporal Relations, Causality and Co-Existence. IJCAI-71.
- Fikes, R.E. Automatic Planning from a Frames Point of View. TINLAP-75.
- Fikes R.E., Hart P.E. and Nilsson H.J. Learning and Executing Generalized Robot Plans. in *AIJ*, vol.3, 1972.
- Firbas, J. On Defining the Theme in Functional Sentence Analysis. in *Travaux Linguistiques de Prague*, vol.1, 1964.
- Firschein O. and Fischler M.A. A Study in Descriptive Representation of Pictorial Data. IJCAI-71.
- Foster, J.M. Automatic Syntactic Analysis. American Elsevier, NY, 1970.
- Friedman, J. A Computer Model of Transformational Grammar. American Elsevier, NY, 1971.
- Friedman J., Moran D.B. and Warren D.S. Evaluating English Sentences with a Logical Model. COLING-78.
- Garfman, H. Dependency Systems and Phrase Structure Systems. in *Information and Control*, vol.8, P.304, 1965.
- Gardin, J. and Syntol, Rutgers Series on Systems for the Intellectual Organisation of Information. Ortandi (ed), New Jersey, 1965.
- Geach, P. Reference and Generality. New York, Cornell Univ. Press, 1962.
- Geens, D. A Venture in Computational Analysis of English Syntax. in *Katholieke Universiteit Te Leuven*, Dept. Linguistiek, 1975.
- Gershman, A. Analysing English Noun Group for their Conceptual Content. in *Comp. Science Research Report*, no.110, Yale Univ., new Haven, CT, 1977.
- Ginsburg, S. The Mathematical Theory of Context-Free Languages. Mc Graw-Hill, NY, 1966.
- Go Guen, J. Concept Representation in Natural and Artificial Languages: Axioms, Extensions and Applications for Fuzzy Sets. in *International Journal of Man-Machine Studies*, vol.6, no.5, pp.513-552, 1974.

- Goldman, N.M. The Boundaries of Language Generation. TINLAP-75.
- Goldman, N. Sentence Paraphrasing from a Conceptual Base. in CACM, vol.18, no.2, Feb. 1975.
- Goldstein, I. and all, Artificial Intelligence, Language and the Study of Knowledge. in Artificial and Natural Language Comprehension, Washington: National Institute of Education, pp.27-78, 1976.
- Grandjean, E. Conception et réalisation d'un dictionnaire pour un analyseur interactif en langues naturelles. in Memoire GNAM, Univ. de Grenoble, 1975.
- Green B., Wolf A., Chomsky C. and Langherty K. Baseball: an Automatic Question-Answerer. in Computers and Thought, Feigenbaum E. and Feldman J. (eds), Mc Graw Hill Birk Company, vol.4, NY, pp.183-208, 1963.
- Green C.C. and Raphael B. Research on Intelligent Question-Answering Systems. in Proceeding of ACM 23rd Nat. Conf., Brandon Systems Press, Princeton, NJ, pp.169-181, 1968.
- Green, C.C. Theorem Proving by Resolution as a Bases for Question-Answering Systems. in MI, vol.4, pp.183-208, 1969.
- Greif I. and Hewitt C. Actor Semantics of Planner-73. in Proc. of ACM SIGLAW-SIGACT Conference, Palo Alto Calif., Jan. 1975.
- Grice, H.P. Logic and Conversation. in Syntax and Semantics: Speech Acts, Cole P. and Morgan J.L. (eds), vol.3, Academic Press, NY, 1975.
- Greines, J.E. Topic Levels. TINLAP-78, 1978.
- Crimes, J.E. The Thread of Discourse. The Hague: Mouton, 1975.
- Grober E. and Loftus, E. F. Semantic Memory: Searching for Attributes Versus Searching for Names. in Memory and Cognition, vol.2, pp.413-416, 1974.
- Cross, M. Méthodes en syntaxe. Hermann, Paris, 1975.
- Cross M. et Lentin A. Notions sur les grammaires formelles. Gauthier-Villars, Coll: Programmation, Paris, 1970.
- Grossman, R.W. Representing the Semantics of Natural Language as Constraint Expressions. in MIT-AI Working Paper 87, Ja. 1975.
- Grosz, B.J. Focussing on Dialog. TINLAP-78.
- Grosz, B.J. The Representation and the Use of Focus in Dialogue Understanding. in SRI Tech. Note 151, Menlo Park, Calif., 1977.
- Gundel, J. The Role of Topic and Comment in Linguistic Theory. Ph.D., Univ.of Texas, 1974.
- Hajicova, E. Negation and Topic VS Comment. in Philologica Pragensia 16, pp.81-93, 1973.



- Hajicova E. and Sgall P. A Semantic Representation of Topic and Focus. COLING-74.
- Hajicova E., Krizek P. and Sgall P. A Generative Approach to Semantics. in *Linguistica Silesiana* 1, pp.17-31, 1975.
- Hajicova E. and Sgall P. Topic and Focus in Transformational Grammar. in *Papers in Linguistics* 8, 1975.
- Hahn, W.V. and al. HAM-RPM: Natural Dialog with an Artificial Partner. in *Proceedings of the AISB Conf. on AI, Hamburg, 1973.*
- Hayes, P.J. Computation and Deduction. in *Proc. of Symposium on the Mathematical Basis of Computation CZECH, Academy of Sciences, 1973.*
- Hays, D.G. Dependency Theory, a Formalism and Some Observations. in *Language*, vol.40, no.4, P.511, 1964.
- Hays, D.G. Introduction to Computational Linguistics. 1st Edn, Mc Donald, London, England, 1967.
- Hays, D.G. Types of Processes on Cognitive Networks. COLING-73.
- Harris, L.R. Natural Language Data Base Query: Using the Data Base Itself as the Definition of World Knowledge and as an Extension of the Dictionary. in *Tech. Report TR-77-2, Math. Dept., Dartmouth College, Hanover, NH, 1977.*
- Harris, Z. Discourse Analysis. in *Language* 28, pp.1-30, 1952.
- Harris, Z. Discourse Analysis Reprints. The Hague: Mouton et CO, 1963.
- Harris, Z. Mathematical Structures of Languages. Intersciences Publishers, NY, 1968.
- Heidorn, G.E. Augmented Phrase Structure Grammars. TI-NLAP-75.
- Heidorn, G.E. Natural Language Inputs to a Simulation Programming System. in *NPS-55HD, Naval Post Graduate School, Monterey, Calif., 1972.*
- Hempel, C. Vagueness and Logic. in *Philosophy of Science*, vol.6, pp.163-180, 1939.
- Heny, F.W. Sentence and Predicate Modifiers in English. in *Syntax and Semantics*, Kimball, J. (ed), vol.2, Seminar Press, NY, pp.217-245, 1973.
- Hesnard A. et Jerbel J. Analyse sémantique d'un champ lexicologique se rapportant à des objets en vue d'une traduction automatique dans un langage formalisé. COLING-73.
- Herskovits, A. The Generation of French from a Semantic Representation. in *Report CS-334, Computer Science Dept., and in Stanford Artificial Intelligence Laboratory Memo AIM-212.*
- Hewitt, C. Description and Theoretical Analysis of PLANNER: a Language for Proving Theorems and Manipulating Models in a Robot. Ph.D. Thesis, MIT, Camb., Mass., 1971.



- Hewitt, C. PLANNER: a Language for Manipulating Models and Proving Theorems in a Robot. IJCAI-69.
- Hewitt, C. and al. Actor Induction and Meta-Evaluation. Conference Record of ACM Symposium on Principles of Programming Language, Boston, Oct. 1973.
- Hewitt, C. Protection and Synchronisation in Actor Systems. in MIT Working Paper, Nov. 1974.
- Hewitt, C. The Semantics of Actions and the Semantics of Truth. IJCAI-75.
- Hewitt, C. Stereotypes as an Actor Approach Towards Solving the Problem of Procedural Attachment in Frame Theories. TINLAP-75.
- Hobbs, J. A Computational Approach to Discours Analysis. in Research Report 76-2, Dept. of Computer Science, City College, City Univ. of NY, 1976.
- Hobbs, J.R. What the Nature of Natural Language tell us about how to make Natural-Language-like Programming Languages More Natural. SIGART-77.
- Hobbs, J.R. Coherence in English Discourse. COLING-78.
- Hoffman, T. Semantic in Aid of Automatic Translation. COLING-76.
- Horning, G.E. A Study of Grammatical Inference. in Tech. Report CS-139, Computer Science Dept., Stanford Univ., 1969.
- Howe W. and Krulee G. A Logic of English Questions. in IBM Research RC3490, Yorktown Heights, NY, 1971.
- Huffman, D.A. Impossible Objects as Nonsense Sentences. in MI, Michie D. and Meltzer B. (eds), vol.6, Edimburgh: Edinburgh Univ. Press, 1972.
- Isabelle P. et all. TAUN-METEO. 5 Congrès de linguistique appliquée, Montréal 78.
- Isard S. and Longuet-Higging H.C. Question-Answering in English. in MI, vol.6, pp.243-254, 1971.
- Jackendoff, R. A System of Semantic Primitives. TINLAP-75.
- Jackendoff, R. An Argument About the Composition of Conceptual Structure. TINLAP-78.
- Jackson, P.D. Introduction to Artificial Intelligence. Petrocelli Books, NY, 1974.
- Jacobs R.A. and Rosenbaum P.S. English Transformational Grammar. Blaisdell Publ., Waltham, Mass., 1968.
- Johnson-Laird, P. Memory for Words. in Nature, 1974.
- Johnson-Laird, P. On Understanding Logically Complex Sentences. in Quaterly Journal of Experimental Psychology, vol.21, pp.1-13, 1969.
- Joloboff, V. Interprétation sémantique d'énoncés en langue naturelle. COLING-78.
- Jordan, S. Learning to Use Contextual Patterns in Language Processing. in Tech. Report no.152, Univ. of Wisconsin, Madison, Wisconsin, 1972.

- Joski A.K. and Rosenschein S.J. A Formalism for Relating Lexical and Pragmatic Information: its Relevance to Recognition and Generation. TINLAP-75 .
- Joski A.K., Kaplan J. and Lee R.M. Approximate Responses from a Data Base Query System: an Application of Inferencing in Natural Language. IJCAI-77 .
- Joski, A.K. A Note on Partial Match of Descriptions: Can One Simultaneously Question (retrieve) and Inform (update)? . TINLAP-78.
- Joski A.K., Kaplan S. and Sag I.A. Cooperative Responses: Why Query Systems Stonewall. COLING-78.
- Kaplan, R.M. ATNs as Psychological Models of Sentence Comprehension. in AI 3:77-100, 1972.
- Kaplan, R.M. A General Syntactic Processor. in Natural Language Processing, Rustin R. (ed), NY: Algorithmics Press, 1973.
- Kaplan, R.M. Transient Processing Load in Relative Clauses. Doctoral Dissertation, Psychology Dept., Harvard Univ., 1974.
- Kaplan, S.J. Cooperative Responses from a Natural Language Data Base Query System: Preliminary Report. in Tech. Report, Dept. of Computer and Information Science, Moore School, Univ. of Pennsylvania, Philadelphia, PA, 1977.
- Kaplan, S.J. Indirect Responses to Loaded Questions. TINLAP-78.
- Kaplan, R. Augmented Transition Networks as a Model of Sentence Comprehension. in AIJ, vol.3, no.2, pp.77-100, 1972.
- Karttunen, L. Discourse Referents. in Syntax and Semantics, Mc Cawley, J. (ed), vol.7, New York Academic Press, 1976.
- Karttunen L. and Peters S. Requiem for Presupposition. in Third Annual Meeting of the Berkeley Linguistics Society, Berkeley, Calif., 1977.
- Karttunen, L. Whichever Antecedent. Squib Anthology, Chicago Linguistics Society, 1977.
- Katz J. and Fodor J. The Structure of Semantic Theory. in The Structure of Language, Fodor J. and Katz J. (eds), Prentice-Hall, Englewood Cliffs, New-Jersey, 1964.
- Katz, J. Semantic Theory. New York: Harper Row, 1972.
- Kay, K. Experiments with a Powerful Parser. in RN-5452-PR, The Rand Corporation, Santa Monica, Calif., 1967.
- Kay M. and Martin G.R. The Mind System: Structure of the Semantic File. in RM-6265/2 PR, The Rand Corporation, Santa Monica, 1970.
- Kay, M. The Mind System. in Natural Language Processing, Rustin R. (ed), Courant Computer Science Symposium 8, Dec. 1971.

- Kay, M. Syntactic Processing and Functional Sentence Perspective. in TINLAP-75.
- Kellog, C.H. On-Live Translation of Natural Language Questions Into Artificial Language Queries. in SP-2827/000/00 SDC, Santa Monica, 1967.
- Kellog, C.H. A Natural Language Compiler for On-Live Data Management. in Proc. of AFIPS, FJCC, vol.33, AFIPS Press, Montvale, New Jersey, pp.473-493, 1968.
- Keenan E.L. and Hull R.D. The Logical Presuppositions of Questions and Answers. in Prasuppositionen in Philosophie Und Linguistik, Pctofi and Frank (eds), Albenaun Verlag, Frankfurt, 1973.
- Keenan E. and Scheffelen B. Topic as a Discourse Notion. in Subject and Topic, Li (ed), Academic Press, 1976.
- Kempson, R.M. Presupposition and the Delimitation of Semantics. Camb., Camb. Univ. Press, 1975.
- Kittredge R. et al. TAUM. Projet TAUM, Univ. de Montréal, 1973.
- Kittredge, R. Textual Cohesion within Sublanguages: Implications for Automatic Analysis and Synthesis. COLING-78.
- Klein, S. Automatic Paraphrasing in Essay Format. in Mechanical Translation, vol.8, no.2 et 3, 1965.
- Klein, S. Control of Style with a Generative Grammar. in Language, vol.41, no.4, 1965.
- Klein, S. Current Research in the Computer Simulation of Historical Change in Language. in Actes Du 10 ième Congrès Inter. Des Linguistes, Bucarest, 1967.
- Klein, S. Computer Simulation of Language Contact Models. in UWCS Tech. Report/67, Also in Proc. SECOL 8 Conf., Georgetown Univ., 26-29 Oct. 1972.
- Klein S. and al. Automatic Novel Writing: a Status Report. in TR 186, Univ. of Wisconsin, Aug. 1973.
- Klein, S. Automatic Inference of Semantic Deep Structure Rules in Generative Semantic Grammars. in COLING-73.
- Klein S., Lieman S.L. and Lindstrom G.E. DISEMINER: a Distributional Semantic Inference Maker. in Computer Studies in the Humanities Verbal Behavior, vol.1, no.1, 1966.
- Klein S., Davis B., Fabens., Herriot R., Katke W., Kuppen H.A., and Towster A. AUTOLING: an Automated Linguistic Fieldworker. COLING-67.
- Klein S., Fabens W., Herriot R., Katke W., Kuppen H.A. and Towster A. The AUTOLING System. in UWCS Tech. Report 43, Univ. of Wisconsin, Computer Science Dept., 1968.
- Klein S., Price L.A., Aeschlemann J.F., Balseger D.A. and Curtis E.J. A Meta-Symbolic Simulation Model for Five Myths from Levi-Strauss "The Raw and the Cooked". ICCH-2.
- Knuth, D.E. Top-Down Syntax Analysis. in Acta Informatica 1, pp.79-110, 1971.

- Knuth, D.E. The Art of Computer Programming. vol.1 et 3, Addison Wesley, 1973.
- Kripke, S. Naming and Necessity. in SONL, 1972.
- Kuno S. and Oetlinger A.G. Multiple-Path Syntactic Analyser. in Information Processing 62, P.306, 1962.
- Kuno, S. Some Characteristics of the Multiple-Path Syntactic Analyser. in Language Data Processing, Camb. Mass.: The Computation Lab. of Harvard Univ., C-6-1-C-6-8, 1964.
- Kuno, S. A System for Transformational Analysis. in Rep. NSF-15, Comput. Lab., Harvard Univ., Camb., Mass., 1965.
- Kuno, S. The Predictive Analyser and a Path Elimination Technique. in CACM, vol.8, no.7, pp.453-462, 1965.
- Kuno, S. Some Properties S. of Non-Referential Noun Phrases. in Studies in General and Oriental Linguistics, Jakobson R. and Kanamoto S. (eds), Tokyo, Japan: TEC Company Ltd, 1973.
- Kuipers, B. An Hypothesis-Driven Recognition System of the Blocksworld. in MIT-AI Working Paper 83, March, 1974.
- Kuipers, B. A Frame for Frames. in Representation and Understanding, Bobrow D. and Collins A.(eds), NY Academic Press, 1975.
- Labov, W. What is a Linguistic Fact? Lisse: Peter de Ridder Press, 1975.
- Lakoff, G. Generative Semantic. in Semantics, an Interdisciplinary Reader in Philosophy, Linguistic, Anthropology and Psychology, Camb. Univ. Press, London, 1969.
- Lakoff, G. Linguistics and Natural Logic. in SONL, 1972.
- Lakoff, G. Hedges: a Study in Meaning Criteria and the Logic of Fuzzy Concepts. in Journal of Philosophical Logic, vol.2, pp.458-508, 1
- Lehnert, W. Human and Computational Question-Answering. in Journal of Cognitive Science, vol. 1, no. 1, Sept. 1976.
- Lehnert, W. The Process of Question-Answering. Ph. D. Dissertation, dept. of Computer Science, Yale Univ., New Haven, Conn., 1977.
- Levin J.A. and Moore J.A. Dialogue-Games: Metacommunication Structures for Natural Language Interaction. in Cognitive Science 1:4, 1977.
- Levin J. and Goldman N. Process Models of Reference. Unpublished Ms., Marina Del Rey, Information Sciences Institute, 1978.
- Levison, M. QUELL: A Programming Language for Literary Text Analysis. COLING-73.
- Lewiss, D. General Semantics. in SONL, 1972.

- Leviss, D. Coulewiss, D. Counterfactuals. Harvard Univ. Press, Camb., Mass., 1973.
- Lindsay, R. Inferential Memory as the Basis of Machines which Understand Natural Thought. in CAT, 1973.
- Lytle E.G., Packard D., Melby A.K., Billings F. and Gibb P.K. Junction Grammar as a Base for Natural Language Processing. in AJCL Microfiche 26, 1975.
- Londe D. and Olney J. PAIRS: a Postprocessor for the Kuno-Oettinger English Analyser. in Santa Monica: SDC Document.
- Longuet-Higgins, C. The Algorithmic Description of Natural Language. in Proc. Royal Society, London pp.255-276, London.
- Lux, A. Etude d'un modèle abstrait pour une machine LISP et de son implantation. Thèse de troisième cycle, Univ. de Grenoble, Mars 1975.
- Lyons, J. Semantics. England: Camb. Univ. Press, 1977.
- Maegard B. and Spang-haussen E. Segmentation of French Sentences. COLING-73.
- Mann, W.C. Improving Methodology in Natural Language Processing. in TINLAP-75.
- Mann W., Moore J. and Levin J. A Comprehension Model for Human Dialogue. IJCAI-77.
- Marcus, M. Wait and See Strategies for Parsing Natural Language. in Working Paper 75: MI Lab, 1974.
- Marcus, M. Diagnosis as a Notion of Grammar. TINLAP-75.
- Marcus, M. A Design for a Parser for English. ACM National Conference, 1976.
- Marcus, M. Theory of Syntactic Recognition for Natural Language. MIT, Ph.D. Thesis, Dept. of Electrical Engineering and Computer Science, 1977.
- Marcus, M. A Computational Account of Some Constraints on Language. TINLAP-78.
- Markgold, E. Current Approaches to the Inference of Phrase Structures Grammars. Presented at Phrase Structure Conference, Univ. de Montréal, 22-23 Mars 1973.
- Marr D. and Hewitt C. Video Ergo Scio: an Essay on Some Thing we would like a Vision System to Know. MIT AI Lab Working Paper 60, Nov. 1973.
- Marron, J.L. A propos d'une sémantique de la méthode. in Revue Internationale de Philosophie, 103, 27e Avenue, Fasc.1, 1973.
- Medema P. and al. PHLIQAI: Multilevel Semantics in Question-Answering. in AJCL Microfiche 32, 1975.
- Melby, A.K. Pitch Contour Generation in Speech Synthesis: a Junction Grammar Approach. Ph.D. Dissertation, AJCL Microfiche 60, Spring 1977.

- Melby, A. Design and implementation of a Computer-Assisted Translation System. COLING-78.
- Nephin, M.S. L'ordinateur et l'analyse grammaticale. Doc. Centre International de Recherche Sur Le Bilinguisme, Univ. de Laval, Québec.
- Meunier, J.C. Experiments in Conceptual Analysis of Theoretical Discourses. COLING-76.
- Meunier, J.C., Rolland S., Daoust F. A System for Text and Context Analysis. Computers and the Humanities; VOL 10, pp 281-286; Pergamon Press, 1976.
- Miller, G.A. Comments on Lexical Analysis. TINLAP-75.
- Minsky, M.L. Minsky Frame System Theory. TINLAP-75.
- Minsky, M. Form and Content in Computer Science. in Journal of ACM, vol.17, no.2, ACM Turing Lecture, 2 April 1970.
- Minsky, M. A Framework for Representing Knowledge. in MIT-AI-Lab Memo no.306, 1974.
- Mitre, English Preprocessor Manual. in Rep. SR-132 The Mitre Corp., Bedford, Mass..
- Montague, R. English as Formal Language. in Formal Philosophy: Selected Papers of Richard Montague, Thomason, R. (ed), Yale Univ. Press, New Haven, Conn., 1969.
- Montague, R. The Proper Treatment of Quantification in Ordinary English. Dept. of Philosophy, Univ. of Calif., Los Angeles, 1970.
- Montague, R. Pragmatics and Intensional Logic. in SONL, 1972.
- Moore, R. D-Script: a Computational Theory of Descriptions. in Memo no.278, Artificial Intelligence Lab. MIT, Camb., Mass., 1973.
- Moore J. Newell A. How Can MERLIN Understand? Dept. of Computer Science, Carnegie-Mellon Univ., 1973.
- Moore J. and Newell A. How Can MERLIN Understand? in Knowledge and Cognition, Greag (ed), Baltimore, Md: Laurence, Erlbaum, Associates, 19
- Moran, J.L. Toward a Rational Model of Discourse Comprehension. TINLAP-78.
- Mc Calla, G. MUSE: a Model to Understand Simple English. in CACM, vol.15, no.1, pp.29-40, 1972.
- Mc Carthy J. and Hayes P. Some Philosophical Problems from the Standpoint of Artificial Intelligence. in MI, vol.4, 1969.
- Mc Carthy, J. Programs with Common Sense. in Proc. of the Symposium on the Mechanization of Thought Processes, Teddington HMSO, London.
- Mc Carthy, J. Review of Lightfill, Artificial Intelligence. in Artificial Intelligence, vol.5, pp.198-202, 1974.
- Mc Conlogue K. and Simmons R. An Analysing English Syntax with a Pattern Learning Parser. in CACM, vol.8, no.11, pp.637-698, 1965.

- Mc Dermott, D.V. Assimilation of New Information by a Natural Language-Understanding System. in MIT AI LAB, AI-TR-291, 1974.
- Mc Donald, A Framework for Writing Generation Grammars for Interactive Computer Program. 13th Annual Meeting Proc., ACL, 1975.
- Mc Donald, D. Preliminary Report on a Program for Generating Natural Language. in MIT AI-Lab, 1975.
- Mc Donald, D. Language Generation: Automatic Control Grammatical Detail. COLING-73.
- Mc Donald, D. Subsequent Reference: Syntactic and Rhetorical Constraints. TINLAP-73.
- Nedobekive M., Quezel-Ambrunaz and Jaeger D. Etat actuel du système de traduction automatique multilingue (Grenoble). COLLING-78.
- Newell, A. Artificial Intelligence and the Concept of Mind. CMTL, 1973.
- Newell, A. and al. Speech Understanding Systems: Final Report of a Study Group. North Holland Publishing CO., Amsterdam, 1973.
- Orthony, A. How Episodic is Semantic Memory. TINLAP-78.
- Ouellette, F. JEUDÉMO. Centre de calcul, Univ. de Montréal; 1972.
- Paillet, J.P. Computational Linguistics and Linguistic Theory. COLING-73.
- Palme, J. Making Computers Understand Natural Language. in Artificial Intelligence and Heuristic Programming, Findler N. and Meltzer B. (eds), American Elsevier Publishing Company, NY, pp.199-244, 1971.
- Panaccio, C. Des phoques et des hommes, Autopsie d'un débat idéologique. 5 ième Congrès de la Société de Philosophie du Québec à Ottawa; Mai 1978; aussi dans Philosophiques, vol. VI, numéro 1, avril 1979.
- Plante, P. Proposition d'algorithme pour le dépistage de relations de dépendance contextuelle dans un texte. Thèse de Maîtrise, dépt. de Philosophie, Univ. du Québec à Montréal, août 1974.
- Plante, P. Le dépistage automatique des structures de surface et l'analyse des textes philosophiques. CIRPHO vol.3 1975 76 no. 2.
- Plante, P. Le dépistage automatique des structures de surface. ACFAS, 44 ième congrès, Mai 1976.
- Plante, P. Le système Déredec et ses applications. Doc ALTP0 PP-3; Dépt. de Philosophie, Univ. du Québec à Montréal, 111 pages, Juin 1976.
- Plante, P. Le Système Déredec. Third International Conference on Computing in the Humanities. Waterloo, 1977.
- Plante, P. Le Déredec, manuel de l'utilisateur. (première édition: 1977, seconde édition: 1979), Service de l'Informatique de l'univ. du Québec à Montréal.

