



HAL
open science

Analyse du français comme langage de commande dans un système de construction graphique

Bruno de La Fayette

► **To cite this version:**

Bruno de La Fayette. Analyse du français comme langage de commande dans un système de construction graphique. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1976. Français. NNT: . tel-00287141

HAL Id: tel-00287141

<https://theses.hal.science/tel-00287141>

Submitted on 11 Jun 2008

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

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

THESE

présentée à

Université Scientifique et Médicale de Grenoble

Institut de Recherche en Mathématiques Avancées

pour obtenir le grade de
DOCTEUR DE TROISIEME CYCLE
EN INFORMATIQUE

ANALYSE DU FRANCAIS COMME LANGAGE DE COMMANDE
DANS UN SYSTEME DE CONSTRUCTION GRAPHIQUE



Bruno DE LA FAYOLLE



Thèse soutenue le 16 novembre 1976 devant la Commission d'Examen

Président : J. KUNTZMANN
A. COLMERAUER
Examineurs : M. LUCAS
B. VAUQUOIS

GRUPE D'ÉTUDES POUR LA TRADUCTION AUTOMATIQUE

Equipe Associée au C.N.R.S.

B.P. 53 - 38 041 GRENOBLE-CEDEX



UNIVERSITE SCIENTIFIQUE
ET MEDICALE DE GRENOBLE

Monsieur Gabriel CAU : Président
Monsieur Pierre JULLIEN : Vice-Président

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

PROFESSEURS TITULAIRES

MM. ARNAUD Paul	Chimie
AUBERT Guy	Physique
AYANT Yves	Physique approfondie
Mme BARBIER Marie-Jeanne	Electrochimie
MM. BARBIER Jean-Claude	Physique Expérimentale
BARBIER Reynold	Géologie appliquée
BARJON Robert	Physique nucléaire
BARNOUD Fernand	Biosynthèse de la cellulose
BARRA Jean-René	Statistiques
BARRIE Joseph	Clinique chirurgicale
BEAUDOING André	Clinique de Pédiatrie et Puériculture
BERNARD Alain	Mathématiques Pures
Mme BERTRANDIAS Françoise	Mathématiques Pures
MM. BERTRANDIAS Jean-Paul	Mathématiques Pures
BEZES Henri	Pathologie chirurgicale
BLAMBERT Maurice	Mathématiques Pures
BOLLIET Louis	Informatique (IUT B)
BONNET Georges	Electrotechnique
BONNET Jean-Louis	Clinique ophtalmologique
BONNET-EYMARD Joseph	Clinique gastro-entérologique
Mme BONNIER Marie-Jeanne	Chimie générale
MM. BOUCHERLE André	Chimie et toxicologie
BOUCHEZ Robert	Physique nucléaire
BOUSSARD Jean-Claude	Mathématiques Appliquées
BOUTET DE MONTVEL Louis	Mathématiques Pures
BRAVARD Yves	Géographie
CABANEL Guy	Clinique rhumatologique et hydrologique
CALAS François	Anatomie
CARLIER Georges	Biologie végétale
CARRAZ Gilbert	Biologie animale et pharmacodynamie
CAU Gabriel	Médecine légale et toxicologie
CAUQUIS Georges	Chimie organique
CHABAUTY Claude	Mathématiques Pures
CHARACHON Robert	Clinique Oto-rhino-laryngologique
CHATEAU Robert	Clinique de neurologie
CHIBON Pierre	Biologie animale
COEUR André	Pharmacie chimique et chimie analytique
CONTAMIN Robert	Clinique gynécologique
COUDERC Pierre	Anatomie pathologique
Mme DEBELMAS Anne-Marie	Matière médicale
MM. DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DELORMAS Pierre	Pneumophtisiologie

MM. DEPORTES Charles	Chimie minérale
DESRE Pierre	Métallurgie
DESSAUX Georges	Physiologie animale
DODU Jacques	Mécanique appliquée (IUT A)
DOLIQUE Jean-Michel	Physique des plasmas
DREYFUS Bernard	Thermodynamique
DUCROS Pierre	Cristallographie
DUGOIS Pierre	Clinique de dermatologie et syphiligraphie
GAGNAIRE Didier	Chimie physique
GALLISSOT François	Mathématiques Pures
GALVANI Octave	Mathématiques Pures
GASTINEL Noël	Analyse numérique
GAVEND Michel	Pharmacologie
GEINDRE Michel	Electroradiologie
GERBER Robert	Mathématiques Pures
GERMAIN Jean-Pierre	Mécanique
GIRAUD Pierre	Géologie
JANIN Bernard	Géographie
KAHANE André	Physique générale
KLEIN Joseph	Mathématiques Pures
KOSZUL Jean-Louis	Mathématiques Pures
KRAVTCHENKO Julien	Mécanique
KUNTZMANN Jean	Mathématiques Appliquées
LACAZE Albert	Thermodynamique
LACHARME Jean	Biologie végétale
Mme LAJZEROWICZ Janine	Physique
MM. LAJZEROWICZ Joseph	Physique
LATREILLE René	Chirurgie générale
LATURAZE Jean	Biochimie pharmaceutique
LAURENT Pierre-Jean	Mathématiques Appliquées
LEDRU Jean	Clinique médicale B
LLIBOUTRY Louis	Géophysique
LOISEAUX Pierre	Sciences nucléaires
LONGEQUEUE Jean-Pierre	Physique nucléaire
LOUP Jean	Géographie
Mlle LUTZ Elisabeth	Mathématiques Pures
MM. MALGRANGE Bernard	Mathématiques Pures
MALINAS Yves	Clinique obstétricale
MARTIN-NOËL Pierre	Clinique cardiologique
MAZARE Yves	Clinique médicale A
MICHEL Robert	Minéralogie et Pétrographie
MICOUD Max	Clinique maladies infectieuses
MOURIQUAND Claude	Histologie
MOUSSA André	Chimie nucléaire
MULLER Jean-Michel	Thérapeutique (Néphrologie)
NEEL Louis	Physique du Solide
OZENDA Paul	Botanique
PAYAN Jean-Jacques	Mathématiques Pures
PEBAY-PEYROULA Jean-Claude	Physique
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
REVOL Michel	Urologie
RINALDI Renaud	Physique
DE ROUGEMONT Jacques	Neuro-chirurgie
SEIGNEURIN Raymond	Microbiologie et Hygiène
SENGEL Philippe	Zoologie

MM. SIBILLE Robert	Construction mécanique (IUT A)
SOUTIF Michel	Physique générale
TANCHE Maurice	Physiologie
TRAYNARD Philippe	Chimie générale
VAILLANT François	Zoologie
VALENTIN Jacques	Physique nucléaire
VAUQUOIS Bernard	Calcul électronique
Mme VERAIN Alice	Pharmacie galénique
MM. VERAIN André	Physique
VEYRET Paul	Géographie
VIGNAIS Pierre	Biochimie médicale
YOCCOZ Jean	Physique nucléaire théorique

PROFESSEURS ASSOCIES

MM. CLARK Gilbert	Spectrométrie physique
CRABBE Pierre	CERMO
ENGLMAN Robert	Spectrométrie physique
HOLTZBERG Frédéric	Basses températures
DEMBICKI Eugéniuz	Mécanique
MATSUSHIMA Yozo	Mathématiques Pures

PROFESSEURS SANS CHAIRE

Mlle AGNIUS-DELORE Claudine	Physique pharmaceutique
ALARY Josette	Chimie analytique
MM. AMBROISE-THOMAS Pierre	Parasitologie
BELORIZKY Elie	Physique
BENZAKEN Claude	Mathématiques Appliquées
BIAREZ Jean-Pierre	Mécanique
BILLET Jean	Géographie
BOUCHET Yves	Anatomie
BRUGEL Lucien	Energétique (IUT A)
BUISSON René	Physique (IUT A)
BUTEL Jean	Orthopédie
COHEN ADDAD Pierre	Spectrométrie physique
COLOMB Maurice	Biochimie
CONTE René	Physique (IUT A)
DEPASSEL Roger	Mécanique des fluides
FONTAINE Jean-Marc	Mathématiques Pures
GAUTHIER Yves	Sciences Biologiques
GAUTRON René	Chimie
GIDON Paul	Géologie et Minéralogie
GLENAT René	Chimie organique
GROULADE Joseph	Biochimie médicale
HACQUES Gérard	Calcul numérique
HOLLARD Daniel	Hématologie
HUGONOT Robert	Hygiène et Médecine préventive
IDELMAN Simon	Physiologie animale
JOLY Jean-René	Mathématiques Pures
JULLIEN Pierre	Mathématiques Appliquées
Mme KAHANE Josette	Physique
MM. KRAKOWIAK Sacha	Mathématiques Appliquées
KUHN Gérard	Physique (IUT A)
LE ROY Philippe	Mécanique (IUT A)
LUU DUC Cuong	Chimie organique

MM. MAYNARD Roger	Physique du solide
Mme MINIER Colette	Physique (IUT A)
MM. PELMONT Jean	Biochimie
PERRIAUX Jean-Jacques	Géologie et Minéralogie
PFISTER Jean-Claude	Physique du solide
Mlle PIERY Yvette	Physiologie animale
MM. RAYNAUD Hervé	M.I.A.G.
REBECQ Jacques	Biologie (CUS)
REYMOND Jean-Charles	Chirurgie générale
RICHARD Lucien	Biologie végétale
Mme RINAUDO Marguerite	Chimie macromoléculaire
MM. ROBERT André	Chimie papetière
SARRAZIN Roger	Anatomie et chirurgie
SARROT-REYNAULD Jean	Géologie
SIROT Louis	Chirurgie générale
Mme SOUTIF Jeanne	Physique générale
MM. STREGLITZ Paul	Anesthésiologie
VIALON Pierre	Géologie
VAN CUTSEM Bernard	Mathématiques Appliquées

MATTRES DE CONFERENCES ET MATTRES DE CONFERENCES AGREGES

MM. AMBLARD Pierre	Dermatologie
ARMAND Gilbert	Géographie
ARMAND Yves	Chimie (IUT A)
BACHELOT Yvan	Endocrinologie
BARGE Michel	Neuro-chirurgie
BARJOLLE Michel	M.I.A.G.
BEGUIN Claude	Chimie organique
Mme BERIEL Hélène	Pharmacodynamie
MM. BOST Michel	Pédiatrie
BOUCHARLAT Jacques	Psychiatrie adultes
Mme BOUCHE Liane	Mathématiques (CUS)
MM. BRODEAU François	Mathématiques (IUT B)
CHAMBAZ Edmond	Biochimie médicale
CHAMPETIER Jean	Anatomie et organogénèse
CHARDON Michel	Géographie
CHERADAME Hervé	Chimie papetière
CHIAVERINA Jean	Biologie appliquée (EFP)
CONTAMIN Charles	Chirurgie thoracique et cardio-vasculaire
CORDONNIER Daniel	Néphrologie
COULOMB Max	Radiologie
CROUZET Guy	Radiologie
CYROT Michel	Physique du solide
DELOBEL Claude	M.I.A.G.
DENIS Bernard	Cardiologie
DOUCE Roland	Physiologie végétale
DUSSAUD René	Mathématiques (CUS)
Mme ETERRADOSSI Jacqueline	Physiologie
MM. FAURE Jacques	Médecine légale
FAURE Gilbert	Urologie
GAUTIER Robert	Chirurgie générale
GENSAC Pierre	Botanique
GIDON Maurice	Géologie
GROS Yves	Physique (IUT A)

MM. GUITTON Jacques	Chimie
HICTER Pierre	Chimie
IVANES Marcel	Electricité
JALBERT Pierre	Histologie
JUNIEN-LAVILLAVROY Claude	O.R.L.
KOLODIE Lucien	Hématologie
LE NOC Pierre	Bactériologie-virologie
LEROY Philippe	IUT A
MACHE Régis	Physiologie végétale
MAGNIN Robert	Hygiène et médecine préventive
MALLION Jean-Michel	Médecine du travail
MARECHAL Jean	Mécanique (IUT A)
MARTIN-BOUYER Michel	Chimie (CUS)
MICHOULIER Jean	Physique (IUT A)
NEGRE Robert	Mécanique (IUT A)
NEMOZ Alain	Thermodynamique
NOUGARET Marcel	Automatique (IUT A)
PARAMELLE Bernard	Pneumologie
PECCOUD François	Analyse (IUT B)
PEFFEN René	Métallurgie (IUT A)
PERRET Jean	Neurologie
PERRIER Guy	Géophysique - Glaciologie
PHELIP Xavier	Rhumatologie
RACHAIL Michel	Médecine interne
RACINET Claude	Gynécologie et obstétrique
RAMBAUD André	Hygiène et hydrologie
RAMBAUD Pierre	Pédiatrie
Mme RENAUDET Jacqueline	Bactériologie
MM. ROBERT Jean-Bernard	Chimie Physique
ROMIER Guy	Mathématiques (IUT B)
SHOM Jean-Claude	Chimie générale
STOEBNER Pierre	Anatomie pathologique
VROUSOS Constantin	Radiologie

MAITRE DE CONFERENCES ASSOCIES

M. COLE Antony

Sciences nucléaires

Fait à SAINT MARTIN D'HERES, AVRIL 1976.

Je remercie Monsieur le Professeur KUNTZMANN qui m'a fait l'honneur de présider le jury ;
Monsieur le Professeur VAUQUOIS qui a dirigé l'ensemble de ce travail ; Messieurs COLMERAUER et LUCAS qui ont eu
à s'occuper des autres aspects du système dans lequel ce travail s'insère.

Je remercie aussi les personnes avec qui j'ai eu l'occasion de travailler, en particulier
Messieurs GUILLAUME et QUEZEL-AMBRUNAZ qui m'ont aidé à utiliser les systèmes ATEF et CETA et Monsieur LALOUM
qui a écrit le système déductif.

Je remercie enfin Mademoiselle CURINIER et Madame BLANCHARD qui eurent le mérite de dactylographier
mon manuscrit, sans oublier le service de reprographie.

TABLE DES MATIERES

CHAPITRE I : INTRODUCTION

1. Langues naturelles, traduction automatique et intelligence artificielle	p. 2
2. Etudes et réalisation dans le domaine de la communication HM	p. 2
3. But du projet	p. 4
4. Organisation du traitement	p. 5

PREMIERE PARTIE

CHAPITRE II : LANGAGE EXTERNE

1. Le monde étudié	p. 8
1.1. Les objets	p. 8
1.2. Les actions	p. 9
1.3. Les spécifications	p. 9
1.4. L'axiomatisation du monde	p. 11
2. Syntaxe du français utilisé	p. 11
2.1. Grammaire	p. 11
2.2. Remarques	p. 13

CHAPITRE III : STRUCTURE PROFONDE

1. Représentation générale	p. 14
2. Le système déductif	p. 15
3. La représentation du sens	p. 16
3.1. Grammaire du langage de sortie	p. 16
3.2. Représentation des objets	p. 17
3.2.1. Objets créés	p. 18
3.2.2. Objets référencés	p. 18
3.3. Représentation des propriétés	p. 18
3.3.1. Propriétés absolues	p. 19
3.3.2. Relations	p. 20
3.3.3. Dimensions et emplacements	p. 21
3.4. Représentation des actions	p. 24
3.4.1. Création d'objets	p. 24
3.4.2. Destruction d'objets	p. 24
3.4.3. Modifications d'objets	p. 24

CHAPITRE IV : SEMANTIQUE DU LANGAGE EXTERNE

1. Les verbes	p. 27
1.1. Verbes exprimant une modification de la scène	p. 27
1.1.1. Verbes de création	p. 27
1.1.2. Verbes de destruction	p. 27
1.1.3. Verbes de transformation	p. 27
1.1.4. Autres verbes d'actions	p. 29
1.2. Verbes introduisant ou exprimant des propriétés	p. 29
1.2.1. Verbes auxiliaires	p. 29
1.2.2. Verbes d'appartenance	p. 30
1.2.3. Autres verbes	p. 30

2. Les noms communs	p. 30
2.1. Noms d'objets	p. 30
2.2. Noms de dimensions	p. 30
2.3. Autres noms communs	p. 31
3. Les adjectifs	p. 31
3.1. Adjectifs temporels	p. 31
3.2. Adjectifs spaciaux	p. 32
3.3. Adjectifs de comparaison	p. 33
3.4. Adjectifs de forme	p. 33
4. Les prépositions	p. 34
4.1. Prépositions d'emplacement	p. 34
4.2. Prépositions de position	p. 35
4.3. Autres prépositions	p. 35
5. Les détermineurs	p. 36
6. Les adverbes	p. 37
7. Les pronoms	p. 38
7.1. Pronoms personnels	p. 38
7.2. Pronoms relatifs	p. 39
8. Les apositions	p. 40
9. Les connecteurs logiques	p. 40
10. Les nombres	p. 41
11. Les signes de ponctuation	p. 41

DEUXIEME PARTIE

p. 43

CHAPITRE V : ANALYSE DU TEXTE

p. 44

1. Morphologie, syntaxe et sémantique	p. 45
2. Outils informatiques utilisés	p. 47
2.1. Mode de représentation du modèle d'une langue	p. 47
2.2. Systèmes du GETA	p. 47
2.3. Le système ATEF	p. 48
2.4. Le système CETA	p. 49

CHAPITRE VI : ANALYSE MORPHOLOGIQUE

p. 52

1. Les dictionnaires	p. 53
1.1. Dictionnaire des désinences et affixes	p. 53
1.2. Dictionnaire des bases	p. 53
1.3. Dictionnaire des tournures	p. 53
2. Les variables	p. 54
2.1. Variables exclusives	p. 54
2.2. Variables non exclusives	p. 55
2.3. Unité lexicale	p. 56
2.4. Remarques sur les variables	p. 56
3. La grammaire	p. 56

CHAPITRE VII : ANALYSE DE LA PHRASE	p. 60
1. La grammaire générale	p. 61
1.1. Forme-cible et stratégie d'analyse	p. 61
1.2. Structure générale de la grammaire	p. 61
1.3. Variables	p. 63
2. Les sous-grammaires	p. 64
2.1. Phase de prétraitement	p. 64
2.1.1. Forme-cible	p. 64
2.1.2. Grammaire de prétraitement	p. 65
2.1.3. Grammaire de présyntaxe	p. 66
2.2. Phase d'analyse	p. 66
2.2.1. Forme-cible	p. 66
2.2.2. Grammaire GNE	p. 69
2.2.3. Grammaire MS	p. 70
2.2.4. Grammaire GV	p. 73
2.3. Phase de post-traitement	p. 74
2.3.1. Structure cible	p. 74
2.3.2. Grammaires G41 et G42	p. 74
3. Exemples de traitements	p. 75
4. Sur les ambiguïtés	p. 82
 CHAPITRE VIII : CONCLUSION	 p. 84
1. Elargissement du domaine d'étude	p. 85
2. Système question-réponse	p. 86
2.1. Utilité	p. 86
2.2. Représentation des questions	p. 86
2.3. Génération des réponses	p. 87
3. Ajout d'informations	p. 87
3.1 Utilité	p. 87
3.2. Représentation de l'information	p. 89
 BIBLIOGRAPHIE	 p. 90
 ANNEXES	 p. 93

CHAPITRE I

INTRODUCTION

1 - LANGUES NATURELLES, TRADUCTION AUTOMATIQUE ET INTELLIGENCE ARTIFICIELLE

Dans le domaine général du traitement des langues naturelles, c'est la traduction automatique qui fut l'application pionnière, puisque dès le début des années 50, des études sur ce sujet étaient effectuées. c'est au début des années 60 que les premiers résultats ont pu être obtenus (Georgetown University [8] pour russe → anglais, Kulagina et Mel'chuk [10] pour français → russe), les études se poursuivant depuis.

Cependant, les résultats obtenus n'ont pas été aussi satisfaisants qu'on l'espérait au début. La raison profonde vient de la difficulté énorme qu'il y a pour saisir toute la complexité du phénomène linguistique dans le domaine très général qui est celui de la TA.

Parallèlement à ce domaine, l'intelligence artificielle se créait et se développait avec pour objectif de simuler sur ordinateur une fonction intellectuelle humaine, généralement très précise. On peut citer les problèmes de jeux, en particulier le jeu d'échecs ou le domaine récent de la robotique (voir par exemple Warplan [16]).

Enfin, depuis une dizaine d'années, les études portant sur le traitement automatique des langues naturelles ont souvent pour but la communication homme-machine : on veut pouvoir communiquer un message en langue naturelle à l'ordinateur et recevoir de lui une réaction cohérente par rapport à ce message. La forme de cette communication homme-machine peut être un système de question-réponse, les données sur lesquelles porteront les questions ayant, ou non, été auparavant fournies à l'ordinateur également en langue naturelle. La communication peut aussi prendre la forme d'ordres donnés à la machine, exprimant des modifications à faire subir au monde qu'elle contrôle. Dans toutes les réalisations, on peut noter plusieurs points communs, en opposition avec les méthodes utilisées en TA :

- utilisation d'un monde restreint à l'intérieur duquel le dialogue devra se cantonner. Les concepts étant bien délimités, il devient possible de structurer et de formaliser le monde accessible. On peut alors développer certains axes de recherche comme l'inférence à partir d'une base de données ou l'analyse sémantique.

- définition d'une structure profonde représentant le sens du discours. On transformera la phrase du locuteur humain en une structure de ce type (ou le processus inverse pour "produire" une réponse en langue naturelle de l'ordinateur). C'est à partir de cette structure qu'on calculera la réponse à la question.

On peut remarquer que les systèmes visant à la communication homme-machine se servent à la fois des méthodes développées en TA, pour l'analyse de la structure des mots ou des phrases et des outils utilisés en IA, en particulier l'usage de méthodes déductives et l'exploitation d'une base de données reflétant l'état du monde à chaque instant.

2 - ETUDES ET REALISATIONS DANS LE DOMAINE DE LA COMMUNICATION HOMME-MACHINE

Nous allons examiner quelques-uns des travaux les plus intéressants dans ce domaine, à la fois du point de vue des objectifs poursuivis et des moyens employés pour la réalisation.

- LE SYSTEME QUESTION-REPOSE DE WOODS

Woods [18] a écrit un système QR dont le domaine d'application est celui des lignes aériennes. Les données sont les heures de départ et d'arrivée des avions, les villes-étapes, les numéros des vols, ... Pour obtenir une réponse, la question est transformée par une analyse syntaxique, puis sémantique en une structure profonde formée de fonctions et de prédicats, primitives comprises par un déducteur de réponses (par exemple : connect (x_1, x_2, x_3) : le vol x_1 relie la ville x_2 à la ville x_3). Dans ce système, la base de données est définie au départ et reste statique.

- LES RESEAUX DE TRANSITION AUGMENTES

Woods [19] a par la suite défini un outil destiné à l'analyse des langues naturelles, les réseaux de transition augmentés : c'est un prolongement des méthodes d'analyse des langages décrits par une grammaire hors-contexte, permettant des transformations de chaîne à arbre sans limitation de puissance. Ce formalisme permet de construire une structure profonde en utilisant des règles de construction ou de transformation d'arborescence à mesure que sont analysés les différents composants syntaxiques. L'analyse se fait en testant des variables affectées aux mots ou aux composants et pouvant avoir une valeur "sémantique" (traits comme "animé", "humain",...).

Ce formalisme, simple et puissant, a été adopté comme outil pour la phase d'analyse du texte dans plusieurs systèmes de traitement de langue naturelle.

- LES RESEAUX SEMANTIQUES DE SIMMONS

Simmons a écrit un système QR qui utilise les réseaux de Woods d'une part pour l'analyse du texte [13], d'autre part pour la génération de langue naturelle à partir d'une structure profonde [14]. Cette structure profonde est constituée par les "réseaux sémantiques" (qui sont pratiquement des arborescences). Un tel réseau représente l'ensemble des concepts contenus dans la phrase, joints par un nombre limité de relations. Le processus de communication se compose de deux parties.

D'abord on peut donner de l'information en LN à l'ordinateur, celui-ci la transformant en réseau sémantique. Ensuite, on peut poser des questions, toujours en LN, sur les informations que connaît alors la machine. La question sera aussi transformée en réseau et la correspondance entre la question et les réponses-candidates sera assurée par des règles d'inférences servant à traiter les paraphrases. Par exemple, si on a donné l'information "Wellington battit Napoléon à la bataille de Waterloo", la réponse à la question "qui perdit une bataille ?" sera fournie grâce à la règle "x bat y à z \leftrightarrow y perd z face à x".

Ce système traite des données très générales, mais pour pouvoir répondre aux questions, il faudra avoir auparavant écrit sous forme de règles d'inférence toute la sémantique contenue dans le sujet traité. Ce système ne pourra donc être utilisé que pour des applications portant sur des sujets précis.

On voit que ce système est plus puissant que le système QR de Woods, puisque non seulement il peut répondre à des questions en LN, mais aussi accepter de l'information nouvelle sur laquelle les questions pourront porter.

- LE SYSTEME DE COMMUNICATION HM DE MARSEILLE

Ce système [7] est un système QR portant sur un texte en LN préalablement communiqué à l'ordinateur. Le système est tout entier écrit dans un même langage : PROLOG [11], spécialement adapté à la démonstration automatique.

La phase d'analyse du texte, question ou information, consistera donc à transformer celui-ci en un énoncé en logique du premier ordre. Quand il s'agira d'une question, c'est la démonstration de ce théorème qui fournira la réponse.

Le domaine d'études n'est pas limité : on peut en particulier employer n'importe quel verbe ou substantif. Il n'y a en effet pas de dictionnaire et seuls les mots auxiliaires (prépositions, articles,...) ont un traitement sémantique spécial.

En contrepartie, pour que le système puisse faire des déductions liant le sens de certains mots (comme "battre" et "perdre" chez Simmons), il faudra que l'utilisateur indique lui-même les liens existant entre ces mots. Pour une application donnée, le locuteur devra donc préalablement à l'énoncé de faits et de questions portant sur ces faits et exprimées par des mots différents, définir par des énoncés en LN les liens entre ces mots. Par exemple, pour pouvoir répondre à la question "Qui n'aime par Jean?" à partir de la donnée "Pierre hait Jean", il faudra avoir indiqué que "Chaque personne qui hait une personne ne l'aime pas". On voit que la puissance de ce système est équivalente à celle du système de Simmons.

- LE ROBOT DE WINOGRAD

Enfin, la tentative la plus intéressante du point de vue des objectifs poursuivis est, en ce qui nous concerne celle de Winograd [17]. Son système permet de donner des ordres à un robot, de lui poser des questions, soit sur l'état du monde, soit même sur les raisons ayant motivé ses actions, et de lui donner certaines définitions dont il devra par la suite tenir compte. Le monde auquel le robot aura accès est un monde de jouets géométriques : cubes et pyramides.

Dans ce système, comme dans le précédent, le traitement d'une phrase se fera en deux étapes : analyse du texte qui sera transformé en un énoncé logique représentant un théorème à démontrer, et exécution de l'action qui correspondra à la démonstration de ce théorème, cette démonstration se traduisant par une réponse s'il s'agit d'une question, ou l'exécution d'une série d'ordres, sur l'écran représentant la scène, s'il s'agit d'une commande.

Le domaine d'études se caractérise par sa réduction. Comme le nombre de types d'objets et d'actions à effectuer (des translations) est très limité, la formalisation du monde peut être très poussée et la sémantique être incorporée au modèle.

On voit que c'est ce système qui est le plus complet puisqu'il permet de donner des informations, de poser des questions et de donner des ordres exécutables.

3 - BUT DU PROJET

Le cadre général dans lequel s'inscrit ce projet est celui de la communication HM en LN, pour nous le français. L'étude porte sur un domaine très précis. Il s'agira de décrire une scène ou de la modifier de la manière que l'on voudra en utilisant un certain nombre d'objets élémentaires. Le monde que l'on pourra ainsi créer et gérer de la manière la plus générale possible sera représenté sur un terminal graphique (console de visualisation) et c'est sur celui-ci que l'on verra le résultat des demandes formulées.

Pour les pouvoirs de ce système, on se cantonnera à un seul type de requêtes : on donnera en français des ordres à la machine et celle-ci les exécutera, s'ils ont une interprétation acceptable par rapport à la scène existant au moment de l'expression de ces ordres. Dans le système actuel, on ne peut ni poser de questions portant sur l'état du monde accessible, ni donner de nouvelles informations qui auraient alors le sens de théorèmes. Cependant, l'organisation du traitement et la nature des outils utilisés permettraient d'insérer ces deux types de textes. On discutera dans le dernier chapitre l'intérêt que présenteraient ces nouveaux pouvoirs et les moyens de les réaliser. Néanmoins, ces objectifs sont secondaires par rapport au but recherché qui est d'avoir un moyen d'utiliser le français, ou du moins un sous-ensemble assez large de celui-ci, pour pouvoir décrire des scènes plus ou moins complexes.

Si on compare ce modèle à ceux décrits auparavant, il est clair que c'est de celui de Winograd qu'il se rapproche le plus.

4 - ORGANISATION DU TRAITEMENT

On distingue trois phases qui s'enchaînent dans le traitement :

- analyse du texte. Cette analyse devra fournir une représentation du sens du texte en LN.
- déduction des valeurs. En utilisant la base de données représentant l'état de la scène, il faudra déduire les valeurs des actions et des objets. Par exemple, si on a parlé d'"un gros cube", il faudra déduire de la représentation de cet objet de quel élément existant il s'agit.
- réalisation des ordres graphiques. Un système adapté au terminal graphique se chargera de la réalisation effective des ordres sur l'écran.

Le traitement étant divisé en 3 étapes enchaînées, il faut définir les interfaces entre ces différentes parties dont les objectifs et les outils de réalisation sont très différents.

Pour l'interface entre le système d'analyse et le système déductif, on partira du fait qu'il s'agit d'un système logique (il s'agit de PROLOG). On aura donc une structure profonde représentant le sens du texte sous forme arborescente d'un énoncé logique formé de fonctions et de prédicats. Le sens d'un ordre sera donc considéré comme un théorème à démontrer.

Pour l'interface entre le système déductif et le système graphique, le principe est de donner à ce dernier une description minimale des objets et des actions pour permettre la réalisation. Par exemple, pour la création d'un parallélépipède rectangle, on pourra donner l'ordre C(PAR,20,(100,200,300),100,200,50) où on donne l'ordre (création), le type d'objet, son nom interne, les coordonnées de son centre et ses trois dimensions.

A propos de cette représentation, on peut faire une remarque : il existe des langages "graphiques" qui permettent en s'intégrant à la syntaxe d'un langage de haut-niveau de manipuler un terminal graphique. C'est en particulier le cas d'EUCLID [2] dont la syntaxe générale est celle de FORTRAN et qui permet de créer ou de modifier des objets géométriques. Les ordres de manipulation d'objets ont, dans ce langage une forme analogue à la représentation servant d'interface entre les systèmes déductif et graphique. On voit qu'on peut alors parler de "programmation en langue naturelle" pour le système décrit ici.

Dans l'étude qui suit, on ne s'intéressera qu'au premier point (analyse du texte et représentation en une structure profonde). On ne parlera des autres aspects que quand ils permettront de comprendre les raisons des choix adoptés. Le système déductif est développé dans une autre thèse à paraître, dont on pourra voir les principes essentiels dans [20].

L'étude qu'on fera est divisée en-deux parties :

Dans la première partie, on définira de façon précise l'entrée et la sortie du système d'analyse : dans le chapitre II on indiquera quels concepts on manipulera et quel sous-ensemble du français permettra de les exprimer ; dans le chapitre III on définira la structure profonde ; dans le chapitre IV on indiquera le lexique choisi et la représentation qu'auront les mots ou les composants du langage dans la structure profonde, donc la sémantique du français utilisé comme langage de commande.

Dans la seconde partie, on s'intéressera à la réalisation effective permettant de calculer le sens d'une phrase, c'est-à-dire au système d'analyse : dans le chapitre V on parlera des principes généraux qui président au système d'analyse et dans les chapitres VI et VII on définira les analyseurs morphologique et syntaxo-sémantique utilisés.

Enfin, dans le chapitre conclusif, on verra quels élargissements peuvent s'offrir à ce système et quels problèmes restent à résoudre.

PREMIERE PARTIE

On décrira dans cette partie le sous-ensemble du français qu'on a choisi pour exprimer les ordres et la représentation profonde correspondante.

CHAPITRE II

LANGAGE EXTERNE

Par "langage externe" nous entendons la fraction de l'ensemble du français que nous utiliserons comme entrée. La réduction du français général au sous-ensemble que nous prendrons provient de la réduction du monde général que permet de manipuler le français à un monde bien précis pour lequel une partie du français s'avère suffisante. Examinons de façon plus précise ce que l'on se propose de réaliser et quels concepts on devra considérer.

On dispose d'un terminal graphique où est représentée une scène. Contrairement au système de Winograd dans lequel la scène existe dès le départ et reste figée, à des translations d'objets près, on se place dans le cas le plus général. Au départ, la scène sera vide et c'est l'interlocuteur qui créera la scène qu'il désire. Il pourra aussi modifier celle-ci à sa guise en déplaçant des objets, en en détruisant, etc... On peut ainsi construire, à partir d'objets et d'actions élémentaire, des scènes complexes ou les faire évoluer de diverses manières. En fait, les objets manipulables et les actions qu'on peut effectuer sont réduits aux cas les plus simples.

1 - LE MONDE ETUDIE

1.1. - LES OBJETS

Les objets élémentaires qu'on prendra seront les objets géométriques qui ont une définition mathématique précise et qu'on sait donc exactement comment représenter. Les seuls "objets" proprement dits seront les corps géométriques à 3 dimensions. On pourra aussi utiliser les éléments à 0, 1 ou 2 dimensions (points, lignes, surface), mais ils ne seront pas manipulés de la même façon que les objets, la sémantique du français se rapportant au monde réel à 3 dimensions, et ils serviront surtout d'auxiliaires pour la description des "vrais" objets : par exemple, créer un point qui servira ensuite de centre ou un segment qui servira d'axe de rotation.

D'une façon générale, on pourrait utiliser tous les corps géométriques ayant un nom en français : cube, sphère, cône, dodécaèdre, etc... On n'utilisera néanmoins que des polyèdres, objets dont la surface est formée de polygones, car ce sont les seuls objets facilement représentables sur l'écran : une sphère, par exemple est représentée par un ensemble de polygones approchant le plus possible sa surface. Cependant, comme cette limitation n'est due qu'aux caractéristiques du matériel graphique utilisé, l'étude faite ne dépendra pas des objets effectivement utilisés.

Dans la réalisation effectuée on a pris des parallélépipèdes rectangles (appelés "bloc" en français pour raccourcir l'expression), dont le cube est un cas particulier, des pyramides à base rectangulaire et des prismes à face triangulaire isocèle.

Il est évident qu'on pourrait facilement ajouter d'autres formes si on le voulait. Par exemple avec les objets indiqués, on pourra représenter la figure suivante :

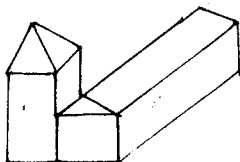


fig. 1

1.2. - LES ACTIONS

On peut distinguer, dans le langage externe, 3 formes d'ordres qui correspondent sur l'écran à 3 sortes de modifications de la scène :

- création d'objet (apparition d'un nouvel objet sur l'écran) ;
- destruction d'objet (disparition de l'objet mentionné) ;
- modifications d'objet. Ces modifications peuvent être de différentes natures :

Toutes les transformations géométriques simples peuvent être envisagées : translation, rotation, homothétie... On aura même des transformations non géométriques, en particulier le fait de changer la forme même d'un objet ("transformer le cube en pyramide"). On s'intéressera donc, de façon générale, à toutes les modifications exprimables de façon simple en français. C'est ainsi que, par exemple, nous n'avons pas considéré le cas des rotations d'objet, d'une part parce que l'expression en français d'une telle transformation n'est pas évidente, si on veut l'indiquer de façon précise (il faudrait en particulier définir l'axe de rotation), d'autre part et surtout parce que la représentation d'objets dotés d'inclinaisons quelconques par rapport aux axes de référence et l'exploitation de ces angles posent au système déductif de très sérieux problèmes. Pour les translations, déplacements qui respectent les angles, et les homothéties, que des verbes comme "avancer" ou "raccourcir" permettent d'exprimer, l'un et l'autre de ces problèmes ne se posent pas. C'est pourquoi on s'intéressera surtout à ces 2 types de transformations.

1.3. - LES SPECIFICATIONS

Par spécification, on entend tous les éléments du langage qui précisent les valeurs des objets ou des actions. Pour cela, nous utiliserons les adjectifs qui permettent de donner des ordres de grandeurs pour les dimensions. On pourra donner des noms aux objets, ce qui permettra ensuite de les référencer de façon précise. On pourra aussi préciser les objets par leur place sur la scène ou par rapport à d'autres objets (le cube qui est à droite, entre A et B). D'autre part, on utilisera les nombres pour désigner tout un groupe d'objets (créez 10 cubes).

On pourra aussi préciser la valeur des actions : par exemple, pour les translations, des prépositions permettront de dire le lieu de translation (mettez A sur B).

Enfin, on se donne la possibilité d'utiliser des valeurs réelles, c'est-à-dire adaptées au terminal graphique, ce qui permettra de donner des valeurs très précises puisque numériques. On pourra dire par exemple : "Créez un cube centré en (100, 100, 100)" ou : "Augmenter la longueur de A de 10 unités". On ajoutera ainsi au français habituel les possibilités d'un langage de programmation. Il est bien évident que ce n'est pas sur cet aspect que l'étude du langage sera axée.

A titre d'exemple, indiquons quelle suite d'ordres on pourrait donner pour faire représenter la figure 1 indiquée précédemment, qui pourrait figurer une église avec son clocher :

- Créez un bloc B plus large que long (fig. 2).
- Créez un prisme aussi long et large que B, sur le bloc (fig. 3).
- Créez un cube C et un cube P aussi haut que C (fig. 4).
- Augmentez la hauteur de C et transformez P en pyramide (fig. 5).
- Mettez C à gauche de B, contre lui, les faces avant de C et de B étant alignées (fig. 6).
- Posez la pyramide sur le bloc le plus haut (fig. 7).

Il est évident que pour représenter cette scène, on aurait pu choisir une autre suite de phrases, peut être plus simple, en créant par exemple directement la pyramide, mais cet exemple montre quel genre de phrase on prendra.

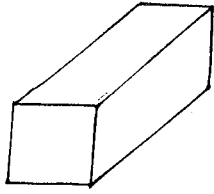


fig. 2

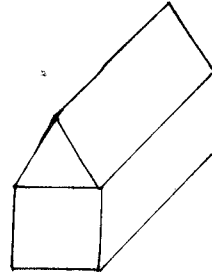


fig. 3

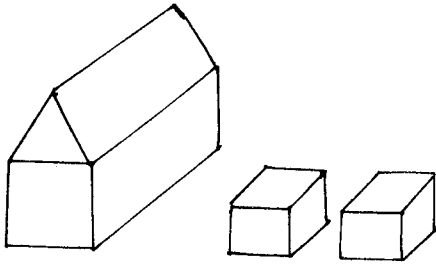


fig. 4

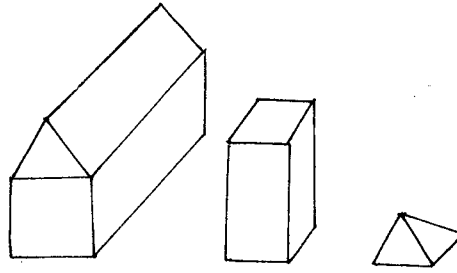


fig. 5

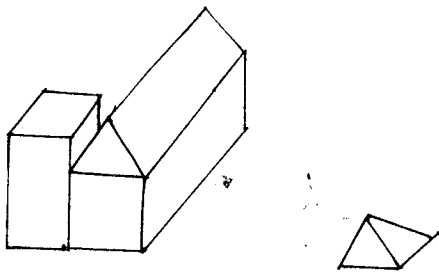


fig. 6

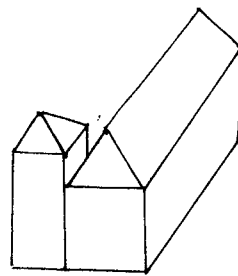
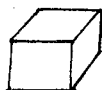


fig. 7

1.4. - L'AXIOMATISATION DU MONDE

Le monde que nous voulons décrire et voir représenté sur l'écran n'est pas un monde réel, mais un monde géométrique. Ce n'est donc pas le monde réel qui sera axiomatisé et par exemple des objets pourront avoir des parties communes.

Si on dit : "Créez 2 cubes", il est possible que ces 2 cubes soient construits au même endroit et avec les mêmes dimensions. On ne verra alors qu'un seul cube et les 2 objets ne pourront être distingués visuellement que si l'un d'eux est déplacé.



"Créez 2 cubes"



"Déplacez un cube vers la droite"

Cependant, le langage que nous utilisons servant à décrire le monde réel, l'axiomatisation réalisée devra rendre compte du monde réel. Par exemple, quand on dit de poser un objet x sur un objet y, il faudra que les emplacements de x et y soient tels que x soit au-dessus de y, contre lui, sans qu'il y ait d'intersection entre eux. En revanche si on dit de placer x au-dessus de y, la représentation résultante ne tiendra aucun compte des lois physiques existantes (pesanteur, intersection vide pour des objets pleins,...).

2 - SYNTAXE DU FRANCAIS UTILISE

Nous allons définir le sous-ensemble du français qui nous permettra de manipuler le monde ainsi défini. Ce français est réduit dans la mesure où on se limite aux phrases simples suffisant à décrire les commandes permettant de manipuler notre monde. Il doit être cependant assez étendu pour que les ordres puissent être exprimés d'une part avec une précision suffisante, d'autre part de la façon la plus naturelle possible.

De toute façon, cette grammaire n'a rien de figé : elle peut être modifiée si on veut ajouter certaines possibilités et rendre compte de phénomènes non encore pris en compte. La grammaire qui suit ne servira donc qu'à indiquer de façon générale quelle est la partie du français qu'on peut appréhender.

2.1. - GRAMMAIRE

Par convention, nous indiquerons entre parenthèses les éléments, terminaux ou non, dont la présence est facultative, ceci pour réduire le nombre de règles.

Le signe # représente le blanc séparateur de mots. Enfin, les symboles non terminaux sont écrits entre crochets.

`<texte> ::= <phrase> . | <phrase> . # <texte> | <phrase> # <conj|> # <texte>`

`<conj|> ::= et | ou | mais |,`

`<phrase> ::= <gn> # <gv> | <gv>`

`<gn> ::= <gn|> | <gn|> # <conj|> # <gn>`

`<gn|> ::= (<apos>,) # (<det>) # (<nb>) # (<nbordi>) # (<gal>) # <nmc> # ((, # <apos>) # (<mod>) |`

`<apos> | <prp> | <vi>`

`<apos> ::= "<slc>"`

`<slc> ::= <lc> | <lc> <slc>`

`<lc> ::= 0 | ... | 9 | a | ... | z`

<det> ::= un | le | toutes # les | ...
 <nb> ::= <n> | <n> <nb>
 <n> ::= 0 | ... | 9
 <nbordi> ::= <nb>ième | premier | dernier
 <gal> ::= <adj> | <signe> # <adj>
 <signe> ::= plus | moins
 <adj> ::= grand | petit | long | ...
 <nmc> ::= cube | centre | hauteur | ...
 <mod> ::= pas # <mod> | <mod> # <conj1> # <mod> | <conj2> # <mod> # <conj3> # <mod> | <mod1>
 <conj2> ::= ni | à # la # fois | soit
 <conj3> ::= ni | et | soit
 <mod1> ::= <ga2> | <gp> | <gvp> | <rel>
 <ga2> ::= <adjpos> | <det1> # <signe> # <adjpos> | (<nb> # fois) # <signel> # <adjpos> # que # <gnadj> |
 <adj> # <gp1>
 <adjpos> ::= <adj> | <pos>
 <pos> ::= en # avant | à # droite | ...
 <det1> ::= le | la | les
 <signel> ::= <signe> | aussi
 <gnadj> ::= <gn> | <adjpos>
 <gp1> ::= de # <nb> # (unité)
 <gp> ::= <prep> # <gn>
 <prep> ::= sur | sous | entre | en # avant # de | à | de | en | ...
 <gvp> ::= <vbp> # <GN> | <vbp> # <mod>
 <vbp> ::= passant | étant | centré | ... (verbe au participe passé ou présent)
 <rel> ::= <prel> # <phrase> | qui # <gv>
 <prel> ::= que | dont | où | ...
 <gv> ::= <vb> # <gn> | <vb> # <mod> | <vb> # <gn> # <gp>
 <vb> ::= <vbl> | ne # <vbl> # pas
 <vbl> ::= créer | détruire | être | ...
 <prp> ::= le | elle | en | ...
 <vi> ::= (<nb>, <nb>, <nb>) (ici les parenthèses sont des symboles terminaux).

2.2. - REMARQUES

1. Le langage accepté n'est qu'un sous-ensemble du langage qu'engendre cette grammaire (celle-ci engendre des phrases n'ayant aucun sens, comme : "créez les 5 largeurs plus hautes que cube centré en pyramide" !). Si on voulait écrire la grammaire qui n'engendre que le langage qui nous intéresse, il faudrait multiplier le nombre de règles en divisant les catégories syntaxiques (verbe, nom, préposition en particulier) suivant des critères sémantiques, indiquant que par exemple un GV dont le verbe est "créer" ne peut être suivi que d'un GN dont le substantif désigne un objet.

2. En considérant le langage engendré par cette grammaire on peut constater quelle fraction du français elle décrit :

- les restrictions sont importantes dans le domaine de la phrase complète : outre le fait qu'on ne tient pas compte des phrases interrogatives ou déclaratives, on voit qu'une phrase est formée d'une proposition simple, avec éventuellement des relatives ou des participiales à l'intérieur du GN, mais que les subordonnées ne sont pas acceptées.

- en revanche la structure du GN choisie est assez complète. On peut entre autres utiliser le GA pour désigner les dimensions, absolues ou relatives, le GP pour indiquer par exemple les emplacements, l'aposition pour donner un nom aux objets, etc...

La raison de ces choix est dictée :

- d'abord par la nécessité de définir un modèle expérimental assez simple pour qu'il puisse être réalisé pratiquement.

- ensuite par la nature du monde que ce langage sert à décrire. Les ordres qu'on utilise sont assez simples pour que les phrases les exprimant aient elles-mêmes une structure propositionnelle simple. Par contre, comme on veut pouvoir désigner les objets de façon précise et diversifiée, il est nécessaire que la structure du GN soit la plus large possible.

3. A titre d'exemple, pour montrer les différentes structures manipulées, on pourra avoir le texte suivant : "Créez 2 gros cubes, posez le prisme plus long que le cube le plus large sur le bloc centré en (100, 500, 200), de nom B et qui est à la fois plus haut que large et de longueur 100 et augmentez la hauteur du cube qui est à droite".

CHAPITRE III

STRUCTURE PROFONDE

1 - REPRESENTATION GENERALE

La structure choisie sert à représenter la phrase à la fin des transformations constituant la phase d'analyse. Elle sert d'entrée au système déductif et doit donc avoir une forme adaptée à celui-ci. C'est une structure profonde représentant le sens de la phrase au moyen d'un nombre limité de primitives, chacune exprimant un élément de base participant à la description du sens de l'énoncé en français. Il est clair que deux phrases sémantiquement identiques sont décrites par la même structure profonde. Par exemple, les phrases :

"Créez un cube A et un prisme moins haut que A" et :

"Représentez un cube nommé A et un prisme plus bas que A"

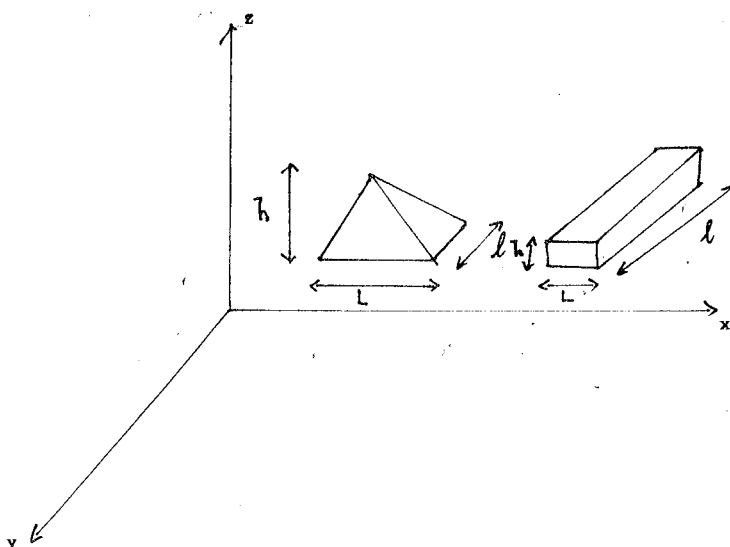
sont représentées de la même façon.

La notation générale employée pour cette structure est fonctionnelle et particulièrement prédicative. Chaque phrase est représentée comme une proposition logique écrite sous forme d'un prédicat avec des variables formelles pour distinguer les différents éléments intervenant dans le prédicat, par exemple : sur (x,y) pour indiquer que l'objet représenté par x est sur l'objet représenté par y. Les arguments des prédicats sont parfois des fonctions numériques quand il s'agit de prédicats de comparaison de nombres. La raison du choix de la structure réside dans la nature du système déductif.

2 - LE SYSTEME DEDUCTIF

Pour justifier le choix adopté, il est nécessaire d'indiquer brièvement le fonctionnement du système déductif. Ce dernier, écrit en PROLOG est un démonstrateur de théorèmes. L'entrée qui lui est fournie est considérée comme un théorème écrit en logique du premier ordre à démontrer. Le système déductif est formé de théorèmes correspondant aux primitives formant la structure d'entrée, qui décrivent la logique du monde traité, celui des objets géométriques. Les différents objets existant à tout moment sont représentés sous forme de clauses PROLOG. La base de données est constituée par un ensemble de telles clauses dont chacune représente toutes les caractéristiques permettant de définir un objet de façon complète. Cette représentation, qu'on pourra étendre pour traiter des cas plus complexes, comprend les éléments suivants :

- le numéro d'ordre, différent pour chaque objet, qui sert de nom interne,
- le nom externe, si on en a donné un,
- la forme (bloc, pyramide, prisme actuellement),
- les coordonnées de son point caractéristique, en l'occurrence le centre, qui permet de localiser l'objet,
- les longueurs des projections sur les trois axes, qui correspondent à la longueur, la largeur et la hauteur des objets, ceux-ci gardant une inclinaison constante par rapport aux axes.



L = longueur
l = largeur
h = hauteur

Pour voir le fonctionnement du système déductif, prenons la phrase "créez un gros bloc". Elle est représentée par une proposition disant qu'il faut créer un objet et que cet objet est caractérisé par deux propriétés : celle d'être gros et celle d'être un bloc. On pourrait donc donner la représentation : créer (x, et (gros(x), bloc (x))).

Pour "prouver" ce théorème le système utilisera les théorèmes créer (x,p), et (p,q), gros (x) et cube (x), où x désigne une variable formelle et p et q sont des prédicats évaluables.

Les théorèmes pourront s'écrire :

+ créer (x,p) - numéro (x) - vrai (p) - taille (x,u) - position (x,v) - nom (x,z) - forme (x,w) - ajout (x)
 + vrai (et(p,q)) - vrai (p) - vrai (q)
 + vrai (bloc(x)) - forme (x, bloc)
 + vrai (gros (x)) - taille (x,u) - inf (300,u)

Le fonctionnement sera le suivant pour l'exemple donné : voulant créer un objet x avec la propriété p, on lui donne d'abord un numéro, par exemple 3 si c'est le 3ième objet créé, puis on cherche à vérifier p qui vaut et (gros (x), cube (x)) ; Pour cela, il faut vérifier successivement les deux prédicats. Pour vérifier gros (x), on prend une taille supérieure à 300, nombre fixé arbitrairement. Pour vérifier bloc (x), on donne à l'objet la forme "bloc" (on ne détaille pas comment s'écrivent numéro (x), taille (x,u), etc). Revenant à "créer", on donne une position à l'objet, le nom restant non déterminé, et on ajoute cette description de l'objet à la base de données.

En réalité la représentation d'une telle commande et le traitement PROLOG correspondant ne sont pas exactement ceux que nous avons indiqués et on va maintenant décrire de façon précise la structure profonde utilisée.

3 - LA REPRESENTATION DU SENS

Avant d'indiquer le rôle des différentes primitives, donnons la grammaire de la structure profonde.

3.1. - GRAMMAIRE DU LANGAGE DE SORTIE

Le langage de sortie représentant le sens des phrases et servant d'entrée au système déductif est engendré par la grammaire suivante :

```

<S>      ::= <CONJ> (<S>, <S>) | créer (<VF>, <N1>, <P>) | od (<VF>, <N>, <P>, destr (<VF>))
<CONJ>   ::= et | ou
<VF>     ::= +1 | ... | + 20
<N1>     ::= des | <NB>
<N>      ::= des | tous | <NB>
<NB>     ::= <CHIF> | <CHIF> <NB>
<CHIF>   ::= 0 | ... | 9
<P>      ::= ref (<VF>, <REF>, <P1>) | <P1> | nil
<P1>     ::= <CONJ> (<P1>, <P1>) | non <P1> | <P2>
  
```

<P2> ::= forme (<VF>, <FORME> |
 nom (<VF>, <NOM>) |
 genre (<VF>, <GENRE>) |
 centre (<VF>, <COORD>) |
 <COMP> (<VAL NUM>, <VAL NUM>) |
 od (<VF>, <N>, <P>, <Q>)

<FORME> ::= cube | bloc | pyramide | prisme

<NOM> ::= "<SLC>"

<SLC> ::= <LC> | <LC> <SLC>

<LC> ::= A | ... | Z | <CHIF>

<REF> ::= dern | avdern | <NB>

<GENRE> ::= mas | fem

<COORD> ::= (<NB>, <NB>, <NB>)

<COMP> ::= sup | égal

<VAL NUM> ::= <OP> (<VAL NUM>, <VAL NUM>) | <NB> | <VAL STAND> | <ADJ POS> (<VF>)

<OP> ::= add | soustr | prod | quot

<VAL STAND> ::= grande | petite | moyenne

<ADJ POS> ::= longueur | largeur | hauteur | grosseur | abscisse | ordonnée | cote | place

<Q> ::= <CONJ> (<Q>, <Q>) | non (<Q>) | <Q1>

<Q1> ::= <COMP> (<VAL NUM>, <VAL NUM>) |
 sur (<VF>, <VF>) |
 à (<VF>, <VF>) |
 entre (<VF>, <VF>, <VF> |
 contre (<VF>, <VF> |
 menobjet (<VF>, <VF>) |
 destr (<VF>) |
 od (<VF>, <N>, <P>, <Q>)

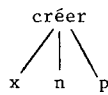
3.2. - REPRESENTATION DES OBJETS

Sur le plan conceptuel, il y a une différence fondamentale entre deux sortes d'objets : ceux qu'on demande de créer, qui n'existent pas au moment de l'ordre, et ceux auxquels on fait référence, qui existent déjà et sont représentés par une clause PROLOG. Le traitement logique est très différent dans les deux cas : quand on crée un objet, on doit créer une clause en utilisant les propriétés qu'il doit vérifier et ajouter cette clause aux autres, alors que quand on fait référence à un objet, il faut trouver dans la base de données la clause le décrivant, grâce encore aux propriétés qu'il doit vérifier.

D'une façon générale, pour représenter les objets, on utilise une variable formelle (VF), un nombre indiquant le cardinal de l'ensemble désigné globalement et un prédicat, plus ou moins complexe représentant les caractéristiques précisant le ou les objets.

3.2.1. Objets créés

La représentation générale est :



x = VF

n = cardinal de l'ensemble des objets créés

p = prédicat que doit vérifier x

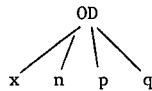
La signification est : "créer n objets x satisfaisant le prédicat p".

Le nombre n appartient à **NU** {des} : c'est un entier quand on indique un nombre déterminé d'objets ; il vaut "des" si on indique un certain nombre non précisé d'objets.

Ex : créer quelques cubes créer (x, des, p1)
 créer 2 gros cubes créer (x, 2, p2)

3.2.2. Objets référencés

La représentation générale est :



x = VF

n = cardinal

p, q = prédicats

La signification est : "trouver n objets satisfaisant le prédicat P et qui satisfassent aussi le prédicat q"

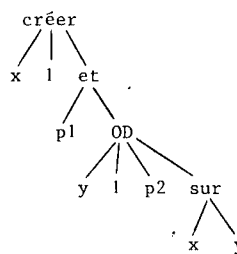
- n appartient à **NU** {des, tous} où n = tous signifie qu'on parle de tous les objets satisfaisant p.

- p est le prédicat permettant de trouver les objets dans la base de données.

- q est le prédicat représentant la propriété que doivent vérifier les objets référencés une fois obtenus :

q sert en particulier à représenter les relations entre objets.

ex : créer l'objet x sur l'objet y :



où p1 et p2 sont les prédicats définissant x et y

3.3. REPRESENTATION DES PROPRIETES

Les propriétés caractérisant les objets sont représentées par un prédicat. D'une façon générale, ce prédicat est une conjonction ou une disjonction de prédicats plus élémentaires. Par exemple, pour "un prisme long ou large", le prédicat est la conjonction de deux prédicats : l'un définissant la forme, l'autre étant la disjonction de deux prédicats définissant les dimensions : et (prisme, ou (long, large)). Ce simple exemple permet d'ailleurs de constater que si la disjonction apparaissant dans la structure profonde est toujours exprimée dans le langage externe, par exemple par la conjonction ou, il peut n'en pas aller de même pour la conjonction : c'est la simple concaténation des éléments spécificateurs des objets qui, dans ce cas, correspond à la conjonction et. Par exemple, du point de vue de l'expression des propriétés, le GN "le gros cube A" équivaut à "l'objet qui est un cube et qui est gros et qui s'appelle A".

On peut diviser l'ensemble des prédicats de propriétés en trois groupes : ceux qui définissent les propriétés absolues qui donnent une indication sur un objet pris en tant que tel et qui n'ont qu'un seul argument formel ; ceux qui définissent les relations permettant de lier les caractéristiques de plusieurs éléments et qui admettent plusieurs arguments formels ; enfin les prédicats définissant les tailles ou les emplacements dont le formalisme est commun, qu'il s'agisse de relations ou de propriété absolue. Enfin, l'absence même de propriété à vérifier sera traduite par le symbole nil et la négation par non (p).

3.3.1. Les propriétés absolues

3.3.1.1. La forme (ou nature)

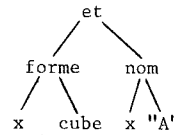
C'est la propriété la plus importante. Le prédicat correspondant est : forme (x, f) où f appartient à $F = \{\text{prisme, cube, bloc, pyramide}\}$. Il indique de quel genre d'objet on parle. Il n'y a que trois cas où ce prédicat n'apparaît pas dans le prédicat général : référence d'objet par pronominalisation (ex : détruisez-le) ; par nominalisation (ex : détruisez A) ou usage d'un mot désignant une forme quelconque.

ex : créer un cube	créer (x, 1, forme (x, cube))
créer un objet	créer (x, 1, nil)

3.3.1.2. Le nom

Le nom est donné aux objets par l'utilisateur pour les référencer de façon simple et précise. Il est représenté par le prédicat nom (x, n) où n est une suite de caractères alphanumériques entre guillemets.

Ex : le cube "A" : le prédicat général est



3.3.1.3. Le genre

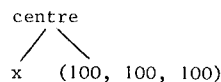
Dans les expressions habituelles (le cube, "A", etc...) le genre n'est pas considéré comme information signifiante et n'est donc pas indiqué dans la structure profonde. On en tient compte dans le seul cas où on réfère un objet par un pronom personnel ayant un genre unique (le, la, elle, ...). Si on dit "détruisez-le" on sait que l'objet indiqué fait partie de l'ensemble (cube, prisme, bloc) : le genre est donc dans ce cas un élément significatif.

La représentation du prédicat correspondant est : genre (x, g) ou $g \in \{\text{mas, fem}\}$

3.3.1.4. Le centre

Cette propriété sert à indiquer la valeur réelle qu'a le centre de l'objet. La forme du prédicat est centre (x, coord) ou coord est la liste des coordonnées du centre.

Ex : un cube centré en (100, 100, 100) :

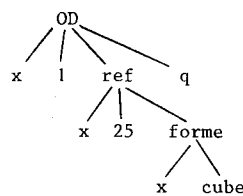


On verra que cette propriété pourrait être représentée dans le formalisme adopté pour les tailles et les emplacements, mais cela est rendu impossible par le fait que l'argument numérique est ici une liste de nombres au lieu d'un nombre simple pour le cas général que nous verrons.

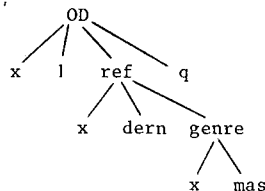
3.3.1.5. La référence

Elle est représentée d'une façon différente des autres propriétés. La représentation est $\text{ref}(x, n, p)$ où n est un nombre et p un prédicat. Elle signifie : x est le n ème des objets créés qui vérifient le prédicat p . Ce prédicat représente le prédicat général définissant les autres caractéristiques de l'objet. Les valeurs possibles du nombre sont les entiers et les deux valeurs spéciales: dernier et avant-dernier dont les sens sont : premier et second objet à partir du dernier objet créé". Cette référence proviendra d'abord des nombres ordinaux (25ième prisme, dernier bloc), mais aussi de références implicites : cas en particulier des pronoms personnels.