- Plante, P. un Apprenti-Lecteur. Joint Conference on Computer Science/Linguistics/Text Processing; Learned Societies; Western Univ. Ontario 73.
- Plante, P. Le Déredéc, logiciel pour le traitement linguistique et l'analyse de contenu des textes. 5th International Congress of Applied Linguistics; Montréal 78.
- Plante, P. APLEC et la chasse aux phoques. Quatrième colloque canadien sur le traitement automatique des textes; Congrès des Sociétés Savantes, Saskatoon mai 1979.
- Paxton, W.H. A Framework for Language Understanding. COLING-76.
- Petrick, S.R. A Recognition Procedure for Transformational Grammars. Ph.D., Dept. Modern Languages, MIT, Camb., Mass., 1965.
- Petrick S.R. and Plath W.J. Semantic Interpretation and Transformational Parsing in the Request System. COLING-73.
- Philips, B. Topic Analysis. COLING-73.
- Phillips, B. A Model for Knowledge and its Application to Discourse Analysis. in Tech. Report KSL-9, Chicago: Knowledge Systems Lab., Univ. of Illinois At Chicago Cercle, 1977.
- Phillips, B. Constraining Inference on Automatic Text Understanding. COLING-73.
- Petofi J.S. and Rieser H. Studies in Text Grammar. Dordrecht: Reidel.
- Polanyi, L. Understanding Stories: a Tripartite Model of Story Structure. COLING-73.
- Prince, E. On the Function of Existential Presupposition in Discourse. in Text VS Sentence, Petofi, J. (ed), Hamburg, Buske Verlag, 1973.
- Quam H. and Diffie W.B. LISP 1.6. Stanford Univ. 1973.
- Quesel-Ambrunaz et Quillaume P. Analyse automatique des textes par un système d'états finis. COLING-73.
- Quilliam, M.R. Semantic Memory. in Semantic Information Processing, Minsky, M. (ed), Camb. Mass. MIT Press, 1968.
- Quilliam, M. The Teachable Language Comprehender. in CACM, vol.12, no.8, pp.459-475, 1969.
- Raphael, B. "SIR: a Computer Program for Semantic Information Retrieval". in Semantic Information Processing, Minsky, M. (ed), MIT Press, Camb., Mass., 1968.
- Raphael, B. The Thinking Computer (Man Inside Matter). W.H. Freeman and CO., San Francisco, 1976.
- Ray L. and Arbizu J. Design and Implementation of a Transfert Component in a Machine-Assisted Translation System. COLING-73.
- Reichman, Conversational Coherency. COLING-73.
- Reiter, R. A Semantically Guided Deductive System for Automatic Theorem Proving. IJCAI-73.



- Reiter, R. A Paradigm for Formal Reasoning. Dept. of Computer Science, Univ. of British-Columbia.
- Reiter, R. Formal Reasoning and Language Understanding Systems. TINLAP-75.
- Reiter, R. On Reasoning by Default. TINLAP.
- Rieger, C.J. Conceptual Memory: a Theory and Computer Program for Processing the Meaning Content of Natural Language Utterances. Unpublished Ph.D. Thesis, Stanford Univ., 1974.
- Rieger, C. Conceptual Overlays: a Mechanism for the Interpretation of Sentence Meaning in Context. IJCAI-75.
- Rieger, C. Understanding by Conceptual Inference. in American Journal of Computational Linguistics, 1975.
- Rieger, C. The Commonsense Algorithm as a Basis for Computer Models of Human Memory, Inference, Belief and Contextual Language Comprehension. TINLAP-75.
- Rieger, C. One System for Two Tasks: a Commonsense Algorithm Memory That Solves Problems and Comprehends Language. in Tech. Report no.435, Univ. of Maryland, Computer Science Dept., 1976.
- Rieger, C. Spontaneous Computation in Cognitive Models. in Cognitive Science, vol.1, no.3, pp.315-357, 1977.
- Rieger, B. Feasible Fuzzy Semantics. COLING-78.
- Ricoeur, P. The Model of the Text: Meaningful Action as a Text. in Social Research, vol.38, no.3, Autumn 1971.
- Riesbeck, C. Computational Understanding: Analysis of Sentences and Context. Ph.D. Thesis, Computer Science Dept., Stanford Univ., Stanford, CA, 1974.
- Riesbeck, C. Computational Understanding. TINLAP-75.
- Riesbeck C. and Schank R. Comprehension by Computer: Expectation-Based Analysis of Sentences in Context. in Research Report no.78, Dept. of Computer Science, Yale Univ., New Haven, CT, 1976.
- Riesbeck, C. Computer Analysis of Natural Language in Context. Ph.D. Thesis, Computer Science Dept., Stanford Univ., Stanford, CA.
- Robinson, J.J. Pointed Questions and Purposeful Answers. COLING-78.
- Robinson, A.E. Speech, Language and Artificial Intelligence. COLING-78.
- Ross, D. Jr. Beyond the Concordance: Algorithms for the Description of English Clauses and Phrases. in The Computer and Literary Studies, Aitken, A.J. and al. (eds), Edinburgh: Univ. of Edinburgh Press, pp.85-99, 1973.
- Ross, D.J.R. and Robert H.R. Eyeball: a Computer Program for Description of Style. in Chum 6:213-21, 1972.
- Ross, D. The Use of World-Class Distribution Data for Stylistics. COLING-76.

- Ross, J.R. On Declarative Sentences. in Readings in English Trans. Grammar, Jacobs and Rosenbaum (eds), Waltham, 1970.
- Roussel, P. PROLOC, Manuel de référence et d'utilisation. Groupe D'intelligence Artificielle, VER de Marseille, France, 1975.
- Rulefson J.F., Derksen J.A. and Waldinger R.J. QA4: a Procedural Calculus for Intuitive Reasoning. Ph.D., Stanford, Nov. 1972.
- Rumelhart D. and Norman D. Active Semantic Networks as a Model of Human Memory. IJCAI-73.
- Rumelhart D., Lindsay P. and Norman D. A Process Model for Long Term Memory. in Organisation of Memory, Tulving E. and Donaldson W. (eds), Academic Press, NY, pp.198-221, 1973.
- Rumelhart, D. Notes on a Schema for Stories. in Representing and Understanding, Bobrow and Collins (eds), Academic Press, NY, pp.211-236, 1975.
- Rumelhart D. and Orthonoy A. The Representation of Knowledge in Memory. in CHIP Memo no.55, Jabolla: Centre for Human Information Processing, 1976.
- Salkoff, M. Une Grammaire en chaîne du français. Dunod, Paris, 1973.
- Salkoff, M. Sur la confection d'un lexique pour l'analyse automatique Sandewall, E.J. "Concepts and Methods for Heuristic Search". IJCAI-69.
- Sandewall, E.J. a Set-Oriented Property-Structure Representation for Binary Relations. in MI, vol.5, pp.237-252, 1970.
- Sandewall, E.J. Representing Natural Language Information in Predicate Calculus. in MI, Michie D. and Meltzer B. (eds), vol.6, Edinburgh: Edinburgh Univ. Press, 1972.
- Sandewall, E.J. Formal Methods in the Design of Question-Answering Systems. in AIJ, vol.2, no.2, pp.129-145, 1971.
- Salton, G. The SMART Retrieval System. Prentice Hall, 1971.
- Scha, R.J.H. Semantic Types in PHLIQAI. COLING-76.
- Scha, R.J.H. A Formal Treatment of Some Aspects of Quantification in English. COLING-78.
- Schanck, R. Conceptual Dependency: a Theory of Natural Language Understanding. in Cognitive Psychology, vol.3, pp.552-631, 1972.
- Schanck, R. Identification of Conceptualizations Underlying Natural Language. in CMTL.
- Schanck, R. Conceptual Dependency: a Theory of Natural Language Understanding. CMTL.
- Schanck, R. Adverbs of Belief. in Lingua, vol.33, North Holland Publishing Company, pp.45-67, 1974.

- Schanck, R. Understanding Paragraphs. in Tech. Report 5, Istituto Per Gli Studi Semantecie Cognitivi, Castagnola, Switzerland, 1974.
- Schanck, R. The Role of Memory in Language Processing. in Yale Univ. Tech. Report, New Haven, Conn., 1975.
- Schanck, R. The Structure of Episodes in Memory. in Representation and Understanding, Bobrow D.G. and Collins A.M. (eds), New York: Academic Press, 1975.
- Schanck, R. The Primitive Acts of Conceptual Dependency. TINLAP-75.
- Schanck, R. Conceptual Information Processing. North Holland Publishing Company, 1975.
- Schanck, R. Using Knowledge to Understand. TINLAP-73.
- Schanck, R. Rules and Topics in Conversation. in Cognitive Science 1:4, pp.421-442, 1977.
- Schanck, R. What Makes Something "ad Hoc"? TINLAP-78.
- Schanck R., Goldman N., Rieger C. and Rusbeck C. Primitive Concepts Underlying Verbs of Thought. in AI Memo 162, Stanford Univ., 1972.
- Schanck R., Goldman N., Rieger C. and Riesbeck C. Margie: Memory, Analysis, Response Generation and Inference on English. in Adv. Papers on the IJCAT-73, Univ. of Stanford, 1973.
- Schanck R. and Rieger C. Inference and the Computer Understanding of Natural Language. in AIM-1973, Computer Science Dept., Stanford Univ., April 1973.
- Schanck, R. and al. Inference and Paraphrase by Computer. in Journal of the ACM, July 1975.
- Schanck R. and Abelson R.P. Scripts, Plans and Knowledge. IJCAI-75.
- Schanck R. and Abelson R.P. Knowledge Structures. Potomac, MD: Lawrence Erlbaum Press, 1976.
- Schanck R. and Abelson R.P. Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Assoc., Hillsdale, NJ, 1977.
- Schubert, L. On the Expressive Adequacy of Semantic Networks. in TR 74-18, Dept. of Computing Science, Univ. of Alberta, Edmonton, 1974.
- Sgall, P. Linguistic and Artificial Intelligence. in The Prague Bulletin of Mathematical Linguistics 24, pp.5-34, 1974.
- Sgall P., Pavenova J. and Machova S. A Linguistic Approach to Information Retrieval. in Inf. Processing and Management 11, pp.147-153, 1975.
- Sgall, P. On the Notion of Semantic Language. COLING-76.
- Sgall, P. On Some Relationships of Linguistics and Information Retrieval. in The Prague Bulletin of Mathematical Linguistics 26, 1976.

- Shapero S. and Woodmansu G. A Net Structure Based Relational Question-Answerer: Description and Examples. IJCAI-69.
- Shapero, S. Path-Based and Node-Based Inference in Semantic Networks. TINLAP-78.
- Siklossy, L. Natural Language Learning by Computer. in Automated Language Processing, Borko, H. (ed), John Wiley and Sons, NY, pp.253-290, 1968.
- Setsuo, O. An Information-Structure Based Method for Extracting Meaning from Input Sentence. COLLING-73.
- Sidver, C.L. The Use of Focus as a Tool for Disambiguation of Deninite Noun Phrases. TINLAP-78.
- Siess, B. Recherche sur la formalisation de l'analyse syntaxique et sémantique dans une projective de déduction automatique. Thèse, Paris, Jan. 1976.
- Simmons, R. Storage and Retrieval of Aspects of Meaning in Directed Graph Structures. in CACM, vol.3, pp.211-215, 1966.
- Simmons, R. Answering English Questions by Computer. in ALP, pp.253-290, 1968.
- Simmons, R. Natural Language Question-Answering Systems: 1969. in CACM, vol.13, no.1, pp.15-30, 1970.
- Simmons, R. Semantic Networks: Their Computation and Use for Understanding English Sentences. in CNTL.
- Simmons R., Klein S. and Mc Conlongue K. Indexing and Dependency Logic for Answering English Questions. in American Documentation, vol.15, no.3, pp.196-204, 1964.
- Simmons R., Burger R. and Schwartz R. A Computerized Model of Verbal Understanding. in FJCC, pp.441-456, 1968.
- Simmons, R. Some Semantic Structures for Representing English Meanings. Univ. of Texas, Austin, Comp. Sci. Diff. Preprint, Nov. 1970.
- Simmons R. and Slocum J. Generating English Discourse from Semantic Networks. in CACM, vol.15, no.10, Oct. 1972.
- Slagle, J. Experiments with a Deductive Question-Answering Program. in CACM, vol.8, no.12, pp.792-798, 1965.
- Slagle, J. Artificial Intelligence: the Heuristic Programming Approach. Mc Graw Hill Book CO, NY, 1971.
- Sloman, A. Interactions Between Philosophy and AI - the Role of Intuition and Non-Logical Reasoning in Intelligence. in Proc. of 2nd IJCAI, Reprinted in AI, vol.2, 1971.
- Sloman, A. Object Frames, Domain Frames and Analogical Representation. Miméo, 1975.
- Spark, Jones K. Automatic Keyword Classification for Information Retrieval. Archon Books, London, 1971.
- Stalnaber, R. Pragmatic Presuppositions. in Proc. of the Texas Conference on Performatives Presuppositions and Implicatures, Rogers A., Wall B. and Murphy J.P. (eds), Arlington, Center for Applied Linguistics, pp.135-148, 1977.

- Stenning, K. Understanding English Articles and Quantifiers. Published Doctoral Dissertation, New York: the Rockefeller Univ., 1975.
- Stewart, G. Le langage de programmation REZO. Rapport de stage informatique, Univ. de Montréal, 1975.
- Stutzman, W.J. Organizing Knowledge for English-Chinese Translation. COLING-76.
- Sussman, G. and al. Microplanner Reference Manual. in AIL MIT, Camb., Mass., 1972.
- Svensson, B. On the Use of Fuzzy Concepts in Description. IJCAI-71.
- Szolovits P., Havkenson L.B. and Martin W.A. An Overview of OWL, an Language for Knowledge Representation. in MIT LCS-TH-86, 1977.
- Tanay P., Korosec T. and Bratko I. Structural and Stylistic Analysis of Texts Via Formalized Sentences of Natural Language. COLING-78.
- Isabelle P. et al. TAUM-Aviation: un Système pour la traduction automatique de manuelles techniques. in 5e Congrès International de Linguistique Appliquée, Montréal, 1978.
- Thorne J., Bratley P. and Dewar H. The Syntactic Analysis of English by Machines. in MI, vol.3, pp.281-309, 1968.
- Thouin, E. Aspects informatiques de la traduction assistée par ordinateur et de la traduction automatique; applications au fonctionnement de systèmes mixtes. COLING-78.
- Tolving E. and Donaldson W. Organisation of Memory. Academic Press, NY, 1972.
- Turing, A. Computing Machinery and Intelligence. in MIND, vol.59, pp.433-460, 1950.
- Unger, S.H. A Global Parser for Context-Free Phrase Structure Grammars. in CACM, vol.11, no.4, P.240, 1968.
- Van Dijk, T.A. Some Aspects of Texts Grammars. The Hague: Mouton, 1972.
- Van Dijk, T.A. (ed) Text Grammar and Narrative Structures. in Poetics 3, 1973.
- Van Dijk T.A. and Petofi (eds) Grammars and Descriptions. Berlin -- New York: de Gruyter, 1974.
- Vauquois, B. La Traduction automatique à Grenoble. in Documents de Ling. quant. 24, Paris, 1975.
- Walker, D.E. The SRI Speech Understanding System. in 1er Symp. on Speech Recognition, Apr. 1974.
- Wellier B.L. and Formal A. Approach to Discourse Anaphora. in Tech. Report 3761, Camb., Mass., BBN, 1978.
- Webber, B.L. Description Formation and Discourse Model Synthesis. TINLAP-78.
- Weizenbaum, J. Contextual Understanding by Computers. in CACM, vol.10, no.8, pp.474-480, 1967.

- Wilks, Y. Computable Semantic Derivations. Systems Development Corp., Memo, 1968.
- Wilks, Y. Decidability and Natural Language. in Mind, 1971.
- Wilks, Y. Grammar, Meaning and the Machine Analysis of Language. London, 1972.
- Wilks, Y. Preference Semantics. in Memo AIM-206, Stanford AI Lab., Publications of Stanford Univ., July 1973.
- Wilks, Y. Natural Language Inference. in Memo AIM-211, Stanford AI Project, Stanford Univ., 1973.
- Wilks, Y. An Artificial Intelligence Approach to Machine Translation. in CMTL.
- Wilks, Y. One Small Head. Foundations of Language, 1974.
- Wilks, Y. Natural Language Systems within the AI Paradigm. in AI Lab, Memo no.337, Stanford Univ., 1975.
- Wilks, Y. Methodology in AI and Natural Language Understanding. TINLAP-75.
- Wilks, Y. Philosophy of Language. in Notes for the Tutorial on Computational Semantics, ISSCO, Castagnola, 1975.
- Wilks, Y. An Intelligent Analyser and Understander of English. in CACM, vol.18, pp.264-274, 1975.
- Wilks, Y. A Preferential Pattern-Matching Semantics for Natural Language Inference. in AI, vol.6, 1975.
- Wilks, Y. Primitives and Works. TINLAP-75.
- Wilks, Y. Frames, Scripts, Stories and Fantasies. CO-LING-76.
- Wilks, Y. What Sort of Taxonomy of Causation do we Need for Natural Language Understanding? in Cognitive Science, vol.1, 1976.
- Wilks, Y. Good and Bad Arguments About Semantic Primitives. in DAI Research Report, no.43, Univ. of Essex, May 1977.
- Wilks, Y. Knowledge Structures and Language Boundaries. IJCAI-77.
- Wilks, Y. Semantic Primitives in Language and Vision. TINLAP-78.
- Winograd, T. PROGRAMMAR: a Language for Writing Grammars. in AI Lab, Memo no.181, MIT, Camb., Mass., 1969.
- Winograd, T. Procedures as a Representation for Data. in a Computer Program for Understanding Natural Language. in Project MAC, TR-84, MIT, Feb. 1971.
- Winograd, T. An Approach to English Morphemic Analysis. in AI Lab, Memo no.241, MIT, Camb., Mass., 1971.
- Winograd, T. Understanding Natural Language. Academic Press, 1972.
- Winograd, T. Five Lectures on Artificial Intelligence. in Stanford AI, Memo, Sept. 1974.

- Winograd, T. Towards a Procedural Understanding of Semantics. in *Revue Internationale de Philosophie*, Fasc. 3-4, pp.117-118, 1976.
- Winograd, T. A Framework for Understanding Discourse. in *AIN-297*, Stanford, Calif., 1977.
- Winograd, T. On Primitives, Prototypes and Other Semantic Anomalies. *TINLAP-78*.
- Woods, W.A. Semantics for a Question-Answering System. Ph.D. The. Rep. NSF-19, Aiken Computation Lab, Harvard Univ., Camb., Mass., 1967.
- Woods, W.A. Transition Network Grammars for Natural Language Analysis. in *Communications of the ACM*, vol.13, no.10, pp.591-606, Oct. 1970.
- Woods, W.A. An Experimental Parsing for Transition Network Grammars. in *BBN Report no.2362*, 15 May 1972.
- Woods W.A., Kaplan R.H. and Nash-Webber B. The Lunar Sciences Natural Language Information System. in *BBN Report no.2378*, 1972.
- Woods W.A. and Maknoul J. Mechanical Reference Problems in Continuous Speech Understanding. in *AI 5*, pp.73-91, 1974.
- Woods, W.A. Some Methodological Issues in Natural Language Understanding Research. *TINLAP-78*.
- Woods, W.A. Semantics and Quantification in Natural Language Question-Answering. in *Advances in Computers*, vol.17, New York: Academic Press, Also in *BBN Report no.3687*, 1977.
- Woods W.A. and Brachman R.J. Research in Natural Language Understanding. in *Quarterly Technical Progress Report no.1*, and *BBN Report no.3742*, 1978.
- Woods, W.A. Taxonomic Lattice Structures for Situation Recognition. *TINLAP-78*.
- Zarri, G.P. Sur le traitement automatique de données biographiques médiévales: le Projet RESEDA. in *Computing in the Humanities*, Univ. of Waterloo Press, Lusignan S., et North J. (eds), 1977.
- Zadeh, L. Fuzzy Sets. in *Information and Control*. v8 pp.338-353, 1965.
- Zadeh, L. A Fuzzy-Set Theoretic Interpretation of Linguistic Hedges. *Journal of Cybernetics*, V2 no. 3 pp.4-34, 1972.