Ex : le 25ième cube



détruisez-le



3.3.2. Les relations

On ne parlera pas dans ce paragraphe des comparaisons de dimensions ou d'emplacement (qu'expriment "plus long que", "à droite de", etc...). Les relations considérées ici sont celles d'appartenance et de position.

3.3.2.1. Relation d'appartenance

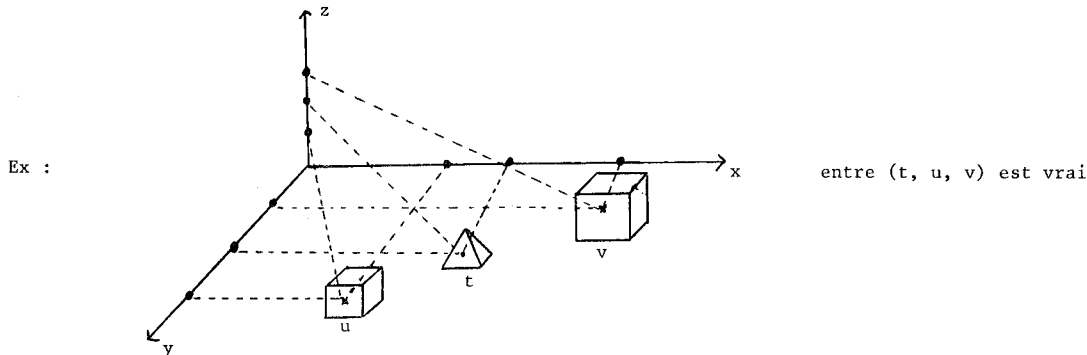
Elle exprime une relation entre deux éléments dont le premier fait partie du second. Elle peut avoir lieu entre deux éléments géométriques dont le nombre de dimension est différent : le sommet d'un cube, l'arête d'une pyramide, etc... On pourrait aussi l'élargir au cas où l'on construit des objets à partir des éléments de base (cube, prisme, ...) et où l'on considère ces constructions comme des tous : on pourrait alors avoir une relation d'appartenance entre l'objet global et un de ses éléments constitutifs.

Le prédicat s'écrit : $\text{à}(x, y)$ (x appartient à y)

3.3.2.2. Relations de position

Les relations de position primitives sont représentées par les prédicats suivants :

- entre (x, y, z) (x est entre y et z). L'interprétation géométrique est la suivante : x est entre y et z si les coordonnées du centre de x appartiennent aux intervalles délimités sur chaque axe par les coordonnées des centres de y et de z.



- sur (x, y), qui représentera aussi bien les prépositions sur que sous, en intervertissant les arguments. Il définit une relation numérique très précise entre les coordonnées des centres et les dimensions des deux objets liés, telle que l'un soit "posé" sur l'autre (contact entre les deux objets sans intersection).

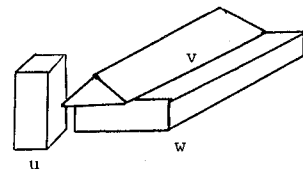


Ce prédicat privilégie un des axes, celui des z, car l'expression des relations équivalentes pour x ou y n'existe pas de façon simple en français. La raison en est que l'axe z correspond à la direction qu'a dans le monde réel la pesanteur qui n'existe pas pour les autres axes et qui se traduit par l'utilisation de concepts spécifiques. Pour représenter "x est à droite de y et contre lui", qui correspond, pour l'axe x, à "x est sur y" pour l'axe z, on aura deux primitives, l'une d'emplacement (pour à droite de), l'autre étant le prédicat contre.

- contre (x, y)

Il exprime une relation équivalente à "sur", mais de façon plus générale. L'objet x est contre l'objet y si une arête d'un d'entre eux appartient, au moins en partie, à la surface de l'autre.

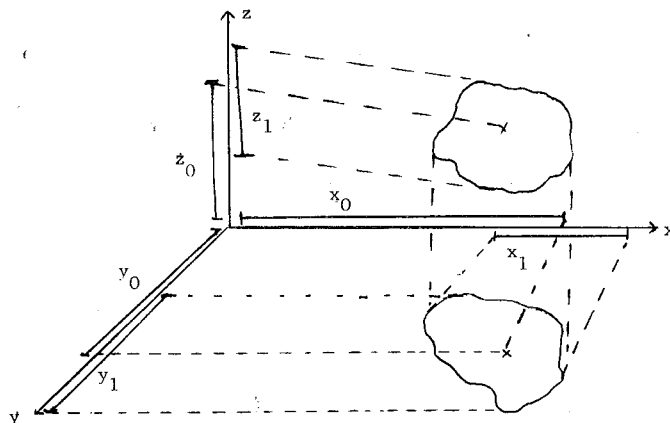
Ex : dans la situation suivante, sur (v, w) et contre (u, v) sont vrais



3.3.3. Les dimensions et les emplacements

On veut un mode de représentation assez général pour rendre compte, dans le domaine des dimensions, de structures syntaxiques très différentes, comme : long, court, plus haut que A, de longueur 100, ou, pour les emplacements : le plus à droite, au-dessus de A, 100 unités en avant de A, etc... Montrons d'abord que la représentation des dimensions et des emplacements peut être analogue.

Rappelons la description de ces deux éléments dans la base de données. La taille est donnée par un triplet (x_1, y_1, z_1) indiquant les trois longueurs des projections de l'objet sur les axes (longueur pour l'axe x, largeur pour l'axe y, hauteur pour l'axe z). Par exemple, pour un cube ou une sphère $x_1 = y_1 = z_1$, correspondant respectivement à l'arête du cube et au diamètre de la sphère. La position est également donnée par un triplet (x_0, y_0, z_0) indiquant les coordonnées du centre, l'emplacement d'un objet étant ramené à celui de son centre. Les six valeurs sont toutes des entiers positifs inférieurs à 1023 (un point de l'écran appartient à $[0, 1023]^3$).



La représentation semblable dans la base de données des dimensions et emplacements induit une représentation semblable de leur sens. Celle-ci obéit aux règles suivantes :

```

< P ← < COMP ' > ( < VALNUM > , < VALNUM > )
< COMP > ← égal | sup
< VALNUM > ← < op > ( < VALNUM > , < VALNUM > ) | < nb > | < valeur standard > | < dimension ou coor > ( < VF > )
< op > ← add | soustr | quot | prod
< nb > ← o | ... | 1023
< valeur standard > ← grande | petite | moyenne
< dimension ou coor > ← longueur | largeur | hauteur | grosseur
                           abscisse | ordonnée | cote | place

```

La forme générale est donc une comparaison de deux valeurs : égalité entre deux arguments numériques ou supériorité du premier sur le second. L'infériorité n'est pas indiquée car $\text{inf}(x, y) = \text{sup}(y, x)$. De même les autres opérateurs de comparaison (\neq , \succ , ...) peuvent être exprimés en utilisant la négation de prédicat non (p).

Les arguments peuvent être les suivants :

- une fonction calculant une dimension (ex : longueur (x)) ou une coordonnée d'objet (ex : cote (x)). Les fonctions donnent les trois dimensions, les trois coordonnées du centre, le volume (indiqué par grosseur (x)) et l'emplacement total (donné par place (x)).
- une valeur numérique entière. Ce peut être une valeur de dimension ou de coordonnée (ex : de largeur 200) ou un argument d'une fonction arithmétique. Ex : de largeur 200 : égal (largeur (x), 200).
- une valeur standard

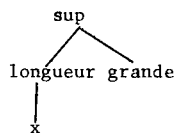
Prenons l'exemple : un bloc long ; la signification de long est : "dont la longueur est importante", c'est-à-dire supérieure à une certaine valeur, appelée "grande". De même, pour "court", la longueur est faible donc inférieure à une certaine valeur appelée "petite".

Que signifient ces valeurs "grande" et "petite" ?

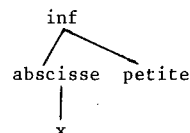
On peut diviser chaque axe en trois parties, séparées par les nombres V1 et V2. Une valeur "petite" est un nombre compris entre 0 et V1, une valeur grande, un nombre supérieur à V2 (et inférieur à 1023). Enfin, V1 et V2 ne sont pas, en général, égaux (un objet long n'est pas forcément un objet qui n'est pas court), et l'intervalle compris entre V1 et V2 correspond aux valeurs "moyennes". Les seuils V1 et V2 sont déterminés arbitrairement par le système déductif.



Ex : (un cube) long :

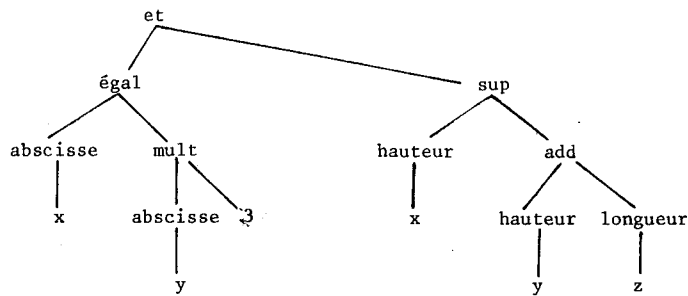


(un cube qui est) à gauche



- une fonction arithmétique, dont les arguments sont des valeurs numériques (fonction de dimension, nombre, fonction arithmétique, etc...). Ces fonctions permettent de définir les dimensions ou les emplacements au moyen de calculs très généraux : on peut ainsi indiquer qu'une dimension d'un certain objet vaut le double de la dimension de même objet, ou à la somme d'une autre dimension de ce même objet et d'une certaine valeur numérique. En fait on se limitera aux quatre opérations élémentaires avec des arguments généralement non composés car l'expression de calculs arithmétiques complexes n'a pas de structures linguistiques simples équivalentes.

Exemple : x trois fois plus à droite que y et de hauteur supérieure à la somme de la hauteur de y et de la longueur de z.



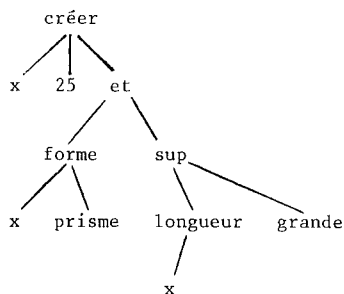
3. 4. REPRESENTATION DES ACTIONS

On a vu que sur le plan externe, on distingue trois sortes d'actions : création, destruction et modification. Indiquons la structure correspondant à chaque cas :

3.4.1. Création d'objets

La structure est : créer (x, n, p) où n représente le cardinal de l'ensemble des objets dont on demande la création et p le prédicat que x doit vérifier.

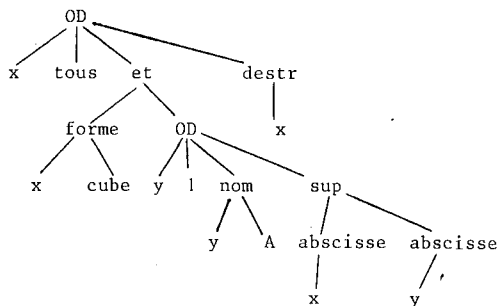
Ex : créer 25 longs prismes



3.4.2. Destruction d'objets

Détruire un objet x correspond à l'opération logique de trouver x grâce au prédicat qu'il doit satisfaire, et lui appliquer l'opération de destruction, prédicat ayant pour effet de faire disparaître la clause représentative. On a donc : OD (x, n, p, destr (x))

Ex : Détruisez tous les cubes qui se trouvent à droite de A



3.4.3. Modification d'objets

Chacune des modifications possibles correspond à la modification d'un des descripteurs d'objet : nom, forme, dimension, emplacement. Les diverses transformations sont donc représentées par le jeu des primitives indiquées : donner un nom utilise la primitive nom, déplacer un objet s'exprime par une primitive de position, dont la forme dépend de la nature du déplacement. De même, si on veut changer la forme d'un objet, c'est le prédicat forme (x, f) qui intervient.

Quelle que soit la nature de la transformation, cette opération est considérée, sur le plan logique, comme la création d'un nouvel objet, dont la transformation précisera les nouvelles propriétés, et la destruction de l'objet servant de base à la transformation. Cette représentation a été choisie parce que, dans le système déductif, on ne peut pas modifier des clauses existantes (la description d'un objet étant une clause PROLOG) mais seulement en ajouter de nouvelles.

Par exemple, poser une pyramide (existante) sur un bloc revient à créer un objet sur un bloc, à lui donner la forme et les dimensions de la pyramide et à supprimer la pyramide primitive.

CHAPITRE IV

SEMANTIQUE DU LANGAGE EXTERNE

On donne dans ce chapitre le lexique des mots utilisés pour pouvoir manipuler le monde qui nous intéresse et qui appartiennent au sous-ensemble du français qu'on a délimité. On donne pour ces mots la signification qu'ils ont dans notre cas et la structure profonde qu'on peut leur faire correspondre. La signification et donc la représentation d'un mot dépendent en général de la structure syntaxique dont il fait partie, celle-ci exprimant le rôle particulier qu'a dans la phrase ce mot. Par exemple, le nombre 100 n'a pas la même interprétation dans "créez 100 cubes" et dans "un bloc long de 100 unités : dans un cas il indique le cardinal d'un ensemble, dans l'autre la mesure d'une dimension.

On étudie successivement les verbes, les noms communs, les adjectifs, les prépositions, les déterminers, les adverbes, les pronoms, les appositions, les conjonctions de coordination, les nombres et les signes de ponctuation.

1 - LES VERBES

Ils servent soit à exprimer une modification de la scène, soit, en général avec d'autres mots, tels les prépositions, à exprimer des relations ou des propriétés. On peut les classer de la façon suivante :

1.1. - VERBES EXPRIMANT UNE MODIFICATION DE LA SCENE

1.1.1. Verbes de création

Ces verbes (créer, construire, etc...) sont considérés comme des synonymes absolus et ont donc tous la même représentation qui est, on l'a vu: créer (x,n,p).

1.1.2. Verbes de destruction

De même, ces verbes (effacer, supprimer, etc...) sont représentés par OD(x,n,p,destr(x)).

1.1.3. Verbes de transformation

Ces verbes modifient une des caractéristiques des objets et, la représentation générale étant : créer(x,n, et (p,OD(y,l,q, et (memobjet(x,y), destr(y))))), le seul élément qui varie selon la transformation est le prédicat p qui définit la modification du descripteur d'objet qu'induit la transformation. On examinera donc pour chaque type de transformation la forme que prendra p.

Affectation ou changement de nom

Les verbes appeler, nommer, etc... permettent de donner un nom à un objet ou de modifier le nom qu'il possède. Ces verbes possèdent 2 arguments : l'objet et le nom qu'on lui donne.

Le prédicat p est $\text{nom}(x,n)$.

Exemple : Appelez "A" la grande pyramide $p=\text{nom}(x,"A")$

Translation

Ces verbes permettent de déplacer les objets. Plusieurs cas sont à distinguer :

a. Translation non directionnelle

Ces verbes (déplacer, mettre, etc...) n'expriment pas la valeur du déplacement. Ils possèdent deux arguments : l'objet à traduire et l'endroit où il devra se trouver après le déplacement.

p exprime ce lieu par une relation de position ou d'emplacement.

Exemple : posez A sur B : $p=\text{OD}(z,l,\text{nom}(z,B), \text{sur}(x,z))$

Mettez A à droite de B : $\text{"}, \text{sup}(\text{abscisse}(x), \text{abscisse}(z))$

Si le second argument du verbe est absent (exemple : déplacez A), sa signification est que le nouvel emplacement doit être différent de l'ancien, ce qui se traduit par la négation d'une égalité de place.

Exemple : Déplacez A : $p=OD(z,l,nom(z,A),non(égal(place(x),place(z))))$.

En fait, z désigne comme y dans la représentation générale, l'objet de départ ; les variables y et z pourront donc être confondues, ce qui fait que la représentation générale sera simplifiée : l'OD permettant de rechercher y devient inutile, le prédicat p ayant permis de l'obtenir et, pour toutes les transformations d'un objet par rapport à lui-même, $OD(y,l,q,et(memobjet(x,y),destr(y)))$ sera remplacé par : $et(memobjet(x,y),destr(y))$.

Déplacez A : $créer(x,l,et(OD(y,l,nom(y,A),non(égal(place(x),place(y))))),et(memobjet(x,y),destr(y)))$

Le second argument du verbe peut être une direction. Celle-ci indique alors l'axe de déplacement (vers la droite \rightarrow abscisse, vers le haut \rightarrow cote, etc...).

Un déplacement vers la droite, par exemple, signifie une augmentation de l'abscisse, tandis que vers le bas, il signifie une diminution de la cote : aussi, suivant le sens dont le déplacement est effectué sur un certain axe, les arguments du prédicat de comparaison correspondant seront inversés par rapport à ce qu'ils seraient dans l'autre sens.

On peut enfin avoir un argument numérique qui sert à valuer le déplacement, le signe étant alors "égal".

Exemple : Déplacez A vers l'arrière : $p=OD(y,l,nom(y,A),sup(ordonnée(y),ordonnée(x)))$

Déplacez A vers la droite de 100 unités : $p=OD(y,l,nom(y,A),égal(abscisse(x),add(abscisse(y),100)))$

b. Translation directionnelle

Ces verbes (avancer, reculer, remonter, descendre) expriment à la fois le déplacement, l'axe (y ou z) et le sens (vers le point origine ou dans le sens contraire). Leur représentation a été indiquée au paragraphe précédent puisque on peut exprimer ces verbes par un verbe de déplacement avec une direction : avancer signifie "déplacer vers l'avant" ou descendre signifie "déplacer vers le bas".

Homothétie

On dispose de verbes exprimant l'augmentation ou la diminution d'une dimension d'objet (longueur, largeur, hauteur) ou de la taille globale (grosseur). La transformation géométrique décrite est donc l'homothétie. Pour chaque dimension, il existe un verbe exprimant l'augmentation et un autre la diminution de valeur ; allonger et raccourcir pour la longueur, élargir et rétrécir pour la largeur, rehausser et aplatir pour la hauteur et enfin agrandir et rapetisser pour la grosseur. Le sens de ces verbes est équivalent à celui des verbes de déplacement directionnel, les dimensions remplaçant les coordonnées dans le prédicat représentatif. On voit par exemple l'analogie entre avancer (augmenter l'ordonnée du centre) et élargir (augmenter la largeur).

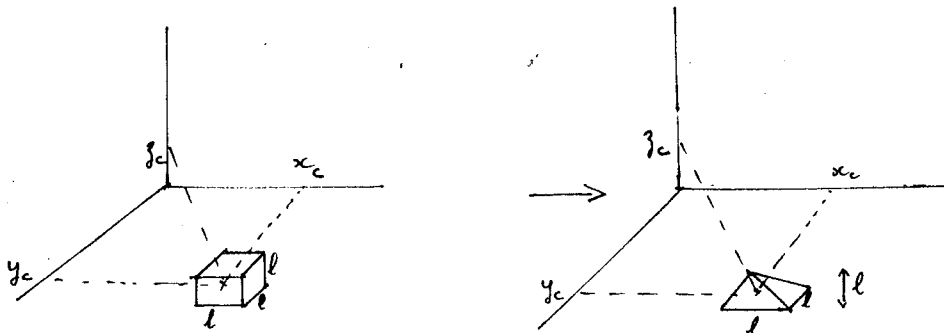
Exemple : élargissez A : $p=OD(y,l,nom(y,A),sup(largeur(x),largeur(y)))$

Il existe d'autre part des verbes (augmenter, diminuer, faire croître, etc...) exprimant la croissance ou la décroissance d'une valeur numérique. Ils possèdent 2 arguments : un objet et un élément descripteur de celui-ci. Il s'agit en général d'une dimension et l'ensemble verbe + dimension est alors équivalent à un verbe d'homothétie : augmenter la longueur de A de 100 = allonger A de 100.

Transformation de forme

On a des verbes (changer, transformer, etc...) qui opèrent une transmutation d'objet, transformant par exemple un prisme en pyramide. Cette transformation, comme toutes les autres ne modifie qu'un seul élément décrivant l'objet, en l'occurrence sa forme, alors que son nom, son numéro, son centre, ses dimensions restent les mêmes. Par exemple, la transformation d'un cube en pyramide donnera à celle-ci une base carrée et une hauteur égale à l'arête de la base.

Exemple : transformez le cube C en pyramide : $p=forme(x,pyramide)$



1.1.4. Autres verbes d'action

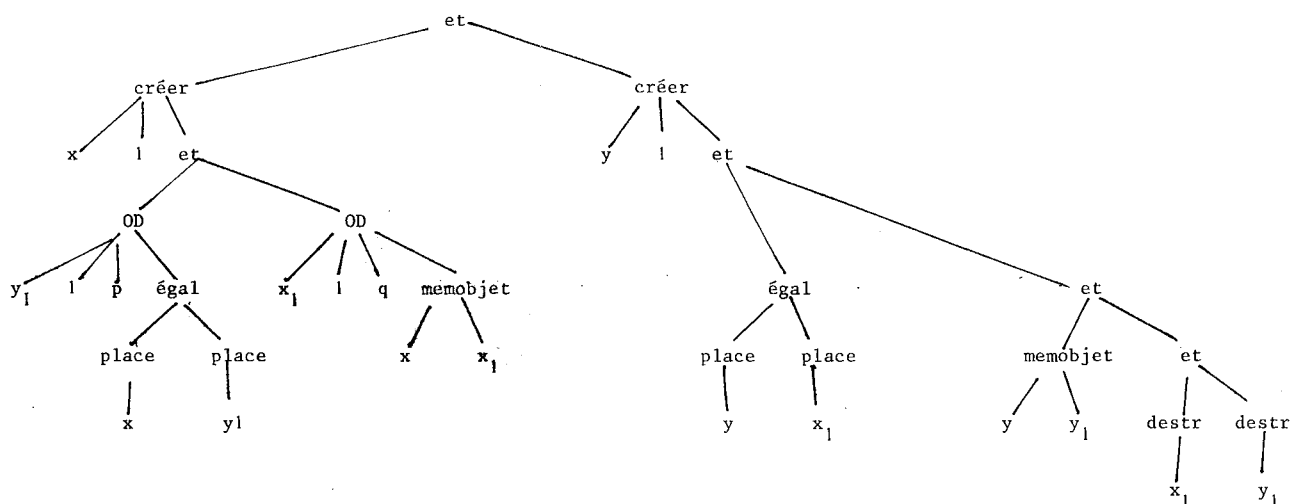
On peut utiliser quelques autres verbes dont le sens est spécifique :

- donner : il permet d'introduire la propriété qu'on veut affecter à un objet (donnez au cube le nom A ; donnez à A la forme d'un cube). Il deux arguments : un objet et la propriété qu'on veut lui donner. Ce verbe peut donc distribuer diverses sortes de modifications et le prédicat p est de nature variée (nom(x,A) ; forme(x,cube),etc...).

- centrer : il précise l'emplacement d'un objet. On a p=centre(x,coord)

- intervertir : ce verbe à 2 arguments-objets (exemple : intervertissez A et B) diffère des autres qui n'agissent que sur un seul objet à la fois en opérant simultanément sur ses deux arguments dont il échange les places. Il signifie en effet : mettre le premier objet à la place du second et réciproquement. La représentation est donc différente de celle des autres transformations : c'est une double création d'objets, placés au même endroit que ceux de départ, les autres propriétés étant échangées, suivie d'une double destruction des objets de départ.

La représentation générale est donc :



Pour "intervertissez A et B", on a p=nom(y₁,A) et q=nom(x₁,B).

On pourrait généraliser l'usage de ce verbe à l'échange d'arguments quelconques au lieu du seul emplacement (exemple : intervertissez les formes de A et de B).

1.2. VERBES INTRODUISANT OU EXPRIMANT DES PROPRIETES

Ces verbes ne sont jamais les verbes principaux de la phrase : on les trouve donc dans les relatives ou au participe passé ou présent.

1.2.1. Verbes auxiliaires (avoir et être)

Examinons les verbes avoir et être qui peuvent tenir dans la phrase des rôles très différents. Ils peuvent d'abord servir d'auxiliaires de conjugaison. Ils servent d'une part à exprimer les temps composés (passe composé : le cube que vous avez créé, etc...). Si, dans un système plus évolué conservant le souvenir des actions effectuées, il faudrait traiter le cas des temps composés, dans le système actuel seul le temps présent est utilisé. On ne s'intéresse donc pas à cet aspect des verbes auxiliaires. D'autre part, le verbe être peut servir à exprimer la voix passive (exemple : le cube qui est surmonté d'une pyramide). Dans ce cas, être n'a qu'un rôle syntaxique et l'usage du passif correspond à un renversement du sujet et de l'objet par rapport à la voix active.

Avoir et être peuvent aussi être utilisés seuls. Dans un cadre général leur signification serait très variable, mais dans notre cadre, on n'a que quelques possibilités qu'on va indiquer à l'aide d'exemples :

Exemple 1 : un bloc qui a une longueur de 100 unités (ou : qui a pour nom A, qui a pour centre (x_1, y_1, z_1) , etc...).

On voit que le verbe introduit un élément caractérisant un objet (qui, en effet, a un nom, un centre, une longueur). Le modifieur de l'objet ainsi introduit s'exprime par les prédicats centre, nom, égal,...

Exemple 2 : un cube qui est long (ou : qui est plus gros que A, qui est sur C, etc...). Le verbe introduit une propriété ou une relation sous forme de GA ou de GP. En fait, le verbe et le pronom relatif le précédant n'ont pas de signification réelle (un cube qui est long \leftrightarrow un cube long) et permettent surtout d'assurer le lien entre un objet et le GA ou GP qui le modifie.

1.2.2. Verbes d'appartenance

Ces verbes, en conjonction avec une préposition introduisant le deuxième argument, expriment la relation d'appartenance $a(x,y)$. On a "appartenir à" et "passer par" dont l'ordre des arguments est inversé dans le prédicat $a(x,y)$.

Exemple : x passant par y : $a(y,x)$

1.2.3. Autres verbes

- Le verbe "surmonter" a le sens de "être sûr" (le prisme qui surmonte A \leftrightarrow le prisme qui est sur A). La représentation est donc sur (x,y) .

- Certains verbes exprimant des modifications peuvent également être utilisés pour indiquer une propriété : C'est le cas par exemple de "centrer" (un cube centré en $(100,100,100)$), des verbes de nomination (le cube appelé "A"),... Le prédicat utilisé reste le même, mais sa place dans l'arborescence est différente.

2 LES NOMS COMMUNS (SUBSTANTIFS)

Le rôle essentiel des noms communs est d'indiquer la forme des objets intervenant dans une phrase, mais ils peuvent avoir d'autres significations.

2.1. NOMS COMMUNS D'OBJETS

Ces noms (cube, bloc, prisme, etc...) expriment 2 concepts différents. D'une part, ils servent d'indicateurs d'objet et sont représentés, en tant que tels, par une VF, argument soit du prédicat créer si on ajoute ce nouvel objet, soit de OD si on désigne par ce mot un objet déjà existant ; d'autre part, ils indiquent une propriété de ces objets, leur forme.

Exemple : "Créer un cube" : créer $(x,l,forme(x,cube))$

Seul, un nom général comme "objet" ne véhicule pas de propriété de forme.

2.2. NOM DE DIMENSION

Ces noms sont dérivés d'adjectifs de dimension : il s'agit des noms hauteur, longueur, largeur (qui expriment les dimensions sur les axes z, x, y), grandeur et grosseur (qui sont synonymes et expriment la taille globale ou volume). Mais alors qu'un adjectif comme "large" exprime d'une part le concept de largeur et d'autre part la valeur "forte" de celle-ci, le nom correspondant (largeur) n'exprime, à lui seul, aucune valeur. Ces noms sont donc accompagnés d'un élément chargé d'exprimer cette valeur de dimension soit sous forme absolue, soit sous forme relative (exemple : un bloc de longueur 100 ; de grande longueur ; dont la longueur est supérieur à la largeur). La représentation de ces noms est donc un prédicat d'égalité ou de supériorité.

Exemple : "de grande longueur" : $sup(longueur(x),grande)$

Le mot "arête" a un sens variable qui dépend de l'objet référé.

Pour un cube, l'arête signifie une quelconque des dimensions et on prendra par exemple la longueur.

Pour un bloc, si le mot "arête" est suivi d'une liste de valeurs, celles-ci représentent respectivement longueur, largeur et hauteur. Pour d'autres objets dont les arêtes et les dimensions ont des valeurs différentes (par exemple, la pyramide dont la hauteur et l'arête reliant le sommet à un des points de la base sont différentes), il faudrait, dans le système déductif, disposer d'outils mathématiques faisant la transformation valeur d'arête \rightarrow valeur de dimension, qui seraient assez complexes ; aussi ce cas n'a pas été retenu.

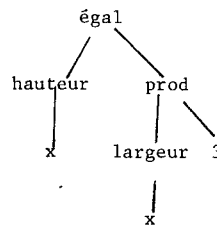
Exemple : (un cube) d'arête 100 : égal (longueur(x), 100).

2.3. AUTRES NOMS COMMUNS

On utilise d'autres noms communs, chacun constituant un cas d'espèce du point de vue sens.

- le mot "unité" derrière un nombre confirme qu'il s'agit de la valeur d'une dimension (exemple : long de 100 unités). En réalité le nombre exprime à lui seul cette signification et l'usage du mot "unité" est redondant.
- les mots : centre, nom, forme, ... (de nom "A", de forme cube, ...) accompagnés d'une valeur correspondent à des prédicats de même nature (de nom "A" : nom(x,A)).
- les mots : place, endroit ("créez A au même endroit que B) correspondent à un prédicat d'emplacement, par exemple égal (place(x), place(y)).
- le mot "fois" n'est employé que précédé d'un nombre (exemple : trois fois plus long que C). Il représente alors l'opération de multiplication, dans un prédicat de dimension ou de place, ses 2 arguments étant la dimension ou la coordonnée qu'indique l'adjectif et le nombre.

Exemple : trois fois plus haut que large :



3 LES ADJECTIFS

Dans une étude linguistique générale, les adjectifs ont des significations de type très différent (couleur, texture, ...). En revanche, dans notre monde on ne trouve que quelques groupes de signification.

3.1. ADJECTIFS TEMPORELS

Ce sont des adjectifs qui se rattachent en fait aux nombres ordinaux, tant par leur signification que par leur place syntaxique dans la phrase. Ce sont les adjectifs dernier et avant-dernier dont la représentation est $\text{ref}(x,n,p)$.

3.2. ADJECTIFS SPACIAUX

Ces adjectifs, les plus importants dans notre monde, indiquent les valeurs de taille. Pour chaque dimension, on a à la fois des adjectifs indiquant une valeur grande et une valeur petite. Pour l'axe x : long et court ; pour l'axe y : large et étroit ; pour l'axe z : haut et bas. Enfin, pour la taille globale : gros ou grand et petit.