Annexe au chapitre -3-



Automate FD-1 utilisé pour la catégorisation de base:

(FD-1

```
(S1 (PHI S2 D) (X S1 G))
(S2 (X (TRANEXA CATEGO) S3))
(S3 ((D1 D2 N V C) (RETURN))
      (X (ARRET) (RETURN))))
```

On obtient CATEGO en faisant préalablement à l'évaluation de FD-1 la commande: (PREPARE (CATEGO.TRA))

#### Fichier CATEGO.TRA

|                     |                    |                   |
|---------------------|--------------------|-------------------|
| (C1 "et")           | (C1 "ni")          | (C1 "ou")         |
| (C21 "d")           | (C21 "de")         | (C21 "des")       |
| (C21 "du")          | (C22 "à")          | (C22 "afin")      |
| (C22 "au")          | (C22 "aux")        | (C22 "avec")      |
| (C22 "dès")         | (C22 "dans")       | (C22 "lors")      |
| (C22 "lorsqu")      | (C22 "lorsque")    |                   |
| (C22 "malgré")      | (C22 "outre")      | (C22 "par")       |
| (C22 "parce")       | (C22 "parmi")      | (C22 "pource")    |
| (C22 "puisque")     | (C22 "puisque")    | (C22 "sans")      |
| (C22 "selon")       | (C22 "sur")        | (C31 ",")         |
| (C32 "!"            | (C32 ".")          | (C32 ":",")       |
| (C32 ";")           | (C32 "?")          | (D11 "ailleurs")  |
| (D11 "ainsi")       | (D11 "alors")      | (D11 "assez")     |
| (D11 "auparavant")  | (D11 "aussi")      | (D11 "aussitot")  |
| (D11 "autant")      | (D11 "autrefois")  | (D11 "beaucoup")  |
| (D11 "bientôt")     | (D11 "ca")         | (D11 "ci-dessus") |
| (D11 "ci-devant")   | (D11 "déjà")       | (D11 "davantage") |
| (D11 "delà")        | (D11 "derechef")   |                   |
| (D11 "donc")        | (D11 "enfin")      | (D11 "ensuite")   |
| (D11 "etc")         | (D11 "guère")      | (D11 "ici")       |
| (D11 "jamais")      | (D11 "jusqu")      | (D11 "jusques")   |
| (D11 "là")          | (D11 "maintenant") | (D11 "mais")      |
| (D11 "mieux")       | (D11 "n")          | (D11 "ne")        |
| (D11 "non")         | (D11 "partout")    | (D11 "peut-être") |
| (D11 "plutôt")      | (D11 "puisque")    | (D11 "quasi")     |
| (D11 "quelquefois") | (D11 "souvent")    | (D11 "surtout")   |
| (D11 "tôt")         | (D11 "tant")       | (D11 "tantôt")    |
| (D11 "toutefois")   | (D11 "très")       | (D11 "trop")      |
| (D12 "ces")         | (D12 "cet")        | (D12 "cette")     |
| (D12 "chaque")      | (D12 "mes")        | (D12 "mon")       |
| (D12 "nos")         | (D12 "notre")      | (D12 "quelqu")    |
| (D12 "quelque")     | (D12 "quelques")   | (D12 "sa")        |
| (D12 "ses")         | (D2 "car")         | (D2 "combien")    |
| (D2 "comme")        | (D2 "comment")     | (D2 "encore")     |
| (D2 "pourquoi")     | (D2 "qu")          | (D2 "quand")      |
| (D2 "que")          | (D12 "na")         | (N211 "elle")     |
| (N211 "elles")      | (N211 "il")        | (N211 "ils")      |

|                       |                      |                    |
|-----------------------|----------------------|--------------------|
| (N211 "j")            | (N211 "je")          | (N211 "on")        |
| (N212 "lui")          | (N212 "m")           | (N212 "me")        |
| (N212 "se")           | (N212 "y")           | (N22 "ceci")       |
| (N22 "cela")          | (N22 "celle")        | (N22 "celle-là")   |
| (N22 "celles")        | (N22 "celles-ci")    | (N22 "celui")      |
| (N22 "celui-ci")      | (N22 "ceux")         | (N22 "ceux-ci")    |
| (N22 "ceux-là")       | (N22 "elles-mêmes")  | (N22 "eux")        |
| (N22 "eux-mêmes")     | (N22 "moi")          | (N22 "moi-même")   |
| (N22 "quelques-unes") | (N22 "quelques-uns") | (N22 "quoi")       |
| (N22 "soi")           | (N22 "soi-même")     | (N22 "unes")       |
| (N22 "uns")           | (N23 "auquel")       | (N23 "auxquelles") |
| (N23 "auxquels")      | (N23 "desquels")     | (N23 "dont")       |
| (N23 "duquel")        | (N23 "laquelle")     | (N23 "lequel")     |
| (N23 "lesquelles")    | (N23 "lesquels")     | (N23 "où")         |
| (N23 "qui")           | (N23 "quiconque")    | (T102 "puis")      |
| (T11 "l")             | (T11 "la")           | (T11 "le")         |
| (T11 "les")           | (T11 "leur")         | (T11 "leurs")      |
| (T12 "en")            | (T131 "ce")          | (T131 "son")       |
| (T132 "aucun")        | (T132 "aucune")      | (T132 "aucunes")   |
| (T132 "aucuns")       | (T132 "quelle")      | (T132 "quelles")   |
| (T132 "tous")         | (T132 "un")          | (T132 "une")       |
| (T133 "autres")       | (T133 "autres")      | (T133 "certain")   |
| (T133 "certaine")     | (T133 "chacun")      | (T133 "chacune")   |
| (T133 "divers")       | (T133 "diverses")    | (T133 "plusieurs") |
| (T133 "quel")         | (T133 "quels")       | (T133 "tel")       |
| (T133 "telle")        | (T133 "telles")      | (T133 "tels")      |
| (T134 "seule")        | (T134 "seules")      | (T134 "sculs")     |
| (T141 "s")            | (T142 "si")          | (T151 "contre")    |
| (T151 "envers")       | (T151 "pour")        | (T152 "mal")       |
| (T1521 "moins")       | (T1521 "peu")        | (T1521 "plus")     |
| (T1521 "rien")        | (T1522 "abord")      | (T1522 "bien")     |
| (T1523 "or")          | (T1523 "pas")        | (T1523 "point")    |
| (T1531 "même")        | (T1531 "mêmes")      | (T1531 "seul")     |
| (T1532 "toute")       | (T1532 "toutes")     | (T1533 "tout")     |
| (T162 "pourvu")       | (T162 "vu")          | (T17 "après")      |
| (T17 "au-dessous")    | (T17 "au-dessus")    | (T17 "au-devant")  |
| (T17 "avant")         | (T17 "dedans")       | (T17 "depuis")     |
| (T17 "devant")        | (T17 "par-dessus")   | (T17 "pendant")    |
| (T17 "personne")      | (T18 "nous")         | (T18 "vous")       |
| (T133 "mien")         | (T133 "mienne")      | (T133 "miennes")   |
| (T133 "nôtre")        | (T133 "nôtres")      | (T133 "sien")      |
| (T133 "sienne")       | (T133 "siennes")     | (V1 "a")           |
| (V1 "ait")            | (V1 "aura")          | (V1 "est")         |
| (V1 "sont")           |                      |                    |

## Séries des automates FD0:

```

(FD00 (S1 (PHI S2 D) (X S1 G))
      (S2 (T102 S3 G) (C32 (E FD01 S1 G) (RETURN)) (X S2 D))
      (S3 ((PHI C3) S4 D) (X S5 D))
      (S4 (T102 (O G) (F D D11) S2 D))
      (S5 (T102 (O G) (F D V1) S2 D)))
(FD01 (S1 (PHI S2 D) (X S1 G))
      (S2 (T11 S3 D) (C32 (E FD02 S1 G) (RETURN)) (X S2 D))
      (S3 ((V1 V21 V22 N212 T12 T162) S4 G)
          (T11 S8 G)
          (D11 S5 D)
          (X S7 G))
      (S4 (T11 (O G) (F D N212) S2 D))
      (S5 (N212 S5 D) (V21 S6 G) (X S7 G))
      (S6 (T11 (O G) (F D N212) S2 D) ((N212 D11) S6 G))
      (S7 (T11 (O G) (F D D12) S2 D) (X S7 G))
      (S8 (T11 (O G) (F D D12) S9 D))
      (S9 (T11 (O G) (F D N22) S2 D)))
(FD02 (S1 (PHI S2 D) (X S1 G))
      (S2 (T12 S3 D) (C32 (E FD031 S1 G) (RETURN)) (X S2 D))
      (S3 ((T152 D11) S3 D) ((V1 V21) S4 G) (X S5 G))
      (S4 (T12 (O G) (F D N212) S2 D) (X S4 G))
      (S5 (T12 (O G) (F D C22) S2 D) (X S5 G)))
(FD031 (S1 (PHI S2 D) (X S1 G))
      (S2 (T131 S3 D) (C32 (E FD032 S1 G) (RETURN)) (X S2 D))
      (S3 ((N22 T1521 N1) S5 G) ((D11 V23) S3 D) (X S4 G))
      (S4 (T131 (O G) (F D N22) S2 D) (X S4 G))
      (S5 (T131 (O G) (F D D12) S2 D) (X S5 G)))
(FD032 (S1 (PHI S2 D) (X S1 G))
      (S2 (T132 S3 D) (C32 (E FD033 S1 G) (RETURN)) (X S2 D))
      (S3 ((T142 T134 T1532 D11 D12 T1533) S3 D)
          (C1 S7 D)
          ((T152 T16 T151 T1531 T133 N1 N22) S4 G)
          (V23 S6 D)
          (X S5 G))
      (S4 (T132 (O G) (F D D12) S2 D) (X S4 G))
      (S5 (T132 (O G) (F D N22) S2 D) (X S5 G))
      (S6 ((T152 T16 T1531 T151 T133 N1 N22) S4 G)
          (V23 S6 D)
          (C1 S7 D)
          (X S5 G))
      (S7 ((D11 V23) S3 D) (X S5 G)))
(FD033 (S1 (PHI S2 D) (X S1 G))
      (S2 (T133 S3 D) (C32 (E FD034 S1 G) (RETURN)) (X S2 D))
      (S3 ((D11 V23) S3 D)
          ((N1 N22 T161 T1533 T1522) S4 G)
          (T133 S11 D)
          (D2 S19 G)
          (X S5 G))

```

```

(S4 (T133 S9 G) (X S4 G))
(S5 (T133 S6 G) (X S5 G))
(S6 ((N1 D11 T152) S8 D) (C1 S17 G) (X S7 D))
(S7 (T133 (O G) (F D N22) S2 D) (X S7 D))
(S8 (T133 (O G) (F D V23) S2 D))
(S9 ((D12 V23) S8 D) (X S10 D))
(S10 (T133 (O G) (F D D12) S2 D))
(S11 ((T161 T1522 T1533 N1 N22) S13 G) (X S12 G))
(S12 (T133 (O G) (F D N22) S14 G))
(S13 (T133 (O G) (F D V23) S15 G))
(S14 (T133 (O G) (F D D12) S16 D))
(S15 (T133 (O G) (F D N22) S16 D))
(S16 (X S2 D))
(S17 (V23 S18 D) (X S7 D))
(S18 (C1 S8 D))
(S19 (T133 (O G) (F D N22) S2 D) (X S19 G)))
(FD034 (S1 (PHI S2 D) (X S1 G))
(S2 (T134 S3 G) (C32 (E FD041 S1 G) (RETURN)) (X S2 D))
(S3 ((D11 V N22 N23 T162) S6 D) (D12 S5 D) (X S4 D))
(S4 (T134 (O G) (F D N22) S2 D))
(S5 (T134 S7 D))
(S6 (T134 (O G) (F D V23) S2 D))
(S7 ((T16 N1 N22 T15) S9 G) ((D11 V23) S7 D) (X S8 G))
(S8 (T134 (O G) (F D N22) S2 D) (X S8 G))
(S9 (T134 (O G) (F D V23) S2 D) (X S9 G)))
(FD041 (S1 (PHI S2 D) (X S1 G))
(S2 (T141 S3 D) (C32 (E FD042 S1 G) (RETURN)) (X S2 D))
(S3 ((N212 D11) S3 D) ((T162 V1 V21 V22) S5 G) (X S4 G))
(S4 (T141 (O G) (F D C22) (O D) (F D D2) S2 D) (X S4 G))
(S5 (T141 (O G) (F D N212) S2 D) (X S5 G)))
(FD042 (S1 (PHI S2 D) (X S1 G))
(S2 (T142 S3 D) (C32 (E FD051 S1 G) (RETURN)) (X S2 D))
(S3 ((T191 V23
      T151
      T1521
      T1522
      T1523
      T1531
      T1532
      T1533
      D11
      T162)
      S5
      G)
      (X S4 G))
(S4 (T142 (O G) (F D C22) (O D) (F D D2) S2 D))
(S5 (T142 (O G) (F D D11) S2 D)))
(FD051 (S1 (PHI S2 D) (X S1 G))
(S2 (T151 S3 G) (C32 (E FD052 S1 G) (RETURN)) (X S2 D))
(S3 (D12 S5 D) (X S4 D))
(S4 (T151 (O G) (F D C22) S2 D))

```

```

(S5 (T151 (O G) (F D N22) S2 D)))
(FD052 (S1 (PHI S2 D) (X S1 G))
(S2 (T152 S3 C) (C32 (E FD053 S1 G) (RETURN)) (X S2 D))
(S3 (V23 S4 D)
(D12 S9 G)
(C21 S11 D)
(D11 S8 D)
(X S5 D))
(S4 (D12 S4 D)
((T1522 T1523) (O G) (F D N1) S2 D)
(T1521 S6 D))
(S5 (T152 (O G) (F D D11) S2 D))
(S6 ((T162 V1 V2 D1 T17 N23) S5 G) (X S7 G))
(S7 (T1521 (O G) (F D N22) S2 D))
(S8 ((T1521 T1523) (O G) (F D D11) S2 D) (T1522 S13 D))
(S9 (C22 S10 D) (X S4 D))
(S10 (D12 S10 D)
(T1521 (O G) (F D D11) S2 D)
((T1522 T1523) (O G) (F D N1) S2 D))
(S11 (T1521 S12 D) ((T1522 T1523) (O G) (F D D11) S2 D))
(S12 (V1 S7 G) (X S5 G))
(S13 ((V23 D2) S14 G) (X S15 G))
(S14 (T1522 (O G) (F D D11) S2 D))
(S15 (T1522 (O G) (F D V23) S2 D))
(S16 (D2 S7 G) (X S8 G)))
(FD053 (S1 (PHI S2 D) (X S1 G))
(S2 (T153 S3 C) (C32 (E FD06 S1 G) (RETURN)) (X S2 D))
(S3 (D12 S4 D) ((D11 V23) S3 G) (X S5 D))
(S4 (T153 S6 D) (X S4 D))
(S5 (T153 S10 D) (X S5 D))
(S6 (N1 S8 G) ((D11 V23) S7 G) (X S9 G))
(S7 (T153 (O G) (F D D11) S2 D) (X S7 G))
(S8 (T153 (O G) (F D V23) S2 D) (X S8 G))
(S9 (T153 (O G) (F D N22) S2 D))
(S10 ((D12 N1 T161 T162 N22) S11 G) (X S12 G))
(S11 (T1531 (O G) (F D D11) S2 D)
((T1532 T1533) (O G) (F D D12) S2 D))
(S12 ((T1531 T1533) S13 G)
(T1532 (O G) (F D D11) S2 D)
(X S12 G))
(S13 (D2 S16 D) ((N1 N22) S14 D) (X S15 D))
(S14 ((T1531 T1532 T1533) (O G) (F D V23) S2 D))
(S15 (T1533 S17 D) (T1531 (O G) (F D D11) S2 D))
(S16 (T1531 (O G) (F D D11) S2 D)
(T1533 (O G) (F D N22) S2 D))
(S17 (C2 S16 G) (X S18 G))
(S18 (T1533 (O G) (F D D11) S2 D)))
(FD06 (S1 (PHI S2 D) (X S1 G))
(S2 ((T161 T162) S3 D)
(C32 (E FD07 S1 G) (RETURN))
(X S2 D))

```

```

(S3 (D11 S3 D) (D2 S5 G) (X S4 G))
(S4 (T161 (O G) (F D N1) S2 D)
  (T162 (O G) (F D V23) S2 D)
  (D11 S4 G))
(S5 (T162 S8 G) (T161 S6 G) (D11 S5 G))
(S6 (C22 S7 D) (X S4 D))
(S7 (T161 (O G) (F D C22) S2 D))
(S8 (C31 S10 D) (X S9 D))
(S9 (T162 (O G) (F D V23) S2 D))
(S10 (T162 (O G) (F D C22) S2 D)))
(FD07 (S1 (PHI S2 D) (X S1 G))
  (S2 (T17 S3 G) (C32 (E FD08 S1 G) (RETURN)) (X S2 D))
  (S3 (D11 S8 G) (X S5 D))
  (S4 (T17 (O G) (F D D11) S2 D) (X S4 D))
  (S5 (T17 S6 D))
  (S6 ((V1 V21 V22 C2 D2 D12 N) S7 G) (X S4 G))
  (S7 (T17 (O G) (F D C22) S2 D))
  (S8 (D11 S8 G) (PHI S9 D) (X S4 D))
  (S9 (T17 S6 D) (X S9 D)))
(FD08 (S1 (PHI S2 D) (X S1 G))
  (S2 (T18 S3 D) (C32 (E FD101 S1 G) (RETURN)) (X S2 D))
  (S3 (V1 S4 G) ((V21 V22 D2 C3 N23) S6 G) (X S5 D))
  (S4 (T18 S9 G) (N22 S8 G) (D11 S7 G) (X S4 G))
  (S5 ((D2 C3 N23) S6 G) (V1 S4 G) (X S5 D))
  (S6 (T18 (O G) (F D N212) S2 D) (X S6 G))
  (S7 (T18 S30 G) (X S7 G))
  (S8 (T18 (O G) (F D N212) S2 D) (X S8 G))
  (S9 ((D1 V23) S9 G)
    ((D2 C32) S10 D)
    ((C2 N211 N1 N23) S11 D)
    (T18 (O G) (F D N211) S12 D)
    (X S13 D))
  (S10 ((D1 V23) S10 D) (T18 (O G) (F D N211) S2 D))
  (S11 ((D1 V23) S11 D) (T18 (O G) (F D N212) S2 D))
  (S12 ((D11 V23) S12 D) (T18 (O G) (F D N212) S2 D))
  (S13 (T18 S15 G) (X S13 D))
  (S14 (T18 (O G) (F D N212) S2 D))
  (S15 (D11 S14 D) (PHI S18 D) (X S16 G))
  (S16 (PHI S18 D) (V1 S17 D) (X S16 G))
  (S17 (T18 S19 G) (X S17 D))
  (S18 (T18 (O G) (F D N211) S2 D) (X S18 D))
  (S19 (C1 S19 G) (C31 S21 G) (X S20 D))
  (S20 (T18 (O G) (F D N212) S2 D) (X S20 D))
  (S21 ((PHI C3) S24 D) (N23 S22 D) (X S21 G))
  (S22 (N212 S23 D) (D11 S22 D) (X S24 D))
  (S23 (X S20 D))
  (S24 (T18 S25 G) (X S24 D))
  (S25 (PHI S27 D) (V1 S26 G) (X S25 G))
  (S26 (PHI S27 D) (D2 S28 G) (X S26 G))
  (S27 (T18 (O G) (F D N211) S2 D) (X S27 D))
  (S28 ((N1 N22) S29 G) ((C31 V23 D11) S28 G) (X S27 D))

```

(S29 ((N1 D1 V23 C2) S29 G) (V1 S27 D) (X S20 D))  
(S30 (D2 S32 G) (X S31 D))  
(S31 (T18 (O G) (F D N211) S2 D) (X S31 D))  
(S32 (D11 S33 G) (X S31 D))  
(S33 (D11 S33 G) ((V1 V21 V22) S20 D) (X S18 D)))

Texte exemple. Les premières phrases de la première partie du "Discours de la méthode" de René Descartes. Sortie de la série d'automates FD0:

```
(N1 NIL "Discours" )
(C21 NIL "de" )
(D12 NIL (T11 NIL "la" ))
(N1 NIL "méthode" )
(C22 NIL (T151 NIL "pour" ))
(D11 NIL (T1522 NIL "bien" ))
(V21 NIL "conduire" )
(D12 NIL "sa" )
(N1 NIL "raison" )
(C1 NIL "et" )
(V21 NIL "chercher" )
(D12 NIL (T11 NIL "la" ))
(N1 NIL "vérité" )
(C22 NIL "dans" )
(D12 NIL (T11 NIL "les" ))
(N1 NIL "sciences" )
(C32 NIL "." )
(C22 NIL (T142 NIL "si" ))
(D2 NIL NIL)
(D12 NIL (T131 NIL "ce" ))
(N1 NIL "discours" )
(V1 NIL "semble" )
(D11 NIL "trop" )
(V23 NIL "long" )
(C22 NIL (T151 NIL "pour" ))
(V21 NIL "être" )
(D11 NIL (T1533 NIL "tout" ))
(V23 NIL "lu" )
(C22 NIL (T12 NIL "en" ))
(D12 NIL (T132 NIL "une" ))
(N1 NIL "fois" )
(C31 NIL "," )
(N211 NIL "on" )
(N212 NIL (T11 NIL "le" ))
(V1 NIL "pourra" )
(V21 NIL "distinguer" )
(C22 NIL (T12 NIL "en" ))
(V23 NIL "six" )
(N1 NIL "parties" )
(C32 NIL "." )
(C1 NIL "Et" )
(C31 NIL "," )
(C22 NIL (T12 NIL "en" ))
(D12 NIL (T11 NIL "la" ))
(N1 NIL "première" )
(C31 NIL "," )
```



(N211 NIL "on" )  
 (V1 NIL "trouvera" )  
 (D12 NIL (T133 NIL "diverses" ))  
 (N1 NIL "considérations" )  
 (V22 NIL "touchant" )  
 (D12 NIL (T11 NIL "les" ))  
 (N1 NIL "sciences" )  
 (C32 NIL "." )  
 (C22 NIL (T12 NIL "En" ))  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "seconde" )  
 (C31 NIL "," )  
 (D12 NIL (T11 NIL "les" ))  
 (V23 NIL "principales" )  
 (N1 NIL "règles" )  
 (C21 NIL "de" )  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "méthode" )  
 (D2 NIL "que" )  
 (D12 NIL (T11 NIL "1" ))  
 (N1 NIL "auteur" )  
 (V1 NIL "a" )  
 (V23 NIL "cherchée" )  
 (C32 NIL "." )  
 (C22 NIL (T12 NIL "En" ))  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "3" )  
 (C31 NIL "," )  
 (N22 NIL "quelques-unes" )  
 (C21 NIL "de" )  
 (N22 NIL "celles" )  
 (C21 NIL "de" )  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "morale" )  
 (D2 NIL "qu" )  
 (N211 NIL "il" )  
 (V1 NIL "a" )  
 (V23 NIL "tirée" )  
 (C21 NIL "de" )  
 (D12 NIL "cette" )  
 (N1 NIL "méthode" )  
 (C32 NIL "." )  
 (C22 NIL (T12 NIL "En" ))  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "4" )  
 (C31 NIL "," )  
 (D12 NIL (T11 NIL "les" ))  
 (N1 NIL "raisons" )  
 (C22 NIL "par" )  
 (N23 NIL "lesquelles" )  
 (N211 NIL "il" )

(V1 NIL "prouve" )  
 (D12 NIL (T11 NIL "1" ))  
 (N1 NIL "existence" )  
 (C21 NIL "de" )  
 (N1 NIL "dieu" )  
 (C1 NIL "et" )  
 (C21 NIL "de" )  
 (D12 NIL (T11 NIL "1" ))  
 (N1 NIL "âme" )  
 (V23 NIL "humaine" )  
 (C31 NIL ", " )  
 (N23 NIL "qui" )  
 (V1 NIL "sont" )  
 (D12 NIL (T11 NIL "les" ))  
 (N1 NIL "fondements" )  
 (C21 NIL "de" )  
 (D12 NIL "sa" )  
 (N1 NIL "métaphysique" )  
 (C32 NIL "." )  
 (C22 NIL (T12 NIL "En" ))  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "5" )  
 (C31 NIL ", " )  
 (D12 NIL (T11 NIL "1" ))  
 (N1 NIL "ordre" )  
 (C21 NIL "des" )  
 (N1 NIL "questions" )  
 (C21 NIL "de" )  
 (N1 NIL "physique" )  
 (D2 NIL "qu" )  
 (N211 NIL "il" )  
 (V1 NIL "a" )  
 (V23 NIL "cherchées" )  
 (C31 NIL ", " )  
 (C1 NIL "et" )  
 (D11 NIL "particulièrement" )  
 (D12 NIL (T11 NIL "1" ))  
 (N1 NIL "explication" )  
 (C21 NIL "du" )  
 (N1 NIL "mouvement" )  
 (C21 NIL "du" )  
 (N1 NIL "coeur" )  
 (C1 NIL "et" )  
 (C21 NIL "de" )  
 (D12 NIL "quelques" )  
 (V23 NIL (T133 NIL "autres" ))  
 (N1 NIL "difficultés" )  
 (N23 NIL "qui" )  
 (V1 NIL "appartiennent" )  
 (C22 NIL "à" )  
 (D12 NIL (T11 NIL "la" ))

(N1 NIL "médecine" )  
 (C31 NIL "," )  
 (D11 NIL (T102 NIL "puis" ))  
 (D11 NIL "aussi" )  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "différence" )  
 (N23 NIL "qui" )  
 (V1 NIL "est" )  
 (C22 NIL "entre" )  
 (D12 NIL "notre" )  
 (N1 NIL "âme" )  
 (C1 NIL "et" )  
 (N22 NIL "celle" )  
 (C21 NIL "des" )  
 (N1 NIL "bêtes" )  
 (C32 NIL "." )  
 (C1 NIL "Et" )  
 (C22 NIL (T12 NIL "en" ))  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "dernière" )  
 (C31 NIL "," )  
 (D2 NIL "quelles" )  
 (N1 NIL "choses" )  
 (N211 NIL "il" )  
 (V1 NIL "croît" )  
 (V21 NIL "être" )  
 (V23 NIL "requisés" )  
 (C22 NIL (T151 NIL "pour" ))  
 (V21 NIL "aller" )  
 (D11 NIL (T1521 NIL "plus" ))  
 (D11 NIL (T17 NIL "avant" ))  
 (C22 NIL (T12 NIL "en" ))  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "recherche" )  
 (C21 NIL "de" )  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "nature" )  
 (D2 NIL "qu" )  
 (N211 NIL "il" )  
 (D11 NIL "n" )  
 (V1 NIL "a" )  
 (V23 NIL "été" )  
 (C31 NIL "," )  
 (C1 NIL "et" )  
 (D2 NIL "quelles" )  
 (N1 NIL "raisons" )  
 (N212 NIL (T11 NIL "1" ))  
 (V1 NIL "ont" )  
 (V23 NIL "fait" )  
 (V21 NIL "écrire" )  
 (C32 NIL "." )

(D12 NIL (T11 NIL "Le" ))  
 (V23 NIL "bon" )  
 (N1 NIL "sens" )  
 (V1 NIL "est" )  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "chose" )  
 (C21 NIL "du" )  
 (N1 NIL "monde" )  
 (D12 NIL (T11 NIL "la" ))  
 (D11 NIL "mieux" )  
 (V23 NIL "partagée" )  
 (C32 NIL ":" )  
 (D2 NIL "car" )  
 (N22 NIL (T133 NIL "chacun" ))  
 (V1 NIL "pense" )  
 (N212 NIL (T12 NIL "en" ))  
 (V21 NIL "être" )  
 (D11 NIL (T142 NIL "si" ))  
 (D11 NIL (T1522 NIL "bien" ))  
 (V23 NIL "pourvue" )  
 (C31 NIL "," )  
 (D2 NIL "que" )  
 (N22 NIL "ceux" )  
 (V23 NIL (T1531 NIL "même" ))  
 (N23 NIL "qui" )  
 (V1 NIL "sont" )  
 (D12 NIL (T11 NIL "les" ))  
 (D11 NIL (T1521 NIL "plus" ))  
 (V23 NIL "difficiles" )  
 (C22 NIL "à" )  
 (V21 NIL "contenter" )  
 (C22 NIL (T12 NIL "en" ))  
 (D12 NIL (T1532 NIL "toute" ))  
 (D12 NIL (T133 NIL "autre" ))  
 (N1 NIL "chose" )  
 (C31 NIL "," )  
 (D11 NIL "n" )  
 (V1 NIL "ont" )  
 (D11 NIL (T1523 NIL "point" ))  
 (N1 NIL "coutume" )  
 (C21 NIL "d" )  
 (N212 NIL (T12 NIL "en" ))  
 (V21 NIL "désirer" )  
 (D11 NIL (T1521 NIL "plus" ))  
 (D2 NIL "qu" )  
 (N211 NIL "ils" )  
 (N212 NIL (T12 NIL "en" ))  
 (V1 NIL "ont" )  
 (C32 NIL "." )  
 (C22 NIL (T12 NIL "En" ))  
 (N22 NIL "quoi" )

(N211 NIL "il" )  
 (D11 NIL "n" )  
 (V1 NIL "est" )  
 (D11 NIL (T1523 NIL "pas" ))  
 (V23 NIL "vraisemblable" )  
 (D2 NIL "que" )  
 (N22 NIL (T132 NIL "tous" ))  
 (N212 NIL "se" )  
 (V1 NIL "trompent" )  
 (C32 NIL ";" )  
 (D11 NIL "mais" )  
 (D11 NIL "plutôt" )  
 (N22 NIL "cela" )  
 (V1 NIL "témoigne" )  
 (D2 NIL "que" )  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "puissance" )  
 (C21 NIL "de" )  
 (D11 NIL (T1522 NIL "bien" ))  
 (V21 NIL "juger" )  
 (C31 NIL "," )  
 (C1 NIL "et" )  
 (V21 NIL "distinguer" )  
 (D12 NIL (T11 NIL "le" ))  
 (N1 NIL "vrai" )  
 (C21 NIL "d" )  
 (C22 NIL "avec" )  
 (D12 NIL (T11 NIL "le" ))  
 (N1 NIL "faux" )  
 (C31 NIL "," )  
 (N23 NIL "qui" )  
 (V1 NIL "est" )  
 (D11 NIL "proprement" )  
 (N22 NIL (T131 NIL "ce" ))  
 (D2 NIL "qu" )  
 (N211 NIL "on" )  
 (V1 NIL "nomme" )  
 (D12 NIL (T11 NIL "le" ))  
 (V23 NIL "bon" )  
 (N1 NIL "sens" )  
 (C1 NIL "ou" )  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "raison" )  
 (C31 NIL "," )  
 (V1 NIL "est" )  
 (D11 NIL "naturellement" )  
 (V23 NIL "égale" )  
 (C22 NIL (T12 NIL "en" ))  
 (D12 NIL (T132 NIL "tous" ))  
 (D12 NIL (T11 NIL "les" ))  
 (N1 NIL "hommes" )

(C32 NIL ";" )  
 (C1 NIL "et" )  
 (D11 NIL "ainsi" )  
 (D2 NIL "que" )  
 (D12 NIL (T11 NIL "la" ))  
 (N1 NIL "diversité" )  
 (C21 NIL "de" )  
 (D12 NIL "nos" )  
 (N1 NIL "opinions" )  
 (D11 NIL "ne" )  
 (V1 NIL "vient" )  
 (D11 NIL (T1523 NIL "pas" ))  
 (C21 NIL "de" )  
 (N22 NIL (T131 NIL "ce" ))  
 (D2 NIL "que" )  
 (D12 NIL (T11 NIL "les" ))  
 (N22 NIL "uns" )  
 (V1 NIL "sont" )  
 (D11 NIL (T1521 NIL "plus" ))  
 (V23 NIL "raisonnables" )  
 (D2 NIL "que" )  
 (D12 NIL (T11 NIL "les" ))  
 (N22 NIL (T133 NIL "autres" ))  
 (C31 NIL "," )  
 (D11 NIL "mais" )  
 (D11 NIL "seulement" )  
 (C21 NIL "de" )  
 (N22 NIL (T131 NIL "ce" ))  
 (D2 NIL "que" )  
 (N211 NIL (T18 NIL "nous" ))  
 (V1 NIL "conduisons" )  
 (D12 NIL "nos" )  
 (N1 NIL "pensés" )  
 (C22 NIL "par" )  
 (D12 NIL (T133 NIL "diverses" ))  
 (N1 NIL "voies" )  
 (C31 NIL "," )  
 (C1 NIL "et" )  
 (D11 NIL "ne" )  
 (V1 NIL "considérons" )  
 (D11 NIL (T1523 NIL "pas" ))  
 (D12 NIL (T11 NIL "les" ))  
 (V23 NIL (T133 NIL "mêmes" ))  
 (N1 NIL "choses" )  
 (C32 NIL "." )  
 (D2 NIL "Car" )  
 (N22 NIL (T131 NIL "ce" ))  
 (D11 NIL "n" )  
 (V1 NIL "est" )  
 (D11 NIL (T1523 NIL "pas" ))  
 (D11 NIL "assez" )

(C21 NIL "d" )  
 (V21 NIL "avoir" )  
 (D12 NIL (T11 NIL "1" ))  
 (N1 NIL "esprit" )  
 (V23 NIL "bon" )  
 (C31 NIL ", " )  
 (D11 NIL "mais" )  
 (D12 NIL (T11 NIL "le" ))  
 (N1 NIL "principal" )  
 (V1 NIL "est" )  
 (C21 NIL "de" )  
 (N212 NIL (T11 NIL "1" ))  
 (V21 NIL "appliquer" )  
 (D11 NIL (T1522 NIL "bien" ))  
 (C32 NIL "." )  
 (D12 NIL (T11 NIL "Les" ))  
 (D11 NIL (T1521 NIL "plus" ))  
 (V23 NIL "grandes" )  
 (N1 NIL "âmes" )  
 (V1 NIL "sont" )  
 (V23 NIL "capables" )  
 (C21 NIL "des" )  
 (D11 NIL (T1521 NIL "plus" ))  
 (V23 NIL "grands" )  
 (N1 NIL "vices" )  
 (C31 NIL ", " )  
 (D11 NIL "aussi" )  
 (D11 NIL (T1522 NIL "bien" ))  
 (D2 NIL "que" )  
 (C21 NIL "des" )  
 (D11 NIL (T1521 NIL "plus" ))  
 (V23 NIL "grandes" )  
 (N1 NIL "vertus" )  
 (C32 NIL ";" )  
 (C1 NIL "et" )  
 (N22 NIL "ceux" )  
 (N23 NIL "qui" )  
 (D11 NIL "ne" )  
 (V1 NIL "marchent" )  
 (D2 NIL "que" )  
 (D11 NIL "fort" )  
 (D11 NIL "lentement" )  
 (C31 NIL ", " )  
 (V1 NIL "peuvent" )  
 (V21 NIL "avancer" )  
 (D11 NIL "beaucoup" )  
 (D11 NIL "davantage" )  
 (C31 NIL ", " )  
 (C22 NIL (T141 NIL "s" ))  
 (D2 NIL NIL)  
 (N211 NIL "ils" )

(V1 NIL "suivent" )  
 (D11 NIL "toujours" )  
 (D12 NIL (T11 NIL "le" ))  
 (V23 NIL "droit" )  
 (N1 NIL "chemin" )  
 (C31 NIL ", " )  
 (D2 NIL "que" )  
 (D11 NIL "ne" )  
 (V1 NIL "font" )  
 (N22 NIL "ceux" )  
 (N23 NIL "qui" )  
 (V1 NIL "courent" )  
 (C31 NIL ", " )  
 (C1 NIL "et" )  
 (N23 NIL "qui" )  
 (N212 NIL (T141 NIL "s" ))  
 (N212 NIL (T12 NIL "en" ))  
 (V1 NIL "éloignent" )  
 (C32 NIL "." )  
 (C22 NIL (T151 NIL "Pour" ))  
 (N22 NIL "moi" )  
 (C31 NIL ", " )  
 (N211 NIL "je" )  
 (D11 NIL "n" )  
 (V1 NIL "ai" )  
 (D11 NIL "jamais" )  
 (V23 NIL "présupposé" )  
 (D2 NIL "que" )  
 (D12 NIL "mon" )  
 (N1 NIL "esprit" )  
 (V1 NIL "fût" )  
 (C22 NIL (T12 NIL "en" ))  
 (D11 NIL (T1521 NIL "rien" ))  
 (D11 NIL (T1521 NIL "plus" ))  
 (V23 NIL "parfait" )  
 (D2 NIL "que" )  
 (N22 NIL "ceux" )  
 (C21 NIL "du" )  
 (N1 NIL "commun" )  
 (C32 NIL ";" )



## Série des automates FD1:

Note: à partir de cette étape des catégories bidons sont indexées (par exemple GV23X GNX ...). Ces catégories ne servent que de points de repères dans l'algorithme.

```
(FD101 (S1 (PHI S2 D) (X S1 G))
      (S2 (D11 S3 D) (C32 (E FD102 S1 G) (RETURN)) (X S2 D))
      (S3 (D11 S3 D) ((V21 V23) S5 G) (X S4 G))
      (S4 (D11 S8 G))
      (S5 ((V1 V21 V22) S7 G) (D11 S5 G) (X S6 D))
      (S6 (D11 S6 D) (X S4 G))
      (S7 (GD11 S11 D) (N212 S7 G) (X S10 D))
      (S8 (D11 S8 G) (X (O D) S9 D))
      (S9 (D11 S9 D) (X (F G GD11) S2 G))
      (S10 (D11 S6 D) (X S10 D))
      (S11 ((V1 V21 V22 N212) S11 D)
            (D11 (O G) (F D GD11) S2 D)))

(FD102 (S1 (PHI S2 D) (X S1 G))
      (S2 (C1 S3 D) (C32 (E FD111 S1 G) (RETURN)) (X S2 D))
      (S3 (GD11 S4 G) (X S2 D))
      (S4 (GD11 (O G) S6 D) (C1 S4 G) (X S5 D))
      (S5 (C1 S2 D))
      (S6 (C1 S6 D) (GD11 (F D GD11CO) S2 D)))

(FD111 (S1 (PHI S2 D) (X S1 G))
      (S2 (V23 S3 G) (C32 (E FD112 S1 G) (RETURN)) (X S2 D))
      (S3 (GD11 S4 G) (X S7 D))
      (S4 ((V1 V21 V22) S6 G) (X S5 D))
      (S5 (GD11 (O G) S8 D) (X S5 D))
      (S6 (N212 S6 G) (GD11 S7 D) (X S5 D))
      (S7 (V23 (O G) S9 D) (X S7 D))
      (S8 (V23 S9 D))
      (S9 (GD11 S10 D) (X (F G GV23) S2 G))
      (S10 (V S11 G) (X (F G GV23) S2 G))
      (S11 (GD11 (F G GV23) S2 D)))

(FD112 (S1 (PHI S2 D) (X S1 G))
      (S2 (C1 S3 D) (C32 (E FD121 S1 G) (RETURN)) (X S2 D))
      (S3 (GV23 S4 G) (X S2 D))
      (S4 (C1 S4 G) (GV23 (O G) S6 D) (X S5 D))
      (S5 (C1 S2 D))
      (S6 (GV23 (F D GV23CO) S2 D) (C1 S6 D)))

(FD121 (S1 (PHI S2 D) (X S1 G))
      (S2 (D2 S3 D) (C32 (E FD122 S1 G) (RETURN)) (X S2 D))
      (S3 ((GD11 GV23) S4 G)
            (C32 (E FD122 S1 G) (RETURN))
            (X S2 D))
      (S4 (D2 S6 G))
      (S5 (GD11 (F D GD11) S2 D) (GV23 (F D GV23) S2 D))
      (S6 (GD11 S6 G) ((V1 V21 V22) S7 D) (X S8 D)))
```

```

(S7 (D2 (O G) S5 D) (X S7 D))
(S8 (D2 S2 D) (X S8 D)))
(FD122 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD131 S1 G) (RETURN)) (C1 S3 D) (X S2 D))
(S3 (GD11 S4 G) (X S2 D))
(S4 (GD11 (O G) S6 D) (C1 S4 G) (X S5 D))
(S5 (C1 S2 D))
(S6 (C1 S6 D) (GD11 (F D GD11CO) S2 D)))
(FD131 (S1 (PHI S2 D) (X S1 G))
(S2 (C1 S3 D) (C32 (E FD132 S1 G) (RETURN)) (X S2 D))
(S3 (V1 S4 G) ((V21 V22) S5 G) (X S2 D))
(S4 (C1 S4 G) (V1 (O G) S7 D) (X S6 D))
(S5 (C1 S5 G) (V21 S8 D) (V22 S10 D) (X S6 D))
(S6 (C1 S2 D))
(S7 (C1 S7 D) (V1 (F D V1CO) S2 D))
(S8 (C1 S8 D) (V21 (F D) S9 G) (X S2 D))
(S9 (V21 (O G V21CO) S2 D) (X S9 G))
(S10 (C1 S10 D) (V22 (F D) S11 G) (X S2 D))
(S11 (V22 (O G V22CO) S2 D) (X S11 G)))
(FD132 (S1 (PHI S2 D) (X S1 G))
(S2 (V1 S3 D) (C32 (E FD133 S1 G) (RETURN)) (X S2 D))
(S3 (GD11 (O G) (F D GD11X) S5 D) (X (F G) S4 G))
(S4 (V1 S4 G) (GD11 (O G GV1) S2 D) (X (O D GV1) S2 D))
(S5 (C32 S9 G) (D2 S6 G) (X S5 D))
(S6 (C31 S8 G) (X S7 D))
(S7 (D2 S5 D))
(S8 (GD11X (F G) S4 G) (X S8 G))
(S9 (GD11X (F D) S4 G) (X S9 G)))
(FD133 (S1 (PHI S2 D) (X S1 G))
(S2 (C1 S3 D) (C32 (E FD134 S1 G) (RETURN)) (X S2 D))
(S3 (GV1 S5 G) (X S2 D))
(S4 (C1 S2 D))
(S5 (C1 S5 G) (GV1 (O G) S6 D) (X S4 D))
(S6 (C1 S6 D) (GV1 (F D GV1CO) S2 D)))
(FD134 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD141 S1 G) (RETURN))
((V21 V22) S3 D)
(X S2 D))
(S3 (GD11 (O G) (F D GD11X) S4 D) (X (F G) S9 G))
(S4 (GD11X S7 G) (C32 S8 G) (D2 S5 G) (X S4 D))
(S5 (C31 S7 G) (X S6 D))
(S6 (D2 S4 D))
(S7 (GD11X (F G) S9 G) (X S7 G))
(S8 (GD11X (F D) S9 G) (X S8 G))
(S9 (V21 S11 G) (V22 S10 G))
(S10 (GD11 (O G GV22) S2 D) (X (O D GV22) S2 D))
(S11 (GD11 (O G GV21) S2 D) (X (O D GV21) S2 D)))
(FD141 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD142 S1 G) (RETURN)) (C1 S3 D) (X S2 D))
(S3 (N1 S4 G) (X S2 D))
(S4 (C1 S4 G) (N1 (O G) S6 D) (X S5 D))

```

```

(S5 (C1 S2 D))
(S6 (C1 S6 D) (N1 (F D GNCO) S2 D)))
(FD142 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD143 S1 G) (RETURN))
((N1 GN) S3 G)
(X S2 D))
(S3 (GV23 (O G) S4 D) (X S5 D))
(S4 ((N1 GN) (F D GN) S2 D))
(S5 ((N1 GN) S2 D)))
(FD143 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD151 S1 G) (RETURN))
((GN N1 N22) S3 G)
(X S2 D))
(S3 (D12 (O G) S5 D) (X S4 D))
(S4 ((N22 N1 GN) (O G) (F D GN) S2 D))
(S5 (D12 (O G) S5 D) ((GN N1 N22) (F D GN) S7 G))
(S6 (GN S2 D))
(S7 (D12 (O G) S5 D) (X S6 D)))
(FD151 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD16 S1 G) (RETURN)) (D12 S3 D) (X S2 D))
(S3 ((GD11 GV23) S4 G) (X S2 D))
(S4 (D12 (O G) S5 D))
(S5 (GD11 (F D GD11) S2 D) (GV23 (F D GV23) S2 D)))
(FD16 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD17 S1 G) (RETURN)) (GN S3 D) (X S2 D))
(S3 (GV23 (O G) (F D GV23X) S4 D)
(C32 (E FD17 S1 G) (RETURN))
(GN S3 D)
(X S2 D))
(S4 (C22 S2 D) (X S10 G))
(S5 ((GV1 GV21 GV22 GV23X) S7 D)
((N23 PHI D2 C31) S8 D)
(GN S13 D)
(X S5 G))
(S6 (GV23 (F D GN) S2 D))
(S7 (GV23X S11 D) (X S7 D))
(S8 (GV23X S9 G) (X S8 D))
(S9 (GN (O G) S6 D))
(S10 (GV23X S10 G) (GN S5 G))
(S11 (C32 (E FD17 S1 G) (RETURN))
(C31 S11 D)
((N23 D2) S12 G)
(GN S3 D)
(X S2 D))
(S12 (GV23X S9 G) (X S12 G))
(S13 (C21 S14 G) (X S8 D))
(S14 (GN S5 G)))
(FD17 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD18 S1 G) (RETURN)) (GD11 S3 D) (X S2 D))
(S3 (C22 S4 D) (C32 (E FD18 S1 G) (RETURN)) (X S2 D))
(S4 (GD11 S5 C) (C32 (E FD18 S1 G) (RETURN)) (X S2 D))

```

```

(S5 (GD11 (O G) S6 D) (X S5 G))
(S6 (GD11 (F D GD11) S2 D) (X S6 D)))
(FD18 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD19 S1 G) (RETURN)) (C1 S3 D) (X S2 D))
(S3 (GN S7 D) (X S2 D))
(S4 ((C1 C31) S4 G) (GN (O G) S6 D) (X S5 D))
(S5 (C1 S2 D) (X S5 D))
(S6 ((C1 C31) S6 D) (GN (F D GNCO) S2 D))
(S7 (C2 S2 D) (X S8 G))
(S8 (GN S4 G)))
(FD19 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD20 S1 G) (RETURN)) (C1 S3 D) (X S2 D))
(S3 (C21 S8 D) (GV23 S4 G) (C31 S3 D) (X S2 D))
(S4 ((C1 C31) S4 G) (GV23 (O G) S6 D) (X S5 D))
(S5 (C1 S2 D) (X S5 D))
(S6 ((C21 C1 C31) S6 D) (GV23 (F D GV23CO) S2 D))
(S7 (GV23 S6 D))
(S8 (GV23 S9 G) (X S2 D))
(S9 ((C21 C1 C31) S9 G) (GV23 S10 G) (X S5 D))
(S10 (C21 (O D) S7 D) (X S11 D))
(S11 (GV23 S5 D)))

```

Même texte exemple à la sortie de la série FD1:

```
(GN NIL (N1 NIL "Discours" ))
(C21 NIL "de" )
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "méthode" ))
(C22 NIL (T151 NIL "pour" ))
(GV21 NIL
  (GD11 NIL (D11 NIL (T1522 NIL "bien" )))
  (V21 NIL "conduire" ))
(GN NIL (D12 NIL "sa" ) (N1 NIL "raison" ))
(C1 NIL "et" )
(GV21 NIL (V21 NIL "chercher" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "vérité" ))
(C22 NIL "dans" )
(GN NIL
  (D12 NIL (T11 NIL "les" ))
  (N1 NIL "sciences" ))
(C32 NIL "." )
(C22 NIL (T142 NIL "Si" ))
(D2 NIL NIL)
(GN NIL
  (D12 NIL (T131 NIL "ce" ))
  (N1 NIL "discours" ))
(GV1 NIL (V1 NIL "semble" ))
(GV23 NIL
  (GD11 NIL (D11 NIL "trop" ))
  (V23 NIL "long" ))
(C22 NIL (T151 NIL "pour" ))
(GV21 NIL (V21 NIL "être" ))
(GV23 NIL
  (GD11 NIL (D11 NIL (T1533 NIL "tout" )))
  (V23 NIL "lu" ))
(C22 NIL (T12 NIL "en" ))
(GN NIL (D12 NIL (T132 NIL "une" )) (N1 NIL "fois" ))
(C31 NIL ",")
(N211 NIL "on" )
(N212 NIL (T11 NIL "le" ))
(GV1 NIL (V1 NIL "pourra" ))
(GV21 NIL (V21 NIL "distinguer" ))
(C22 NIL (T12 NIL "en" ))
(GN NIL
  (GN
    NIL
    (GV23 NIL (V23 NIL "six" ))
    (N1 NIL "parties" )))
(C32 NIL "." )
(C1 NIL "Et" )
(C31 NIL ",")
(C22 NIL (T12 NIL "en" ))
(GN NIL
  (D12 NIL (T11 NIL "la" ))
```

```

      (N1 NIL "première" ))
(C31 NIL "," )
(N211 NIL "on" )
(GV1 NIL (V1 NIL "trouvera" ))
(GN NIL
  (D12 NIL (T133 NIL "diverses" ))
  (N1 NIL "considérations" ))
(GV22 NIL (V22 NIL "touchant" ))
(GN NIL
  (D12 NIL (T11 NIL "les" ))
  (N1 NIL "sciences" ))
(C32 NIL "." )
(C22 NIL (T12 NIL "En" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "seconde" ))
(C31 NIL "," )
(GN NIL
  (D12 NIL (T11 NIL "les" ))
  (GN
    NIL
    (GV23 NIL (V23 NIL "principales" ))
    (N1 NIL "règles" )))
(C21 NIL "de" )
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "méthode" ))
(D2 NIL "que" )
(GN NIL (D12 NIL (T11 NIL "1" )) (N1 NIL "auteur" ))
(GV1 NIL (V1 NIL "a" ))
(GV23 NIL (V23 NIL "cherchée" ))
(C32 NIL "." )
(C22 NIL (T12 NIL "En" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "3" ))
(C31 NIL "," )
(GN NIL (N22 NIL "quelques-unes" ))
(C21 NIL "de" )
(GN NIL (N22 NIL "celles" ))
(C21 NIL "de" )
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "morale" ))
(D2 NIL "qu" )
(N211 NIL "il" )
(GV1 NIL (V1 NIL "a" ))
(GV23 NIL (V23 NIL "tirée" ))
(C21 NIL "de" )
(GN NIL (D12 NIL "cette" ) (N1 NIL "méthode" ))
(C32 NIL "." )
(C22 NIL (T12 NIL "En" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "4" ))
(C31 NIL "," )
(GN NIL (D12 NIL (T11 NIL "les" )) (N1 NIL "raisons" ))
(C22 NIL "par" )
(N23 NIL "lesquelles" )
(N211 NIL "il" )
(GV1 NIL (V1 NIL "prouve" ))
(GN NIL (D12 NIL (T11 NIL "1" )) (N1 NIL "existence" ))

```

```

(C21 NIL "de" )
(GN NIL (N1 NIL "dieu" ))
(C1 NIL "et" )
(C21 NIL "de" )
(GN NIL
  (GN NIL (D12 NIL (T11 NIL "1" )) (N1 NIL "âme" ))
  (GV23X NIL (GV23 NIL (V23 NIL "humaine" ))))
(C31 NIL ", " )
(N23 NIL "qui" )
(GV1 NIL (V1 NIL "sont" ))
(GN NIL
  (D12 NIL (T11 NIL "les" ))
  (N1 NIL "fondements" ))
(C21 NIL "de" )
(GN NIL (D12 NIL "sa" ) (N1 NIL "métaphysique" ))
(C32 NIL "." )
(C22 NIL (T12 NIL "En" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "5" ))
(C31 NIL ", " )
(GN NIL (D12 NIL (T11 NIL "1" )) (N1 NIL "ordre" ))
(C21 NIL "des" )
(GN NIL (N1 NIL "questions" ))
(C21 NIL "de" )
(GN NIL (N1 NIL "physique" ))
(D2 NIL "qu" )
(N211 NIL "11" )
(GV1 NIL (V1 NIL "a" ))
(GV23 NIL (V23 NIL "cherchées" ))
(C31 NIL ", " )
(C1 NIL "et" )
(GD11 NIL (D11 NIL "particulièrement" ))
(GN NIL
  (D12 NIL (T11 NIL "1" ))
  (N1 NIL "explication" ))
(C21 NIL "du" )
(GN NIL (N1 NIL "mouvement" ))
(C21 NIL "du" )
(GN NIL (N1 NIL "coeur" ))
(C1 NIL "et" )
(C21 NIL "de" )
(GN NIL
  (D12 NIL "quelques" )
  (GN NIL
    (GV23 NIL (V23 NIL (T133 NIL "autres" )))
    (N1 NIL "difficultés" )))
(N23 NIL "qui" )
(GV1 NIL (V1 NIL "appartiennent" ))
(C22 NIL "à" )
(GN NIL
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "médecine" ))