Il y a dans l'interprétation que nous avons prise de ces adjectifs (tout comme d'ailleurs dans celle des noms et des verbes qui dérivent de ces adjectifs, tels longueur ou allonger par rapport à l'adjectif long) une certaine part d'arbitraire dans la mesure où le sens de mots comme "long", "petit", "gros",... n'a pas, en réalité, une définition très précise. On pourrait par exemple imaginer que la longueur désigne la plus grande dimension quelle qu'elle soit. On pourrait aussi interpréter le longueur comme étant la dimension mesurée sur l'axe de profondeur, c'est à dire l'axe y. De même, l'adjectif grand pourrait signifier aussi bien la hauteur que le volume. Comme aucune de ces interprétations n'est plus exacte qu'une autre a priori, il est bien évident qu'une d'entre elles à due être choisie, qui s'accorde avec la structure profonde adoptée. On verra que pour les prépositions des problèmes analogues de choix du sens de certains mots de la langue courante se posent à nouveau.

Pour la représentation de ces adjectifs, on a vu qu'on utilise des prédicats de comparaison (égalité ou supériorité) de valeurs numériques.

- Adjectifs absolus ("cube long", "bloc bas"). La représentation est une supériorité entre une dimension de l'objet et les valeurs standards "grande" et "petite".

long : $\text{sup}(\text{longueur}(x), \text{grande})$

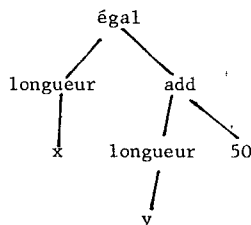
bas : $\text{sup}(\text{petite}, \text{hauteur}(x))$

- Comparatif : il est utilisé dans deux cas différents : comparaison des dimensions d'un même objet ("aussi haut que large"), ou comparaison d'une dimension de plusieurs objets ("aussi haut que A est large"). On peut aussi combiner ces 2 possibilités en comparant des dimensions différentes d'objets différents ("aussi haut que A est large"). La signification est une comparaison entre les dimensions (éventuellement la même) des objets (éventuellement le même). Le signe de la comparaison est "égal" pour le comparatif "aussi" et "sup" dans les autres cas, les arguments étant inversés suivant que le comparatif est de supériorité ou d'infériorité.

Exemple : "plus long que "A" : $\text{sup}(\text{longueur}(x), \text{largeur}(y))$
 "moins haut que large" : $\text{sup}(\text{largeur}(x), \text{hauteur}(x))$
 "aussi long que A est large" : $\text{égal}(\text{longueur}(x), \text{largeur}(y))$

On peut aussi dans un comparatif évaluer la valeur de la comparaison. L'expression "n fois" ("3 fois plus long que large") introduit une multiplication du deuxième argument par le nombre indiqué, tandis que le signe sup cède la place au signe égal, la multiplication permettant en effet de savoir exactement en quoi le premier argument est supérieur au second (dans l'exemple, la longueur vaut 3 fois sa largeur). On peut aussi mesurer la différence entre les deux dimensions comparées par l'expression "de n unités" ("plus long que large de 50 unités"). Par rapport au cas précédent, la multiplication est remplacée par l'addition (dans l'exemple, la longueur est égale à la somme de la largeur et de 50).

Exemple : plus long que A de 50 unités



Quand l'adjectif utilisé exprime une valeur faible (court, étroit, bas, petit), les deux arguments formels ont leur place échangée par rapport aux adjectifs de valeur forte.

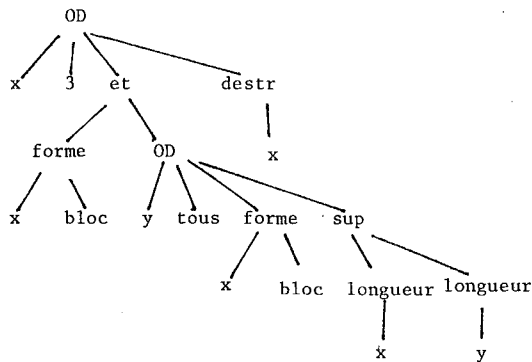
x plus court que y = y plus long que x

x moins court que y = y moins long que x = x plus long que y

- Superlatif

Dans un superlatif, on fait implicitement référence à des objets. Par exemple, "le plus long cube" peut se paraphraser en "le cube plus long que tous les cubes" et la représentation devient celle adoptée pour le comparatif.

Exemple : "Supprimez les 3 plus longs blocs"



Remarque :

Si au lieu de "longs", on avait en "courts", les arguments de sup auraient été intervertis.

- Adjectif avec valeur numérique

On peut enfin utiliser les adjectifs, uniquement sous leur forme "forte", avec un nombre. La représentation est alors une égalité entre la dimension correspondante et la valeur numérique.

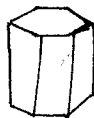
Exemple : long de 100 unités : égal (largeur(x),100).

3.3. ADJECTIFS DE COMPARAISON

On dispose d'adjectifs servant à comparer des valeurs : même, égal, supérieur, inférieur, ("2 cubes de même taille, un bloc dont la longueur est supérieure à 100",...). La représentation de ces adjectifs est évidemment celle d'un prédicat de comparaison.

3.4. ADJECTIFS DE FORME

Dans un système plus complet, où les formes des objets pourraient être beaucoup plus générales, des adjectifs de formes se révéleraient utiles pour donner la description de la forme de l'objet (exemple : un cylindre à base hexagonale qui serait représenté par :



Des adjectifs tels que carré (une pyramide à base carrée), triangulaire (une pyramide à base triangulaire), octogonale, isocèle, pentagonale, ... seraient alors représentés comme une forme d'objet générale avec une caractéristique spéciale tel le nombre de côtés de la base ou l'égalité de certaines valeurs.

4. LES PREPOSITIONS

Parmi les prépositions que l'on utilise, on peut distinguer trois groupes : les prépositions d'emplacement, de position, et les autres qui introduisent propriétés et relations.

4.1. PREPOSITIONS D'EMPLACEMENT

On a un certain nombre de prépositions, et surtout de locutions prépositionnelles qui exprimeront les positions des objets par rapport aux 3 axes de références : pour l'axe x, on a à droite de et à gauche de, pour l'axe y, devant et derrière ou en avant de et en arrière de ; pour l'axe z, au-dessus de et au-dessous de ou en haut de et en bas de. La signification que nous donnons à ces prépositions est analogue à celle que nous donnons aux adjectifs (ou substantifs) de dimension. C'est ainsi que nous disons que "l'objet x est à droite de l'objet y" si la mesure de l'abscisse du centre de x est supérieure à la mesure de l'abscisse du centre de y (figure 1), ce qui équivaut pour les dimensions à la signification de "x est plus long que y".

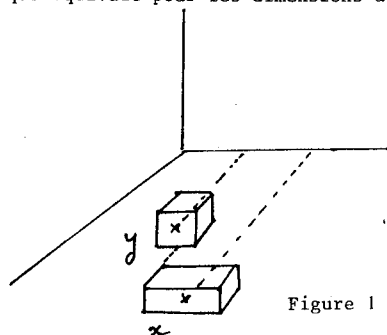


Figure 1

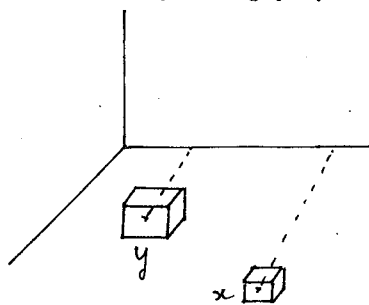


Figure 2

"x est à droite de y"

Le sens qu'on a donné à cette préposition, de même qu'aux autres, peut poser un problème. Un autre sens, peut être plus proche de l'usage habituel qu'on donne à ces mots, serait le suivant : x est à droite de y si l'abscisse du point le plus à gauche de x est supérieure à l'abscisse du point le plus à droite de y (figure 2).

On pourrait dans un système plus évolué choisir cette signification : le système déductif ne prendra pas dans ce cas les coordonnées des centres des objets, mais celles des points les plus à droite ou les plus à gauche qu'on calcule aisément à partir des coordonnées du centre et des trois dimensions ; cela entraînerait d'ailleurs peu de modifications de la représentation choisie. L'inconvénient de cette interprétation est qu'on ne saurait pas exactement comment définir l'emplacement d'un objet. Par exemple la proposition "x est à droite de y ou y est à droite de x" ne serait pas toujours vraie. De même, il ne serait pas évident de savoir ce qu'on entend par "le cube le plus à droite" (Comment le calculerait-on ? Existerait-il toujours ?).

La signification que nous avons choisie, au contraire, est basée sur une définition claire de l'emplacement d'un objet et permet de représenter les prépositions d'emplacement d'une façon simple et cohérente, la même que celle adoptée pour les indicateurs de dimensions.

On doit simplement noter le fait général que la formalisation d'une langue n'est pas entièrement possible de façon satisfaisante, à priori, dans la mesure où le sens donné à certains mots (par exemple : long, contre, à droite de, ...) n'est pas le même pour tout le monde et diffère également suivant le sujet qui est traité : dans la mesure où on manipule un monde géométrique et non pas réel, le sens des mots permettant cette manipulation sera particulier.

On verra dans le chapitre conclusif comment éviter cette difficulté en augmentant les possibilités du système pour permettre à l'utilisateur de définir sa propre sémantique. On voit que les mêmes possibilités que pour les adjectifs de tailles existent pour les prépositions d'emplacement. Celles-ci peuvent exister sous deux formes : sous forme de locution adjectivale : le cube qui est à droite, le bloc le plus en avant ; elles peuvent aussi exister sous forme de préposition : le cube à droite de A. On peut aisément leur faire correspondre la forme adjectivale équivalente qui induit une représentation de même nature que pour les adjectifs dimensionnels. Par exemple, "x est à droite de y" équivaut à "x est plus à droite que y".

Exemple : qui est à droite : sup(abscisse(x), grande)
 qui est à gauche de y : sup(abscisse(y), abscisse(x))
 qui est plus à droite qu'en avant : sup(abscisse(x), ordonnée(x))

4.2. PREPOSITIONS DE POSITIONS

Ce sont les prépositions sur, sous, entre et contre dont on a vu au chapitre précédent le sens et la représentation. Exemple : x sous y : sur (y,x).

4.3. AUTRES PREPOSITIONS

Elles introduisent des propriétés ou des relations. Ce sont :

à : après le verbe appartenir il contribue à la relation d'appartenance ; suivi d'une liste de coordonnées, il indique un emplacement et est alors exprimé par le prédicat centre.

Exemple : posez x à (x_1, y_1, z_1) : centre $(x, (x_1, y_1, z_1))$

par : après le verbe passer il contribue à la relation d'appartenance en inversant les arguments.

en : il a deux sens possibles : il peut, comme à, marquer un emplacement (centré en (x_1, y_1, z_1)) ; il sert aussi, quand on fait une transformation de forme, à introduire la nouvelle forme (transformez x en pyramide).

pour-comme : ils introduisent une propriété d'objet sous forme d'un de ses éléments descripteur accompagné de sa valeur (donnez-lui comme nom A et pour longueur 100).

de : il a les sens suivants :

- il exprime une relation d'appartenance quand il lie deux éléments de dimensions différentes (l'arête du cube). On pourrait élargir cette possibilité aux éléments constitutifs d'une figure (le clocher de l'église).

- il peut servir à introduire un argument numérique complément d'un adjectif ou substantif de dimension (haut de 100 unités ; d'une hauteur de 100). La préposition sépare alors les deux arguments, dimension et nombre, qui constituent les arguments du prédicat égal correspondant.

- il peut enfin introduire un ensemble spécifiqueur-valeur donnant une des caractéristiques de l'objet : une de ses dimensions, son nom, son centre.

Exemple : un cube de nom A et de longueur 100.

5 LES DETERMINEURS

Ce terme englobe à la fois les articles définis et indéfinis (le, un,...), et les adjectifs possessifs (son,...), démonstratifs (ces,...) et indéfinis (certaines,...).

Afin de voir la représentation qu'on donne à ces différents mots, voyons les différents éléments qui les caractérisent et la signification qu'on peut leur donner.

- le premier élément est le genre (masculin ou féminin). Ce genre provient directement du genre du nom que le déterminateur précède et qui n'a en général, en français aucune signification (un cube n'a pas plus de virilité qu'une pyramide n'a de féminité). Le genre ne servira donc qu'à vérifier la cohérence du groupe nominal et ne sera pas représenté dans la structure profonde.

- le second élément est le nombre. On distingue généralement le singulier et le pluriel. Comme le nombre sert ici à désigner le cardinal de l'ensemble des objets désignés simultanément, on prend des valeurs plus significatives. Ces valeurs sont : 1 (quand le déterminateur désigne un seul élément : le, un, ce,...), des (s'il désigne un nombre indéterminé d'objets : des, certains,...) et tous (dans les autres cas, c'est-à-dire s'il désigne tous les objets à la fois : tous, les (en effet "détruyez les gros cubes" par exemple signifie "détruyez tous les gros cubes")).

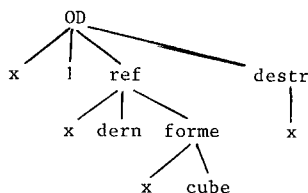
Ce nombre est représenté comme le deuxième élément caractérisant un objet, après la VF qui le désigne, soit dans le prédicat créer (x,n,p), soit dans le prédicat OD (x,n,p,p,q).

Il est évident que le nombre, ainsi considéré, n'a de sens que si le déterminateur appartient à un GN dont le noyau est un substantif d'objet et n'intervient pas dans des structures telles : "la longueur", "le centre", où les déterminateurs sont non significatifs.

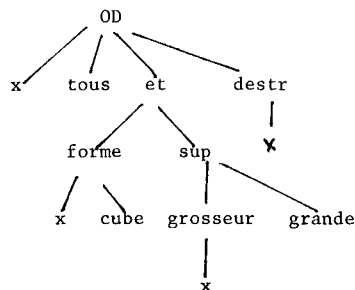
- le troisième élément est la référence qu'il comporte éventuellement, qui permet de définir de quels éléments on parle. Pour un article indéfini ou certains adjectifs indéfinis (certains, plusieurs,...), cette référence n'existe pas.

Pour les autres, il y a référence implicite au dernier objet mentionné (ou aux derniers si on en mentionne plusieurs). Quand on dit "le cube", on sous-entend "le dernier cube". En fait, cet élément de référence n'est significatif que quand l'objet n'est pas modifié de manière importante par ailleurs. En effet, quand on dit : "le gros cube", il s'agit bien du dernier gros cube, mais si on dit : "le plus gros cube", le superlatif est une indication permettant à elle seule de désigner l'objet et la référence au "dernier plus gros cube" n'a plus aucun sens. De même, dans "le nième cube", la référence explicite qui est faite montre clairement qu'il n'y a pas référence au dernier objet. Dans la mesure où le rôle référent des déterminateurs existe, celui-ci se traduit par le prédicat ref(x,dern,p).

Exemple ¶ : "détruyez le (ou ce) cube"



Exemple 2 : "détruisez tout gros cube"



Cet exemple montre que le genre syntaxique (ici le singulier) peut être différent du genre sémantique (ici tous qui implique le pluriel). En effet, l'exemple donné peut se paraphraser en : "détruisez tous les gros cubes".

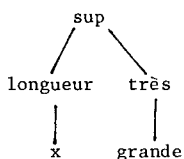
6 LES ADVERBES

Parmi les adverbes on distingue les adverbes modifiant les verbes et les adverbes modifiant les adjectifs. Nous ne nous occupons pas du premier type, ces adverbes n'intervenant pas dans notre application.

Les adjectifs qui peuvent être modifiés sont les adjectifs de dimension (long, haut,...), ainsi que les prépositions d'emplacement jouant le rôle d'adjectif (en haut, en arrière, à droite,...), l'adverbe ayant alors pour rôle de préciser de façon relative, la valeur de la dimension qu'indique l'adjectif. Les adverbes dont nous usons sont les adverbes de comparaison (plus, moins, aussi) intervenant dans les comparatifs et les superlatifs et dont nous avons déjà vu la représentation en termes de prédicats d'égalité ou de supériorité de deux valeurs.

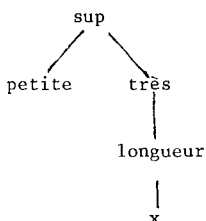
Il y a aussi l'adverbe "très" (très haut, très à gauche,...) dont la signification est de renforcer la valeur qu'exprime l'adjectif. On peut donc considérer "très" comme un opérateur unaire qui augmente la valeur qu'il a pour argument. C'est le système déductif qui déterminera l'expression de cette augmentation (par exemple : multiplication de cette valeur par 2). Si l'adjectif modifié est de valeur forte (exemple : long), c'est la valeur "grande" qui sert d'argument à l'opérateur "très".

Exemple : "un objet très long" signifie : "dont la longueur est supérieure à la valeur standard "grande" même augmentée. D'où :



Si au contraire l'adjectif exprime une valeur faible (exemple : court) c'est la dimension de l'objet qui sert d'argument à "très". Par exemple, un objet est très court si la longueur de cet objet augmentée d'une certaine façon reste inférieure à la valeur standard "petite" (ce qui revient en fait à dire que "très", appliqué à une valeur faible diminue la valeur "petite").

Exemple : très court



7 LES PRONOMS

Les seuls pronoms qu'on considère sont les pronoms personnels et les pronoms relatifs.

7.1. PRONOMS PERSONNELS

Ils servent à remplacer des objets précédemment évoqués. Quoique, dans un cas général, les objets ainsi référencés puissent se trouver dans une position quelconque devant le pronom, dans le cas qui nous intéresse, les pronoms personnels remplacent les objets créés les plus proches (exemple : "créez un cube et une pyramide". "Détruisez-le").

Comme les noms communs de forme, les pronoms personnels sont représentés à deux niveaux différents. D'une part, ils servent à représenter un objet (ou des objets). Ils correspondent donc à une certaine VF et à un OD puisque les pronoms référencent des objets déjà mentionnés, donc existants.

D'autre part, ils sont vecteurs des propriétés qui caractérisent ces objets et du cardinal de l'ensemble d'objets qu'ils évoquent. En effet, contrairement à un nom, un pronom ne peut pas être modifié ou précisé par des éléments tels que : détermineurs, nombre, groupe adjectival, ... et si l'objet qu'ils représentent est OD(x,n,p,q), seul le prédicat q sera en partie exprimé par une autre partie du discours que le seul pronom.

Intéressons nous donc aux deux types de signification qu'ils comportent : cardinalité et propriétés.

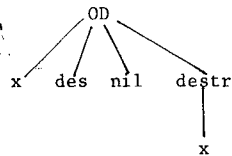
Pour ce qui concerne le cardinal, on aura les mêmes valeurs que pour les détermineurs : 1 pour les pronoms au singulier (le, il, elle, ...), "des" pour les pronoms indéfinis, en particulier "en" (exemple : "détruisez-en"), "tous" pour les pronoms définis correspondant aux articles définis pluriels.

Les propriétés sont le genre et la référence.

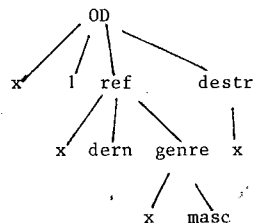
- Ce genre est une propriété signifiante pour les pronoms ayant un genre unique (le, elle, mais pas les, en). Il permet de délimiter la nature des objets auxquels on fait référence en divisant en deux classes l'ensemble des formes d'objets, d'après leur genre.

- La référence au dernier objet est exprimée par les pronoms définis comme le, la, elle, lui.

Exemple : détruisez-en



détruisez-le



7.2. PRONOMS RELATIFS

Ceux que nous utilisons sont : qui, que, dont, où. Nous négligeons donc des pronoms comme quoi, lequel, desquels, que nous devrions considérer dans un cadre plus général (par exemple, pour la question : sur quoi A est-il posé ?).

Pour les pronoms relatifs, le problème de la référence ne se pose pas car l'objet qu'ils représentent est toujours placé juste devant eux, dans la même phrase. On peut donc savoir quel est l'élément référencé dès la phrase d'analyse et la référence n'est pas indiquée dans la structure du sens.

Les pronoms relatifs ont du reste un rôle très différent des autres pronoms : ils introduisent des propositions relatives, structures emboîtées dans la proposition principale.

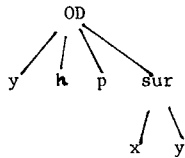
Exemple : Détruisez le cube que surmonte une pyramide dont le nom est A.

La proposition relative introduite permet de préciser une propriété de l'objet que représente le pronom relatif. Celui-ci n'a donc pas de signification en lui-même. Selon le pronom utilisé, les propriétés exprimées par la propriété relative ont une forme et donc une représentation différente.

Examinons le cas des 4 pronoms relatifs :

- le pronom qui est le sujet syntaxique de la proposition relative. Il introduit en tant que tel une propriété modifiant l'objet remplacé : x qui est gros = x est gros

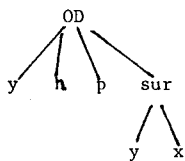
Exemple : (x)qui surmonte y



C'est le pronom le plus utilisé car il permet d'insérer de façon simple une propriété.

- le pronom que est l'objet syntaxique de la proposition. Celle-ci exprime en général une relation entre le sujet syntaxique et son objet représenté par "que" avec un renversement des arguments par rapport au cas précédent.

Exemple : (x)que surmonte y



- le pronom où remplace un circonstanciel de lieu (exemple : l'endroit où est A). Il remplace donc un nom d'emplacement et la proposition relative se traduit par un prédicat d'emplacement comme égal(place(x), place(y)).

- le pronom dont a un rôle syntaxique très particulier. Il remplace un GP introduit par "de", ce GP pouvant dépendre d'un élément quelconque de la proposition relative.

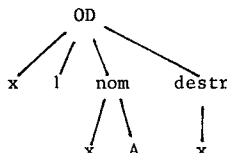
Dans notre cas, le sujet est le seul élément auquel dont peut se rapporter : "le cube dont la longueur est 100" signifie "la longueur du cube est 100". La proposition relative sert alors à préciser une valeur d'une spécification de l'objet : nom, longueur, forme, etc...

Exemple : (x)dont le nom est A : nom(x,A)

8 LES APOSITIONS (OU NOMS PROPRES)

On distingue par ce terme les noms qu'on donne aux objets, même s'ils n'ont pas le rôle syntaxique d'aposition dans la phrase. On les écrit entre guillemets pour les distinguer des autres mots lors de l'analyse. La représentation en est nom(x,n). Outre cette propriété, l'aposition, quand elle tient à elle seule le rôle de GN, désigne aussi l'objet qu'elle référence par son nom.

Exemple : détruisez "A"

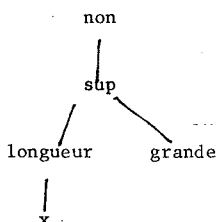


9 LES CONNECTEURS LOGIQUES

9.1. LA NEGATION

La seule locution négative utilisée est ne ... pas entourant un verbe. On peut aussi utiliser "pas" seul devant une description de propriété (un cube pas plus gros que ...). La négation sert donc à nier une propriété et est exprimé par non (p).

Exemple : x qui n'est pas long



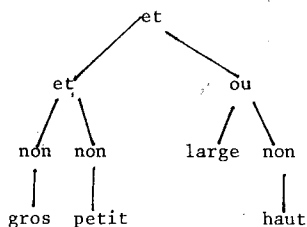
9.2. LES CONJONCTIONS DE COORDINATION

On utilise : et, ou, ni, mais, soit, à la fois (exemple : ni haut et large, ni long). Elles renferment deux concepts :

- la positivité : elles peuvent être négatives (cas de ni) ou positives (cas des autres conjonctions).
- le connecteur logique correspondant : les conjonctions et, ni, mais, à la fois correspondent au connecteur "et" ; ou, soit correspondent au connecteur "ou".

La représentation de ces conjonctions est celle des connecteurs binaires et (p,q) et ou (p,q), avec éventuellement une négation sur les prédicats.

Exemple : un cube à la fois ni gros, ni petit, et soit large, soit pas haut :



(en simplifiant la représentation des adjectifs).

10 LES NOMBRES

On les utilise sous deux formes : ordinale ou cardinale.

10.1. NOMBRES ORDINAUX

Ecrits sous la forme d'un nombre et du suffixe "ième", ils ne sont utilisés que dans le GN dont le noyau est un substantif d'objet. La représentation est $\text{ref}(x,n,p)$ où p désigne toutes les autres propriétés de x .

10.2. NOMBRES CARDINAUX

Leur sens varie selon la structure syntaxique dont ils font partie.

- ils peuvent faire partie d'une liste de nombres entre parenthèses qui désignent les coordonnées d'un point. Ils forment généralement le deuxième argument du prédicat centre (x, coord).

- ils peuvent être l'argument numérique d'un adjectif, substantif ou verbe de dimension ou d'emplacement. Le nombre est alors un des arguments du prédicat sup ou égal, ou d'un opérateur arithmétique (allonger x de 100 \rightarrow addition ; 10 fois plus court que A \rightarrow division).

- ils peuvent désigner le cardinal d'un ensemble et formant alors le deuxième argument des prédicats créer ou OD.

- enfin, quand un élément (verbe, préposition,...) demande un certain nombre d'arguments-objets, on utilise un nombre pour regrouper ceux-ci s'ils s'expriment d'une manière identique. (Exemple : entre deux cubes, intervertissez deux prismes). On divise alors les arguments regroupés pour obtenir la signification : entre deux cubes signifie : entre un cube et un cube.

11 LES SIGNES DE PONCTUATION

Quoiqu'ils ne soient pas vraiment des mots puisqu'ils n'existent pas phonologiquement, leur rôle est important dans l'articulation du discours écrit où ils pallient à l'impossibilité d'exprimer certaines nuances, d'intonation par exemple, qui existent dans le langage parlé et dont l'importance est réelle (par exemple, une phrase interrogative ne se prononce pas comme une phrase exclamative).

On utilise les signes de ponctuation suivants :

- le point : c'est un marqueur de fin de phrase. Quand il sépare deux phrases d'un même texte, il signifie la conjonction des deux prédicats représentant les phrases, comme les autres séparateurs de phrases (Créez un cube. Détruisez A ; Créez un cube et détruisez A).

On remarque cependant que le sens de ce connecteur diffère du connecteur et (p,q), qui est commutatif, dans la mesure où l'ordre des arguments doit être respecté puisqu'il décrit l'ordre des événements exprimés.

- la virgule : son usage est moins bien défini que celui des autres signes. On peut distinguer deux rôles différents : celui de "remplaçant" d'une conjonction exprimée plus loin dans le texte (Créez un cube, un bloc et un prisme : la virgule remplace un "et" ; Détruisez A, B ou C : la virgule remplace un "ou") et celui de séparateur de mots ou de structures syntaxiques : la virgule permet alors de rendre plus compréhensible le sens de la phrase et de supprimer les ambiguïtés que peuvent contenir certaines constructions. Ainsi : "Créez une pyramide sur le bloc, et un cube" n'a qu'une seule interprétation. On voit que dans ce rôle, la virgule n'a aucun rôle sémantique.

- le point d'exclamation : on l'interprète de la même façon que le point.

- le point d'interrogation : il marque que la phrase est une question. Son utilité est en fait très réduite puisqu'une question s'exprime d'une façon particulière (est-ce que le cube est gros ? ou : le cube est-il gros ?). Il serait cependant significatif si on acceptait des formulations comme : le cube est gros ? ou seul le ? indique qu'il s'agit d'une question.

DEUXIEME PARTIE

La première partie ayant permis d'établir d'une part le monde qu'on se propose de manipuler, les concepts qui le composent et le sous-ensemble du français qui permet de les exprimer, d'autre part la représentation qui sert à indiquer le sens de la phrase décomposée en ses divers éléments significatifs, il reste à indiquer de quelle manière s'opère le passage d'une phrase d'entrée (commande en français) à sa représentation en structure profonde.

On parlera d'abord de la méthode générale adoptée : différentes phases en lesquelles peut se diviser le processus global d'analyse et de calcul du sens d'une phrase et outils informatiques utilisés.

Les deux chapitres suivants traitent, le premier de l'analyse du mot (morphologie) et le second de l'analyse de la phrase et de sa représentation (syntaxe et sémantique).

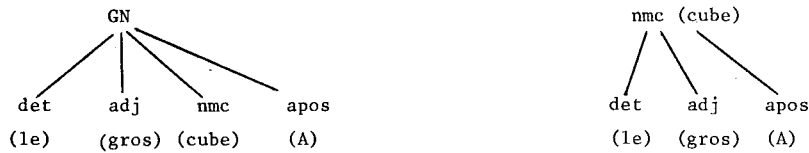
CHAPITRE V

ANALYSE DU TEXTE

1 - MORPHOLOGIE, SYNTAXE ET SEMANTIQUE

Dans tout système opérant un traitement de langue naturelle (en particulier une analyse), on a coutume de distinguer trois niveaux pour la représentation des informations linguistiques : le niveau morphologique qui donne la structure des mots, le niveau syntaxique qui donne la structure de la phrase en ses différents constituants et le niveau sémantique qui donne une représentation du sens de la phrase. Chacun de ces niveaux est obtenu au moyen d'une analyse spécifique : l'analyse morphologique a pour entrée un mot du texte d'entrée et se charge de remplacer celui-ci par toutes les informations qui le caractérisent (identification lexicale, variables grammaticales de différentes natures). L'analyse syntaxique utilise les syntagmes élémentaires ainsi obtenus pour donner, sous forme d'arbre syntaxique, la structure de surface de la phrase en la décomposant en constituants tels que GN, GV, etc. Deux représentations de la structure syntaxique sont possibles : la représentation sous forme de constituants ou la représentation sous forme de dépendance ou un élément de chaque constituant est choisi pour être le gouverneur des autres éléments : le nom pour le GN, le verbe pour le GV, ...

Ex : "le gros cube A" peut être représenté soit par :



Ces deux formes sont de toute façon équivalentes (on peut toujours passer de l'une à l'autre).

L'analyse sémantique enfin, a pour but de donner l'interprétation de la phrase de départ : elle transforme pour cela la structure syntaxique en une certaine "structure profonde" représentant le sens de la phrase. Cette structure sémantique peut être de forme très variée : on peut avoir une représentation linguistique. C'est le cas, par exemple du langage pivot développé par le GETA [1], qui se propose de donner sous une forme unique le sens des phrases écrites dans les langages les plus différents. C'est aussi le cas de Simons avec "ses réseaux sémantiques" où les concepts sont reliés entre eux par des relations sémantiques telles que "source", "but", "agent",... et de Schank [12] dont le mode de représentation est celui de la "dépendance conceptuelle" où un certain nombre de concepts de base sont liés par des relations profondes.