```

```

(C31 NIL ", " )
(GD11 NIL
  (D11 NIL (T102 NIL "puis" ))
  (D11 NIL "aussi" ))
(GN NIL
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "différence" ))
(N23 NIL "qui" )
(GV1 NIL (V1 NIL "est" ))
(C22 NIL "entre" )
(GN NIL (D12 NIL "notre" ) (N1 NIL "âme" ))
(C1 NIL "et" )
(GN NIL (N22 NIL "celle" ))
(C21 NIL "des" )
(GN NIL (N1 NIL "bêtes" ))
(C32 NIL "." )
(C1 NIL "Et" )
(C22 NIL (T12 NIL "en" ))
(GN NIL
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "dernière" ))
(C31 NIL ", " )
(D2 NIL "quelles" )
(GN NIL (N1 NIL "choses" ))
(N211 NIL "il" )
(GV1 NIL (V1 NIL "croît" ))
(GV21 NIL (V21 NIL "être" ))
(GV23 NIL (V23 NIL "requises" ))
(C22 NIL (T151 NIL "pour" ))
(GV21 NIL
  (V21 NIL "aller" )
  (GD11X NIL
    (GD11 NIL
      (D11 NIL (T1521 NIL "plus" ))
      (D11 NIL (T17 NIL "avant" ))))))
(C22 NIL (T12 NIL "en" ))
(GN NIL
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "recherche" ))
(C21 NIL "de" )
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "nature" ))
(D2 NIL "qu" )
(N211 NIL "il" )
(GV1 NIL (GD11 NIL (D11 NIL "n" )) (V1 NIL "a" ))
(GV23 NIL (V23 NIL "été" ))
(C31 NIL ", " )
(C1 NIL "et" )
(D2 NIL "quelles" )
(GN NIL (N1 NIL "raisons" ))
(N212 NIL (T11 NIL "l" ))
(GV1 NIL (V1 NIL "ont" ))

```



```

(GV23 NIL (V23 NIL "fait" ))
(GV21 NIL (V21 NIL "écrire" ))
(C32 NIL "." )
(GN NIL
  (D12 NIL (T11 NIL "Le" ))
  (GN NIL
    (GV23 NIL (V23 NIL "bon" ))
    (N1 NIL "sens" )))
(GV1 NIL (V1 NIL "est" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "chose" ))
(C21 NIL "du" )
(GN NIL (N1 NIL "monde" ))
(GV23X NIL
  (GV23 NIL
    (D12 NIL (T11 NIL "la" ))
    (GV23 NIL
      (GD11 NIL (D11 NIL "mieux" ))
      (V23 NIL "partagée" )))))
(C32 NIL ":" )
(D2 NIL "car" )
(GN NIL (N22 NIL (T133 NIL "chacun" )))
(GV1 NIL (V1 NIL "pense" ))
(N212 NIL (T12 NIL "en" ))
(GV21 NIL (V21 NIL "être" ))
(GV23 NIL
  (GD11 NIL
    (D11 NIL (T142 NIL "si" ))
    (D11 NIL (T1522 NIL "bien" ))))
  (V23 NIL "pourvue" ))
(C31 NIL "," )
(D2 NIL "que" )
(GN NIL
  (GN NIL (N22 NIL "ceux" ))
  (GV23X NIL (GV23 NIL (V23 NIL (T1531 NIL "même" )))))
(N23 NIL "qui" )
(GV1 NIL (V1 NIL "sont" ))
(GV23 NIL
  (D12 NIL (T11 NIL "les" ))
  (GV23 NIL
    (GD11 NIL (D11 NIL (T1521 NIL "plus" ))
      (V23 NIL "difficiles" ))))
(C22 NIL "à" )
(GV21 NIL (V21 NIL "contenter" ))
(C22 NIL (T12 NIL "en" ))
(GN NIL
  (D12 NIL (T1532 NIL "toute" ))
  (GN NIL
    (D12 NIL (T133 NIL "autre" ))
    (N1 NIL "chose" )))
(C31 NIL "," )
(GV1 NIL

```

```

(GD11 NIL (D11 NIL "n" ))
(V1 NIL "ont" )
(GD11X NIL (GD11 NIL (D11 NIL (T1523 NIL "point" )))))
(GN NIL (N1 NIL "coutume" ))
(C21 NIL "d" )
(N212 NIL (T12 NIL "en" ))
(GV21 NIL
  (V21 NIL "désirer" )
  (GD11X NIL (GD11 NIL (D11 NIL (T1521 NIL "plus" )))))
(D2 NIL "qu" )
(N211 NIL "ils" )
(N212 NIL (T12 NIL "en" ))
(GV1 NIL (V1 NIL "ont" ))
(C32 NIL "." )
(C22 NIL (T12 NIL "En" ))
(GN NIL (N22 NIL "quoi" ))
(N211 NIL "il" )
(GV1 NIL
  (GD11 NIL (D11 NIL "n" ))
  (V1 NIL "est" )
  (GD11X NIL (GD11 NIL (D11 NIL (T1523 NIL "pas" )))))
(GV23 NIL (V23 NIL "vraisemblable" ))
(D2 NIL "que" )
(GN NIL (N22 NIL (T132 NIL "tous" )))
(N212 NIL "se" )
(GV1 NIL (V1 NIL "trompent" ))
(C32 NIL ";" )
(GD11 NIL (D11 NIL "mais" ) (D11 NIL "plutôt" ))
(GN NIL (N22 NIL "cela" ))
(GV1 NIL (V1 NIL "témoigne" ))
(D2 NIL "que" )
(GN NIL
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "puissance" ))
(C21 NIL "de" )
(GV21 NIL
  (GD11 NIL (D11 NIL (T1522 NIL "bien" )))
  (V21 NIL "juger" ))
(C31 NIL "," )
(C1 NIL "et" )
(GV21 NIL (V21 NIL "distinguer" ))
(GN NIL (D12 NIL (T11 NIL "le" )) (N1 NIL "vrai" ))
(C21 NIL "d" )
(C22 NIL "avec" )
(GN NIL (D12 NIL (T11 NIL "le" )) (N1 NIL "faux" ))
(C31 NIL "," )
(N23 NIL "qui" )
(GV1 NIL
  (V1 NIL "est" )
  (GD11X NIL (GD11 NIL (D11 NIL "proprement" ))))
(GN NIL (N22 NIL (T131 NIL "ce" )))

```

```

(D2 NIL "qu" )
(N211 NIL "on" )
(GV1 NIL (V1 NIL "homme" ))
(GNCO NIL
  (GN NIL
    (D12 NIL (T11 NIL "le" ))
    (GN NIL
      (GV23 NIL (V23 NIL "bon" ))
      (N1 NIL "sens" )))
    (C1 NIL "ou" )
    (GN NIL
      (D12 NIL (T11 NIL "la" ))
      (N1 NIL "raison" )))
  (C31 NIL ", " )
  (GV1 NIL (V1 NIL "est" ))
  (GV23 NIL
    (GD11 NIL (D11 NIL "naturellement" ))
    (V23 NIL "égale" ))
  (C22 NIL (T12 NIL "en" ))
  (GN NIL
    (D12 NIL (T132 NIL "tous" ))
    (GN NIL
      (D12 NIL (T11 NIL "les" ))
      (N1 NIL "hommes" )))
  (C32 NIL "; " )
  (C1 NIL "et" )
  (GD11 NIL (D11 NIL "ainsi" ))
  (D2 NIL "que" )
  (GN NIL
    (D12 NIL (T11 NIL "la" ))
    (N1 NIL "diversité" ))
  (C21 NIL "de" )
  (GN NIL (D12 NIL "nos" ) (N1 NIL "opinions" ))
  (GV1 NIL
    (GD11 NIL (D11 NIL "ne" ))
    (V1 NIL "vient" )
    (GD11X NIL (GD11 NIL (D11 NIL (T1523 NIL "pas" )))))
  (C21 NIL "de" )
  (GN NIL (N22 NIL (T131 NIL "ce" )))
  (D2 NIL "que" )
  (GN NIL (D12 NIL (T11 NIL "les" )) (N22 NIL "uns" ))
  (GV1 NIL (V1 NIL "sont" ))
  (GV23 NIL
    (GD11 NIL (D11 NIL (T1521 NIL "plus" )))
    (V23 NIL "raisonnables" ))
  (D2 NIL "que" )
  (GN NIL
    (D12 NIL (T11 NIL "les" ))
    (N22 NIL (T133 NIL "autres" )))
  (C31 NIL ", " )
  (GD11 NIL (D11 NIL "mais" ) (D11 NIL "seulement" ))

```

```

(C21 NIL "de" )
(GN NIL (N22 NIL (T131 NIL "ce" )))
(D2 NIL "que" )
(N211 NIL (T18 NIL "nous" ))
(GV1 NIL (V1 NIL "conduisons" ))
(GN NIL (D12 NIL "nos" ) (N1 NIL "pensés" ))
(C22 NIL "par" )
(GN NIL
  (D12 NIL (T133 NIL "diverses" ))
  (N1 NIL "voies" ))
(C31 NIL ", " )
(C1 NIL "et" )
(GV1 NIL
  (GD11 NIL (D11 NIL "ne" ))
  (V1 NIL "considérons" )
  (GD11X NIL (GD11 NIL (D11 NIL (T1523 NIL "pas" )))))
(GN NIL
  (D12 NIL (T11 NIL "les" ))
  (GN NIL
    (GV23 NIL (V23 NIL (T133 NIL "mêmes" )))
    (N1 NIL "choses" )))
(C32 NIL "." )
(D2 NIL "Car" )
(GN NIL (N22 NIL (T131 NIL "ce" )))
(GV1 NIL
  (GD11 NIL (D11 NIL "n" ))
  (V1 NIL "est" )
  (GD11X NIL
    (GD11 NIL
      (D11 NIL (T1523 NIL "pas" ))
      (D11 NIL "assez" ))))
(C21 NIL "d" )
(GV21 NIL (V21 NIL "avoir" ))
(GN NIL (D12 NIL (T11 NIL "1" )) (N1 NIL "esprit" ))
(GV23X NIL (GV23 NIL (V23 NIL "bon" )))
(C31 NIL ", " )
(GD11 NIL (D11 NIL "mais" ))
(GN NIL
  (D12 NIL (T11 NIL "le" ))
  (N1 NIL "principal" ))
(GV1 NIL (V1 NIL "est" ))
(C21 NIL "de" )
(N212 NIL (T11 NIL "1" ))
(GV21 NIL
  (V21 NIL "appliquer" )
  (GD11X NIL (GD11 NIL (D11 NIL (T1522 NIL "bien" )))))
(C32 NIL "." )
(GN NIL
  (D12 NIL (T11 NIL "Les" ))
  (GN NIL
    (GV23 NIL

```

```

        (GD11 NIL (D11 NIL (T1521 NIL "plus" )))
        (V23 NIL "grandes" ))
    (N1 NIL "âmes" )))
(GV1 NIL (V1 NIL "sont" ))
(GV23 NIL (V23 NIL "capables" ))
(C21 NIL "des" )
(GN NIL
  (GN      NIL
    (GV23 NIL
      (GD11 NIL (D11 NIL (T1521 NIL "plus" )))
      (V23 NIL "grands" ))
      (N1 NIL "vices" )))
(C31 NIL ", " )
(GD11 NIL
  (D11 NIL "aussi" )
  (D11 NIL (T1522 NIL "bien" )))
(D2 NIL "que" )
(C21 NIL "des" )
(GN NIL
  (GN      NIL
    (GV23 NIL
      (GD11 NIL (D11 NIL (T1521 NIL "plus" )))
      (V23 NIL "grandes" ))
      (N1 NIL "vertus" )))
(C32 NIL "; " )
(C1 NIL "et" )
(GN NIL (N22 NIL "ceux" ))
(N23 NIL "qui" )
(GV1 NIL
  (GD11 NIL (D11 NIL "ne" ))
  (V1 NIL "marchent" ))
(GD11X NIL
  (GD11 NIL
    (D2 NIL "que" )
    (GD11 NIL
      (D11 NIL "fort" )
      (D11 NIL "lentement" ))))
(C31 NIL ", " )
(GV1 NIL (V1 NIL "peuvent" ))
(GV21 NIL (V21 NIL "avancer" ))
(GD11X NIL
  (GD11 NIL
    (D11 NIL "beaucoup" )
    (D11 NIL "davantage" )))
(C31 NIL ", " )
(C22 NIL (T141 NIL "s" ))
(D2 NIL NIL)
(N211 NIL "ils" )
(GV1 NIL (V1 NIL "suivent" ))
(GD11X NIL (GD11 NIL (D11 NIL "toujours" )))
(GN NIL

```

```

(D12 NIL (T11 NIL "le" ))
(GN      NIL
  (GV23 NIL (V23 NIL "droit" ))
  (N1 NIL "chemin" )))
(C31 NIL ", " )
(D2 NIL "que" )
(GV1 NIL (GD11 NIL (D11 NIL "ne" )) (V1 NIL "font" ))
(GN NIL (N22 NIL "ceux" ))
(N23 NIL "qui" )
(GV1 NIL (V1 NIL "courent" ))
(C31 NIL ", " )
(C1 NIL "et" )
(N23 NIL "qui" )
(N212 NIL (T141 NIL "s" ))
(N212 NIL (T12 NIL "en" ))
(GV1 NIL (V1 NIL "éloignent" ))
(C32 NIL "." )
(C22 NIL (T151 NIL "Pour" ))
(GN NIL (N22 NIL "moi" ))
(C31 NIL ", " )
(N211 NIL "je" )
(GV1 NIL
  (GD11 NIL (D11 NIL "n" ))
  (V1 NIL "ai" )
  (GD11X NIL (GD11 NIL (D11 NIL "jamais" ))))
(GV23 NIL (V23 NIL "présumé" ))
(D2 NIL "que" )
(GN NIL (D12 NIL "mon" ) (N1 NIL "esprit" ))
(GV1 NIL (V1 NIL "fût" ))
(C22 NIL (T12 NIL "en" ))
(GV23 NIL
  (GD11 NIL
    (D11 NIL (T1521 NIL "rien" ))
    (D11 NIL (T1521 NIL "plus" )))
  (V23 NIL "parfait" ))
(D2 NIL "que" )
(GN NIL (N22 NIL "ceux" ))
(C21 NIL "du" )
(GN NIL (N1 NIL "commun" ))
(C32 NIL "; " )

```

## Série d'automates FD2:

```

(FD20 (S1 (PHI S2 D) (X S1 G))
      (S2 (C32 (E FD21 S1 G) (RETURN)) (D2 S3 D) (X S2 D))
      (S3 (D2 S5 D) (X S4 G))
      (S4 (D2 (O G) (F D GD2) S2 D))
      (S5 (D2 S5 D) (X (F G) S6 G))
      (S6 (D2 S6 G) (X (O D GD2) S7 D))
      (S7 (GD2 (O G) (F D CP4) S2 D)))

(FD21 (S1 (PHI S2 D) (X S1 G))
      (S2 (C32 (E FD22 S1 G) (RETURN)) (N23 S3 G) (X S2 D))
      (S3 ((C2 C1) S3 G) (C31 (O G) S5 D) (X (O D) S4 D))
      (S4 (N23 (F D CP11) S6 D) (X S4 D))
      (S5 (N23 (F D CP12) S7 D) (X S5 D))
      (S6 (C31 S8 D) (X S2 D))
      (S7 (C31 S9 D) (X S2 D))
      (S8 (GD2 S10 G) (X S2 D))
      (S9 (GD2 S11 G) (X S2 D))
      (S10 (CP11 (O G) S12 D) (X S10 G))
      (S11 (CP12 (O G) S13 D) (X S11 G))
      (S12 (GD2 (F D CP11) S2 D) (X S12 D))
      (S13 (GD2 (F D CP12) S2 D) (X S13 D)))

(FD22 (S1 (PHI S2 D) (X S1 G))
      (S2 (C32 (E FD23 S1 G) (RETURN)) (GD2 S3 D) (X S2 D))
      (S3 ((GD11 C22) S4 D) (C31 S3 D) (X S2 D))
      (S4 (GD2 S5 G)
          (C32 (E FD23 S1 G) (RETURN))
          ((GD11 C22) S4 D)
          (X S2 D))
      (S5 ((C31 GD11 GV23 C22) S5 G) (GD2 (O G) S6 D))
      (S6 (GD2 (F D CP4) S2 D) (X S6 D)))

(FD23 (S1 (PHI S2 D) (X S1 G))
      (S2 (C32 (E FD24 S1 G) (RETURN)) (GD2 S3 G) (X S2 D))
      (S3 ((C1 C22) S4 D)
          (C31 S3 G)
          ((GV1 GV2 GN GD) S6 D)
          (X S5 D))
      (S4 ((C1 C31 C22) S4 D) (GD2 (O G) (F D CP2) S2 D))
      (S5 (GD2 S2 D) (X S5 D))
      (S6 (GD2 S7 D) (C31 S6 D))
      (S7 ((C3 CP GD2) S9 G) (GV1 S8 G) (X S7 D))
      (S8 (GD2 (O G) (F D CP2) S2 D) (X S8 G))
      (S9 (GD2 S2 D) (X S9 G)))

(FD24 (S1 (PHI S2 D) (X S1 G))
      (S2 (C32 (E FD25 S1 G) (RETURN)) (GV1 S3 D) (X S2 D))
      (S3 (CP S2 D)
          (C32 (E FD25 S1 G) (RETURN))
          (GV1 S4 G)
          (X S3 D))
      (S4 (N211 S7 G))

```

```

(GN S6 G)
(GV1 S5 D)
(C31 S12 D)
(X S4 G))
(S5 (GV1 S10 G) (X S5 D))
(S6 (C31 S8 D)
(C2 S11 G)
(GV1 S5 D)
(GD2 (O G) (F D CP2) S2 D)
(X S7 G))
(S7 ((GD2 C31) S8 D) (GV1 S5 D) (X S7 G))
(S8 (C1 (O D) (F D CP32) S2 D) (X S9 G))
(S9 (X (O G) (F D CP32) S2 D))
(S10 ((N212 GD11) S10 G)
(N211 S14 G)
(GD2 (O G) (F D CP31) S2 D)
(X (O D) (F D CP31) S2 D))
(S11 (GV1 S5 D) (GN S6 G) (C31 S12 D) (X S11 G))
(S12 (C1 (O D) (F D CP31) S2 D) (X S13 G))
(S13 (C31 (O G) (F D CP31) S2 D))
(S14 (GD11 S14 G)
(C1 (O D) (F D CP32) S2 D)
(X (O D) (F D CP32) S2 D)))
(FD25 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD26 S1 G) (RETURN)) (GD2 S3 D) (X S2 D))
(S3 ((GD2 CP C3) S4 G) (GV1 S5 G) (X S3 D))
(S4 (GD2 S2 D) (X S4 G))
(S5 (GD2 (O G) (F D CP2) S2 D) (X S5 G)))
(FD26 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD27 S1 G) (RETURN)) (CP3 S3 D) (X S2 D))
(S3 (C1 S4 D) (X S2 D))
(S4 (GV1 S5 G) (X S4 G))
(S5 (CP S6 D) (X S5 G))
(S6 (X S7 G))
(S7 (CP1 S9 D) (X S8 D))
(S8 (CP3 S2 D) (X S8 D))
(S9 (CP3 (O G) (F D CP12) S2 D) (X S9 D)))
(FD27 (S1 (PHI S2 D) (X S1 G))
(S2 (C32 (E FD30 S1 G) (RETURN))
(GV22 S5 G)
(CP4 S3 D)
(X S2 D))
(S3 ((CP3 GV22) S2 D) (C32 S4 G) (X S3 D))
(S4 (CP4 (O G) (F D C22) (O D) (F D CP2) S2 D) (X S4 G))
(S5 ((C31 N212 GD11) S5 G)
(CP2 (O G) (F D CP4) S6 D)
(X S6 D))
(S6 (GV22 S2 D) (X S6 D)))

```



Même texte exemple à la sortie des automates de la série FD2:

```
(GN NIL (N1 NIL "Discours" ))
(C21 NIL "de" )
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "méthode" ))
(C22 NIL (T151 NIL "pour" ))
(GV21 NIL
  (GD11 NIL (D11 NIL (T1522 NIL "bien" )))
  (V21 NIL "conduire" ))
(GN NIL (D12 NIL "sa" ) (N1 NIL "raison" ))
(C1 NIL "et" )
(GV21 NIL (V21 NIL "chercher" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "vérité" ))
(C22 NIL "dans" )
(GN NIL
  (D12 NIL (T11 NIL "les" ))
  (N1 NIL "sciences" ))
(C32 NIL "." )
(C22 NIL (T142 NIL "Si" ))
(CP2 NIL (GD2 NIL (D2 NIL NIL)))
(GN NIL
  (D12 NIL (T131 NIL "ce" ))
  (N1 NIL "discours" ))
(GV1 NIL (V1 NIL "semble" ))
(GV23 NIL
  (GD11 NIL (D11 NIL "trop" ))
  (V23 NIL "long" ))
(C22 NIL (T151 NIL "pour" ))
(GV21 NIL (V21 NIL "être" ))
(GV23 NIL
  (GD11 NIL (D11 NIL (T1533 NIL "tout" )))
  (V23 NIL "lu" ))
(C22 NIL (T12 NIL "en" ))
(GN NIL (D12 NIL (T132 NIL "une" )) (N1 NIL "fois" ))
(CP32 NIL (C31 NIL "," ))
(N211 NIL "on" )
(N212 NIL (T11 NIL "le" ))
(GV1 NIL (V1 NIL "pourra" ))
(GV21 NIL (V21 NIL "distinguer" ))
(C22 NIL (T12 NIL "en" ))
(GN NIL
  (GN
    NIL
    (GV23 NIL (V23 NIL "six" ))
    (N1 NIL "parties" )))
(C32 NIL "." )
(C1 NIL "Et" )
(C31 NIL "," )
(C22 NIL (T12 NIL "en" ))
(GN NIL
  (D12 NIL (T11 NIL "la" ))
```

```

      (N1 NIL "première" ))
(C31 NIL "," )
(N211 NIL "on" )
(GV1 NIL (V1 NIL "trouvera" ))
(GN NIL
  (D12 NIL (T133 NIL "diverses" ))
  (N1 NIL "considérations" ))
(GV22 NIL (V22 NIL "touchant" ))
(GN NIL
  (D12 NIL (T11 NIL "les" ))
  (N1 NIL "sciences" ))
(C32 NIL "." )
(C22 NIL (T12 NIL "En" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "seconde" ))
(C31 NIL "," )
(GN NIL
  (D12 NIL (T11 NIL "les" ))
  (GN NIL
    (GV23 NIL (V23 NIL "principales" ))
    (N1 NIL "règles" )))
(C21 NIL "de" )
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "méthode" ))
(CP2 NIL (GD2 NIL (D2 NIL "que" )))
(GN NIL (D12 NIL (T11 NIL "1" )) (N1 NIL "auteur" ))
(GV1 NIL (V1 NIL "a" ))
(GV23 NIL (V23 NIL "cherchée" ))
(C32 NIL "." )
(C22 NIL (T12 NIL "En" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "3" ))
(C31 NIL "," )
(GN NIL (N22 NIL "quelques-unes" ))
(C21 NIL "de" )
(GN NIL (N22 NIL "celles" ))
(C21 NIL "de" )
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "morale" ))
(CP2 NIL (GD2 NIL (D2 NIL "qu" )))
(N211 NIL "11" )
(GV1 NIL (V1 NIL "a" ))
(GV23 NIL (V23 NIL "tirée" ))
(C21 NIL "de" )
(GN NIL (D12 NIL "cette" ) (N1 NIL "méthode" ))
(C32 NIL "." )
(C22 NIL (T12 NIL "En" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "4" ))
(C31 NIL "," )
(GN NIL (D12 NIL (T11 NIL "les" )) (N1 NIL "raisons" ))
(CP11 NIL (C22 NIL "par" ) (N23 NIL "lesquelles" ))
(N211 NIL "11" )
(GV1 NIL (V1 NIL "prouve" ))
(GN NIL (D12 NIL (T11 NIL "1" )) (N1 NIL "existence" ))
(C21 NIL "de" )

```

```

(GN NIL (N1 NIL "dieu" ))
(C1 NIL "et" )
(C21 NIL "de" )
(GN NIL
  (GN NIL (D12 NIL (T11 NIL "1" )) (N1 NIL "âme" ))
  (GV23X NIL (GV23 NIL (V23 NIL "humaine" ))))
(CP12 NIL (C31 NIL "," ) (N23 NIL "qui" ))
(GV1 NIL (V1 NIL "sont" ))
(GN NIL
  (D12 NIL (T11 NIL "les" ))
  (N1 NIL "fondements" ))
(C21 NIL "de" )
(GN NIL (D12 NIL "sa" ) (N1 NIL "métaphysique" ))
(C32 NIL "." )
(C22 NIL (T12 NIL "En" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "5" ))
(C31 NIL "," )
(GN NIL (D12 NIL (T11 NIL "1" )) (N1 NIL "ordre" ))
(C21 NIL "des" )
(GN NIL (N1 NIL "questions" ))
(C21 NIL "de" )
(GN NIL (N1 NIL "physique" ))
(CP2 NIL (GD2 NIL (D2 NIL "qu" )))
(N211 NIL "il" )
(GV1 NIL (V1 NIL "a" ))
(GV23 NIL (V23 NIL "cherchées" ))
(C31 NIL "," )
(C1 NIL "et" )
(GD11 NIL (D11 NIL "particulièrement" ))
(GN NIL
  (D12 NIL (T11 NIL "1" ))
  (N1 NIL "explication" ))
(C21 NIL "du" )
(GN NIL (N1 NIL "mouvement" ))
(C21 NIL "du" )
(GN NIL (N1 NIL "coeur" ))
(C1 NIL "et" )
(C21 NIL "de" )
(GN NIL
  (D12 NIL "quelques" )
  (GN
    NIL
    (GV23 NIL (V23 NIL (T133 NIL "autres" )))
    (N1 NIL "difficultés" )))
(CP11 NIL (N23 NIL "qui" ))
(GV1 NIL (V1 NIL "appartiennent" ))
(C22 NIL "à" )
(GN NIL
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "médecine" ))
(C31 NIL "," )
(GD11 NIL

```

```

      (D11 NIL (T102 NIL "puis" ))
      (D11 NIL "aussi" ))
(GN NIL
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "différence" ))
(CP11 NIL (N23 NIL "qui" ))
(GV1 NIL (V1 NIL "est" ))
(C22 NIL "entre" )
(GN NIL (D12 NIL "notre" ) (N1 NIL "3me" ))
(C1 NIL "et" )
(GN NIL (N22 NIL "celle" ))
(C21 NIL "des" )
(GN NIL (N1 NIL "bêtes" ))
(C32 NIL "." )
(C1 NIL "Et" )
(C22 NIL (T12 NIL "en" ))
(GN NIL
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "dernière" ))
(C31 NIL "," )
(CP2 NIL (GD2 NIL (D2 NIL "quelles" )))
(GN NIL (N1 NIL "choses" ))
(N211 NIL "il" )
(GV1 NIL (V1 NIL "croît" ))
(GV21 NIL (V21 NIL "être" ))
(GV23 NIL (V23 NIL "requisés" ))
(C22 NIL (T151 NIL "pour" ))
(GV21 NIL
  (V21 NIL "aller" )
  (GD11X NIL
    (GD11 NIL
      (D11 NIL (T1521 NIL "plus" ))
      (D11 NIL (T17 NIL "avant" ))))))
(C22 NIL (T12 NIL "en" ))
(GN NIL
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "recherche" ))
(C21 NIL "de" )
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "nature" ))
(CP2 NIL (GD2 NIL (D2 NIL "qu" )))
(N211 NIL "il" )
(GV1 NIL (GD11 NIL (D11 NIL "n" )) (V1 NIL "a" ))
(GV23 NIL (V23 NIL "été" ))
(C31 NIL "," )
(C1 NIL "et" )
(CP2 NIL (GD2 NIL (D2 NIL "quelles" )))
(GN NIL (N1 NIL "raisons" ))
(N212 NIL (T11 NIL "1" ))
(GV1 NIL (V1 NIL "ont" ))
(GV23 NIL (V23 NIL "fait" ))
(GV21 NIL (V21 NIL "écrire" ))

```

```

(C32 NIL "." )
(GN NIL
  (D12 NIL (T11 NIL "Le" ))
  (GN NIL
    (GV23 NIL (V23 NIL "bon" ))
    (N1 NIL "sens" )))
(GV1 NIL (V1 NIL "est" ))
(GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "chose" ))
(C21 NIL "du" )
(GN NIL (N1 NIL "monde" ))
(GV23X NIL
  (GV23 NIL
    (D12 NIL (T11 NIL "la" ))
    (GV23 NIL
      (GD11 NIL (D11 NIL "mieux" ))
      (V23 NIL "partagée" )))))
(C32 NIL ":" )
(CP2 NIL (GD2 NIL (D2 NIL "car" )))
(GN NIL (N22 NIL (T133 NIL "chacun" )))
(GV1 NIL (V1 NIL "pense" ))
(N212 NIL (T12 NIL "en" ))
(GV21 NIL (V21 NIL "être" ))
(GV23 NIL
  (GD11 NIL
    (D11 NIL (T142 NIL "si" ))
    (D11 NIL (T1522 NIL "bien" )))
  (V23 NIL "pourvue" ))
(C31 NIL "," )
(GD2 NIL (D2 NIL "que" ))
(GN NIL
  (GN NIL (N22 NIL "ceux" ))
  (GV23X NIL (GV23 NIL (V23 NIL (T1531 NIL "même" )))))
(CP11 NIL (N23 NIL "qui" ))
(GV1 NIL (V1 NIL "sont" ))
(GV23 NIL
  (D12 NIL (T11 NIL "les" ))
  (GV23 NIL
    (GD11 NIL (D11 NIL (T1521 NIL "plus" ))
      (V23 NIL "difficiles" ))))
(C22 NIL "à" )
(GV21 NIL (V21 NIL "contenter" ))
(C22 NIL (T12 NIL "en" ))
(GN NIL
  (D12 NIL (T1532 NIL "toute" ))
  (GN NIL
    (D12 NIL (T133 NIL "autre" ))
    (N1 NIL "chose" )))
(CP31 NIL (C31 NIL "," ))
(GV1 NIL
  (GD11 NIL (D11 NIL "n" ))
  (V1 NIL "ont" ))

```

```

      (GD11X NIL (GD11 NIL (D11 NIL (T1523 NIL "point" )))))
(GN NIL (N1 NIL "coutume" ))
(C21 NIL "d" )
(N212 NIL (T12 NIL "en" ))
(GV21 NIL
  (V21 NIL "désirer" )
  (GD11X NIL (GD11 NIL (D11 NIL (T1521 NIL "plus" )))))
(CP2 NIL (GD2 NIL (D2 NIL "qu" )))
(N211 NIL "ils" )
(N212 NIL (T12 NIL "en" ))
(GV1 NIL (V1 NIL "ont" ))
(C32 NIL "." )
(C22 NIL (T12 NIL "En" ))
(GN NIL (N22 NIL "quoi" ))
(N211 NIL "il" )
(GV1 NIL
  (GD11 NIL (D11 NIL "n" ))
  (V1 NIL "est" )
  (GD11X NIL (GD11 NIL (D11 NIL (T1523 NIL "pas" )))))
(GV23 NIL (V23 NIL "vraisemblable" ))
(CP2 NIL (GD2 NIL (D2 NIL "que" )))
(GN NIL (N22 NIL (T132 NIL "tous" )))
(N212 NIL "se" )
(GV1 NIL (V1 NIL "trompent" ))
(C32 NIL ";" )
(GD11 NIL (D11 NIL "mais" )) (D11 NIL "plutôt" ))
(GN NIL (N22 NIL "cela" ))
(GV1 NIL (V1 NIL "témoigne" ))
(GD2 NIL (D2 NIL "que" ))
(GN NIL
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "puissance" ))
(C21 NIL "de" )
(GV21 NIL
  (GD11 NIL (D11 NIL (T1522 NIL "bien" )))
  (V21 NIL "juger" ))
(C31 NIL "," )
(C1 NIL "et" )
(GV21 NIL (V21 NIL "distinguer" ))
(GN NIL (D12 NIL (T11 NIL "le" )) (N1 NIL "vrai" ))
(C21 NIL "d" )
(C22 NIL "avec" )
(GN NIL (D12 NIL (T11 NIL "le" )) (N1 NIL "faux" ))
(CP12 NIL (C31 NIL "," ) (N23 NIL "qui" ))
(GV1 NIL
  (V1 NIL "est" )
  (GD11X NIL (GD11 NIL (D11 NIL "proprement" ))))
(GN NIL (N22 NIL (T131 NIL "ce" )))
(CP2 NIL (GD2 NIL (D2 NIL "qu" )))
(N211 NIL "on" )
(GV1 NIL (V1 NIL "bonne" ))

```

```

(GNCO NIL
  (GN NIL
    (D12 NIL (T11 NIL "le" ))
    (GN NIL
      (GV23 NIL (V23 NIL "bon" ))
      (N1 NIL "sens" )))
    (C1 NIL "ou" )
    (GN NIL
      (D12 NIL (T11 NIL "la" ))
      (N1 NIL "raison" )))
  (CP31 NIL (C31 NIL "," ))
  (GV1 NIL (V1 NIL "est" ))
  (GV23 NIL
    (GD11 NIL (D11 NIL "naturellement" ))
    (V23 NIL "égale" ))
  (C22 NIL (T12 NIL "en" ))
  (GN NIL
    (D12 NIL (T132 NIL "tous" ))
    (GN NIL
      (D12 NIL (T11 NIL "les" ))
      (N1 NIL "hommes" )))
  (C32 NIL ";" )
  (C1 NIL "et" )
  (GD11 NIL (D11 NIL "ainsi" ))
  (CP2 NIL (GD2 NIL (D2 NIL "que" )))
  (GN NIL
    (D12 NIL (T11 NIL "la" ))
    (N1 NIL "diversité" ))
  (C21 NIL "de" )
  (GN NIL (D12 NIL "nos" ) (N1 NIL "opinions" ))
  (GV1 NIL
    (GD11 NIL (D11 NIL "ne" ))
    (V1 NIL "vient" )
    (GD11X NIL (GD11 NIL (D11 NIL (T1523 NIL "pas" )))))
  (C21 NIL "de" )
  (GN NIL (N22 NIL (T131 NIL "ce" )))
  (CP2 NIL (GD2 NIL (D2 NIL "que" )))
  (GN NIL (D12 NIL (T11 NIL "les" )) (N22 NIL "uns" ))
  (GV1 NIL (V1 NIL "sont" ))
  (GV23 NIL
    (GD11 NIL (D11 NIL (T1521 NIL "plus" ))
    (V23 NIL "raisonnables" ))
  (GD2 NIL (D2 NIL "que" ))
  (GN NIL
    (D12 NIL (T11 NIL "les" ))
    (N22 NIL (T133 NIL "autres" )))
  (C31 NIL "," )
  (GD11 NIL (D11 NIL "mais" ) (D11 NIL "seulement" ))
  (C21 NIL "de" )
  (GN NIL (N22 NIL (T131 NIL "ce" )))
  (CP2 NIL (GD2 NIL (D2 NIL "que" )))

```

```

(N211 NIL (T18 NIL "nous" ))
(GV1 NIL (V1 NIL "conduisons" ))
(GN NIL (D12 NIL "nos" ) (N1 NIL "pensés" ))
(C22 NIL "par" )
(GN NIL
  (D12 NIL (T133 NIL "diverses" ))
  (N1 NIL "voies" ))
(C31 NIL "," )
(C1 NIL "et" )
(CP31 NIL)
(GV1 NIL
  (GD11 NIL (D11 NIL "ne" ))
  (V1 NIL "considérons" )
  (GD11X NIL (GD11 NIL (D11 NIL (T1523 NIL "pas" )))))
(GN NIL
  (D12 NIL (T11 NIL "les" ))
  (GN NIL
    (GV23 NIL (V23 NIL (T133 NIL "mêmes" )))
    (N1 NIL "choses" )))
(C32 NIL "." )
(CP2 NIL (GD2 NIL (D2 NIL "Car" )))
(GN NIL (N22 NIL (T131 NIL "ce" )))
(GV1 NIL
  (GD11 NIL (D11 NIL "n" ))
  (V1 NIL "est" )
  (GD11X NIL
    (GD11 NIL
      (D11 NIL (T1523 NIL "pas" ))
      (D11 NIL "assez" ))))
(C21 NIL "d" )
(GV21 NIL (V21 NIL "avoir" ))
(GN NIL (D12 NIL (T11 NIL "1" )) (N1 NIL "esprit" ))
(GV23X NIL (GV23 NIL (V23 NIL "bon" )))
(CP32 NIL (C31 NIL "," ))
(GD11 NIL (D11 NIL "mais" ))
(GN NIL
  (D12 NIL (T11 NIL "le" ))
  (N1 NIL "principal" ))
(GV1 NIL (V1 NIL "est" ))
(C21 NIL "de" )
(N212 NIL (T11 NIL "1" ))
(GV21 NIL
  (V21 NIL "appliquer" )
  (GD11X NIL (GD11 NIL (D11 NIL (T1522 NIL "bien" )))))
(C32 NIL "." )
(GN NIL
  (D12 NIL (T11 NIL "Les" ))
  (GN NIL
    (GV23 NIL
      (GD11 NIL (D11 NIL (T1521 NIL "plus" )))
      (V23 NIL "grandes" ))

```



```

        (N1 NIL "âmes" )))
(GV1 NIL (V1 NIL "sont" ))
(GV23 NIL (V23 NIL "capables" ))
(C21 NIL "des" )
(GN NIL
  (GN      NIL
    (GV23 NIL
      (GD11 NIL (D11 NIL (T1521 NIL "plus" )))
      (V23 NIL "grands" ))
      (N1 NIL "vices" )))
(C31 NIL ", " )
(GD11 NIL
  (D11 NIL "aussi" )
  (D11 NIL (T1522 NIL "bien" )))
(GD2 NIL (D2 NIL "que" ))
(C21 NIL "des" )
(GN NIL
  (GN      NIL
    (GV23 NIL
      (GD11 NIL (D11 NIL (T1521 NIL "plus" )))
      (V23 NIL "grandes" ))
      (N1 NIL "vertus" )))
(C32 NIL "; " )
(C1 NIL "et" )
(GN NIL (N22 NIL "ceux" ))
(CP11 NIL (N23 NIL "qui" ))
(GV1 NIL
  (GD11 NIL (D11 NIL "ne" ))
  (V1 NIL "marchent" ))
(GD11X NIL
  (GD11 NIL
    (D2 NIL "que" )
    (GD11 NIL
      (D11 NIL "fort" )
      (D11 NIL "lentement" ))))
(CP31 NIL (C31 NIL ", " ))
(GV1 NIL (V1 NIL "peuvent" ))
(GV21 NIL (V21 NIL "avancer" ))
(GD11X NIL
  (GD11 NIL
    (D11 NIL "beaucoup" )
    (D11 NIL "davantage" )))
(C31 NIL ", " )
(C22 NIL (T141 NIL "s" ))
(CP2 NIL (GD2 NIL (D2 NIL NIL)))
(N211 NIL "ils" )
(GV1 NIL (V1 NIL "suivent" ))
(GD11X NIL (GD11 NIL (D11 NIL "toujours" )))
(GN NIL
  (D12 NIL (T11 NIL "le" ))
  (GN      NIL
    (GV23 NIL
      (GD11 NIL (D11 NIL (T1521 NIL "plus" )))
      (V23 NIL "grands" ))
      (N1 NIL "vices" )))

```

```

        (GV23 NIL (V23 NIL "droit" ))
        (N1 NIL "chemin" )))
(C31 NIL ", " )
(CP2 NIL (GD2 NIL (D2 NIL "que" )))
(GV1 NIL (GD11 NIL (D11 NIL "ne" )) (V1 NIL "font" ))
(GN NIL (N22 NIL "ceux" ))
(CP11 NIL (N23 NIL "qui" ))
(GV1 NIL (V1 NIL "courent" ))
(CP12 NIL
  (C31 NIL ", " )
  (C1 NIL "et" )
  (N23 NIL "qui" ))
(N212 NIL (T141 NIL "s" ))
(N212 NIL (T12 NIL "en" ))
(GV1 NIL (V1 NIL "éloignent" ))
(C32 NIL ". " )
(C22 NIL (T151 NIL "Pour" ))
(GN NIL (N22 NIL "moi" ))
(C31 NIL ", " )
(N211 NIL "je" )
(GV1 NIL
  (GD11 NIL (D11 NIL "n" ))
  (V1 NIL "ai" )
  (GD11X NIL (GD11 NIL (D11 NIL "jamais" ))))
(GV23 NIL (V23 NIL "présuné" ))
(CP2 NIL (GD2 NIL (D2 NIL "que" )))
(GN NIL (D12 NIL "mon" ) (N1 NIL "esprit" ))
(GV1 NIL (V1 NIL "fût" ))
(C22 NIL (T12 NIL "en" ))
(GV23 NIL
  (GD11 NIL
    (D11 NIL (T1521 NIL "rien" ))
    (D11 NIL (T1521 NIL "plus" )))
  (V23 NIL "parfait" ))
(GD2 NIL (D2 NIL "que" ))
(GN NIL (N22 NIL "ceux" ))
(C21 NIL "du" )
(GN NIL (N1 NIL "commun" ))
(C32 NIL "; " )

```

## Série d'automates FD3-4-5:

```

(FD30 (S1 (C32 S2 G) (X S1 D))
      (S2 (PHI (E FD40 S1 D) (RETURN) S2 D)
          (GN (E FD311 S1 D) S3 G)
          ((GV21 GV22) (E FD321 S2 G) S4 G)
          (CP (E FD331 S1 D) S5 G)
          (GV23X S2 G)
          (GV23 (O G) (E FD341 S2 D) S6 G)
          (X S2 G))
      (S3 (GN S7 D))
      (S4 ((GV21 GV22) S2 G))
      (S5 (GP S2 G))
      (S6 (GV23 S2 G))
      (S7 ((GP31 GP1) S9 G) (C31 S7 D) (GP32 S15 G) (X S8 G))
      (S8 (GN S2 G) (X S8 G))
      (S9 (GN S10 G) (C31 S9 G))
      (S10 ((C2 GD2 GD11 C1) S10 G)
          ((GV1 GV2 GN) S11 D)
          (C31 S14 G)
          (X S13 D))
      (S11 (GN S2 G) (X S11 D))
      (S12 ((GN GV1 GV21 GV22 C) S11 D) (GV23 S12 G) (X S13 D))
      (S13 (GP32 (E FD332 S1 G) S5 G)
          (GP1 (E FD312 S1 G) S3 G)
          (GP31 (E FD333 S1 G) S5 G)
          (X S13 D))
      (S14 (CP11 S11 D) (X S13 D))
      (S15 (GNY S10 G) (C31 S15 G) (X S2 G)))
(FD311 (S1 (GV21 (R I GN G) S4 G)
          (C21 S5 D)
          (C22 S8 G)
          ((GP2 GP11) S29 G)
          (GV22 S19 G)
          (GP12 S13 G)
          (X S25 G))
      (S2 (GN (O G) S3 D))
      (S3 (C2 S3 D)
          (GNZ (F D GN) S1 D)
          (X (F D GN) (E FD314 S1 D) (RETURN) S1 D))
      (S4 (GN (O G) S3 D))
      (S5 ((GV1 GV21 GV22) (R D GN G) S6 G)
          (GN S8 G)
          (X S7 G))
      (S6 (C2 S6 G) (GN (O G) S3 D))
      (S7 (GN (E FD314 S1 D) (RETURN)) (X S7 G))
      (S8 (GN (O G) (F D GNX) S9 G) (X S8 G))
      (S9 ((CP PHI C31) S10 D)
          ((GV1 GV21 GV22) S11 D)
          (X S9 G))

```

```

(S10 (GNX S12 D) (X S10 D))
(S11 (GNX (E FD314 S1 D) (RETURN)) (X S11 D))
(S12 (C2 S12 D) ((GN GV21 GV22 GV1) (R D GN G) S6 G))
(S13 (GN S14 G))
(S14 ((GV22 GD2) S26 G) (C2 S15 G) (X S16 D))
(S15 (GN S25 D) ((C2 C1 GV23 GV22) S15 G) (X S16 D))
(S16 (C32 S30 G) (GP12 (R D GN G) S17 G) (X S16 D))
(S17 (GN (O G) S18 D))
(S18 (GP12 (F D GNY) (E FD314 S1 D) (RETURN) S1 D))
(S19 (GN S28 G))
(S20 ((GP2 GV22 GP11) (R D GN G) S24 G) (X S20 D))
(S21 (GN (O G) S22 D) (X S20 D))
(S22 (C21 S22 D) (GN (R D GN G) (F D GN) S23 D))
(S23 (GV22 (R D GN G) S24 G))
(S24 (GN (O G) S3 D))
(S25 (GN (E FD314 S1 D) (RETURN)) (X S25 D))
(S26 (N212 S26 G) (C31 S27 D) (X S16 D))
(S27 (GP12 S25 G) (X S27 D))
(S28 (C21 S21 G) (X S20 D))
(S29 (GNZ (E FD314 S1 D) (RETURN)) (X S20 D))
(S30 (GN (E FD314 S1 D) (RETURN)) (X S30 G))
(FD312 (S1 (GN (O G) S2 D) (X S1 G))
        (S2 (GP1 (R D GN G)
                (F D GN)
                (E FD314 S1 D)
                (RETURN)
                S1
                D)
                (X S2 D)))
(FD313 (S1 (X S2 G))
        (S2 (GP32 S2 G) (CP2 (O G) S3 D) (X S4 D))
        (S3 (GP32 (F D GP2) S5 D))
        (S4 (GP32 S5 D))
        (S5 (X (RETURN))))
(FD314 (S1 (C31 S1 D) (C1 S2 D) (X S3 G))
        (S2 (C21 S4 D) (C22 S5 D) (GN S6 G) (X S3 G))
        (S3 (GN S16 D) (X S3 G))
        (S4 (GN S8 G) (X S3 G))
        (S5 (GN S9 G) (X S3 G))
        (S6 (C S6 G) (GN S22 G))
        (S7 (GN (F D GNCO) S16 D) (X S7 D))
        (S8 (GN S10 G) (C S8 G))
        (S9 (GN S11 G) (C S9 G))
        (S10 (C21 S19 G) (X S3 D))
        (S11 (C21 S19 G) (C22 (O D) S13 D) (X S3 D))
        (S12 (C2 S14 D) (X S12 D))
        (S13 (C22 S15 D) (X S13 D))
        (S14 (GN (F D GNCO) S16 D))
        (S15 (GN (F D GNCO) S16 D))
        (S16 (X (RETURN)))
        (S17 (GN (O G) S18 D) (X S23 D))

```

```

(S18 (GN (R D GN G) (F D GN) S1 D) (X S18 D))
(S19 (GN S21 G) (X S20 D))
(S20 (C21 (O D) S12 D) (GN S20 D))
(S21 (C2 S17 D) (X S20 D))
(S22 (C21 S17 G) (X S23 D))
(S23 (GN (O G) S7 D) (X S23 D)))
(FD321 (S1 (X S2 D))
(S2 (GV21 (O G) (F D GV21X) S3 D)
(GV22 (O G) (F D GV22X) S3 D)
(X S2 D))
(S3 ((GD2 GD11 GV23X C30 C21 GP12 GP11) S3 D)
((GN GP2) (R P1 (GV21 GV22) G) S3 D)
(C31 S7 G)
(C1 S6 D)
(C22 S4 D)
(X S23 G))
(S4 ((C2 GD2 GD11) S4 D)
((GN GV2 GP2) (R P2 (GV21 GV22) G) S5 D)
(X S23 G))
(S5 ((GV23 C21) S5 D)
(C1 S8 D)
(GN (R P1 (GV21 GV22) G) S5 D)
(C31 S7 G)
(C22 S4 D)
(X S23 G))
(S6 (GN (R P1 (GV21 GV22) G) S6 D)
((C21 GD2 GV23) S6 D)
(X S23 G))
(S7 ((GV21 GV22) S9 G) (X S7 G))
(S8 ((GV23 GD11 GD2) S8 D)
((GN GP2) (R P2 (GV21 GV22) G) S5 D)
(X S23 G))
(S9 ((C1 C22 W2) S9 G)
(C31 S10 D)
(C21 S32 G)
(X S11 D))
(S10 ((GV21 GV22) S12 D) (X S10 D))
(S11 ((GV21 GV22) S13 D) (X S11 D))
(S12 (C31 S23 G) (X S12 D))
(S13 (C31 S14 D) (X S13 D))
(S14 (GD11 S14 D)
(C1 S23 G)
((GD2 GP2 C22) S16 G)
(X S15 G))
(S15 (GD11 S15 G) (C31 (O G) (F D C30) S33 D))
(S16 (C31 (O G) (F D C31X) S17 G) (X S16 G))
(S17 (CP S18 D) ((PHI GV22) S21 D) (X S17 G))
(S18 (X S19 G))
(S19 ((CP1 CP2) S20 G) (X S21 D))
(S20 (GNZ S21 D) ((GN GV2) S22 D) (X S21 D))
(S21 (GV1 S27 D) (C31X (O G) (F D C30) S33 D) (X S21 D))