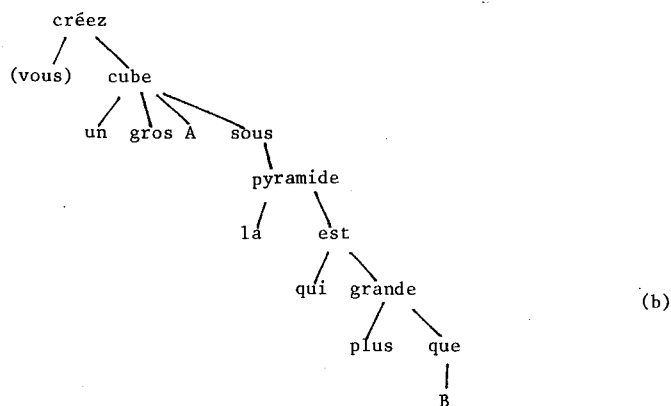
On peut aussi représenter le sens d'une phrase comme un théorème, donc un énoncé logique, des fonctions et des prédicats évaluables exprimant les éléments de sens élémentaires. C'est, nous l'avons vu, le cas du système de Winograd, celui du système de Marseille et aussi celui que nous décrivons ici : c'est donc le mode généralement adopté pour les systèmes de communication HM portant sur un sujet délimité, la nature et le rôle des primitives utilisées dépendant du sujet étudié dans chaque cas.

Par exemple, dans le monde qu'on s'est fixé, si on considère la phrase : "créez un gros cube A sous la pyramide qui est plus haute que B" on pourra donner, pour le niveau morphologique, la suite de syntagmes :

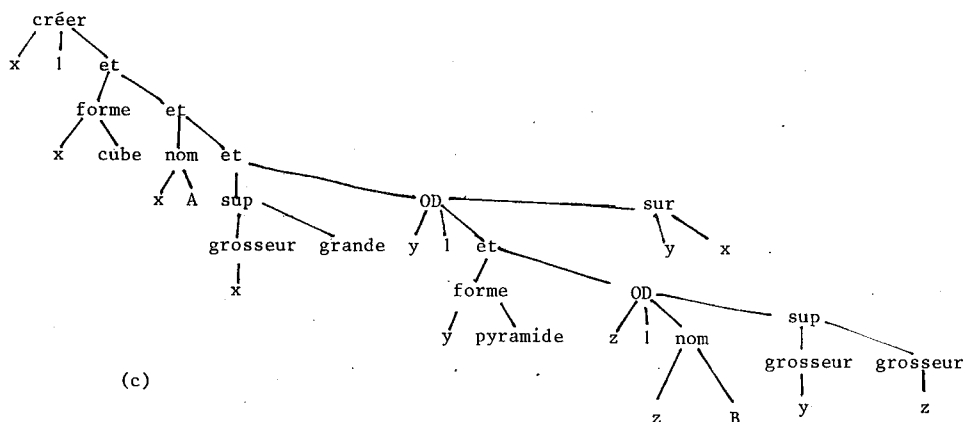
catégorie (verbe)	,	catégorie (article)	,	catégorie (adjectif)	,	...
unité lexicale (créer)		nombre (sin)		genre (mas)		
personne (2)				nombre (sin, plu)		

(a)

L'analyse syntaxique donnera la structure syntaxique de la phrase, qui, sous forme dépendentielle, pourra avoir la forme suivante :



L'analyse sémantique, enfin fournira une structure correspondant à celle que nous avons décrite au chapitre III.



En réalité, cette conception de l'analyse du texte en trois phases distinctes et s'enchaînant peut se modifier quelque peu pour les deux dernières étapes : alors que chez Winograd, on calcule effectivement une structure syntaxique correspondant à la structure de surface, avant de la transformer, au moyen de "spécialistes sémantiques", en une structure profonde prédictive, chez Simmons, au contraire, l'utilisation de réseaux de Woods permet de construire une structure profonde au cours même de l'analyse des différents constituants du texte.

De même, dans l'analyse du texte qu'on pratique, on ne cherche pas à construire l'arbre syntaxique de la phrase, tel qu'on l'a montré en (b) et on construit progressivement la structure profonde (c). On verra cependant (chapitre VII) que l'analyse du texte se fait en plusieurs phrases successives et qu'on obtient, avant la structure logique finale une représentation proche de celle représentée en (b). Mais cette structure est obtenue au cours du même traitement logique que la structure profonde finale et c'est déjà, en réalité une représentation du sens, des étiquettes indiquant le rôle sémantique des différents éléments.

On a donc un traitement en deux phases distinctes successives : analyse morphologique et analyse syntaxo-sémantique.

2 - LES OUTILS INFORMATIQUES UTILISES

On va maintenant définir les outils qui nous permettent d'effectuer l'analyse des phrases exprimées dans le français qu'on a défini :

2.1. MODE DE REPRESENTATION DU MODELE D'UNE LANGUE

Il existe deux approches pour l'écriture de systèmes permettant de traiter une langue naturelle. On peut utiliser un langage de programmation, comme LISP ou une des ses extensions, tel PROGRAMMAR, plus spécialement destinée à traiter des problèmes de cette nature, et écrire dans ce langage un programme qui décrit à la fois la structure de la langue étudiée et l'algorithme qu'on utilise pour cette étude (par exemple la simulation d'un automate à pile si on veut faire l'analyse syntaxique d'un langage hors-contexte). L'inconvénient d'une telle méthode, où algorithme et grammaire sont mélangés, réside en ce que, pour modifier le modèle de la langue, il faut modifier le programme et on risque vite de se retrouver dans une situation inextricable.

Dans l'autre approche, on sépare les deux éléments distincts en l'algorithme, qui décrit le processus de calcul employé, et le modèle, qui décrit le langage employé et ses particularités, que ce soit au niveau morphologique, syntaxique, ... La grammaire devient donc une donnée au lieu d'être un programme. Les systèmes qu'on utilise ont été conçus selon cette optique.

On pourrait utiliser un système unique pour les différentes étapes de l'analyse (comme dans les systèmes-Q [6] où même les articles du dictionnaire sont écrits sous forme de règles), mais il faudrait alors employer le système algorithmique le plus compliqué pour les phases de traitement les plus simples. On a donc deux systèmes qui réaliseront un algorithme de puissance très différente.

2.2. LES SYSTEMES DU GETA

Pour une description générale de ces systèmes, on peut consulter [15]. On a :

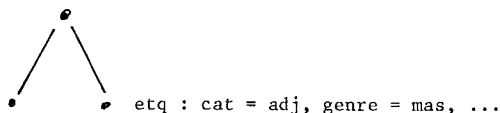
- pour l'analyse morphologique, le système ATEF (Analyse de Textes en Etat Fini : [3],[4]), dont la puissance est limitée à celle d'un transducteur d'état fini non déterministe. Il transforme la chaîne d'entrée en une arborescence.

- pour l'analyse syntaxique et sémantique, le système CETA (Contrôle Et Transduction d'Arborescence:[5], [4]), qui effectue des calculs sur des arborescences dans la limite de puissance des fonctions récursives primitives: on peut donc faire des calculs de haut niveau, sans avoir la puissance d'une machine de Turing, pour s'assurer que le processus ait une fin.

Ces systèmes ont été écrits essentiellement dans un but de traduction automatique, mais on peut les utiliser dans tous les traitements où l'on demande les mêmes puissances de calcul. Ils sont en particulier, bien adaptés à notre cas où la chaîne d'entrée doit être transformée en une arborescence représentant, sous une forme appropriée, le sens de la phrase : la phase d'analyse morphologique est faite par une grammaire écrite en ATEF, la phase d'analyse de la phrase par des grammaires écrites en CETA.

Avant de décrire les deux systèmes employés, pour montrer ensuite comment on s'en servira, il faut parler d'un concept qui leur est commun : celui d'arborescence étiquetée : c'est à la fois la forme en laquelle ATEF transforme la chaîne d'entrée, et l'objet de base que manipule CETA et la forme finale de sortie. Les éléments qu'on manipule sont des arborescences dont chaque noeud est doté d'une étiquette. Celle-ci est une ensemble de valeurs décrivant la nature et le rôle de chaque point de l'arborescence. Ces valeurs sont supportés par des variables.

On peut par exemple avoir :

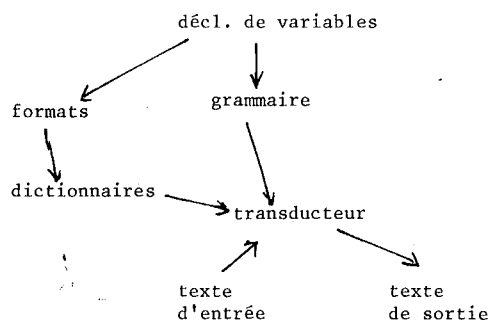


Les règles de grammaire écrites dans ATEF et CETA ont pour but de calculer et de faire évoluer ces étiquettes.

2.3. LE SYSTEME ATEF

C'est ce système, qui simule un transducteur d'état fini, qui sert à l'analyse morphologique. Afin de comprendre la manière dont fonctionne ce système, prenons un exemple, le mot "créés". Ce mot n'est pas référencé en tant que tel dans les dictionnaires. Il est formé de deux parties (appelées segments) : "cré", base du verbe créer, et "és" désinence participiale au masculin pluriel d'un verbe du premier groupe ; ces deux segments appartiennent à des dictionnaires. Le système fait deux sortes d'opérations : le découpage de la forme d'entrée (cette forme correspondant en général à un mot, sera tout de suite de caractères entourée par des blancs) en ses différents segments, ceux-ci étant listés dans des dictionnaires et une forme pouvant être composée d'un nombre quelconque de segments ; puis le contrôle de la cohérence des segments, pour refuser tous les découpages qui n'ont pas de sens (tels que base de verbe suivie de désinence de nom), accompagné du calcul de l'étiquette composée de toutes les informations décrivant la forme. Par exemple, l'analyse de la désinence "és" donne le temps de conjugaison (participe passé), le genre et le nombre (masculin, pluriel), l'analyse de la base "cré" donne l'unité lexicale (créer), l'auxiliaire que peut avoir ce verbe (avoir), le système de transitivité, ... La combinaison de ces différents éléments permet d'obtenir pas à pas l'étiquette complète caractérisant le mot.

Pour avoir une idée plus précise du fonctionnement de l'automate, définissons la description des différents composants du modèle morphologique de la langue. L'organisation générale du système est :



2.3.1. Variables

Les variables servent à porter les renseignements élémentaires de toutes natures qui composent les étiquettes des noeuds des arborescences. Ces variables servent soit au traitement morphologique lui-même (tests de cohérence entre les segments), soit au traitement ultérieur par CETA. Les variables sont divisées d'une part en "variables syntaxiques" et "variables morphologiques" et d'autre part en variables exclusives et non-exclusives selon qu'elles peuvent prendre une seule valeur ou un ensemble de valeurs. Pour chaque variable, on déclare son nom et la liste des valeurs qu'elle peut prendre.

Exemple : -exc-
cat=(nmc, adj, verbe, ...)
-nex-
genre = (mas, fem)

2.3.2. Formats

Un format permet d'indiquer un ensemble de valeurs de variables. Ces formats servent à définir dans les dictionnaires les différents segments : au lieu d'avoir dans les dictionnaires la liste des valeurs à associer à chaque entrée, on a deux noms de formats. Les formats morphologiques, formés de variables morphologiques servent à l'activation des règles de grammaire. Les formats syntaxiques, composés essentiellement des variables grammaticales permettent de compléter l'étiquette remplaçant la forme d'entrée.

2.3.3. Dictionnaire

Les différents segments qui forment le vocabulaire utilisable sont rangés dans un certain nombre de dictionnaires. Dans les dictionnaires où les entrées ne correspondent pas à une unité lexicale (affixes, désinences), on a pour chaque article, la liste des caractères formant ce segment et les noms des formats morphologique et syntaxique. Pour les dictionnaires contenant les racines des mots (bases), on a aussi une unité lexicale.

Exemple : dic 1	dic 2
és == FM1 (FS1)	cré == FM2 (FS2, créer)

où les formats sont formés d'une liste de valeurs de variables décrivant ces segments. D'une manière générale deux segments ayant le même comportement, c'est-à-dire décrits par les mêmes valeurs, sont référencés par un même format.

2.3.4. Grammaire

Alors que les dictionnaires définissent le vocabulaire d'entrée, la grammaire décrit la fonction de transition du transducteur. Elle a donc pour but de calculer l'étiquette des mots, le vocabulaire de sortie étant composé de l'ensemble des valeurs de variables. A chaque forme, on associe un ensemble de registres pour contenir ces valeurs de variables, en particulier un registre pour l'état courant (c) décrivant l'étiquette après l'analyse de chaque segment et un registre pour l'état argument (A) décrivant les valeurs du segment en cours d'analyse.

La grammaire est composée d'un certain nombre de règles, chacune d'elle étant formée de deux parties. La partie gauche est une liste de formats morphologiques : la règle ne s'appliquera que si le FM du segment analysé appartient à cette liste. La partie droite comprend une partie condition permettant de restreindre l'usage d'une règle aux seuls cas possibles : la condition porte en effet sur les valeurs des registres courant ou argument ce qui permet de contrôler la cohérence de l'argument avec la partie déjà analysée de la forme. La partie droite comprend aussi un ensemble d'opérations permettant de faire évoluer l'étiquette de l'état courant.

On verra des exemples d'analyse au chapitre suivant.

2.3.5. Fonctions spéciales

Dans ATEF, le modèle décrivant la morphologie d'une langue est une donnée et non un programme. L'algorithme général n'est donc pas accessible à l'utilisateur. Pour éviter les inconvénients qu'une telle rigidité impose, celui-ci dispose cependant d'un certain nombre de fonctions spéciales lui permettant d'intervenir sur l'algorithme général. On peut par exemple modifier la forme courante d'entrée (par transformation de chaîne) ou accélérer l'algorithme de segmentation général : des fonctions comme -final- ou -arrêt- permettent d'arrêter le processus de découpage quand on pense que les solutions pour une certaine forme ont toutes été obtenues.

2.4. LE SYSTEME CETA

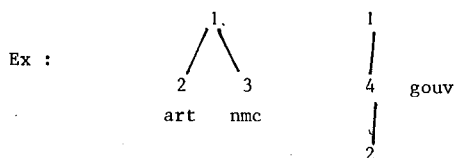
Ce système est un transducteur d'arborescence. Il est conçu pour permettre la définition et la manipulation de grammaires transformationnelles (au sens général de transformations d'arborescence et non au sens de Chomsky). La puissance de ce modèle permet à l'utilisateur de définir la stratégie qu'il désire pour l'écriture d'une grammaire, l'algorithme pouvant fonctionner selon des modes différents. Du point de vue de la définition d'un modèle, on peut distinguer trois parties.

2.4.1. Les variables

Ces variables ont, comme dans ATEF, pour but de composer les étiquettes des noeuds des arborescences. On commencera donc par indiquer les variables qu'on utilise et les valeurs qu'elles peuvent prendre. Certaines de ces variables sont celles définies et utilisées lors de l'analyse morphologique en ATEF. Les variables qu'on n'indique pas sont alors supprimées des étiquettes des noeuds de l'arborescence de sortie, ce qui permet d'éliminer les variables purement morphologiques n'ayant servi qu'au contrôle de la cohérence des parties d'une forme. On définit aussi d'autres variables qui apparaîtront au cours des calculs suivant les besoins qu'on en aura.

2.4.2. Les règles

Les règles de transformation constituent l'élément de base de la grammaire. Une règle est formée de deux parties : la partie gauche indique quelle sous-arborescence de l'arborescence d'entrée voit s'appliquer la transformation : on indique pour cela la structure de la sous-arborescence et des conditions portant sur les étiquettes des différents points concernés, pour restreindre l'application des règles aux seuls cas nécessaires ; la partie droite indique la manière dont la transformation est faite : on a successivement la nouvelle structure formelle, la fonction de transfert permettant de définir la projection des descendants des points du schéma (indiqué en partie gauche) sur les points de l'arborescence résultante et les transformations sur l'étiquette des points. Pour chaque point de l'arborescence résultante, on précise la nouvelle valeur de l'étiquette en indiquant la provenance de l'étiquette, et les modifications qu'on veut lui faire subir.



On écrira : $R1 : 1 (2, 3) / 2 : \text{cat} - e - \text{art} ; 3 : \text{cat} - e - \text{nmc} / \text{genre} (3) - e - \text{genre} (2) = = 1 (4(2)) / 1 \leftarrow 1 ; 2 \leftarrow 2 ; 4 \leftarrow 3 / 1 : 1 ; 2:2 ; 4:3 ; \text{fonction} := \text{gouverneur} .$

On distingue donc, dans l'écriture des règles entre la transformation de la structure et la transformation des étiquettes. Toutefois quand un sommet a le même nom en parties gauche et droite, la fonction de transfert et la fonction d'étiquetage sont implicites si elles ne sont pas indiquées. La partie droite de la règle peut s'écrire plus simplement : $1 (3(2)) // 3:3, \text{fonction} = \text{gouverneur} .$

2.4.3. La grammaire

La grammaire a pour but d'indiquer la manière dont les règles de transformation sont activées. C'est en effet l'utilisateur qui indique l'ordre dans lequel les règles sont appliquées. Pour l'écriture de la grammaire d'une certaine application, on peut distinguer deux niveaux : une grammaire élémentaire est formée d'un ensemble de règles de transformations, tandis que la grammaire générale est composée de l'enchaînement des grammaires élémentaires.

2.4.3.1. Grammaire élémentaire

Une grammaire élémentaire est formée d'un ensemble de règles (ex : $G1 : R1, R2, \dots Rn.$) On peut l'appliquer de deux façons différentes. Si on indique un paramètre d'unitarité ($G1 (U)$), on applique les règles de la grammaire une seule fois sur l'arborescence d'entrée (en général plusieurs règles ou une même règle s'appliquent simultanément sur des sous-arborescences qui doivent être indépendantes). Quand plusieurs règles peuvent concurremment s'appliquer, c'est leur ordre d'écriture qui prévaut.

Si on indique un paramètre d'exhaustivité ($G1 (E)$), après chaque application des règles de la grammaire, on essaie une nouvelle application. Pour que le processus ait toujours une fin, on impose une condition : un point racine d'une transformation ne pourra plus être racine de la même transformation, ainsi que tous ses descendants. On peut aussi écrire les règles de grammaire sous forme récursive, ce qui permet d'indiquer quelles règles doivent être appliquées après une certaine règle.

2.4.3.2. Grammaire générale

Le résultat d'une grammaire élémentaire peut servir d'entrée à une autre grammaire. Pour cela, on indique pour chaque grammaire après la liste de ses règles, des conditions que l'arborescence de sortie doit réaliser et les grammaires qui s'enchaîneront si la condition est satisfaite.

La grammaire générale est composée en fait d'une arborescence de sous-grammaires. Cette organisation permet de diviser le traitement en plusieurs parties indépendantes, chaque partie étant réalisée par un sous-grammaire différente. On peut par exemple écrire des sous-grammaires chargées d'analyser des structures syntaxiques particulières et qui ne sont pas appelées dans le cas général.

On voit donc que cette organisation du traitement permet à l'utilisateur de choisir sa propre stratégie d'analyse en organisant sa grammaire de la manière qu'il juge souhaitable.

CHAPITRE VI

ANALYSE MORPHOLOGIQUE

Nous allons définir les différents composants du système d'analyse morphologique.

1 - LES DICTIONNAIRES

On définit d'abord les dictionnaires qui comportent les différents segments formant les mots de notre vocabulaire. On utilise trois dictionnaires : un dictionnaire de désinences et de suffixes, un dictionnaire de bases et un dictionnaire de tournures. On ne trouve pas dans ces dictionnaires un vocabulaire très important : en effet dans le monde étudié, un nombre restreint de mots est suffisant pour décrire des objets ou des actions de la façon que l'on a vue. Pratiquement le nombre de bases de mots est d'environ 200.

1.1. DICTIONNAIRES DES DESINENCES ET AFFIXES

Ces segments servent à ajouter des informations de nature variée à la base d'un mot. Ils ne possèdent donc pas d'unité lexicale.

Les désinences sont des segments servant à infléchir les mots-racines. Seules les désinences usuelles seront utilisées :

- désinences de verbes à la 3e personne du singulier et du pluriel et la deuxième personne du pluriel (pour s'adresser au système), les seuls temps conjugués étant le présent de l'indicatif et le présent de l'impératif. On a aussi les désinences de l'infinitif et des participes passé et présent : mettez, créer, passant, se trouve, ...

- désinences de nom. Les seules désinences sont celles du pluriel (ex. : cube s). Les désinences indiquant à la fois le genre et le nombre ne sont pas utilisées ici.

- désinences d'adjectif. On a cinq sortes de désinences : mas - sin (ex : égal), mas - plu (ex : haut s) , fem - sin (ex : haut e), fem - plu (ex : derni ères qu'on écrira dernielres, le l remplaçant l'accent) et plu (ex : large s).

On remarque que toutes ces désinences servent à indiquer le genre, le nombre et pour les verbes le mode, le temps et la personne.

On utilise aussi certains affixes, mais en nombre très limité : on n'utilise aucun préfixe et parmi les suffixes (ment, able, ation, ité, ...) on n'en considère que deux, les seuls utiles dans le monde manipulé:

- le suffixe ième, exprimant la dérivation nombre cardinal → nombre ordinal (ex : 25ième)
- les suffixes eur et esse, exprimant la dérivation adjectif → nom, qui permettent d'obtenir des noms de tailles à partir des adjectifs : grandeur, hauteur, étroiteesse, ...

1.2. DICTIONNAIRE DES BASES

C'est le dictionnaire proprement dit, où se trouve tout le volume du vocabulaire dont on a besoin. Les mots sont écrits sous forme d'une racine, c'est-à-dire la partie qui reste fixe par rapport aux désinences ou affixes, qu'on peut ajouter : le verbe passer s'écrit pass, l'adjectif égal s'écrit ég, ...

On y trouve les mots indiqués au chapitre IV. Le nombre des verbes, noms et adjectifs qui forment l'essentiel d'un vocabulaire est très réduit alors que les autres mots sont proportionnellement bien plus nombreux.

1.3. DICTIONNAIRE DES TOURNURES

Ce dictionnaire particulier renferme toutes les suites de formes que l'on peut considérer comme formant un tout. En particulier les locutions prépositionnelles comme "à droite de " ou "en avant du" dont le comportement syntaxique et sémantique peut être comparé à des prépositions comme "devant" ou "sur". Le dictionnaire des tournures renferme toutes les telles suites de formes. Dans notre cas outre certaines tournures comme "à la fois", on a essentiellement des locutions prépositionnelles, soit sous forme prépositionnelles (ex : à droite de), soit sous forme

adjectivale (ex : en avant). Comme les tests de tournures faits dans la grammaire comparent les formes d'entrée avec les suites de formes référencées dans ce dictionnaire, il faut non seulement indiquer les tournures "de base" telles à droite de, mais aussi ces tournures où "de" est remplacé par "du", "des", ... (à droite du, en avant des, ...)

2 - LES VARIABLES

Pour chacun des dictionnaires indiqués on a pour chaque entrée, une suite de caractères (ou une suite de telles suites pour les tournures) indiquant les segments, comme : détrui, le, s, à droite, une unité lexicale pour les dictionnaires de base et de tournure, et deux formats :

ex : crel ==VBMOI (VBA, créer), VBMOI et VBA étant des formats morphologique et syntaxique définissant le segment (disant par exemple qu'il s'agit d'un verbe du premier groupe, que sa catégorie syntaxique est "verbe", qu'il a pour auxiliaire de conjugaison le verbe avoir).

Chacun de ces formats est écrit comme une suite de valeurs de variables ou d'un ensemble de valeurs si les variables sont non-exclusives : par exemple pour l'adjectif large le genre vaut l'ensemble (mas, fem).

2.1. VARIABLES EXCLUSIVES

Le segment : Il indique l'état dans lequel l'automate se trouve en disant la nature du dernier segment analysé. Il permet donc de contrôler la cohérence des découpages effectués.

seg := (DP, DV, DN, B, SUFADJ, SUFNB, CHIF, INVA, GUIL). Les valeurs possibles sont Désinence Participale (é, i, ...), Désinence Verbale (e, ez, ...), Désinence Nominale (pour les désinences de noms et d'adjectifs : s, aux, ...), Base (racine d'un mot admettant des flexions : cube, pass, ...), SUFFixe d'ADJectif (eur, esse), SUFFixe de NomBre (ième), CHIFFre (élément des nombres), INVARIABLE (pour les mots invariables : prépositions, articles, ...) et GUILleMET pour les noms entre guillemets ("B").

Cette variable a un rôle essentiel dans la partie condition des règles. Par exemple, pour les règles correspondant aux bases de verbes, il faut que la valeur de SEG soit DV ou DP, pour les mots invariables il faut que cette variable n'ait pas encore été affectée. Chaque fois qu'un nouveau segment est analysé, la variable SEG est recalculée de façon à suivre l'évolution de l'analyse.

La catégorie syntaxique : Cette variable sera très importante pour l'analyse de la phrase.
cat := (VB, NMC, MINC, PRP, ADJ, PRE, PAPA, AUX, COR, SUB, APOS, NB, NBORDI, VI, PCT, REL, DET). Les catégories possibles sont : VerBe, NoM Commun, Mot INConnu (pour les mots n'ayant pas de découpage correct), PRonom Personnel, ADJectif qualificatif, PRÉposition, PARTICIPE PASSÉ, AUXiliaire (ces deux dernières sont distinctes de VB par raison de commodité), conjonction de COORDINATION, de SUBordination, APOsition (pour les noms propres), NomBre (cardinal), NomBre ORDInal, Valeur Immédiate (coordonnées), PonCTuation, pronom RELatif et DETermineurs (articles, adjectifs démonstratifs, ...).

La sous-catégorie : Cette variable distingue les différents détermineurs
Soucat := (ARTI, ARTD, AJD, AJP, AJI). Les valeurs possibles sont : ARTicle Indéfini, ARTicle Défini, AdJectif Démons-tratif, AdJectif Possessif et AdJectif Indéfini.

La sémantique : Cette variable indique certains "traits sémantiques" de taille ou de lieu.
Sem := (TAILLE, PLACE, POS). Cette variable vaut TAILLE pour les mots, tels long, largeur, rétrécir, ... qui expriment les dimensions, PLACE pour les mots, tels à droite de, descendre, en haut, qui expriment les emplacements et POSition pour les mots, tels sur, contre, entre, ... qui expriment des relations particulières de position entre les objets.

La personne possessive : Cette variable est utilisée pour les adjectifs possessifs.
pPERS := (1S, 2S, 3S, 1P, 2P, 3P). PERS indique la personne et le temps auquel l'adjectif possessif fait référence : lère personne du singulier (mon, ma, mes), lère personne du pluriel (notre, nos), etc...

La négation : Cette variable est utilisée dans la définition des termes de négation (ne... pas, ni).
Nega := (neg).

L'ull : Cette variable donne les valeurs standard pour les dimensions ou les coordonnées.
Ull := (grande, petite, moyenne). Par exemple, dans la définition de "long", on a ull = grande, tandis que pour "reculer", on a ull = petite.

2.2. VARIABLES NON-EXCLUSIVES

La conjugaison verbale : Elle indique à quel groupe de conjugaison appartiennent les désinences et les bases verbales. Cette variable, tout comme SEG, sert donc à contrôler la cohérence de deux segments. Par exemple, la désinence issant (2e groupe) ne pourra être précédé que par des bases de verbes du 2ème groupe (élargir, intervertir, ...).

Les dictionnaires : Cette variable prédéclarée indique quels dictionnaires on pourra utiliser. En particulier, quand une forme d'entrée comporte plusieurs unités lexicales (ex : détruisez-le), on réinitialisera le traitement en indiquant quels dictionnaires on utilisera.

Le genre : GNR := (MAS, FEM). Il indique le genre, MASculin ou FEMin des noms, adjectifs, pronoms, ...

Le nombre :
NBR := (SIN, PLU). Il indique le nombre, SINGulier ou PLUriel des noms, verbes, pronoms, ...

Le mode et le temps de conjugaison : Ces deux éléments sont regroupés en une seule variable.
MOTE := (IDPR, IMPR, PAPR, INF). Les valeurs qu'on considère sont l'Indicatif Présent, l'Impératif Présent, le Participe Présent et l'Infinitif.

La personne : PRS := (1, 2, 3). Elle indique la personne pour les temps conjugués et les pronoms.

Les arguments verbaux : Les variables ARG2 et ARG3 donnent la nature des arguments que peuvent avoir les verbes, c'est-à-dire leurs actants. Ces arguments peuvent être par exemple un nom, un nom précédé de la préposition de ou de la préposition à, un infinitif, ... Par exemple, le verbe créer est caractérisé par arg2 = Nom, appartenir par arg2 = ANom. Ces variables servent dans l'analyse des groupes verbaux pour la recherche des arguments des verbes.

L'auxiliaire : AUX := (A, E) indique quel auxiliaire (Avoir ou Etre) peut posséder un verbe. Cette variable n'aura d'utilité que pour les temps composés.

Le rôle des pronoms : Il indique le rôle syntaxique joué par les pronoms, seuls éléments qui se déclinent en français.
Rôle := (SUJ, OBJ, ATT, REF, PPT, DONT). Les valeurs possibles sont SUJ et (je, qui, ...), OBJ et (le, que, ...), ATT et (lui, nous, ...), REF et (me, se, ...), noyau d'un groupe PréPositionnel (elle, qui, ...), et enfin DONT pour le pronom relatif "dont" au rôle très particulier. On remarque que les différents pronoms ont un nombre de rôles très variable : je est seulement suj, alors que nous est à la fois suj, obj, att, ref et ppt.

L'argument d'un adverbe : Arg = (AJ, VE, AJC). Arg indique quel mot modifie un adverbe. Ce peut être un Adjectif (très), un Verbe (le cube précédemment créé) ou un Adjectif au Comparatif (cube plus gros que...).

L'indicateur d'objet : OBE := (0, 1). Cette variable vaut 1 pour les substantifs d'objets (cube, prisme, ...) et à 0 pour les autres. Il s'agit donc d'un "trait sémantique".

La valeur de la conjonction : CJ := (ET, OU) indique à quels connecteurs logiques les conjonctions de coordination correspondent.

Le nombre des conjonctions : NC := (1, 2) indique combien de fois une conjonction doit se trouver dans un groupe de deux éléments conjugués (nc = 2 pour ni, nc = 1 pour mais, ...).

Le signe : Le signe sert aux prédicats de comparaison de valeurs numériques. SIGNE = (SUP, INF, EGAL). Cette variable figure dans la description de tous les mots exprimant une valeur ou une comparaison de valeurs (aussi \rightarrow égal ; plus, avancer, à droite \rightarrow sup ; moins, descendre \rightarrow inf).