```

```

(S22 (C31X S23 G) (X S22 D))
(S23 ((C1 C2 C31) S23 G)
  (C30 (O G) (F D C31) S23 G)
  (X (F D) S24 D))
(S24 (GV21X S25 G) (GV22X S26 G) (X S24 G))
(S25 ((N212 GD11) S25 G) (X (O D GV21) S29 D))
(S26 ((N212 GD11) S26 G) (X (O D GV22) S29 D))
(S27 (GD11 S22 D) ((GV21 GV22) S31 D) (X S30 D))
(S28 (C31X (O G) (F D C30) S33 D) (X S28 D))
(S29 ((GV21 GV22) (E FD322 S1 D) (RETURN) S1 D))
(S30 ((GV21 GV22) S31 D) (X S30 D))
(S31 (GD11 S22 D)
  (C31X (O G) (F D C30) S33 D)
  (X S28 D))
(S32 (GNZ S9 G) (GN S12 D) (C31 S10 D) (X S11 D))
(S33 ((GD2 C30 C2 GD11 GN GP2 N212 GV2) S33 D)
  (C31 S7 G)
  (C1 S6 D)
  (X S23 G)))
(FD322 (S1 (C31 S1 D) (C1 S2 D) (X S3 G))
  (S2 (GD11 S2 D)
    (C21 S9 D)
    (C22 S10 D)
    (GV21 S5 G)
    (GV22 S6 G)
    (X S3 G))
  (S3 ((GV21 GV22) S4 D) (X S3 G))
  (S4 (X (RETURN)))
  (S5 ((C GD11 GV23) S5 G) (GV21 (O G) S8 D) (X S4 D))
  (S6 ((C GD11 GV23) S6 G) (GV22 (O G) S7 D) (X S4 D))
  (S7 (GV22 (F D GV22CO) S4 D) (X S7 D))
  (S8 (GV21 (F D GV21CO) S4 D) (X S8 D))
  (S9 (GV21 S11 G) (GV22 S12 G) (X S3 G))
  (S10 (GV21 S13 G) (GV22 S14 G) (X S3 G))
  (S11 ((C GD11 GV23) S11 G) (GV21 S15 G) (X S4 D))
  (S12 ((C GD11 GV23) S12 G) (GV22 S16 G) (X S4 D))
  (S13 ((C GD11 GV23) S13 G) (GV21 S17 G) (X S4 D))
  (S14 ((C GD11 GV23) S14 G) (GV22 S18 G) (X S4 D))
  (S15 (C21 (O D) S19 D) (X S3 D))
  (S16 (C21 (O D) S20 D) (X S3 D))
  (S17 (C22 (O D) S21 D) (X S3 D))
  (S18 (C22 (O D) S22 D) (X S3 D))
  (S19 (C2 S23 D) (X S19 D))
  (S20 (C2 S24 D) (X S20 D))
  (S21 (C2 S25 D) (X S21 D))
  (S22 (C2 S26 D) (X S22 D))
  (S23 (GV21 (F D GV21CO) S4 D))
  (S24 (GV22 (F D GV22CO) S4 D))
  (S25 (GV21 (F D GV21CO) S4 D))
  (S26 (GV22 (F D GV22CO) S4 D)))
(FD331 (S1 (GV1 S2 D) ((CP C32 GP3) (F G) S35 G) (X S1 D))

```

```

(S2 ((GP12 GD11 GV23X C30 C21) S2 D)
  (GD2 S18 D)
  (C22 S32 D)
  (C1 S23 G)
  (C31 S3 D)
  (GN (R P1 GV1 G) S33 D)
  (GV2 S33 D)
  (X S29 G))
(S3 ((GD2 GV22 C1 GD11 GP2 GN C22) S5 G)
  (N212 S16 D)
  (X S4 G))
(S4 (C31 (O G) (F D C30) S21 D))
(S5 (CP S6 D) (PHI S7 D) (X S5 G))
(S6 (X S9 G))
(S7 (GV22 S13 D) (GV1 S8 D) (X S7 D))
(S8 (C31 (O G) (F D C30) S21 D) (X S8 D))
(S9 (CP1 S10 G) (CP2 S11 G) (X S7 D))
(S10 ((GN GV1 GV21 GV22 C) S13 D) (X S14 D))
(S11 (C22 S12 D) ((GN GV2) S13 D) (X S7 D))
(S12 (C31 S15 D) (X S12 D))
(S13 (GV1 S19 D) (X S13 D))
(S14 (GV1 S20 D) (X S14 D))
(S15 ((GD2 GP2) S19 G) (X S8 G))
(S16 (GV22 S5 G) (X S17 G))
(S17 (C31 (O G) (F D C30) S21 D) (X S17 G))
(S18 (GP32 S33 D) (X S33 G))
(S19 (C31 (F G) S30 G) (X S19 D))
(S20 (C31 (F D) S30 G) (X S20 D))
(S21 ((GD11 GP1 C30 C2 GV2 GD2 GN N2 GP2) S21 D)
  (C1 S23 G)
  (C31 S3 D)
  (X S29 G))
(S22 (GV1 S8 D) (X S22 D))
(S23 ((GD11C0 GV23C0) S24 D) (X S25 D))
(S24 (C1 S29 G))
(S25 (C1 (O G) (F D C1X) S26 D))
(S26 ((GP3 GP2) S28 G) (C32 S27 G) (X S26 D))
(S27 (C1X S2 D) (X S27 G))
(S28 (C1X S29 G) (X S28 G))
(S29 ((C1 C2) S29 G)
  (C31 (F G) S30 G)
  (GV1 (F D) S36 G)
  (C30 (O G) (F D C31) (F G) S30 G)
  (X (F D) S30 G))
(S30 (GV1 S36 G) ((CP PHI) S31 D) (X S30 G))
(S31 (X S41 G))
(S32 ((GP1 GD2 C2) S32 D)
  (C31 S3 D)
  ((GP2 GN GV2) (R P2 GV1 G) S34 D)
  (GD11 S32 D)
  (C22 S34 D))

```

```

(X S29 G))
(S33 ((C1 GD11 GV23X GP1 C21 GD2) S33 D)
(C31 S3 D)
(C22 S32 D)
(GN (R P1 GV1 G) S34 D)
(GV2 S34 D)
(X S29 G))
(S34 ((GP1 GD11 C22 GV23X GD2 C21 C1) S34 D)
(C31 S3 D)
(GN (R P1 GV1 G) S34 D)
(GV2 S34 D)
(X S29 G))
(S35 ((CP PHI) S31 D) (X S30 G))
(S36 (N212 S36 G) (X (O D GV1) S37 D))
(S37 (GV1 (F D) S38 G))
(S38 (N211 (R I GV1 D) S38 G)
((GD2 N212 GD11 C1) S38 G)
(X S39 D))
(S39 (X S41 G))
(S40 (GP (E FD334 S1 D) (RETURN)) (X S40 D))
(S41 (GD11 S41 G)
((N2 GP1 GP2 GP3 GV2 GN) S42 G)
(GP4 (R P2 GV1 D) S41 G)
(PHI (O D GP0) S40 D)
(CP11 (O G GP11) S40 G)
(CP12 (O G GP12) S40 G)
(CP2 (O G GP2) S40 G)
(CP31 (O G GP31) S40 G)
(CP32 (O G GP32) S40 G)
(CP4 (O G GP4) S40 G)
(X S41 G))
(S42 (C22 S44 D) (X S43 D))
(S43 (X (R I GV1 D) S41 G))
(S44 (X (R P2 GV1 D) S41 G)))
(FD332 (S1 (GN S2 G) (X S1 G))
(S2 ((GD11 GV23) (O G) S5 D) (C22 S3 D) (X S4 D))
(S3 (GN (R P2 GP3 D) (O G) S5 D))
(S4 (GN (R I GP3 D) (O G) S5 D))
(S5 (GP3 (F D GP3) (E FD334 S1 D) (RETURN) S1 D)
(GN (R I GP3 D) S5 D)
(X S5 D)))
(FD333 (S1 (GN (O G) S2 D) (X S1 G))
(S2 (GP31 (R I GN G)
(F D GP32)
(E FD313 S1 D)
(RETURN)
S1
D)
(X S2 D)))
(FD334 (S1 (C31 S1 D) (C1 S2 D) (X S7 G))
(S2 (GP2 S3 G) (GP31 S13 G) (GP32 S14 G) (X S7 G))

```



```

(S3 ((GP2 GP4) (O G) S9 D)
  ((GP3 GP0) S8 D)
  (GP12 (O G) S5 D)
  (GP11 S10 G)
  (X S3 G))
(S4 (GP (F D GP11CO) S8 D) (X S4 D))
(S5 (GP (F D GP12CO) S8 D) (X S5 D))
(S6 (GP (F D GP3CO) S8 D) (X S6 D))
(S7 (GP S8 D) (X S7 G))
(S8 (X (RETURN)))
(S9 (GP (F D GP2CO) S8 D) (X S9 D))
(S10 ((PHI CP11) S11 D) ((CP2 CP4) S12 D) (X S10 G))
(S11 (GP11 (O G) S4 D) (X S11 D))
(S12 (GP11 S8 D) (X S12 D))
(S13 (GP31 (O G) S6 D)
  (GP12 (O G) S5 D)
  (GP11 (O G) S4 D)
  (GP2 S15 G)
  (GP S8 D)
  (X S13 G))
(S14 ((GP0 GP32) (O G) S6 D) (GP S8 D) (X S14 G))
(S15 ((C31 PHI) S16 D) (GV1 S17 D) (X S15 G))
(S16 (GP2 (O G) S9 D) (X S16 D))
(S17 (GP2 S8 D) (X S17 D)))
(FD341 (S1 ((GV23 GP11 GD2 GD11 C21) S1 D)
  (GN (R P1 GV23 G) S2 D)
  (C22 S3 D)
  (C31 S5 G)
  (X S7 G))
  (S2 ((GV2 GP11 C21) S2 D)
    (GN (R P1 GV23 G) S2 D)
    (C22 S3 D)
    (X S7 G))
  (S3 (C2 S3 D)
    ((GP2 GN GV21 GV22) (R P2 GV23 G) S4 D)
    (X S7 G))
  (S4 (GP11 S4 D)
    (GN (R P1 GV23 G) S2 D)
    (C22 S3 D)
    (X S7 G))
  (S5 (GV23CO S6 D)
    (X (F D GV23) (E FD342 S1 D) (RETURN) S1 D))
  (S6 (C31 S1 D))
  (S7 (C S7 G)
    (X (F D GV23) (E FD342 S1 D) (RETURN) S1 D)))
(FD342 (S1 (C31 S1 D) (C1 S2 D) (X S3 G))
  (S2 (C21 S4 D) (C22 S5 D) (GV23 S6 G) (X S3 G))
  (S3 (GV23 S16 D) (X S3 G))
  (S4 (GV23 S8 G) (X S3 G))
  (S5 (GV23 S9 G) (X S3 G))
  (S6 (GV23 (O G) S7 D) (C S6 G))

```

```

(S7 (GV23 (F D GV23CO) S16 D) (X S7 D))
(S8 (GV23 S10 G) (C S8 G))
(S9 (GV23 S11 G) (C S9 G))
(S10 (C21 (O D) S12 D) (X S3 D))
(S11 (C22 (O D) S13 D) (X S3 D))
(S12 (C2 S14 D) (X S12 D))
(S13 (C2 S15 D) (X S13 D))
(S14 (GV23 (F D GV23CO) S16 D))
(S15 (GV23 (F D GV23CO) S16 D))
(S16 (X (RETURN))))
(FD40 (S1 (C32 (E FD41 S1 G) (RETURN))
(GP3 S2 G)
(GV1 (E FD331 S2 D) S1 G)
(X S1 D))
(S2 ((C22 C1 GD2 GD11 N212 C31) S2 G)
(C1 (O D) S8 D)
(N211 (R I GP3 D) S2 G)
(C32 (E FD41 S1 G) (RETURN))
((GV2 GN GP3 GP0 GP2) S3 G)
(X S7 D))
(S3 (C22 S5 D) (X S4 D))
(S4 (C32 (RETURN)) (X (R I GP3 D) S6 G))
(S5 (X (R P2 GP3 D) S6 G))
(S6 ((GV2 GN GP3 GP0 GP2 C1 GD2 GD11 C31 C2) S6 G)
(X S7 D))
(S7 (C1 (O D) S8 D) (GP3 (O G) S9 D) (X (O G) S8 D))
(S8 (GP3 S9 D) (X S8 D))
(S9 ((C21 GP2 GV2 GD2 GD11 GP1 C1 C31) S9 D)
(GP4 (F G GP3) S1 D)
(GP3 (R I GP3 G) (F D GP3) S14 D)
(GN (R P1 GP3 G) S11 D)
(C22 S10 D)
(X S13 G))
(S10 ((GN GV21 GV22 GP2) (R P2 GP3 G) S12 D)
(GP4 (F G GP3) S1 D)
(GP3 (F G GP3) S1 G)
(X S9 D))
(S11 (C32 S13 G)
(GP4 (F G GP3) S1 D)
(GP3 (F G GP3) S1 G)
(GD11 S11 D)
(C22 S10 D)
(X (F D GP3) S14 D))
(S12 (GN (R P1 GP3 G) S11 D)
(GP4 (F G GP3) S1 D)
(GP3 (F G GP3) S1 G)
((C21 GP2 GD2 GD11 GV2 C1 C31) S12 D)
(C22 S10 D)
(X S13 G))
(S13 ((C1 C31) S13 G) (X (F D GP3) S2 D))
(S14 (C32 (E FD41 S1 G) (RETURN)) (X S1 G)))

```

```

(FD41 (S1 (PHI S2 D) (X S1 G))
      (S2 (C32 (E FD42 S1 G) (RETURN)) (GP S3 D) (X S2 D))
      (S3 (C31 S3 D)
          (C1 S4 D)
          (C32 (E FD42 S1 G) (RETURN))
          (X S2 D))
      (S4 (GD11 S4 D)
          (C32 (E FD42 S1 G) (RETURN))
          (GP S5 G)
          (X S2 D))
      (S5 (GP (O G) S6 D) (X S5 G))
      (S6 (GP (F D GPC0) S2 D) (X S6 D)))
(FD42 (S1 (PHI (E FD5 S1 D) (RETURN))
      (GP2 S17 G)
      (GP4 S18 G)
      (GP3 S2 G)
      (X S1 G))
      (S2 (GP4 (R P2 GP3 D) (O G) S3 D)
          (PHI (E FD5 S1 D) (RETURN))
          (X S2 G))
      (S3 (GP3 (F D GP4) S4 D) (X S3 D))
      (S4 (X S5 G))
      (S5 (GP4 S6 G))
      (S6 ((C22 C31) S6 G)
          ((GP2 GP0) S12 G)
          (PHI (E FD5 S1 D) (RETURN))
          ((GP3 GP0) (O G) S7 D)
          (X S1 G))
      (S7 (GP4 (R I (GP3 GP0) G) (F D GP3) S8 D))
      (S8 (X S9 G))
      (S9 (GP3 S2 G))
      (S10 (GP3 (R I (GP3 GP0) G) (F D GP3) S11 D) (X S10 D))
      (S11 (X S9 G))
      (S12 (C22 S14 D) (X S13 D))
      (S13 ((GP2 GP0) (O G) S15 D))
      (S14 ((GP2 GP0) (R P2 GP4 D) (O G) S16 D))
      (S15 (GP4 (R I (GP2 GP0) G) (F D GP4) S6 G) (X S15 D))
      (S16 (GP4 (F D GP4) S6 G) (X S16 D))
      (S17 ((C2 C31 C1) S17 G)
          (PHI (E FD5 S1 D) (RETURN))
          (GP3 S2 G)
          ((GD GV23) (R D GP2 D) S1 G)
          (X S1 G))
      (S18 (GP3 (O G) S19 D)
          (C S18 G)
          (GP2 S17 G)
          (PHI (E FD5 S1 D) (RETURN))
          (X S1 G))
      (S19 (GP4 (R P2 GP3 G) (F D GP3) S2 G) (X S19 D)))
(FD5 (S1 (PHI S2 D) (X S1 G))
      (S2 (X S3 D))

```

```
(S3 (C32 (RETURN)) (X S4 G))  
(S4 (X (O C) S5 D))  
(S5 (C32 (F G GP) (RETURN)) (X S5 D)))
```

Même texte exemple sortie finale de la grammaire descriptive des structures de surface:

```
(GN NIL
  (GNX ((D 2 -)) (GN NIL (N1 NIL "discours" )))
  (C21 NIL "de" )
  (GN ((D -2 +))
    (GNX ((D 2 -))
      (GN NIL
        (D12 NIL (T11 NIL "la" ))
        (N1 NIL "méthode" )))
      (C22 NIL (T151 NIL "pour" ))
      (GV21CO
        ((D -2 +))
        (GV21 NIL
          (GV21X ((P1 1 -))
            (GV21 NIL
              (GD11 NIL
                (D11 NIL
                  (T1522 NIL
                    "bien"
                    )))
                (V21 NIL "conduire" )))
            (GN ((P1 -1 +))
              (D12 NIL "sa" )
              (N1 NIL "raison" )))
          (C1 NIL "et" )
          (GV21 NIL
            (GV21X ((P2 3 -) (P1 1 -))
              (GV21 NIL (V21 NIL "chercher" )))
            (GNX ((P1 -1 +))
              (GN NIL
                (D12 NIL (T11 NIL "la" ))
                (N1 NIL "vérité" )))
              (C22 NIL "dans" )
              (GN ((P2 -3 +))
                (D12 NIL (T11 NIL "les" ))
                (N1 NIL "sciences" ))))))))
  (C32 NIL "." )
  (GP3 NIL
    (C22 NIL (T142 NIL "Si" ))
    (GP2 ((P2 1 +))
      (CP2 NIL (GD2 NIL (D2 NIL NIL)))
      (GN ((I 1 +))
        (D12 NIL (T131 NIL "ce" ))
        (N1 NIL "discours" ))
      (GV1 ((I -1 -))
        (GV1 NIL (V1 NIL "semble" ))
        (CV23 NIL
          (GV23 ((P2 2 -))
```

```

(GD11 NIL (D11 NIL "trop" ))
(V23 NIL "long" ))
(C22 NIL (T151 NIL "pour" ))
(GV21 ((P2 -2 +))
(GV21X NIL
(GV21 NIL
(V21 NIL
"être"
))))))
(GV23 NIL
(GV23 ((P2 2 -))
(GD11 NIL
(D11 NIL
(T1533 NIL "tout" )))
(V23 NIL "lu" ))
(C22 NIL (T12 NIL "en" ))
(GN ((P2 -2 +))
(D12 NIL (T132 NIL "une" ))
(N1 NIL "fois" ))))
(GP32 ((P2 -1 -))
(CP32 NIL (C31 NIL "," ))
(N211 ((I 1 +)) "on" )
(GV1 ((I -1 -))
(N212 NIL (T11 NIL "le" ))
(GV1 NIL (V1 NIL "pourra" ))
(GV21 NIL
(GV21X ((P2 2 -))
(GV21 NIL
(V21 NIL
"distinguer"
)))
(C22 NIL (T12 NIL "en" ))
(GN ((P2 -2 +))
(GN NIL
(GV23 NIL (V23 NIL "six" ))
(N1 NIL "parties" )))))))
(C32 NIL "." )
(GP0 NIL
(C1 NIL "Et" )
(C31 NIL "," )
(C22 NIL (T12 NIL "en" ))
(GN ((P2 3 +))
(D12 NIL (T11 NIL "la" ))
(N1 NIL "première" ))
(C31 NIL "," )
(N211 ((I 1 +)) "on" )
(GV1 ((P2 -3 -) (I -1 -))
(GV1 ((P1 1 -)) (V1 NIL "trouvera" ))
(GN ((P1 -1 +))
(GN ((D 1 -))
(D12 NIL (T133 NIL "diverses" ))

```

```

        (N1 NIL "considérations" ))
      (GV22 ((D -1 +))
        (GV22X ((P1 1 -))
          (GV22 NIL
            (V22 NIL "touchant" )))
        (GN ((P1 -1 +))
          (D12 NIL (T11 NIL "les" ))
          (N1 NIL "sciences" ))))))
(C32 NIL "." )
(GP NIL
  (C22 NIL (T12 NIL "En" ))
  (GN NIL
    (D12 NIL (T11 NIL "1a" ))
    (N1 NIL "seconde" ))
  (C31 NIL "," )
  (GN NIL
    (GNX ((D 2 -))
      (GN NIL
        (D12 NIL (T11 NIL "les" ))
        (GN NIL
          (GV23 NIL (V23 NIL "principales" ))
          (N1 NIL "règles" )))))
    (C21 NIL "de" )
    (GN ((D -2 +))
      (GN ((D 1 -))
        (D12 NIL (T11 NIL "1a" ))
        (N1 NIL "méthode" ))
      (GP2 ((D -1 +))
        (CP2 NIL (GD2 NIL (D2 NIL "que" )))
        (GN ((I 1 +))
          (D12 NIL (T11 NIL "1" ))
          (N1 NIL "auteur" ))
        (GV1 ((I -1 -))
          (GV1 NIL (V1 NIL "a" ))
          (GV23 NIL
            (GV23 NIL
              (V23 NIL
                "cherchée"
                ))))))))
(C32 NIL "." )
(GP NIL
  (C22 NIL (T12 NIL "En" ))
  (GN NIL (D12 NIL (T11 NIL "1a" )) (N1 NIL "3" ))
  (C31 NIL "," )
  (GN NIL
    (GNX ((D 2 -)) (GN NIL (N22 NIL "quelques-unes" )))
    (C21 NIL "de" )
    (GN ((D -2 +))
      (GNX ((D 2 -)) (GN NIL (N22 NIL "celles" )))
      (C21 NIL "de" )
      (GN ((D -2 +))

```

```

(GN ((D 1 -))
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "morale" ))
(GP2 ((D -1 +))
  (CP2 NIL (GD2 NIL (D2 NIL "qu" )))
  (N211 ((I 1 +)) "11" )
  (GV1 ((I -1 -))
    (GV1 NIL (V1 NIL "a" ))
    (GV23 NIL
      (GV23 ((P1 2 -))
        (V23 NIL "tirée" ))
      (C21 NIL "de" )
      (GN ((P1 -2 +))
        (D12 NIL "cette" )
        (N1 NIL
          "méthode"
          ))))))))
(C32 NIL "." )
(GP NIL
  (C22 NIL (T12 NIL "En" ))
  (GN NIL (D12 NIL (T11 NIL "la" )) (N1 NIL "4" ))
  (C31 NIL ", " )
  (GN NIL
    (GN ((D 1 -))
      (D12 NIL (T11 NIL "les" ))
      (N1 NIL "raisons" ))
    (GP11 ((D -1 +))
      (CP11 NIL
        (C22 NIL "par" )
        (N23 NIL "lesquelles" ))
      (N211 ((I 1 +)) "11" )
      (GV1 ((I -1 -))
        (GV1 ((P1 3 -) (P1 1 -))
          (V1 NIL "prouve" ))
        (GNX ((P1 -1 +))
          (GN NIL
            (D12 NIL (T11 NIL "1" ))
            (N1 NIL "existence" )))
        (C21 NIL "de" )
        (GNCO ((P1 -3 +))
          (GN NIL (N1 NIL "dieu" ))
          (C1 NIL "et" )
          (C21 NIL "de" )
          (GN NIL
            (GN NIL
              (D12 NIL (T11 NIL "1" ))
              (N1 NIL "âme" ))
            (GV23X
              NIL
              (GV23 NIL
                (V23 NIL

```



```

"humaine"
))))
(GP12 NIL
  (CP12 NIL
    (C31 NIL "," )
    (N23 NIL "qui" ))
  (GV1 NIL
    (GV1 ((P1 3 -) (P1 1 -))
      (V1 NIL "sont" ))
    (GNX ((P1 -1 +))
      (GN NIL
        (D12 NIL
          (T11 NIL
            "les"
          ))
          (N1 NIL
            "fondements"
          ))
        ))
    (C21 NIL "de" )
    (GN ((P1 -3 +))
      (D12 NIL "sa" )
      (N1 NIL
        "métaphysique"
      ))
    ))))
(C32 NIL "." )
(GP NIL
  (C22 NIL (T12 NIL "En" ))
  (GN NIL (D12 NIL (T11 NIL "1a" )) (N1 NIL "5" ))
  (C31 NIL "," )
  (GN NIL
    (GNX ((D 2 -))
      (GN NIL
        (D12 NIL (T11 NIL "1" ))
        (N1 NIL "ordre" )))
    (C21 NIL "des" )
    (GN ((D -2 +))
      (GNX ((D 2 -)) (GN NIL (N1 NIL "questions" )))
      (C21 NIL "de" )
      (GN ((D -2 +))
        (GN ((D 1 -)) (N1 NIL "physique" ))
        (CP2 ((D -1 +))
          (CP2 NIL (GD2 NIL (D2 NIL "qu" )))
          (N211 ((I 1 +)) "11" )
          (GV1 ((I -1 -))
            (GV1 NIL (V1 NIL "a" ))
            (GV23 NIL
              (GV23 NIL
                (V23 NIL
                  "cherchées"
                ))
              ))
          ))
        ))
      ))
  (C31 NIL "," )

```

```

(C1 NIL "et" )
(GD11 NIL (D11 NIL "particulièrement" ))
(GN      NIL
  (GNX ((D 2 -))
    (GN NIL
      (D12 NIL (T11 NIL "1" ))
      (N1 NIL "explication" )))
  (C21 NIL "du" )
  (GNCO ((D -2 +))
    (GN NIL
      (GN ((D 2 -)) (N1 NIL "mouvement" ))
      (C21 NIL "du" )
      (GN ((D -2 +)) (N1 NIL "coeur" )))
    (C1 NIL "et" )
    (C21 NIL "de" )
    (GN NIL
      (GN ((D 1 -))
        (D12 NIL "quelques" )
        (GN NIL
          (GV23 NIL
            (V23 NIL
              (T133 NIL "autres" )))
            (N1 NIL "difficultés" )))
        (GP11 ((D -1 +))
          (CP11 NIL (N23 NIL "qui" ))
          (GV1 NIL
            (GV1 ((P2 2 -))
              (V1 NIL
                "appartiennent"
              ))
            (C22 NIL "à" )
            (GN ((P2 -2 +))
              (D12 NIL (T11 NIL "la" ))
              (N1 NIL
                "médecine"
              ))))))))
(C31 NIL ", " )
(GD11 NIL
  (D11 NIL (T102 NIL "puis" ))
  (D11 NIL "aussi" ))
(GN      NIL
  (GN ((D 1 -))
    (D12 NIL (T11 NIL "la" ))
    (N1 NIL "différence" ))
  (GP11 ((D -1 +))
    (CP11 NIL (N23 NIL "qui" ))
    (GV1 NIL
      (GV1 ((P1 4 -) (P2 2 -))
        (V1 NIL "est" ))
      (C22 NIL "entre" )
      (GNCO ((P2 -2 +))

```

```

(GN NIL
  (D12 NIL "notre" )
  (N1 NIL "âme" ))
(C1 NIL "et" )
(GNX NIL
  (GN NIL (N22 NIL "celle" ))))
(C21 NIL "des" )
(GN ((P1 -4 +)) (N1 NIL "bêtes" ))))
(C32 NIL "." )
(GP NIL
  (C1 NIL "Et" )
  (C22 NIL (T12 NIL "en" ))
  (GN NIL
    (D12 NIL (T11 NIL "la" ))
    (N1 NIL "dernière" ))
  (C31 NIL ", " )
  (GP2
    NIL
    (CP2 NIL (GD2 NIL (D2 NIL "quelles" )))
    (GN ((I 2 +)) (N1 NIL "choses" ))
    (N211 ((I 1 +)) "11" )
    (GV1
      ((I -2 -) (I -1 -))
      (GV1 NIL (V1 NIL "croit" ))
      (GV21 NIL (GV21X NIL (GV21 NIL (V21 NIL "être" ))))
      (GV23
        NIL
        (GV23 ((P2 2 -)) (V23 NIL "requises" ))
        (C22 NIL (T151 NIL "pour" ))
        (GV21
          ((P2 -2 +))
          (GV21X
            ((P1 4 -) (P2 2 -))
            (GV21
              NIL
              (V21 NIL "aller" )
              (GD11X
                NIL
                (GD11 NIL
                  (D11 NIL (T1521 NIL "plus" ))
                  (D11 NIL (T17 NIL "avant" ))))))
          (C22 NIL (T12 NIL "en" ))
          (GNX
            ((P2 -2 +))
            (GN NIL
              (D12 NIL (T11 NIL "la" ))
              (N1 NIL "recherche" )))
          (C21 NIL "de" )
          (GN ((P1 -4 +))
            (GN ((D 1 -))
              (D12 NIL (T11 NIL "la" ))

```

```

      (N1 NIL "nature" ))
(GP2CO
  ((D -1 +))
  (GP2
    NIL
    (CP2 NIL (GD2 NIL (D2 NIL "qu" )))
    (N211 ((I 1 +)) "11" )
    (GV1
      ((I -1 -))
      (GV1 NIL
        (GD11 NIL (D11 NIL "n" ))
        (V1 NIL "a" ))
        (GV23 NIL (GV23 NIL (V23 NIL "été" )))))
    (C31 NIL ", " )
    (C1 NIL "et" )
    (GP2
      NIL
      (CP2 NIL (GD2 NIL (D2 NIL "quelles" )))
      (GN ((I 1 +)) (N1 NIL "raisons" ))
      (GV1
        ((I -1 -))
        (N212 NIL (T11 NIL "1" ))
        (GV1 NIL (V1 NIL "ont" ))
        (GV23
          NIL
          (GV23 NIL (V23 NIL "fait" ))
          (GV21
            NIL
            (GV21X
              NIL
              (GV21 NIL
                (V21 NIL "écrire" ))))))))
    (C32 NIL "." )
    (GPO NIL
      (GN ((I 1 +))
        (D12 NIL (T11 NIL "Le" ))
        (GN NIL
          (GV23 NIL (V23 NIL "bon" ))
          (N1 NIL "sens" )))
      (GV1 ((I -1 -))
        (GV1 ((P1 3 -) (P1 1 -)) (V1 NIL "est" ))
        (GNX ((P1 -1 +))
          (GN NIL
            (D12 NIL (T11 NIL "la" ))
            (N1 NIL "chose" )))
        (C21 NIL "du" )
        (GN ((P1 -3 +)) (N1 NIL "monde" ))
        (GV23X NIL
          (GV23 NIL
            (D12 NIL (T11 NIL "la" ))
            (GV23 NIL

```

```

(GD11 NIL (D11 NIL "mieux" ))
(V23 NIL "partagée" ))))))
(C32 NIL ":" )
(GP3 NIL
  (GP2 ((I 1 +))
    (CP2 NIL (GD2 NIL (D2 NIL "car" )))
    (GN ((I 1 +)) (N22 NIL (T133 NIL "chacun" )))
    (GV1 ((I -1 -))
      (GV1 NIL (V1 NIL "pense" ))
      (GV21
        NIL
        (N212 NIL (T12 NIL "en" ))
        (GV21X NIL (GV21 NIL (V21 NIL "être" ))))
      (GV23
        NIL
        (GV23
          NIL
          (GD11 NIL
            (D11 NIL (T142 NIL "si" ))
            (D11 NIL (T1522 NIL "bien" )))
            (V23 NIL "pourvue" )))
          (C30 NIL (C31 NIL ", " ))
          (GD2 NIL (D2 NIL "que" ))))
      (GP32
        ((I -1 -))
        (GN ((I 1 -))
          (GN ((D 1 -))
            (GN NIL (N22 NIL "ceux" ))
            (GV23X
              NIL
              (GV23 NIL (V23 NIL (T1531 NIL "même" ))))
            (GP11
              ((D -1 +))
              (CP11 NIL (N23 NIL "qui" ))
              (GV1 NIL
                (GV1 NIL (V1 NIL "sont" ))
                (GV23
                  NIL
                  (GV23
                    ((P2 2 -))
                    (D12 NIL (T11 NIL "les" ))
                    (GV23
                      NIL
                      (GD11 NIL (D11 NIL (T1521 NIL "plus" )))
                      (V23 NIL "difficiles" )))
                    (C22 NIL "à" )
                    (GV21
                      ((P2 -2 +))
                      (GV21X ((P2 2 -))
                        (GV21 NIL (V21 NIL "contenter" )))
                      (C22 NIL (T12 NIL "en" ))

```

```

(GN ((P2 -2 +))
  (D12 NIL (T1532 NIL "toute" ))
  (GN NIL
    (D12 NIL (T133 NIL "autre" ))
    (N1 NIL "chose" )))))))
(GP31
  ((I -1 +))
  (CP31 NIL (C31 NIL "," ))
  (GV1 NIL
    (GV1 ((P1 1 -))
      (GD11 NIL (D11 NIL "n" ))
      (V1 NIL "ont" )
      (GD11X
        NIL
        (GD11 NIL
          (D11 NIL (T1523 NIL "point" )))))
      (GN ((P1 -1 +))
        (GN ((D 2 -)) (N1 NIL "coutume" ))
        (C21 NIL "d" )
        (GV21
          ((D -2 +))
          (N212 NIL (T12 NIL "en" ))
          (GV21X
            ((P1 1 -))
            (GV21
              NIL
              (V21 NIL "désirer" )
              (GD11X
                NIL
                (GD11 NIL
                  (D11 NIL (T1521 NIL "plus" )))))
            (GP2 ((P1 -1 +))
              (CP2 NIL (GD2 NIL (D2 NIL "qu" )))
              (N211 ((I 1 +)) "ils" )
              (GV1 ((I -1 -))
                (N212 NIL (T12 NIL "en" ))
                (GV1 NIL
                  (V1 NIL "ont" )))))))
          (C32 NIL "." )
          (GP NIL
            (GPO NIL
              (C22 NIL (T12 NIL "En" ))
              (GN ((P2 2 +)) (N22 NIL "quoi" ))
              (N211 ((I 1 +)) "il" )
              (GV1 ((P2 -2 -) (I -1 -))
                (GV1 NIL
                  (GD11 NIL (D11 NIL "n" ))
                  (V1 NIL "est" )
                  (GD11X NIL
                    (GD11 NIL
                      (D11 NIL

```

```

(T1523 NIL "pas" )))))
(GV23 NIL
  (GV23 NIL
    (V23 NIL "vraisemblable" )))))
(GP2 NIL
  (CP2 NIL (GD2 NIL (D2 NIL "que" )))
  (GN ((I 1 +)) (N22 NIL (T132 NIL "tous" )))
  (GV1 ((I -1 -))
    (N212 NIL "se" )
    (GV1 NIL (V1 NIL "trompent" )))))
(C32 NIL ";" )
(GP3
  NIL
  (GP0
    ((I 1 +))
    (GD11 NIL (D11 NIL "mais" ) (D11 NIL "plutôt" ))
    (GN ((I 1 +)) (N22 NIL "cela" ))
    (GV1
      ((I -1 -))
      (GV1 ((P1 2 -)) (V1 NIL "témoigne" ))
      (GD2 NIL (D2 NIL "que" ))
      (GN ((P1 -2 +))
        (GN ((D 2 -))
          (D12 NIL (T11 NIL "la" ))
          (N1 NIL "puissance" ))
          (C21 NIL "de" )
          (GV21CO
            ((D -2 +))
            (GV21
              NIL
              (GV21X
                NIL
                (GV21 NIL
                  (GD11 NIL (D11 NIL (T1522 NIL "bien" )))
                  (V21 NIL "juger" )))))
            (C31 NIL "," )
            (C1 NIL "et" )
            (GV21
              NIL
              (GV21X ((P2 4 -) (P1 1 -))
                (GV21 NIL (V21 NIL "distinguer" )))
              (GN ((P1 -1 +))
                (D12 NIL (T11 NIL "le" ))
                (N1 NIL "vrai" ))
                (C21 NIL "d" )
                (C22 NIL "avec" )
                (GN ((P2 -4 +))
                  (D12 NIL (T11 NIL "le" ))
                  (N1 NIL "faux" )))))
              (GP12
                NIL

```

```

(CP12 NIL (C31 NIL "," ) (N23 NIL "qui" ))
(GV1
NIL
(GV1 ((P1 1 -))
(V1 NIL "est" )
(GD11X NIL (GD11 NIL (D11 NIL "proprement" ))))
(GN ((P1 -1 +))
(GN ((D 1 -)) (N22 NIL (T131 NIL "ce" )))
(GP2
((D -1 +))
(CP2 NIL (GD2 NIL (D2 NIL "qu" )))
(N211 ((I 1 +)) "on" )
(GV1
((I -1 -))
(GV1 ((P1 1 -)) (V1 NIL "nomme" ))
(GNCO
((P1 -1 +))
(GN NIL
(D12 NIL (T11 NIL "le" ))
(GN NIL
(GV23 NIL (V23 NIL "bon" ))
(N1 NIL "sens" )))
(C1 NIL "ou" )
(GN NIL
(D12 NIL (T11 NIL "la" ))
(N1 NIL "raison" ))))))))
(CP31
((I -1 -))
(CP31 NIL (C31 NIL "," ))
(GV1
NIL
(GV1 NIL (V1 NIL "est" ))
(GV23
NIL
(GV23 ((P2 2 -))
(GD11 NIL (D11 NIL "naturellement" ))
(V23 NIL "égale" ))
(C22 NIL (T12 NIL "en" ))
(GN
((P2 -2 +))
(D12 NIL (T132 NIL "tous" ))
(GN NIL
(D12 NIL (T11 NIL "les" ))
(N1 NIL "hommes" ))))))))
(C32 NIL ";" )
(GP NIL
(C1 NIL "et" )
(GD11 ((D 1 +)) (D11 NIL "ainsi" ))
(GP2 ((D -1 -))
(CP2 NIL (GD2 NIL (D2 NIL "que" )))
(GN ((I 1 +))
(GNX ((D 2 -))

```



```

(GN NIL
  (D12 NIL (T11 NIL "la" ))
  (N1 NIL "diversité" )))
(C21 NIL "de" )
(GN ((D -2 +))
  (D12 NIL "nos" )
  (N1 NIL "opinions" )))
(GV1 ((I -1 -))
  (GV1 ((P1 2 -))
    (GD11 NIL (D11 NIL "ne" ))
    (V1 NIL "vient" )
    (GD11X
      NIL
      (GD11 NIL
        (D11 NIL (T1523 NIL "pas" ))))))
(C21 NIL "de" )
(GN ((P1 -2 +))
  (GN ((D 1 -)) (N22 NIL (T131 NIL "ce" )))
  (GP2 ((D -1 +))
    (CP2 NIL (GD2 NIL (D2 NIL "que" )))
    (GN ((I 1 +))
      (D12 NIL (T11 NIL "les" ))
      (N22 NIL "uns" ))
    (GV1 ((I -1 -))
      (GV1 ((P1 3 -))
        (V1 NIL "sont" ))
      (GV23
        NIL
        (GV23
          NIL
          (GD11
            NIL
            (D11 NIL
              (T1521 NIL "plus" )))
              (V23 NIL "raisonnables" )))
            (GD2 NIL (D2 NIL "que" ))
            (GN ((P1 -3 +))
              (D12 NIL (T11 NIL "les" ))
              (N22 NIL
                (T133 NIL
                  "autres"
                  ))))))
(C30 NIL (C31 NIL ", " ))
(GD11 NIL
  (D11 NIL "mais" )
  (D11 NIL "seulement" ))
(C21 NIL "de" )
(GN NIL
  (GN ((D 1 -)) (N22 NIL (T131 NIL "ce" )))
  (GP2CO
    ((D -1 +))

```



```

(GD11 NIL (D11 NIL "n" ))
(V1 NIL "est" )
(CD11X
  NIL
  (GD11 NIL
    (D11 NIL (T1523 NIL "pas" ))
    (D11 NIL "assez" ))))
(C21 NIL "d" )
(GV21 NIL
  (GV21X ((P1 1 -))
    (GV21 NIL (V21 NIL "avoir" )))
  (GN ((P1 -1 +))
    (D12 NIL (T11 NIL "1" ))
    (N1 NIL "esprit" ))
  (GV23X NIL
    (GV23 NIL (V23 NIL "bon" ))))))
(GP32 ((I -1 -))
  (CP32 NIL (C31 NIL "," ))
  (GD11 NIL (D11 NIL "mais" ))
  (GN ((I 1 +))
    (D12 NIL (T11 NIL "le" ))
    (N1 NIL "principal" ))
  (GV1 ((I -1 -))
    (GV1 NIL (V1 NIL "est" ))
    (C21 NIL "de" )
    (GV21 NIL
      (N212 NIL (T11 NIL "1" ))
      (GV21X
        NIL
        (GV21 NIL
          (V21 NIL "appliquer" )
          (CD11X
            NIL
            (GD11 NIL
              (D11 NIL
                (T1522 NIL
                  "bien"
                  ))))))))
  (C32 NIL "." )
  (GPO NIL
    (GN ((I 1 +))
      (D12 NIL (T11 NIL "Les" ))
      (GN NIL
        (GV23 NIL
          (CD11 NIL (D11 NIL (T1521 NIL "plus" )))
          (V23 NIL "grandes" ))
          (N1 NIL "âmes" )))
      (GV1 ((I -1 -))
        (GV1 NIL (V1 NIL "sont" ))
        (GV23 NIL
          (GV23 ((P1 2 -)) (V23 NIL "capables" ))

```

```

(C21 NIL "des" )
(GN ((P1 -2 +))
  (GN NIL
    (CV23 NIL
      (GD11 NIL
        (D11 NIL
          (T1521 NIL
            "plus"
          )))
        (V23 NIL "grands" ))
      (N1 NIL "vices" ))))
(C30 NIL (C31 NIL "," ))
(GD11 NIL
  (D11 NIL "aussi" )
  (D11 NIL (T1522 NIL "bien" )))
(GD2 NIL (D2 NIL "que" ))
(C21 NIL "des" )
(GN NIL
  (GN NIL
    (GV23 NIL
      (GD11 NIL
        (D11 NIL
          (T1521 NIL "plus" )))
        (V23 NIL "grandes" ))
      (N1 NIL "vertus" ))))
(C32 NIL ";" )
(GP NIL
  (C1 NIL "et" )
  (GP3
    NIL
    (GP32
      NIL
      (GN ((I 1 -))
        (GN ((D 1 -)) (N22 NIL "ceux" ))
        (GP11
          ((D -1 +))
          (CP11 NIL (N23 NIL "qui" ))
          (GV1
            NIL
            (GV1 NIL
              (GD11 NIL (D11 NIL "ne" ))
              (V1 NIL "marchent" ))
            (GD11X
              NIL
              (GD11
                NIL
                (D2 NIL "que" )
                (GD11 NIL
                  (D11 NIL "fort" )
                  (D11 NIL "lentement" ))))))))
  (GP31

```

```

((I -1 +))
(CP31 NIL (C31 NIL "," ))
(GV1
NIL
(GV1 NIL (V1 NIL "peuvent" ))
(GV21
NIL
(GV21X NIL (GV21 NIL (V21 NIL "avancer" )))
(GD11X
NIL
(GD11 NIL
(D11 NIL "beaucoup" )
(D11 NIL "davantage" )))
(C30 NIL (C31X NIL (C31 NIL "," )))
(C22 NIL (T141 NIL "s" ))
(GP2
NIL
(CP2 NIL (GD2 NIL (D2 NIL NIL)))
(N211 ((I 1 +)) "ils" )
(GV1
((I -1 -))
(GV1 ((P1 2 -)) (V1 NIL "suivent" ))
(GD11X NIL (GD11 NIL (D11 NIL "toujours" )))
(GN ((P1 -2 +))
(D12 NIL (T11 NIL "le" ))
(GN NIL
(GV23 NIL (V23 NIL "droit" ))
(N1 NIL "chemin" )))))
(C30 NIL (C31 NIL "," ))
(GP2
NIL
(CP2 NIL (GD2 NIL (D2 NIL "que" )))
(GV1
NIL
(GV1 ((P1 1 -))
(GD11 NIL (D11 NIL "ne" ))
(V1 NIL "font" ))
(GN ((P1 -1 +))
(GN ((D 1 -)) (N22 NIL "ceux" ))
(GP11
((D -1 +))
(CP11 NIL (N23 NIL "qui" ))
(GV1
NIL
(GV1 NIL (V1 NIL "courent" ))
(GP12
NIL
(CP12 NIL
(C31 NIL "," )
(C1 NIL "et" )
(N23 NIL "qui" ))

```

```

(GV1
  NIL
  (N212 NIL (T141 NIL "s" ))
  (N212 NIL (T12 NIL "en" ))
  (GV1 NIL
    (V1 NIL "éloignent" )))))))))))
(C32 NIL "." )
(GP NIL
  (GP0 NIL
    (C22 NIL (T151 NIL "Pour" ))
    (GN ((P2 3 +)) (N22 NIL "moi" ))
    (C31 NIL "," )
    (N211 ((I 1 +)) "je" )
    (GV1 ((P2 -3 -) (I -1 -))
      (GV1 NIL
        (GD11 NIL (D11 NIL "n" ))
        (V1 NIL "ai" )
        (GD11X NIL
          (GD11 NIL (D11 NIL "jamais" ))))
      (GV23 NIL
        (GV23 NIL (V23 NIL "présumé" )))))
    (GP2 NIL
      (CP2 NIL (GD2 NIL (D2 NIL "que" )))
      (GN ((I 1 +))
        (D12 NIL "mon" )
        (N1 NIL "esprit" ))
      (GV1 ((I -1 -))
        (GV1 ((P1 6 -) (P1 4 -) (P2 2 -))
          (V1 NIL "fût" ))
        (C22 NIL (T12 NIL "en" ))
        (GV23 ((P2 -2 +))
          (GV23 NIL
            (GD11 NIL
              (D11 NIL
                (T1521 NIL "rien" ))
              (D11 NIL
                (T1521 NIL "plus" ))
              (V23 NIL "parfait" )))
            (GD2 NIL (D2 NIL "que" ))
            (GNX ((P1 -4 +)) (GN NIL (N22 NIL "ceux" )))
            (C21 NIL "du" )
            (GN ((P1 -6 +)) (N1 NIL "commun" ))))
          (C32 NIL ";" )

```

## TABLE DES MATIERES

|                                          |    |
|------------------------------------------|----|
| Introduction                             | 2  |
| CHAPITRE 1                               |    |
| Description générale du logiciel         | 12 |
| 1.1 Les automates Déredec                | 12 |
| 1.2 Les fonctions Déredec                | 23 |
| 1.3 Cadre de validation                  | 24 |
| CHAPITRE 2                               |    |
| Description formelle du logiciel         | 28 |
| 2.1 Le méta-langage BNF                  | 28 |
| 2.2 Syntaxe des EXFAD                    | 30 |
| 2.3 Syntaxe des modèles d'exploration    | 32 |
| 2.4 Syntaxe des automates                | 35 |
| 2.5 Les fonctions Déredec                | 49 |
| 2.6 La fonction LEXIQUE                  | 49 |
| 2.7 La fonction TRAFIC                   | 50 |
| 2.8 La fonction DESCRIP                  | 51 |
| 2.9 La fonction IMPRI                    | 52 |
| 2.10 La fonction EXPAF                   | 53 |
| 2.11 La fonction EXPA                    | 53 |
| 2.12 La fonction TRADES                  | 53 |
| 2.13 La fonction SIMULECART              | 54 |
| 2.14 La fonction USAGER                  | 55 |
| 2.15 La fonction SOIT                    | 56 |
| 2.16 La fonction TRADUIRE                | 57 |
| 2.17 La fonction PREPARE                 | 57 |
| 2.18 La fonction PROJEC                  | 58 |
| 2.19 La fonction PROFIL                  | 58 |
| 2.20 La fonction LEXIFORME               | 59 |
| 2.21 La fonction LIRFIC                  | 60 |
| 2.22 La fonction COPIER                  | 60 |
| 2.23 La fonction EMBELLIR                | 60 |
| 2.24 Fonctions LISP couramment utilisées | 61 |
| 2.25 Les erreurs de programmation        | 66 |
| 2.26 Les messages d'erreurs du Déredec   | 67 |
| 2.27 Séance de programmation             | 68 |
| CHAPITRE 3                               |    |
| Une Grammaire de Texte                   | 77 |
| 3.1 Généralités                          | 77 |
| 3.2 L'indexation des catégories de base  | 80 |

|     |                                                  |     |
|-----|--------------------------------------------------|-----|
| 3.3 | Le dépistage des syntagmes et des RDC            | 83  |
| 3.4 | Le traitement des ambiguïtés                     | 86  |
| 3.5 | L'exploration du contenu des structures indexées | 90  |
| 3.6 | Le dépistage automatique des paradigmes          | 103 |

#### CHAPITRE 4

|     |                                             |     |
|-----|---------------------------------------------|-----|
|     | L'apprenti-lecteur (APLEC)                  | 111 |
| 4.1 | Systèmes "top-down" VS systèmes "bottom-up" | 111 |
| 4.2 | APLEC                                       | 114 |
| 4.3 | Premier exemple d'utilisation d'APLEC       | 118 |
| 4.4 | Second exemple d'utilisation d'APLEC        | 121 |

|               |     |
|---------------|-----|
| Bibliographie | 130 |
|---------------|-----|

|                      |     |
|----------------------|-----|
| Annexe au chapitre 3 | 155 |
|----------------------|-----|

|                    |     |
|--------------------|-----|
| Table des matières | 226 |
|--------------------|-----|

-----