L'opérateur : OP := (ADD, SOUSTR, PROD, QUOT). Cette variable permet de définir les opérations arithmétiques élémentaires sur les valeurs numériques. OP figurera dans la description des mots pouvant faire référence à de telles opérations. Par exemple, pour "plus", on aura op = add - u - prod, car plus peut correspondre à une addition (x plus long que y de 10 \Rightarrow longueur (x) = longueur (y) + 10) ou à une multiplication (x 10 fois plus long que y \Rightarrow longueur (x) = longueur (y) * 10) ; pour le verbe diminuer au contraire, on a op = soustr.

2.3. L'UNITE LEXICALE

Cette variable particulière qu'on peut manipuler comme les autres variables n'est pas déclarée de la même façon. C'est dans les dictionnaires de base et de tournure que ses valeurs sont indiquées (une valeur pour chaque entrée). Ces valeurs sont libres et nous les avons choisies de manière à faciliter le traitement ultérieur. Par exemple les synonymes ont la même unité lexicale (ul = destr pour détruire, effacer, ...).

Pour les mots servant à indiquer les emplacements par rapport aux axes x, y, z, on utilise les UL abscisse, ordonnée et cote (à droite, à gauche \rightarrow ul = abscisse ; en haut, descendre \rightarrow ul = cote, ...) ; de même pour "allonger" et "raccourcir" on a ul = longueur.

2.4. REMARQUES SUR LES VARIABLES

On peut d'abord noter que, comme pour les traits de la grammaire systémique, ces variables sont hiérarchisées : par ex, les variables AUX et PRS ne seront affectées que si CAT = VB ou CAT = AUX, OBB seulement si CAT = NMC.

On peut d'autre part classer ces variables par rapport au rôle qu'elles ont dans les différentes phases du traitement :

- variables purement morphologiques, servant à contrôler le découpage des formes lors de l'analyse morphologique. Elles disparaîtront dans la suite du traitement. Ce sont les variables SEG, CV et DICT.

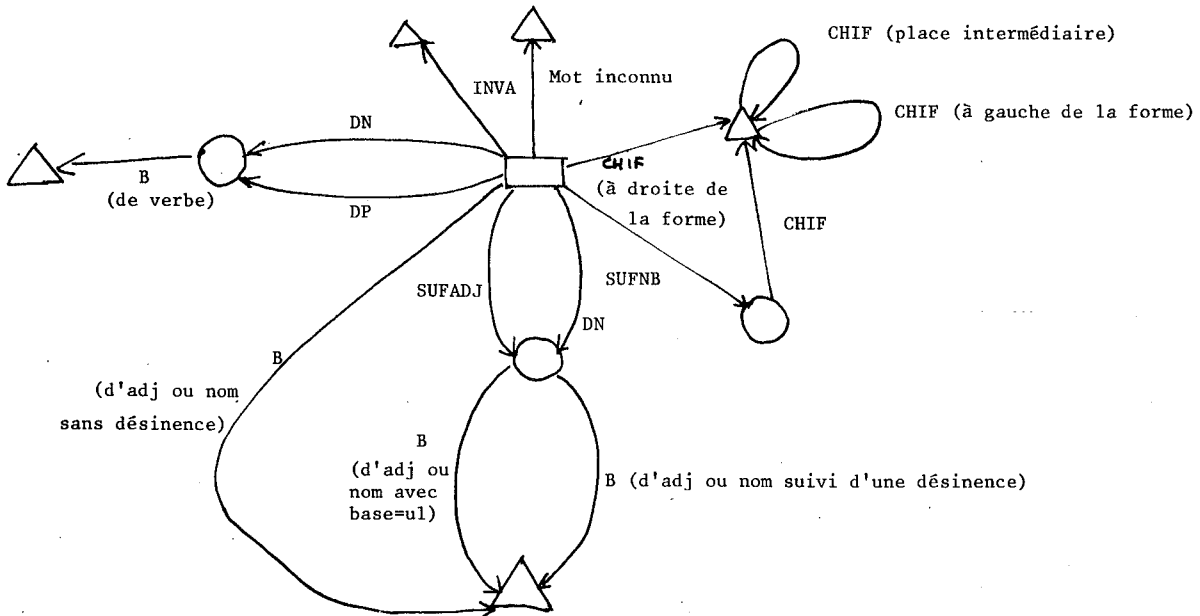
- variables syntaxiques. Ces variables comme CAT, GNR, NBR serviront à l'analyse syntaxique de la phrase.

- variables syntaxo-sémantique, équivalent aux "traits sémantiques". Elles sont adaptées au sujet d'étude et servent à définir le sens qu'ont les mots de notre vocabulaire dans le monde choisi. C'est le cas des variables SEM, OBB, OP. Elles seront traitées de la même façon que les autres variables syntaxiques dans l'analyse de la phrase mais permettront des transformations mettant en valeur le sens des structures analysées.

3 - LA GRAMMAIRE

On décrit par la grammaire le fonctionnement du transducteur. Chaque règle s'écrit : liste de formats morpho == liste d'actions / condition / Transformation de la chaîne d'entrée. Pour un segment donné de la forme d'entrée, la règle s'applique si le format morphologique indiqué dans le dictionnaire pour le segment appartient à la liste de ces formats et si la condition est réalisée. Les actions indiquées ont pour but essentiel de calculer l'étiquette courante à partir de l'argument et de l'ancien état courant.

Indiquons la forme générale du réseau d'état fini décrivant le système morphologique choisi. En simplifiant, en ne tenant pas compte en particulier des tournures, ni des formes contenant plusieurs syntagmes différents (par exemple en cas d'élision d'un article), on a le réseau suivant étant entendu que l'analyse des formes se fera à partir de la droite :



(Sur chaque arc, on indique la valeur de la variable SEG qui est calculé). Prenons quelques exemples illustrant le fonctionnement du transducteur :

Exemple 1 hautes

Dans les dictionnaires on a :

es == D1 (GN4)

haut == F4 (SA5, hauteur)

les formats étant écrits : D1 : seg - e - dn

F4 : seg - e - b

GN4 : gnr - e - fem, nbr - e - plu

SA5 : cat - e - adj, gnr - e - mas, nbr - e - sin, sem - e - taille ,

ull - e - grande, signe' - e - sup.

L'algorithme de découpage de la forme fournit le segment es. Son FM, D1 appelle les règles dont la partie gauche renferme D1, en l'occurrence la règle RD1 :

RD1 : D1 - D2 - ORD - SUFADJ == var (c) := var (A) / seg (c) - e - seg0.

Pour effectuer les opérations, il faut vérifier la condition: pour cela, il suffit que SEG n'ait pas encore été affecté (ce qui revient à dire qu'une désinence doit se trouver à droite de la forme). La condition étant vérifiée on applique les actions qui consistent à prendre les valeurs de l'argument.

Le masque de l'état courant devient donc : seg (DN), nbr (PLU), gnr(FEM).

Ce qui reste de la forme appartient aussi à un dictionnaire : le segment haut, par l'intermédiaire de son FM peut appeler deux règles différentes :

R1 : BAN - F3 - F4 == ... / seg (c) - e - seg0

R3 : F2 - F3 - F4 == vars (c) := vars (A), dict (2), - sol -, - final -, seg (c) := seg (A) / seg (c) - e - DN.

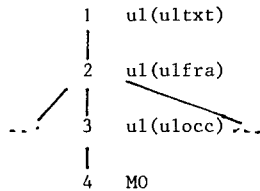
La règle R1 n'est pas appliquée, la condition n'étant pas remplie (elle aurait été appliquée par exemple pour la forme haut).

La règle R3, elle peut s'appliquer, SEG venant d'être affecté à la valeur DN. Les actions effectuées sont les suivantes ; on prend pour l'état courant les variables syntaxiques de l'argument : ces variables sont cat, sem, ull et signe. La variable SEG est aussi affectée pour refléter le nouvel état. La fonction spéciale-final- indique que la solution obtenue par ce découpage est la seule solution et elle arrête tous les autres découpages. La fonction spéciale -sol- fournit le masque de l'état courant comme solution et dict := 2 réinitialise le dictionnaire. Ces deux actions ne servent que quand la forme comprend plusieurs syntagmes.

On obtiendra donc en fin de compte la solution :

MO : seg(B), gnr(FEM), nbr(PLU), cat(ADJ), sem(TAILLE), ull(GRANDE), signe(SUP), ul(HAUTEUR)

La forme de l'arborescence générale est :



Le sommet étiqueté ul = ultxt est la racine générale. Le sommet ul = ulfra domine les différents éléments de chaque phrase composant le texte. Le sommet ul = uocc domine, pour chaque forme les différentes valeurs possibles, correspondant aux différents découpages possibles de cette forme. Dans le cas de "hautes", il n'y a qu'une solution.

Exemple 2 : Détruisez-le. (une seule forme composée de trois mots)

Dans les dictionnaires on trouve les éléments suivants :

```

ez == FMV03      (FSV06)
détruis == VBM05 (VBA, destr).
le == FT2       (F1, le ).
le == FT3       (FG4, le ).
. == INV7       (FA1, . ).
  
```

```

avec : FMV03 : seg - e - dv, cv - e - idpr1 - u - impr1 - u - idpr3 - u - idpr4 - u - impr3.
      FSV06 : mote - e - idpr - u - impr, prs - e - 2, nbr - e - plu.
      VBM05 : seg - e - b, cv - e - idpr4 - u - idil - u - impr3 - u - ifpr3 - u - papr1 .
      VBA   : cat - e - vb, aux - e - a, arg2 - e - n.
      FT2   : seg - e - b.
      F1    : cat - e - det, soucat - e - artd, gnr - e - mas, nbr - e - sin.
      FT3   : seg - e - b.
      FG4   : cat - e - prp, gnr - e - mas, nbr - e - sin, prs - e - 3, role - e - obj.
      INV7  : .
      FA1   : cat - e - pct, cj - e - et, nc - e - l.
  
```

L'algorithme de découpage commençant par la droite de la forme trouve d'abord le segment "." dont le FM appelle LIAME :

LIAME : INV7 == var(c) := var(A), -sol -, dict(c) := 1 - u - 2 /sehaîne(A,0,1) - ne - ' - et - seg(c) - e - seg0

La condition indiquée permet de vérifier que le segment est précédé par d'autres caractères (ponctuation collée à la forme) et qu'aucun autre segment n'a encore été analysé. Elle est vérifiée et les actions fournissent une première solution :

M1 : cat (pct), cj (et), nc(1), ul (.), dict (1-U-2)

Les deux dictionnaires sont réinitialisés et le traitement se poursuit. Le découpage fournit ensuite le segment "le" qui figure deux fois dans le dictionnaire (une fois comme article et l'autre comme pronom).

Si on choisit le premier, on se heurte, dans la règle appelée à la condition schaine (A,0,1)-e-'', spécifiant que le reste de la forme doit être vide, ce qui n'est pas ici le cas. On essaie alors le deuxième "le". Son FM appelle d'abord cette même règle qui ne peut pas s'appliquer. Mais une autre règle contient FT3 en partie gauche :
 XYZ : FT3 == var (c) := var (A), seg (c) := seg 0, dict (c) := 1 - U - 2, - sol - /
 seg (c) - e - seg0 - et - schaine (A,0,1) - e - '-' / tchaine (0,'-',").

La condition est réalisée : un tiret existe juste avant l'argument et la variable seg n'a pas été affectée. On réalise alors une transformation de chaîne qui supprime le tiret du reste de la forme. Les actions affecteront les variables portées par les formats de l'argument, à l'exception de seg restant non affecté, puisqu'un verbe complet reste à analyser. On réinitialise encore les dictionnaires et on crée une nouvelle solution :

M2 : cat (prp), gnr (mas), nbr (sin), prs (3), role (obj), dict (1-U-2), ul (le)

Le segment suivant est "ez". Son FM appelle la règle des désinences verbales :
 VRB : FMV03 - FMV01 - ... == var(c) := var(A) / seg(c) - e - seg0.

La condition est réalisée et les actions créent un masque courant avec les valeurs portées par FMV03 et FSV06.

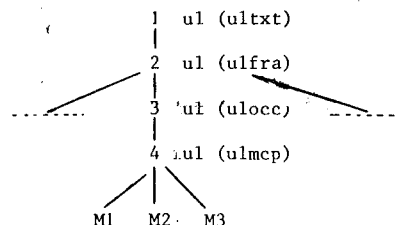
Enfin, le dernier segment découpé dans la forme est "détruis". Son FM appelle les règles traitant des bases verbales :

BV1 ; VBM05 - VBM01 - ... == seg(c) := b, vares(c) := vares(A), varns(c) := varns(c) - U - varns(A), dict(c) := 2,
 - sol -, final - / seg(c) - e - dv - et - cv(c) - I - cv (A) - ne - cv0.

On constate que la condition est encore réalisée, les actions qu'on effectue alors nous fournissent une dernière solution.

M3 : seg(b), dict (2), cat (vb), aux (a), arg2(n), prs(2), nbr(plu), mote(idpr - U - impr),
 cv (idpr1 - U - impr1 - U - ...), ul (destr)

Dans l'arborescence de sortie, les trois solutions que renferme la forme d'entrée sont dominées par un sommet étiqueté ul (ulmcp) :



CHAPITRE VII

ANALYSE DE LA PHRASE

Le but du système d'analyse de la phrase est de transformer l'arborescence représentant le résultat de l'analyse morphologique du texte d'entrée en une structure représentant le sens du texte. Cette structure, arborescente, est bien sûr la structure profonde décrite dans la première partie. Pour la description du modèle choisi, on utilisera le formalisme des grammaires CETA.

1 - LA GRAMMAIRE GENERALE

1.1. FORME CIBLE ET STRATEGIE D'ANALYSE

Nous avons vu (chapitre III) la forme cible en laquelle la structure d'entrée, fournie par ATEF doit être transformée. Le traitement général opérant cette transformation ne sera pas divisé en une phrase de construction de la structure syntaxique de la phrase et une phrase d'analyse sémantique des différents constituants.

On cherchera au contraire à construire pas à pas la structure profonde en transformant par exemple tous les constituants de même sens en une structure cible identique. Pour cela les différentes sous-grammaires composant la grammaire générale participeront toutes au but final et seront orientées en conséquence.

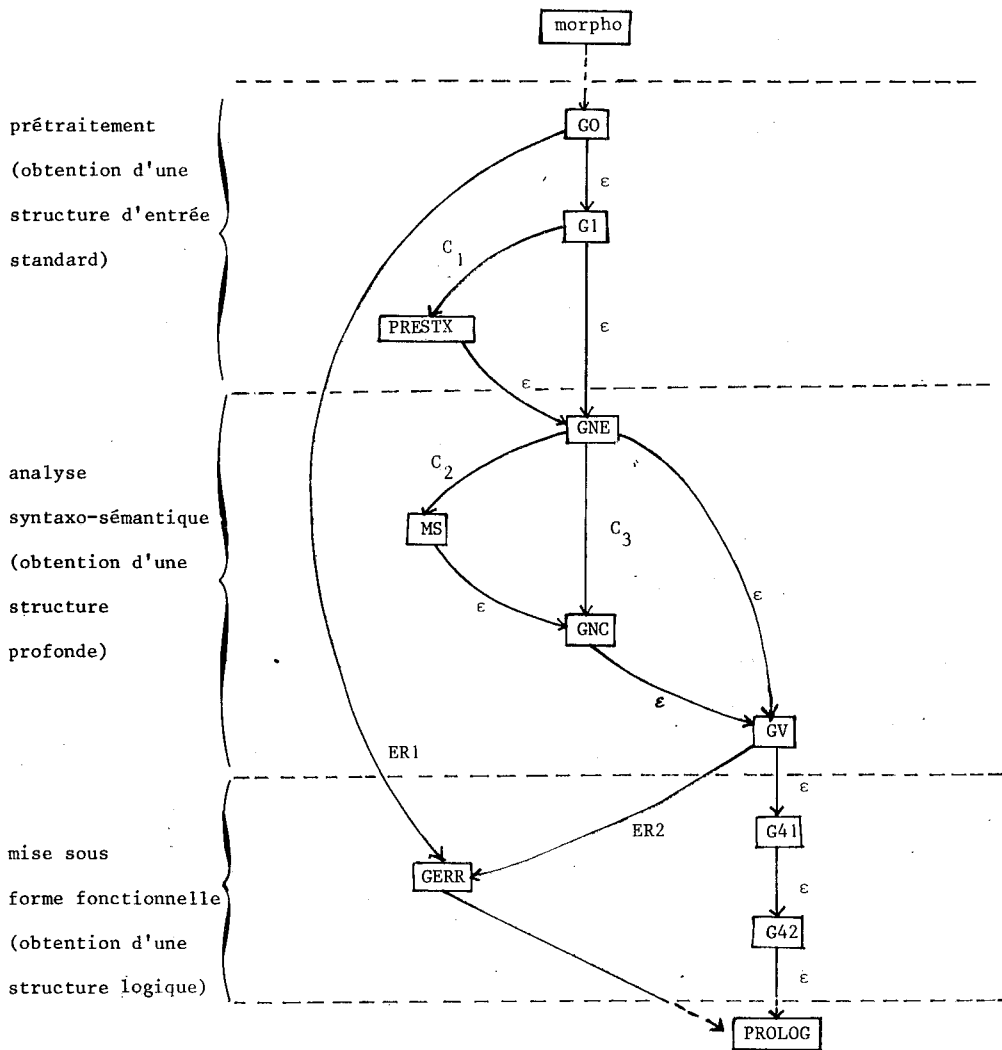
1.2. STRUCTURE GENERALE DE LA GRAMMAIRE

On a vu qu'en CETA une grammaire de transformation peut s'écrire sous la forme d'une arborescence de sous-grammaires, ce qui permet de diviser le traitement global en plusieurs phases qui s'enchaînent les unes aux autres, des conditions sur les arcs permettant de contrôler les transitions d'une sous-grammaire à une autre.

La grammaire utilisée peut être représentée au total comme un enchaînement de trois phases distinctes :

- la première a pour but de transformer l'arborescence fournie par ATEF en une structure standard : il s'agit de résoudre les ambiguïtés lexicales et, d'une façon générale, de simplifier l'arborescence d'entrée.
- la deuxième, qui est la plus importante, est formée de grammaires transformationnelles qui opèrent l'analyse proprement dite de la phrase. Celle-ci est transformée en une arborescence indiquant, sous une certaine forme, le sens du texte.
- On use enfin d'un autre jeu de grammaires pour représenter le sens du texte sous une forme opérationnelle, celle destinée à être utilisée par le système déductif PROLOG. On doit alors obtenir une arborescence composée des primitives décrites au chapitre III.

La structure générale de la grammaire est :



On a donc un réseau sans cycle (équivalent à une arborescence) dont les noeuds sont les sous-grammaires et les arcs sont les conditions permettant le passage entre deux sous-grammaires. la condition vide est indiquée par ϵ .

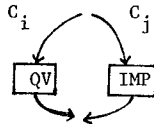
On voit qu'après une grammaire de standardisation de l'arborescence (G1) et, éventuellement de présyntaxe (PRESTX) on passe à la phase d'analyse du texte : les grammaires d'analyse du groupe nominal élémentaire (GNE) et du groupe verbal (GV) sont les deux principales grammaires, le traitement d'une phrase passant toujours par celles-ci. Enfin, les grammaires G41 et G42 transforment la structure intermédiaire obtenue en prédicats PROLOG utilisables pour la suite du traitement.

Si le texte comprend des mots inconnus ou des structures inconnues, on passe directement à la grammaire GERR qui indique à PROLOG l'existence d'un tel cas.

On voit les avantages de cette représentation modulaire de la grammaire :

- d'une part, on utilise des grammaires différentes pour traiter des phases différentes ou des cas différents d'une même phase, ce qui permet une compréhension de l'ensemble de la grammaire et une mise au point plus faciles : on peut modifier la manière d'analyser le GN sans modifier la grammaire analysant le GV par exemple.

- d'autre part, l'emploi de conditions permet de n'activer dans chaque cas que les grammaires nécessaires au traitement du texte. Pour la phase d'analyse syntaxo-sémantique, on n'utilise pour les phrases simples que le seul enchaînement GNE-GV, les phrases plus complexes utilisant le traitement général plus long. De même, on peut imaginer deux grammaires mutuellement exclusives. Si la phrase pouvait être soit sous forme impérative (cas actuel), soit sous forme interrogative, on pourrait remplacer GV par :



où C_i sera une condition portant sur un élément de question, tandis que C_j vérifiera que la phrase commence par un verbe à l'impératif.

On verra au paragraphe suivant le rôle et le fonctionnement des grammaires utilisées à chaque phase du traitement.

1.3. VARIABLES

On va indiquer quelles variables sont nécessaires au traitement général, c'est-à-dire servent à la formation des étiquettes.

1.3.1. Variables fournies par l'AM

On utilise la plupart des variables fournies par l'analyse ATEF. Seules sont éliminées les variables qui n'avaient servi qu'à assurer un découpage cohérent des formes (SEG, CV et DICT). On utilise donc CAT, GNR,... ainsi que UL.

1.3.2. Variables supplémentaires

Ces variables servent à marquer les constituants calculés au cours d'une phase pour des calculs ultérieurs ou à indiquer une valeur finale représentée sur l'arborescence de sortie :

- variables exclusives

etq1 : (modif) marque les différents modifieurs du nom quand on les calcule ;

etq2 : (X,Y,Z,T,U,V,W) est une variable d'état ;

enfin etq a, par exemple, pour valeurs : OD, MEMOBJET, NOM,... et sert dans ce cas à indiquer un élément de l'arborescence finale, ou : ACT, OBJ et sert alors à indiquer, au cours du traitement, le rôle particulier de certains éléments ou constituants : objet, action, adjectif ou position absolu,... Ces valeurs ne se retrouvent pas à la fin du traitement.

- variable arithmétique

VF := (20). Cette variable entière, bornée par + 20 indique en sortie de l'arborescence les variables formelles : sur (2, 10), nom (4, "A1"),... Celles-ci sont en effet des nombres quoique dans les exemples on ait utilisé des lettres (x,y,...) pour se conformer à la notation classique.

2 - LES SOUS-GRAMMAIRES

Pour chaque phase du traitement global on indique la forme-cible, le rôle des sous-grammaires constituant cette phase et le fonctionnement de ces sous-grammaires.

2.1. PHASE DE PRETRAITEMENT

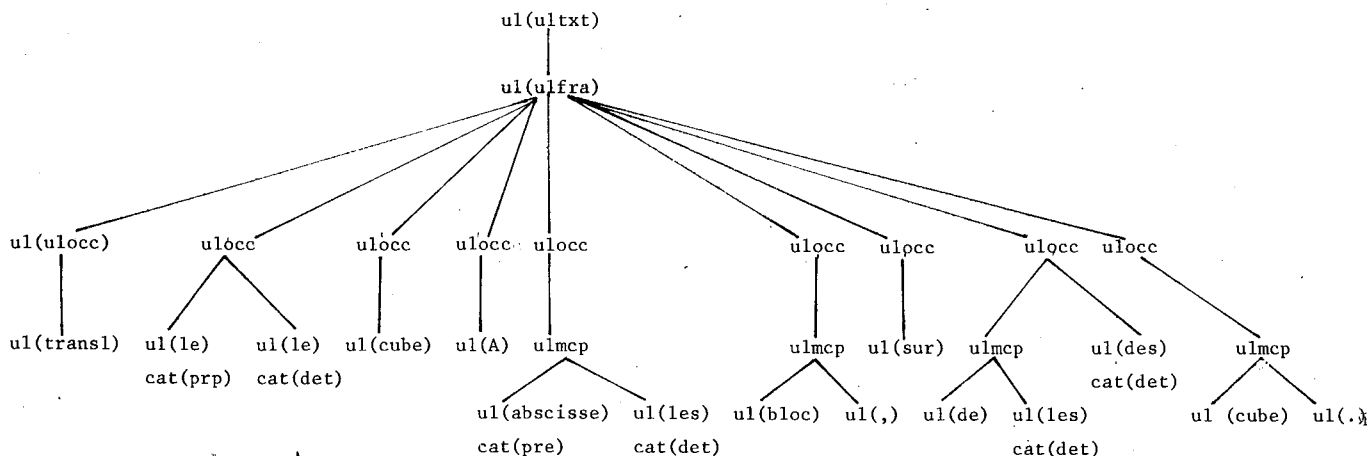
La grammaire GO, formée de la seule transformation identité contrôle l'existence de mot(s) inconnu(s) dans le texte d'entrée. En effet, si de tels mots existent, on se branche directement à la grammaire GERR indiquant les structures non-analysables. PROLOG se contentera, dans ce cas, de faire imprimer un message d'erreur. Sinon, on aura deux grammaires successives tendant toutes deux à simplifier l'arborescence.

2.1.1. Forme-cible

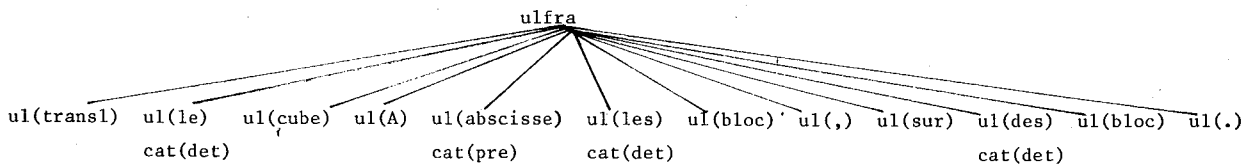
On a vu que le texte d'entrée est fourni par ATEF sous forme d'une arborescence ayant au moins quatre niveaux : ultxt, racine dominant les différentes phases, ulfra dominant les différentes formes, ulocc dominant les différentes occurrences d'une même forme ; enfin, ulocc peut dominer ulmcp pour les formes composées.

Exemple : "Posez le cube "A" à droite des blocs, sur des cubes".

La représentation ATEF est la suivante, en ne représentant comme étiquettes que les valeurs intéressantes cet exemple, donc essentiellement l'UL :



Nous voulons obtenir une arborescence n'ayant que deux niveaux : une racine dominant les différentes unités linguistiques du texte, donc pour l'exemple :



Si on désigne par [...] l'étiquette d'un sommet (exemple : [cat(vb), prs(1), ul(transl)]), on peut écrire la grammaire de l'arborescence qu'on veut obtenir ainsi :

```

<S> ::= [ul(ulfra)] (<LS1>)
<LS1> ::= [<S1>] [<S1>], <LS1>
<S1> ::= <var> (<val>) | <var> (<val>), <S1>
<Var> ::= cat | seg | gnr | ul | ... (ensemble des noms de variables)
<Val> ::= vb | nmc | sin | transl | ... (ensemble des valeurs de variables)

```

L'ordre dans lequel on trouvera les étiquettes des sommets reflètera la grammaire du français choisi, tel qu'il a été défini au chapitre II.

2.1.2. Grammaire de prétraitement

Cette grammaire opère deux sortes de transformations : quand une forme a donné plusieurs découpages (cas d'homographie), il faut éliminer les mauvaises solutions ; d'autre part, il faut éliminer les sommets inutiles (ultxt, ulocc, ulmcp).

2.1.2.1. "Désambiguïsation lexicale"

Dans un système très général (par exemple de TA) il faudrait un nombre important de règles pour sélectionner le bon sens d'un mot (par exemple, ferme peut être adjectif, substantif ou verbe).

Kulagina [9] indique comment résoudre une trentaine de cas différents, la méthode utilisée consistant à contrôler les catégories syntaxiques d'un certain nombre de mots entourant la forme homographe.

Dans notre système, on a les cas suivants : le, la, les (pronom ou article), des (article ou contraction de "de les"), que (pronom relatif ou élément de comparatif).

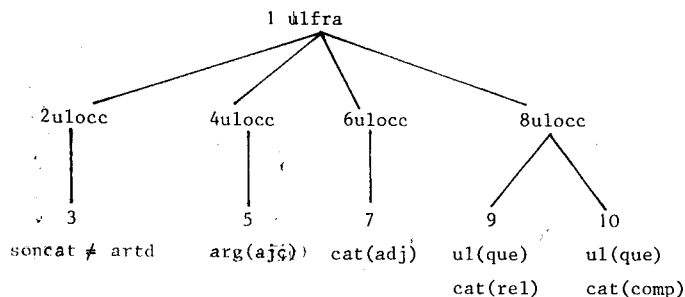
Dans chaque cas, on a deux règles, chacune se chargeant d'éliminer la mauvaise solution.

Par exemple, pour "des" on a les conditions suivantes :

* des est article indéfini si l'occurrence précédente est une préposition, conjonction, ponctuation ou verbe dont le deuxième actant (objet principal) n'est pas un groupe prépositionnel introduit par "de" (cas de "créez des cubes...", mais non de : "ce cube provient des ...).

* sinon c'est la contraction "de les".

De même, la règle : que = relateur de comparatif s'écrit :



Elle s'appliquera dans le cas de : cube plus gros que ..., mais pas dans : cube le plus gros que ...

On peut remarquer que cette résolution des homographes n'est valable que dans le cas général et qu'on peut toujours imaginer des cas où les contraintes devraient être renforcées (exemple : posez sur le cube des pyramides). Dans ces cas-là, il conviendrait d'explorer un contexte éventuellement très large.

2.1.2.2. Suppression des sommets inutiles

Un deuxième ensemble de règles supprime les sommets créés par ATEF et devenus inutiles : ultxt (notre texte est considéré comme une unité globale), ulmcp (les formes composées sont éclatées) et ulocc (devenu inutile après résolution des homographies). On garde ulfra comme racine.

2.1.3. Grammaire de présyntaxe

Cette grammaire, appelée seulement quand elle est nécessaire, effectue certaines simplifications simples de la phrase d'entrée: regroupement de la négation ne ... pas ou des verbes composés, transformation de qui + verbe en verbe au participe présent ou des structures comme "moins étroit" en "plus large".

On transforme par exemple P1 : "le bloc le moins court et qui n'est pas haut" en P2 : "le bloc le plus long et étant pas haut".

Ces quelques transformations faciliteront l'analyse proprement dite.

2.2. PHASE D'ANALYSE

Dans cette phase, on part de la chaîne d'entrée mise sous forme d'arbre à un seul niveau et on veut la transformer en une certaine "structure du sens". Cette structure est différente de la structure profonde finale dans la mesure où elle ne représente pas le sens sous forme de fonctions, prédicatives ou non.

2.2.1. Forme-cible

La structure profonde intermédiaire est déjà une représentation du sens. C'est ainsi que les différents énoncés d'un même concept sont représentés sous une forme unique (par exemple : le cube A, le cube nommé A, le cube qui s'appelle A,...). Cette structure est cependant choisie en fonction de la représentation finale et en est donc très proche.

La grammaire de cette forme est la suivante, les crochets encadrant les étiquettes :

```
<S>      ::= <phrase> | <conj> (<S>,<S>)
<conj>   ::= [cj=et] | [cj=ou]
<phrase> ::= <cd> (<obj>) | <modif> (<obj>,<obj>)
<cd>     ::= [etq=act, ul=<cd1>]
<cd1>    ::= créer | destr
<obj>    ::= [etq=obj, vf=<NB>] (<LSPEC>)
<NB>     ::= <N> | <N><NB>
<N>      ::= 0 | ... | 9
<LSPEC>  ::= <SPEC> | <SPEC>,<LSPEC>
<SPEC>   ::= <conj>(<SPEC>,<SPEC>) | <SPEC1>
```

<SPEC1> ::= <GNRE> | <NBRE> | <REF> | <NOM> | <FORME> | <POSCONST> (<OBJ>) | <POS1> (<OBJ>, <OBJ>) | <SIGNE> (<ARG>, <ARG>)

<GNRE> ::= | [etq=gnre, gnr=<G>]

<G> ::= mas | fem

<NBRE> ::= | [etq=nbre, ul = <ULN>]

<ULN> ::= des | tous | <NB>

<REF> ::= |[etq=ref, ul = <ULR>]

<ULR> ::= dern | avdern | <NB>

<FORME> ::= | [etq=nat, ul = <NAT>, <NEG>]

<NAT> ::= prisme | ... | pyramide

<NEG> ::= | [nega=neg]

<NOM> ::= | [etq=nom, ul = "<NOMA>", <NEG>]

<NOM1> ::= <L> | <L> <NOM1>

<L> ::= A | ... | Z | <N>

<POSCONST> ::= | [etq=pos, ul = <SS>, <NEG>]

<SS> ::= sur | sous | contre

<POS1> ::= | [etq=pos, ul=entre, <NEG>]

<SIGNE> ::= | [etq=signe, signe = <SIG>, <NEG>]

<SIG> ::= sup | inf | égal

<ARG> ::= <OP> (<ARG>, <ARG>) | <ARG1>

<OP> ::= [etq=op, op = <OP1>]

<OP1> ::= add | soustr | prod | quot

<ARG1> ::= <VALSTAND> | <NB1> | <DIM COR> | <DIM COR> (<OBJ>)

<VALSTAND> ::= [etq=taille, ul1 = <VS>]

<VS> ::= grande | petite | moyenne

<NB1> ::= [etq=taille, ul = <NB>]

<DIMCOR> ::= [etq=dim, ul = <DC>]

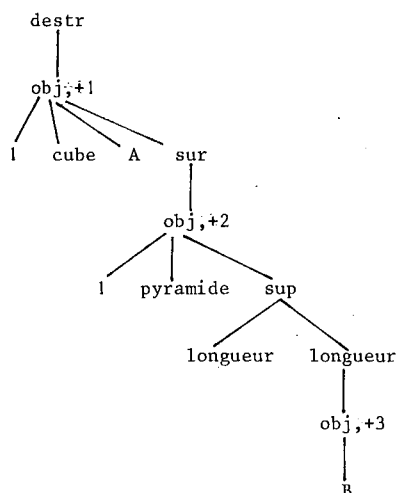
<DC> ::= longueur | largeur | hauteur | grosseur | abscisse | ordonnée | cote | place.

Nous allons décrire cette structure dans la mesure où elle diffère de la représentation finale.

Cette représentation est assez proche d'une structure dépendancielle : pour une phrase élémentaire, le verbe est la racine. Il gouverne ses différents actants : sujet, obj1,...

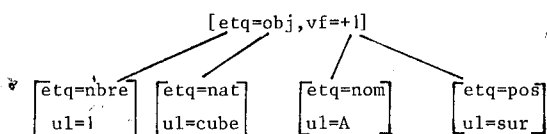
En fait, le sujet implicite est toujours le même : la machine. Il n'est donc pas indiqué, sa valeur étant constante. On remarque que par exemple les structures d'un ordre de création ou de destruction sont les mêmes. Pour les modifications d'objet, cette structure indique à gauche l'objet de "départ" et à droite l'objet d'"arrivée" quelle que soit la transformation indiquée.

Les objets sont les gouverneurs de tout ce qui les caractérise : propriétés, nombre, relations avec d'autres objets. Les propriétés élémentaires sont des feuilles, tandis qu'une relation est représentée par une dépendance entre les objets liés, l'élément (ou les éléments) intercalé entre l'objet gouverneur et les objets dominés marquant la valeur de la relation.



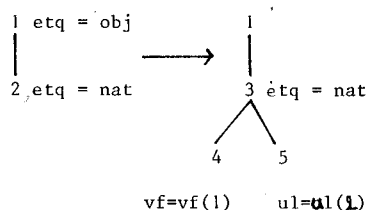
"Détruisez le cube A qui est sur une pyramide plus longue que B".

On n'a indiqué sur l'arborescence que certaines valeurs de variables, par exemple, pour le GN représentant "le cube A qui est sur ...", on a en fait :



Dans cet exemple, la variable etq indique le rôle général des sommets, et ul la valeur particulière.

Il est évident que cette représentation et la structure finale sont très voisines. On passera de l'une à l'autre par des règles telles (pour la forme de l'objet) :



Les deux structures peuvent être considérées comme isomorphes, la dernière structure étant simplement une forme adaptée au système déductif, mais qui n'ajoute rien au sens.

Aussi c'est à cette phase d'analyse que l'on s'intéressera le plus.

2.2.2. La grammaire GNE

Cette première grammaire analyse et calcule les groupes nominaux élémentaires et éventuellement les groupes prépositionnels si un groupe nominal complet a pu être calculé. GNE traite les structures comprenant le substantif et les éléments qui le précèdent dans le GN : les GN comme "le cube qui est gros" ou "le bloc sur la pyramide" où des phénomènes d'imbrication peuvent intervenir, ne sont donc calculés que partiellement. Les structures analysées par GNE sont engendrées par les règles suivantes :

GNS ::= (apos)(det)(nb)(nbordi)(GA)nmc(apos) | prp | apos

GA ::= (comp) adj

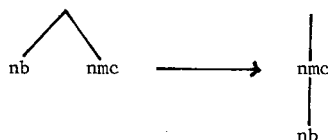
GP ::= prep GN

où : les () indiquent les éléments présents ou non et les terminaux sont représentés par la catégorie syntaxique (qui ne sera pas en fait la seule variable testée dans les règles).

GNE analyse donc les GN complets formés d'un pronom personnel ou d'une apos (nom donné à un objet) et les GN, éventuellement incomplets tels que : un cube A, les deux derniers plus longs prismes, ... Le GP n'est calculé ici que le GN est "simple" et a pu être totalement analysé dans cette grammaire.

Celle-ci est formée de règles analysant le GN pas à pas, à partir du noyau en traitant chacun des éléments qui peuvent y être ajoutés : règle contrôlant et calculant adv+adj+nmc, puis règle portant sur adj+nmc, nb+nmc, etc..., chaque règle pouvant ou non s'appliquer. Chaque règle met en dépendance de l'objet la primitive exprimée.

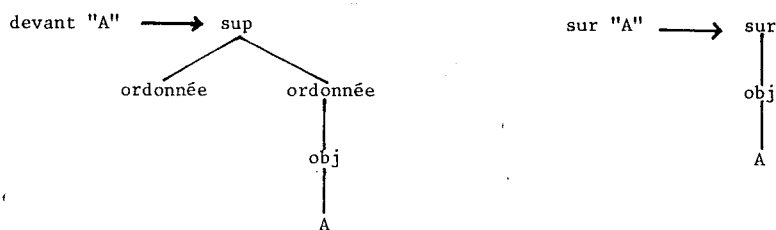
Par exemple, on a :



avec un certain nombre de modifications sur les étiquettes des sommets, indiquant le rôle de ceux-ci aux grammaires suivantes.

La forme générale que donne cette grammaire pour les GN est celle d'un sommet étiqueté "objet" dominant les différents éléments qu'on a pu analyser ; un sommet "nom", un sommet "genre" (dans le cas d'un pronom) ; les adjectifs, absolus ou superlatif sont déjà indiqués sous la forme d'une comparaison de valeurs.

Pour les GP calculés par GNE, la représentation est la préposition dominant l'objet pour les prépositions comme sur, entre, ... ou une comparaison de valeurs pour des prépositions comme "devant" :



Dans GNE, on active chaque règle l'une après l'autre, partout où elle peut s'appliquer (donc sur chaque GN élémentaire). Cet ordre d'application provient de la place fixe qu'occupent les différents éléments en tête d'un GN. On verra que ce n'est pas le cas de toutes les grammaires.

2.2.3. La grammaire MS (modifieurs simples)

Cette grammaire qui n'est appelée que si certaines de ses règles s'appliquent calcule les modifieurs simples des objets calculés partiellement par GNE.

Ces modifieurs sont exprimés par des structures n'admettant pas d'imbrication qui est le cas par exemple des comparatifs : le cube plus long que "A" et sur le bloc plus haut que "...".

On s'occupe donc entre autre des adjectifs absolus ou superlatifs placés derrière le nom, des adjectifs et noms accompagnés d'une valeur numérique (large de 100, de longueur 100), des valeurs de centre (centré en (-,-,-)) ou de nom (nommé "A", de nom "B")...

Chacun de ces éléments peut se calculer immédiatement car il ne peut pas lui-même être modifié ; ici aussi, on applique donc les règles séquentiellement.

Un problème qui se pose est la multiplicité des structures syntaxiques exprimant le même sens : par exemple pour l'indication d'une valeur de dimension, on peut avoir les structures, toutes sémantiquement égales :

- GP introduit par "de" : de largeur 100 ; de 100 unités de large(ur),
- GA formé d'un adjectif modifié par un GP : large de 100 unités,
- Relative : dont la largeur est 100,
- GV au participe : ayant une largeur de 100.

Pour le nom ou le centre, par exemple, on trouve à peu près ces mêmes structures.

Comme la représentation donnée à ces divers modifieurs est différente, on risquerait de devoir multiplier le nombre de règles : aussi use-t-on de deux sortes de règles : les premières projettent les différentes structures d'entrée en une forme canonique ; les secondes calculent effectivement le sens de la modification. En reprenant l'exemple de la donnée numérique de la largeur, on peut choisir la structure : mot de taille d'objet + nombre \Rightarrow "largeur 100" ou "large 100" (des mots comme large et largeur ont une UL commune et les tests ne portent pas seulement sur la catégorie mais aussi sur la variable SEM qu'ils ont en commun). Si on doit calculer la structure "de 100 unités de large", on arrive à la forme "large 100" en utilisant des règles qui suppriment le mot "unité" non-signifiant et les deux prépositions "de" et enfin inversent l'ordre des deux mots restants. Pour "dont la largeur est 100", on supprime le pronom dont et le verbe être. Ces mêmes règles simplifient aussi des constituants comme : de nom A, dont le centre est (-,-,-), dont la forme est cubique,... La structure canonique étant ainsi obtenue, on utilise une règle pour chaque signification dont la représentation est structurellement différente.

Ces différentes règles calculant toutes un élément modificateur de l'objet ne mettent pas ces spécifications sous le sommet représentant l'objet car la structure peut en fait être plus complexe : on peut en particulier avoir des conjonctions de modifications et le modifieur peut n'être pas immédiatement derrière l'objet modifié (exemple : cube plus gros que A et de longueur 100). Toutes les modifications calculées sont dotées de la variable etq=modif qui est utilisée par la grammaire suivante.

2.2.4. La grammaire GNC (groupe nominal complet)

Cette grammaire achève quand c'est nécessaire le calcul complet des GN et des GP. Il s'agit pour GNC de mettre les modifieurs déjà calculés en dépendants du sommet objet auquel ils se rapportent, d'autre part de calculer les modifieurs non encore calculés, ceux-ci exprimant des relations entre objets et pouvant être à un niveau d'imbrication plus ou moins grand.

2.2.4.1. Les modifieurs

Ces modifieurs sont des GA au comparatif : plus long que..., des GP : à droite du cube, sur le bloc, ou des relatives (ou participiales) exprimant des relations de constitution (appartenant au cube) ou introduisant un GA ou un GP : qui est plus long que...

2.2.4.2. Les conjonctions

On traite aussi le cas des connecteurs : négation (pas) ou conjonction (et, ni,...).

Le problème que posent les conjonctions est qu'elles peuvent se placer presque n'importe où dans le discours et qu'elles peuvent donc avoir une portée très variable (ce qui risque d'ailleurs d'entraîner des ambiguïtés, comme on le verra).

Les conjonctions peuvent unir deux phrases (exemple : Créez A et détruisez B), deux GN (Détruisez A et B), deux GP (sur A et sous B), deux GA (long et très étroit), deux modifieurs de structures différentes (un cube plus gros que A et passant par B),... Une des parties conjuguées peut aussi être victime d'une élision :
aussi long que A et que B \iff aussi long que A et aussi long que B ; aussi long mais moins haut que A \iff aussi long que A mais moins haut que A ; passant par A et par B \iff passant par A et passant par B (en considérant que c'est l'ensemble passer + par qui exprime la relation). On voit que dans ces cas, la conjonction met en facteur une partie des éléments communs aux deux structures conjointes. En tenant compte des structures complétées correspondant aux élisions possibles, on peut distinguer deux cas :

- la conjonction unissant deux phrases (dont le cas sera traité par la grammaire suivant analysant l'ensemble de la phrase).

- la conjonction unissant deux modifieurs d'objet, ceux-ci étant calculés soit par cette grammaire, soit par la grammaire précédente, la structure servant à exprimer ces modifications pouvant, comme on l'a vu, être diverse.

La conjonction de deux GN correspond en fait, suivant le cas, à l'une de ces deux possibilités (détruisez A et B \implies détruisez A et détruisez B ; sur A et B \implies sur A et sur B).

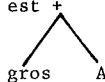
2.2.4.3. Mode d'activation

Contrairement aux grammaires précédentes, les règles de GNC ne s'appliquent pas séquentiellement, mais dans un ordre quelconque. Les règles de GNC sont récursives et on indique dans la liste de récursion les règles susceptibles de s'appliquer après chacune d'elles.

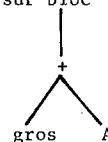
Considérons la phrase : "le cube qui est plus gros que la pyramide qui est sur le bloc qui est plus gros que A".

On applique successivement (en simplifiant la représentation des résultats) :

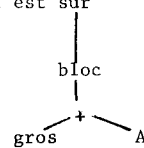
- la règle calculant le comparatif sur A : ... bloc qui est +



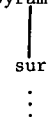
- la règle mettant ce modifieur en dépendant du bloc : ... qui est sur bloc



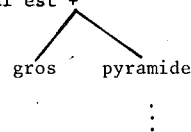
- la règle calculant le GP de position : ... pyramide qui est sur



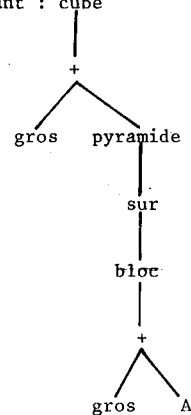
- à nouveau la règle mettant le modifieur en dépendant : ... plus gros que pyramide



- à nouveau la règle calculant le comparatif : le cube qui est +



- enfin, à nouveau la règle mettant le modifieur en dépendant : cube



Pour chacune des règles, il faut faire des tests pour s'assurer qu'elle ne s'applique pas intempestivement : par exemple, la règle calculant le comparatif peut s'appliquer si l'élément suivant le sommet "objet" est un point, mais pas si c'est le début d'une relative. D'une manière générale, on ne peut appliquer une règle ayant pour effet de placer un GN en dépendant d'un autre élément (préposition ou signe de comparaison par exemple) que si le GN a été entièrement calculé. Pour les structures emboîtées, cela revient à commencer le calcul par la droite.

L'introduction des conjonctions complique l'analyse. On utilise un certain nombre de règles complétant les structures élidées de manière à pouvoir effectuer les calculs correspondant aux éléments unis par la conjonction (par exemple : "plus gros que A et plus gros que B", obtenu à partir de "plus gros que A et B", permettra le calcul des deux comparatifs). Les règles sur les conjonctions dont l'effet est de placer la conjonction en gouverneur des éléments qu'elle unit ne peuvent être appliquées que quand les éléments en question ont été totalement calculés. Les règles induites par les conjonctions s'appliquent comme les autres règles de GNC déjà vues, c'est-à-dire dans un ordre variable : dans la phrase "le cube plus gros que A et haut", on applique d'abord la règle calculant le comparatif (l'objet A est entièrement calculé), puis la règle de conjonction sur les deux éléments étiquetés tous deux par etq|=modif alors que dans la phrase "le cube plus gros que le cube qui est gros et haut", l'ordre d'application de ces deux règles est inversé.

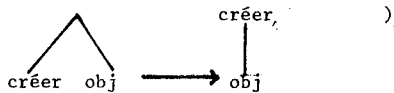
Donc, pour chaque règle de GNC, on doit contrôler le (ou les) élément(s) suivant(s) pour savoir si la transformation est possible. D'autre part, dans la grammaire activant ces règles, on indique pour chacune d'elle la liste des règles susceptibles de s'appliquer : par exemple, dans la liste correspondant à la règle calculant le comparatif figurent la règle calculant la conjonction des modifieurs et la règle mettant le modifieur sous l'objet gouverneur, mais pas la règle calculant le GP de position.

2.2.5. La grammaire GV (groupe verbal)

Cette grammaire calculant le groupe verbal (verbe avec ses arguments objets) analyse en fait toute la phrase compte tenu de la nature impérative du discours pris en compte. Quand on utilise cette grammaire, les GN et GP ont été entièrement calculés et il ne reste plus, suivant la nature des verbes, qu'à effectuer les transformations pour obtenir la structure-cible indiquée au début du paragraphe. Dans la mesure où les grammaires suivantes ne font que des transformations modifiant la représentation du sens sans rien y ajouter, on peut dire que cette grammaire clôt la phase d'analyse et de calcul du sens du texte.

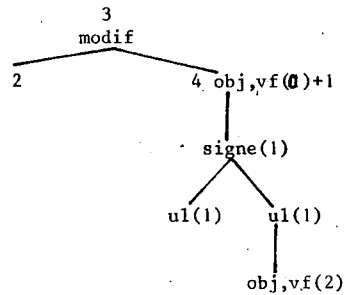
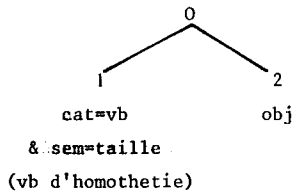
La représentation des actions qu'expriment les verbes imposera deux sortes de règles :

- pour les verbes de création ou de destruction on place le verbe en gouverneur de l'objet créé ou détruit
(exemple :

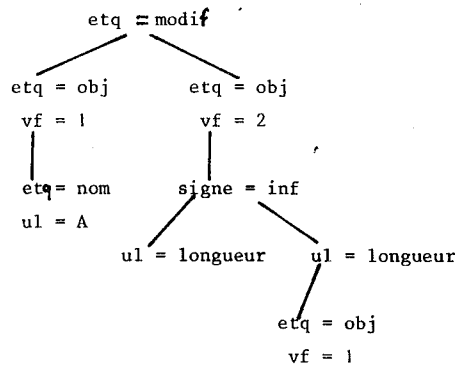


- pour les autres verbes indiquant une modification d'objet (nomination, translation avec éventuellement indication numérique ou indication de sens,...), on cherche à donner le résultat sous forme d'un gouverneur, étiqueté modif, indiquant à gauche l'objet "de départ" (objet à modifier) et à droite l'objet "d'arrivée" (objet obtenu par la modification).

Exemple :



Cette règle donne pour : "Raccourcissez A"



2.3. PHASE DE POST-TRAITEMENT

Cette dernière phase, donne la structure finale.

2.3.1. Structure-cible

C'est bien entendu la structure exposée dans la première partie de cette étude. Cependant, de par la nature du système CETA, on ne donne pas le résultat sous la forme directement assimilable par PROLOG, c'est-à-dire comme un prédicat. Pour chaque sommet, on a en effet une valeur de variable. Par exemple $OD(+1,1,nom(+1,"A"),destr(+1))$, correspondant à "Détruisez "A"", est donné en fait par l'arborescence :

$etq=OD (vf=+1 , nbre=1 , etq=nom (vf=+1 , ul="A") , ul=destr (vf=+1))$.

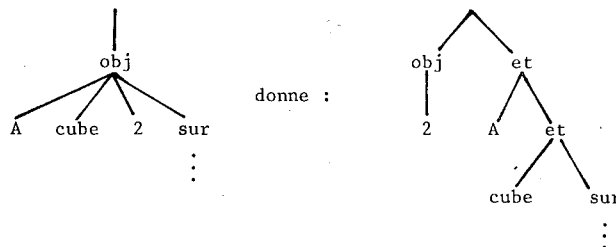
Le système déductif devra donc préalablement au traitement proprement dit, extraire les valeurs de variables représentant les étiquettes des sommets.

2.3.2. Les grammaires G41 et G42

Ces deux grammaires, dont la première s'occupe particulièrement des propriétés et des relations et la deuxième des actions, sont formées de règles qui s'occupent des différentes structures du sens qu'elles transforment suivant la structure profonde définie auparavant. Le traitement qu'elles font subir à l'arborescence est le suivant :

- la liste des éléments gouvernés par un sommet objet, ces éléments représentant ses propriétés, est transformée en une arborescence binaire dont les feuilles sont les éléments de la liste des propriétés et les noeuds des connecteurs "et". Seul le nombre, dont la signification est très particulière, n'est pas touché par ces transformations. Cette arborescence binaire est créée à droite du sommet objet.

Par exemple :



- On crée un sommet étiqueté OD pour tous les objets qu'il ne s'agit pas de créer.
- On transforme les propriétés simples en prédicats, les règles pouvant différer selon la nature de la propriété.
- On transforme également les relations en prédicats.
- On transforme ensuite la représentation des actions : on représentera les verbes comme détruire, modifier, intervertir par des primitives décrivant la nature logique de ces actions.
- Enfin, on effectue un nettoyage pour éliminer de l'arborescence les étiquettes inutiles résiduelles et effectuer des transformations du genre $sous(x,y) \rightarrow sur(y,x)$ pour bien aboutir à la structure profonde cible.

On verra plus précisément sur des exemples complets comment fonctionnent ces grammaires, de même que les autres grammaires définies auparavant.

3 - EXEMPLES DE TRAITEMENTS

On décrira le traitement sur deux exemples. Le premier exemple, simple, verra son traitement être détaillé ; pour le deuxième, plus long, on n'indiquera que les points essentiels.

Exemple 1 :

Soit la phrase : "Détruisez une haute pyramide".

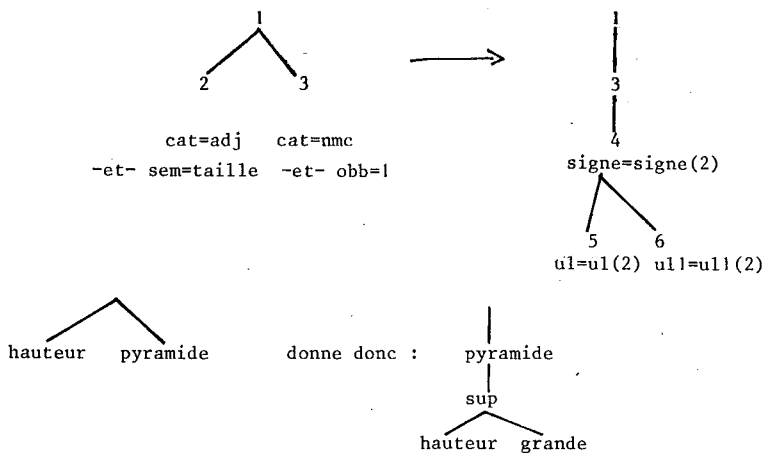
Après application de l'analyse morphologique et de G1, on obtient :



(dans cette arborescence, comme dans les autres, on n'indique que certaines valeurs de variables, celles permettant de suivre le déroulement du traitement : ici c'est l'UL qui est indiquée).

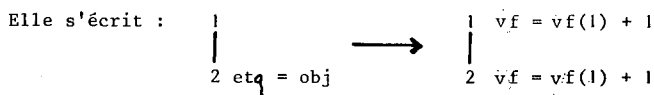
La grammaire PRESTX ne s'applique pas et on appelle GNS, traitant les GN (ici il n'y en a qu'un). 4 règles s'appliquent successivement :

La première s'applique sur les adjectifs de taille :

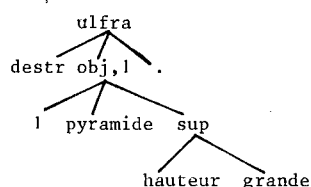


La seconde s'applique sur l'article indéfini singulier devant un nom d'objet : la transformation donne le nombre 1. A ce moment on sait qu'il n'existe plus d'élément appartenant au GN devant le nom. Une règle place alors la nature de l'objet en dépendant de celui-ci.

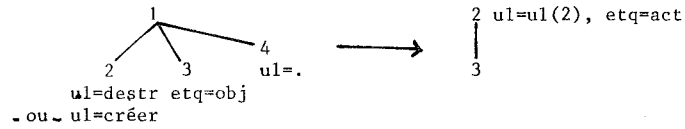
Enfin, une règle place une VF sur le sommet objet. Cette variable est obtenue en incrémentant un compteur placé sur la racine. Cette règle, qu'on pourrait mettre au début de G41 est appliquée à ce niveau pour des raisons de commodité.



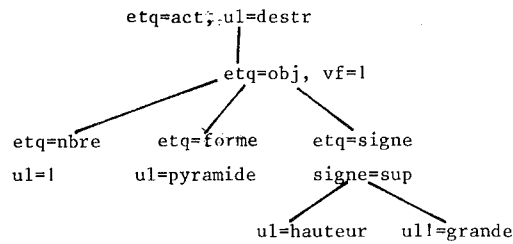
A la sortie de GNS, on a donc :



On fait alors une série de tests pour voir si MS ou GNC s'appliquent. Aucun de ces tests n'étant concluant, on passe directement à GV pour analyser la structure globale de la phrase. Une règle s'applique :

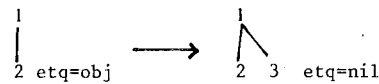


On obtient donc en sortie de la phase d'analyse :

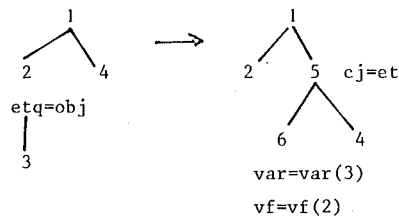


On passe alors aux grammaires finales G41 et G42.

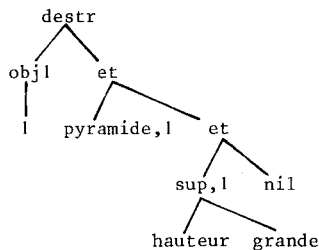
On met d'abord en arborescence les propriétés de l'objet, sauf le nombre, des "et" unissant les propriétés. Le processus est initialisé par une règle plaçant un sommet étiqueté "nil" à côté du sommet objet :



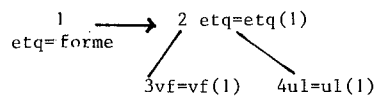
Puis les différentes propriétés sont ajoutées à cette arborescence, accompagnées de la VF propre à l'objet :



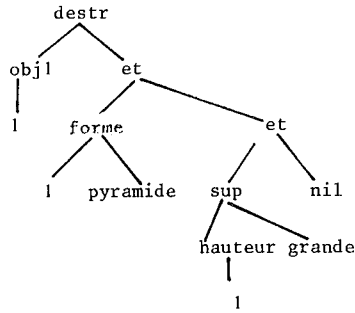
On obtient alors :



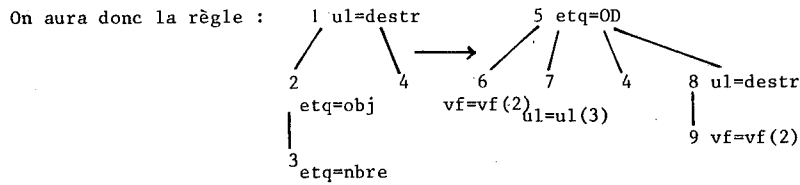
On effectue alors, en une application simultanée, la transformation de ces propriétés en prédicats, les règles étant par exemple, pour la forme :



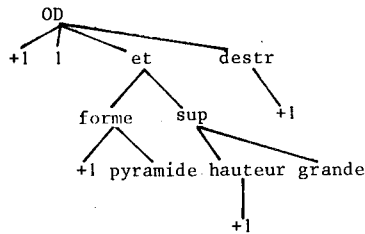
On obtient donc :



On transforme ensuite la représentation de l'action : pour une destruction, il s'agit d'obtenir un certain objet dans la base de données et de le détruire.

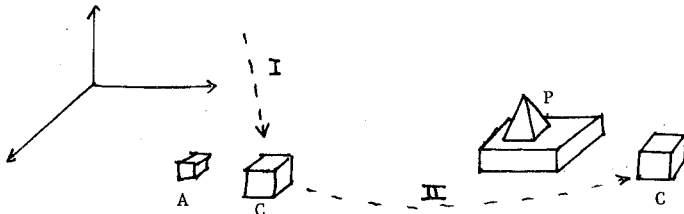


Enfin une dernière règle simplifie les structures $\begin{matrix} \text{et} \\ \swarrow \searrow \\ \text{x} \quad \text{nil} \end{matrix}$ (le sommet nil ne subsistera que pour les objets sans propriété). On obtient alors finalement :



Exemple 2 :

"Créez "c", un cube petit, mais plus grand que "A" et posez-le à droite du bloc qui est soit sous la pyramide "P", soit sur le plus gros cube."



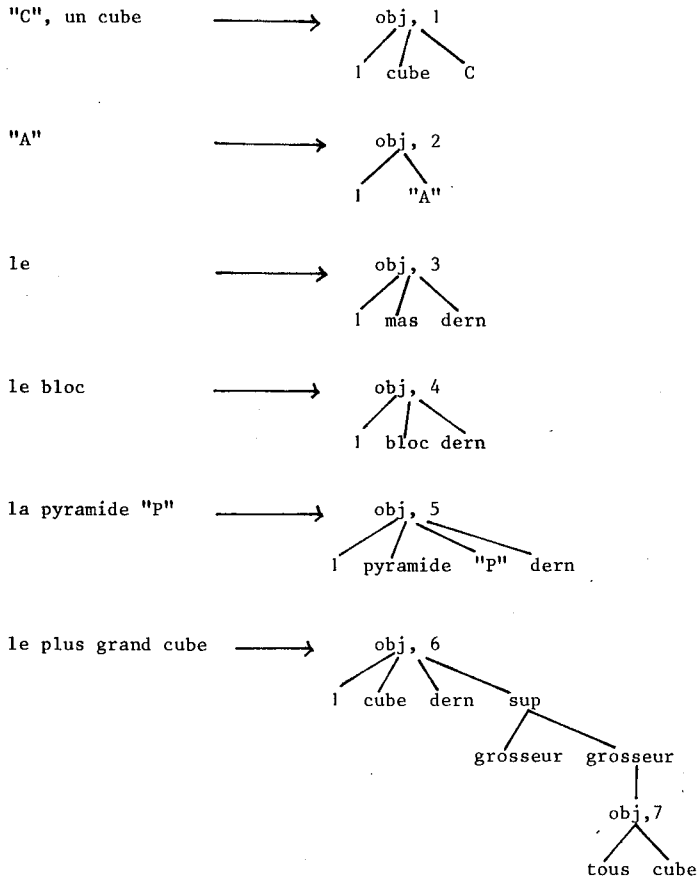
Cette phase représente une succession de deux ordres et correspond donc à deux actions successives, comme il est indiqué sur la figure ci-dessus.

L'analyse se fait de la façon suivante :

Après l'analyse morphologique, on a une arborescence étiquetée où, par exemple grand et gros ont la même étiquette (avec l'ul=grosseur) tandis que "à droite du" est transformé en une occurrence "à droite de" (avec l'ul=abscisse) et une occurrence "le".

La grammaire G1 simplifie cette arborescence de la manière que l'on a vue et PRESTX transforme "qui est" en "étant".

On appelle alors GNS qui s'applique sur 6 structures. Les règles sont du type exposé à l'exemple précédent et on obtient donc :



On applique aussi la règle du GP sur "sous la pyramide P" et sur "sur le plus grand cube". Si on note OB1, ..., OB6 les 6 structures calculées soit totalement (ex : "A", le), soit partiellement (ex : le bloc), on a à la sortie de GNS :

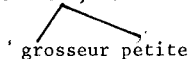
Créer OB1 petit mais plus grand que OB2 et posez OB3 à droite de OB4 étant soit sous, soit sur

```

    graph TD
      OB5[OB5] --- OB4[OB4]
      OB6[OB6] --- OB4
  
```

On a représenté la phrase comme une chaîne pour alléger la structure, mais il est clair qu'on a en fait une arborescence, une racine dominant les différents points.

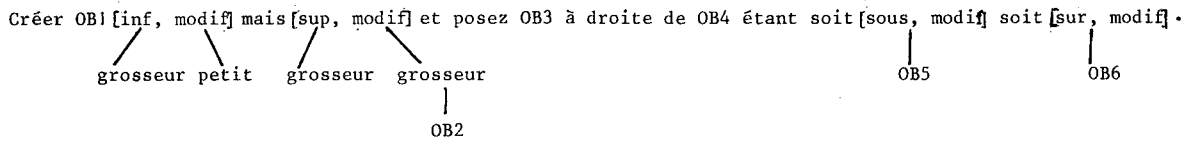
Le test sur MS est satisfait (par l'adjectif petit). On n'applique qu'une seule règle calculant la représentation de cet adjectif, qui est : inf, modif



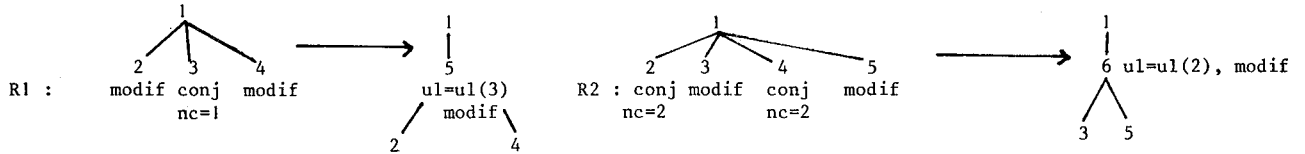
On appelle alors GNC.

La première règle qui s'exécute est celle calculant le comparatif, l'objet OB2 n'étant pas modifié par le reste non analysé du texte (alors que la règle plaçant l'objet sous la préposition ne s'applique pas sur "à droite de OB4", l'objet OB4 n'étant pas totalement calculé).

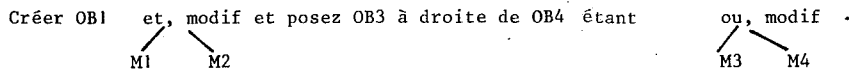
On obtient alors :



On applique alors les deux règles calculant les conjonctions de modif :

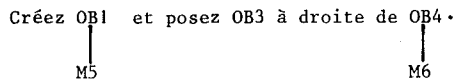


R1 s'applique pour la conjonction mais, d'ul=et et R2 pour les conjonctions soit, d'ul=ou. En notant M1, ..., M4 les quatre sommets modif déjà calculés, on a la structure suivante :

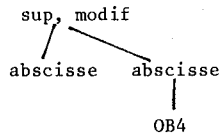


Les règles plaçant les sommets modif en dépendants des objets s'appliquent ensuite, ces éléments étant entièrement calculés. On dispose de deux règles, l'une traite obj + modif, l'autre traite obj + étant + modif.

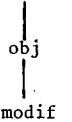
En notant M5 et M6 les deux conjonctions de modif, on obtient :



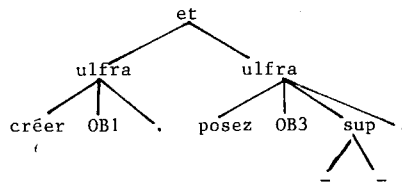
On peut alors enfin appliquer la règle : préposition d'emplacement + obj : cette règle transforme "à droite de OB4" en sup, modif



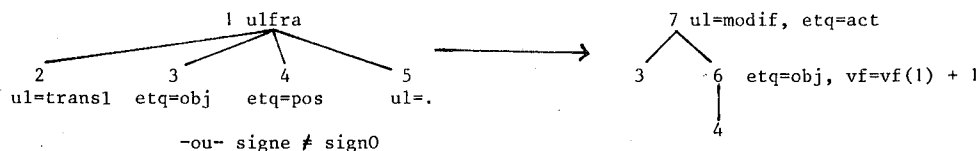
Aucune autre règle de GNC ne s'applique, en particulier pas la règle transformant obj modif en obj le test affectant cette règle prévoyant que le sommet à gauche du sommet obj ne doit pas être un verbe de déplacement. Dans ce cas en effet, le GP est un des arguments du verbe.



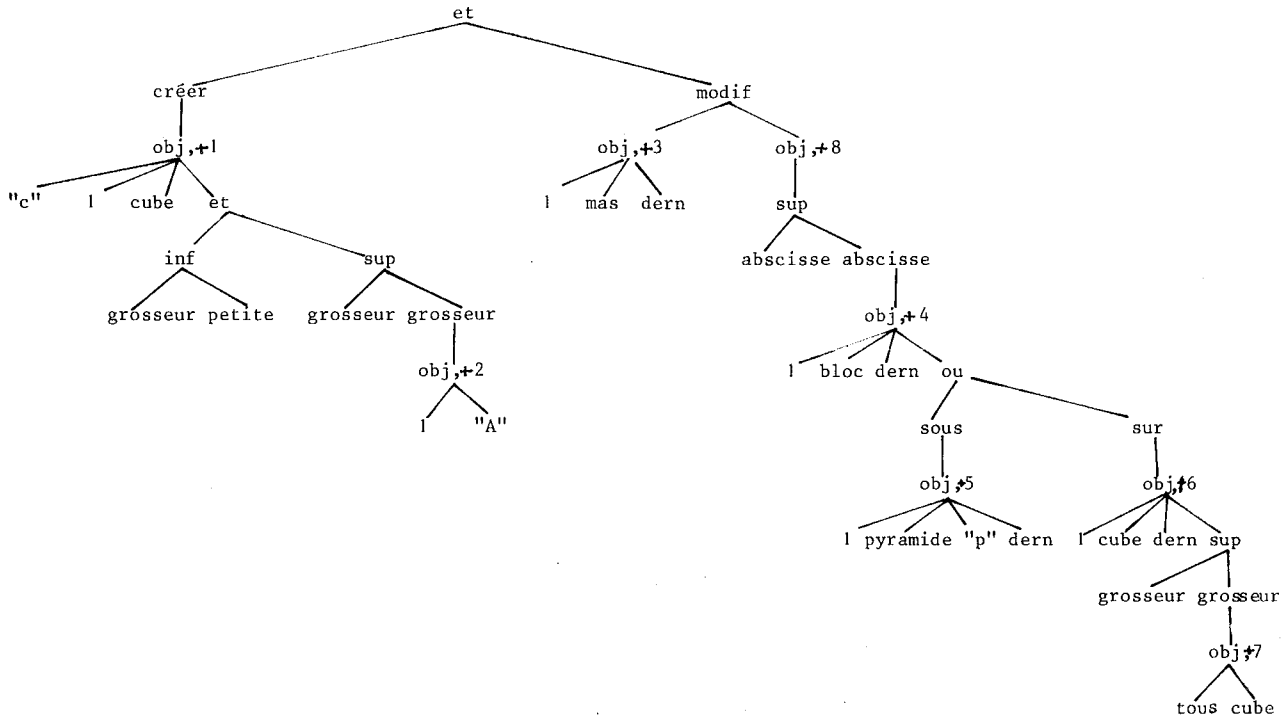
On obtient alors :



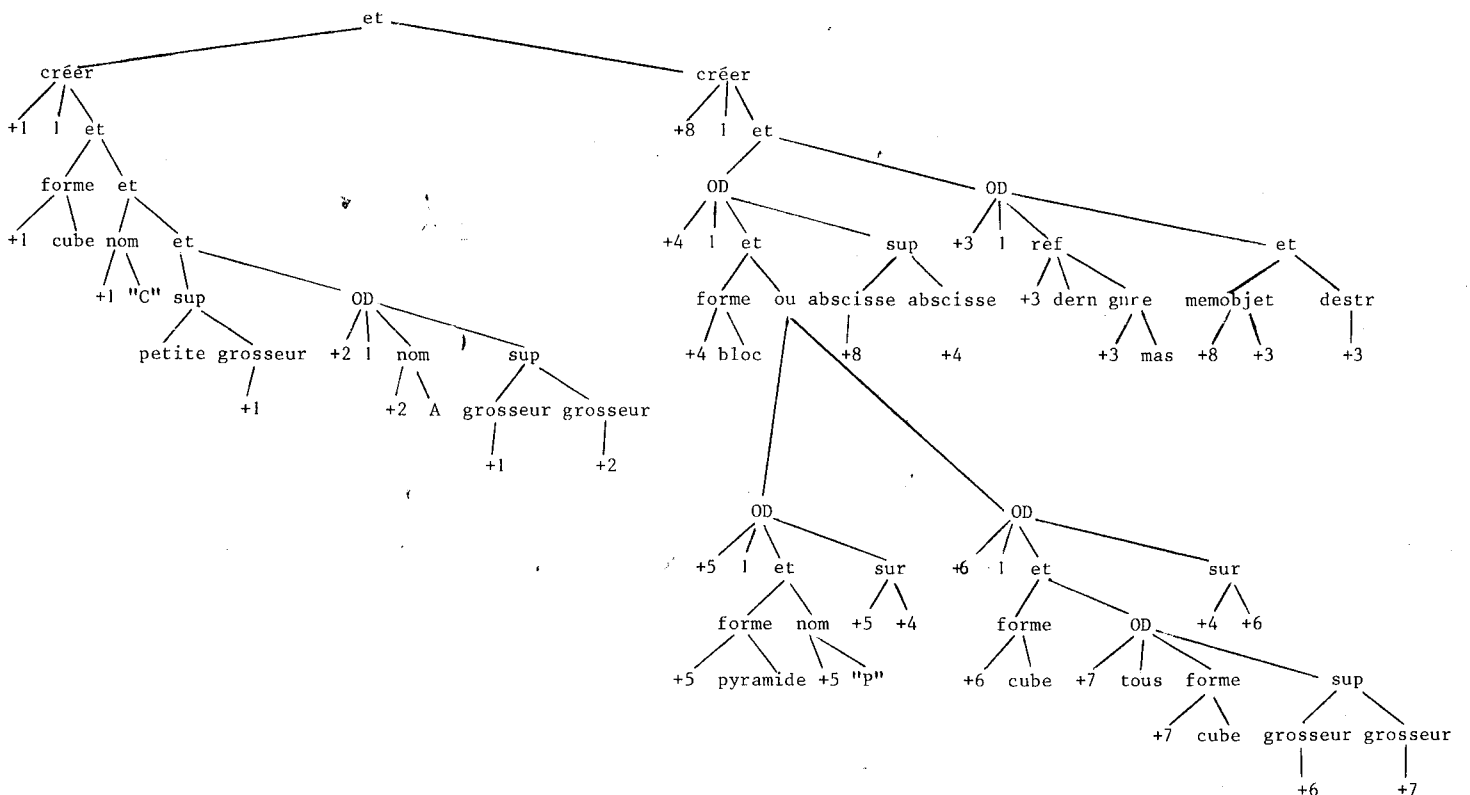
On passe alors aux règles construisant la structure de chaque phrase. Le cas de créer a été vu à l'exemple précédent. Pour les verbes de translations, la règle afférente est :



A la sortie de la phase d'analyse, on a donc :



On décrira plus brièvement le fonctionnement de G41 et G42. On va indiquer directement l'arborescence finale que fourniront G41 et G42 et, en comparant les deux structures, on pourra voir quelles transformations ont été opérées pour effectuer le passage de l'une à l'autre forme.

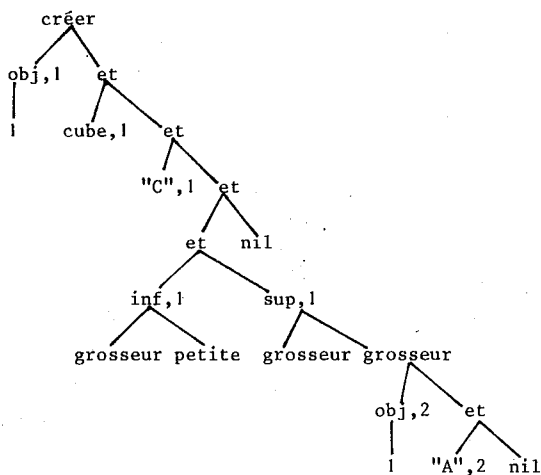


Les opérations conduisant à cela auront été les suivantes :

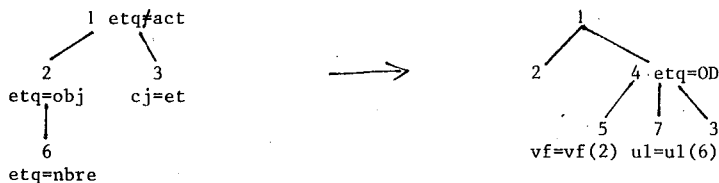
- suppression des etq ref=dern sous les objets 4, 5 et 6, ces objets étant de toute façon définis par ailleurs de façon unique. On aurait d'ailleurs pu faire cette suppression à l'étape précédente.

- affectation d'un sommet "nil" à droite de chaque sommet "obj"

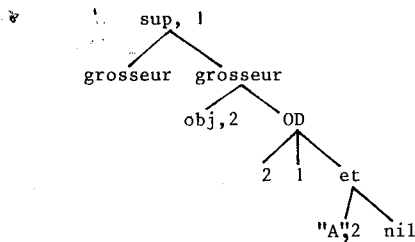
- mise en arborescence binaire des propriétés des objets. Pour l'objet 1, par exemple, on aura donc :



- création d'un sommet OD pour les objets non dominés par un sommet étiqueté "act", cas de "créer" et de "modif". Cette règle qui s'écrit :

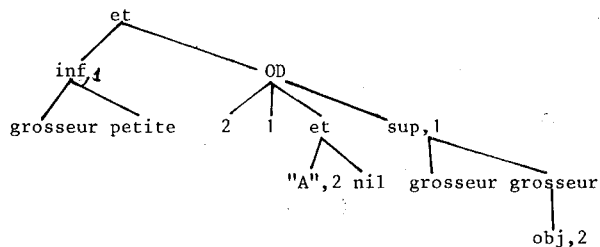


donnera, pour l'objet 2 :

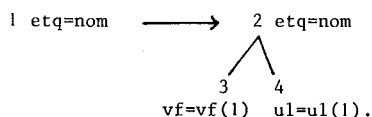


On voit que seul le 4ème dépendant de OD n'est pas encore en place. Pour l'obtenir, on fait "remonter" OD en lui donnant pour 4ème argument le sommet qui le domine, jusqu'à ce que ce sommet soit une conjonction.

Toujours pour l'objet 2, on obtiendra donc :

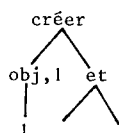


On transforme alors les propriétés et relations en prédicats, par des règles comme :

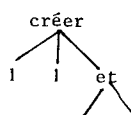


Pour obtenir la partie gauche de l'arborescence, on voit qu'il ne reste que trois transformations à faire :

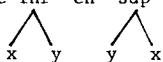
- une règle transforme



en



- une règle transforme inf en sup, inf ne figurant pas parmi les primitives choisies



- une règle enfin supprime les sommets "nil" et les sommets "et" qui les dominent.

Il est évident que les règles ainsi décrites s'appliquent sur toute l'arborescence. On utilisera une règle supplémentaire dans la partie droite, pour transformer le sommet "modif". Cette règle remplacera ce sommet par une création-destruction, en créant les sommets "créer", "memobjet", ...

4 - SUR LES AMBIGUITES

Dans tous les problèmes d'analyse de langue naturelle, le problème des ambiguïtés risque de se poser. Quoique la réduction du nombre des concepts utilisés et du vocabulaire limite dans notre cas la fréquence de ces cas, il est évident que ce problème existe. Dans l'exemple 2 décrit précédemment, il y aurait eu ambiguïté si au lieu d'utiliser les conjonctions soit ..., soit ..., on avait utilisé la seule conjonction équivalente ... ou Dans "posez-le à droite du bloc qui est sous la pyramide P ou sur le plus gros cube", la conjonction peut unir, avec le GP "sur le plus gros cube", soit le GP "à droite du bloc", soit le GP "sous la pyramide P".

Dans la plupart des systèmes, les différentes analyses d'une phrase sont obtenues par un système de backtrack. On choisit alors la solution la plus "vraisemblable" en utilisant par exemple des méthodes heuristiques. Cependant le système CRITA qu'on utilise est un transducteur déterministe : la première solution obtenue est considérée comme la seule solution. Aussi la détection des ambiguïtés et leur traitement peuvent s'avérer difficiles. On n'a donc pas écrit, dans le système d'analyse, de règles analysant ce cas. On peut néanmoins voir quelles solutions pourraient être adoptées.

Prenons deux exemples montrant les problèmes qui se posent :

P1 : Détruisez le cube qui est sur le bloc et la pyramide

P2 : Mettez le cube sur le bloc et la pyramide sur le cube

P1 a deux sens :

P1a : Détruisez le cube qui est sur le bloc et sur la pyramide

ou P2b : Détruisez le cube qui est sur le bloc et détruisez la pyramide

De même pour P2 :

P2a : Mettez le cube qui est sur le bloc et sur la pyramide sur le cube

P2b : Mettez le cube sur le bloc et mettez la pyramide sur le cube

On note deux choses :

- Ces phrases sont rendues ambiguës parce qu'un élément a subi une élision (la préposition pour P1a et P2a

le verbe pour P1b et P2b). Il semble qu'en général, à chacun des sens d'une phrase ambiguë, on peut faire correspondre une phrase non ambiguë et sémantiquement équivalente, ce qui revient à dire qu'on peut toujours éviter les ambiguïtés dans une phrase.

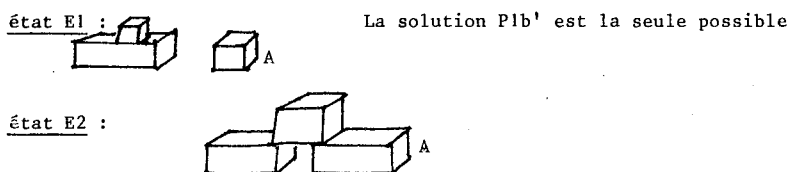
- si on prend, comme signification de "sur", qu'il y a contact entre les faces inférieures et supérieures des 2 objets concernés, ce qui exclut en particulier qu'un objet puisse être sur une pyramide, on voit que seuls P1b et P2b sont possibles.

On a donc deux possibilités :

- soit interdire les phrases ambiguës en obligeant l'utilisateur à employer la phrase non-ambiguë équivalente mais cela revient à supprimer une des caractéristiques fondamentales des langues naturelles.

- soit transmettre les différentes structures possibles d'une phrase ambiguë, le système se chargeant de déterminer quelle solution est la plus vraisemblable, ou la seule possible en examinant la base de données. Une phrase peut en effet être structurellement ambiguë sans l'être par rapport à une situation donnée, c'est-à-dire à l'état du monde au moment de l'émission de la phrase. Si on prend :

P1' : "Détruisez le cube qui est sur le bloc et A", il est clair qu'on a les deux significations P1a' et P1b' où "la pyramide" est remplacée par "A". On va prendre deux états différents du monde :



Ici les deux interprétations sont possibles, mais la solution P1a' est la plus vraisemblable, d'après la structure de la phrase (la conjonction unit en priorité les éléments les plus proches).

On peut aussi imaginer des cas où deux solutions sont également vraisemblables. Il faut cependant signaler un inconvénient de la transmission de l'ensemble des structures ambiguës. Il peut arriver, pour des phrases longues, un nombre très important de solutions, chaque facteur d'ambiguïté conduisant à une multiplication du nombre des solutions (par ex. 16 structures si la phrase compte 4 ambiguïtés à deux significations possibles). Le processus de calcul et de comparaison des différentes structures risque alors de se révéler extrêmement long.

Pour ce qui concerne la détection des ambiguïtés, la méthode pourrait être la suivante : il faudrait d'abord délimiter les cas où le problème pourrait se poser : dans tous les exemples qu'on a vus, il apparaît que c'est la présence de conjonctions qui est en cause. On peut aussi considérer le cas des GP qu'on peut attacher soit à un verbe, soit à un GN. Dans :

P4 : "Mettez le prisme sur le cube sur le bloc", on a deux sens

P4a : "Mettez le prisme qui est sur le cube sur le bloc"

P4b : "Mettez le prisme sur le cube qui est sur le bloc"

Mais en fait, on constate que ce sont les deux phrases non ambiguës qui sont de beaucoup les plus "naturelles" Aussi ce cas pourra-t-il ne pas être considéré.

Une fois délimitées les possibilités d'ambiguïté, il suffira de transformer les règles intéressées (par ex pour P3, les règles calculant le GP et les conjonctions d'objets) qui donneront en sortie les deux structures de phrase possibles, un sommet spécial dominant celles-ci.

CHAPITRE VIII

CONCLUSION

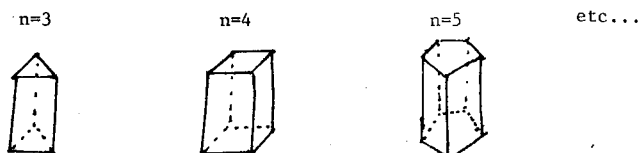
Nous examinerons dans ce chapitre dans quelle mesure le système que l'on vient de décrire pourrait être élargi. En comparant ses pouvoirs avec ceux du système de Winograd qui, pour le domaine d'études, en est le plus proche, on voit que ce dernier traite trois cas différents : les ordres demandant une modification du monde, les questions pour obtenir des renseignements sur l'état du monde et les informations nouvelles que le système intégrera à la connaissance du monde qu'il possède initialement. Dans notre système, seul le premier cas est traité. Nous nous intéresserons à trois possibilités d'élargissement des pouvoirs du système : d'abord l'utilisation d'un monde plus vaste, tant au point de vue des objets manipulés que des ordres ; ensuite l'intégration d'un système de question-réponse ; enfin la possibilité de donner des informations nouvelles au système. Pour chacune de ces possibilités, il faudra essayer de répondre aux questions suivantes :

- leur adjonction serait-elle utile au système ?
- l'implémentation serait-elle facile (en particulier compatible avec ce qui est mis en oeuvre pour la compréhension et l'exécution des commandes), ou non ?

1. ELARGISSEMENT DU DOMAINE D'ETUDE

Il serait tout à fait exclu de prévoir qu'on puisse étendre ce système de façon qu'il puisse manipuler des données se rapportant à un sujet quelconque. On se cantonnera dans le domaine des objets géométriques en voyant comment donner plus de puissance au système. Un des ajouts les plus simples est constitué par l'adjonction de nouvelles formes d'objets : la structure profonde n'aurait pas besoin d'être beaucoup modifiée et c'est uniquement dans le système graphique qu'il faudrait écrire une procédure pour chaque nouvelle forme autorisée. En se cantonnant dans le domaine des polyèdres, on pourrait distinguer deux types de formes : les "pyramides", extension des pyramides à base rectangulaire à une base polygonale quelconque et les "cylindres", extension des prismes. Le nombre n des côtés du polygone de base serait alors un des paramètres de la description des objets :

cylindre :



pyramide :



Un cas plus difficile est celui où, tout en restant dans le monde accessible, on utilise des actions pour lesquelles la structure profonde actuelle s'avère insuffisante. Considérons la phrase : "Faites tourner le cube A de 30 degrés autour de l'axe passant par les centres de B et de C".

Cette phrase pose un certain nombre de problèmes, surtout au point de vue de sa représentation du sens. Il faudra en effet introduire les concepts de rotation, d'angle, d'axe (outil permettant de manipuler les objets géométriques).

Le système déductif devra comporter des théorèmes traitant ces concepts. De plus, les primitives représentant les caractéristiques des objets devraient être complétées : il faudrait alors introduire l'orientation des objets par rapport aux trois axes de référence. Sur le plan de l'analyse du texte, il y aurait moins de difficultés : il faudrait introduire de nouvelles règles pour calculer la représentation des nouveaux concepts, mais la méthode resterait la même : calcul des objets, des modificateurs, des relations entre objets, des spécifications de l'action, etc...

2. SYSTEME QUESTION-REPONSE

2.1. UTILITE

Dans la mesure où le but qu'on s'est fixé est l'étude du français comme langage de commande, permettre de poser des questions apparaît comme un but secondaire, d'autant que la visualisation de la scène permet de se rendre compte immédiatement de l'état de celle-ci. Cependant, à cause précisément de ce moyen de contrôle immédiat, poser des questions permettrait de constater si le système a bien "compris" ce qu'on lui a demandé et c'est dans cette mesure que cette adjonction serait intéressante.

2.2. REPRESENTATION DES QUESTIONS

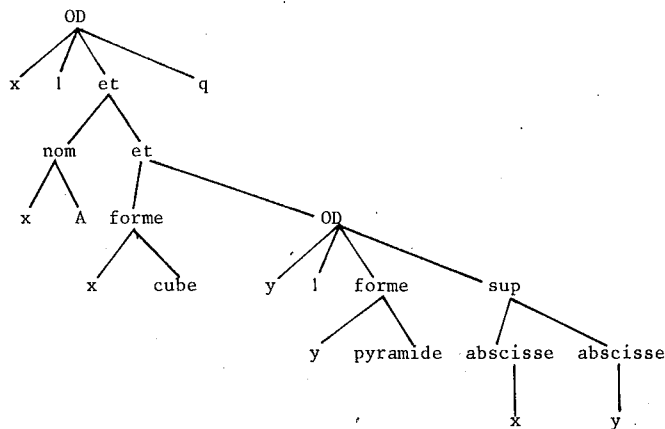
Pour les questions il est facile de se rendre compte que la représentation serait assez proche de la représentation des ordres. Prenons les exemples :

E1 : le cube A est-il à droite d'une pyramide ?

E2 : quel cube se trouve à droite d'une pyramide ?

E1 est une question "oui-non" et E2 une question "wh".

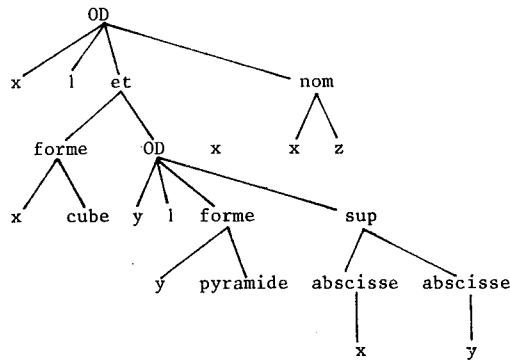
En ce qui concerne E1, on a la description d'un objet et il faut que le système déductif vérifie si l'ensemble des spécifications concernant cet objet est contradictoire ou non, ce qui donnera la réponse à la question (non s'il y a contradiction, par exemple si aucun cube ne s'appelle A, oui sinon). La forme de sortie sera donc :



L'élément q indiquera, sous une forme appropriée, qu'il s'agit d'une question oui-non.

Or, on remarque que c'est la représentation de "le cube A qui est à droite d'une pyramide". Le déducteur agira d'ailleurs dans les deux cas d'une manière analogue, recherchant dans la base de données un élément qui réponde aux spécifications indiquées. On remarque aussi que, quand le déducteur trouve un tel objet, il répond à la question E2 ce qui suggère, pour la représentation de celle-ci une représentation analogue : on aura $OD(x,n,p,q)$ ou p indiquera les propriétés que doit vérifier l'objet sur lequel porte la question et q l'élément questionné.

Si on considère que "quel cube" équivaut à "quel est le nom du cube", E2 sera représenté par :



Le système déductif unira la variable z avec le nom de l'objet trouvé. Le mode de représentation des données limitera de toutes façons la nature des questions que l'on pourra poser : on pourra demander des renseignements sur les caractéristiques (nom, forme, dimension,...) des objets, mais pas par exemple sur l'histoire de la scène (questions "quand" ou "pourquoi").

Su le plan de l'analyse du texte, l'essentiel de la grammaire générale sera utilisé. Il suffira d'ajouter quelques règles pour l'analyse du GN et pour l'analyse de la phrase globale.

2.3. GENERATION DES REPONSES

Dans un système général de QR, il faut un processus inverse de celui de l'analyse : à partir d'une structure profonde représentant la réponse il faudrait engendrer la phrase correspondante en utilisant d'abord une grammaire CETA de génération syntaxique, puis une grammaire de génération morphologique dans un système inverse de ATEF.

Mais dans la mesure où les éléments de réponse sont simples : nom, forme, longueur d'un objet, un système aussi complexe n'est pas nécessaire. les réponses : "A" à la question : "quel est le nom du plus gros cube ?", ou : "cube" à la question : "quelle est la forme de A ?" sont suffisamment explicites.

En revanche, un tel système serait nécessaire si on permettait au système d'intervenir pour obtenir certains renseignements, en particulier en cas d'ambiguïté non résoluble ou pour répondre à des questions plus complexes du genre : Pourquoi ... ?

Pour le système QR on peut donc dire que les difficultés ne seraient pas importantes mais que l'intérêt resterait limité par rapport au but que l'on s'est fixé.

3. AJOUT D'INFORMATIONS

C'est ici que se posent les problèmes les plus intéressants.

3.1. UTILITE

La donnée de nouvelles informations aurait pour but de permettre à l'utilisateur d'adapter le système général à l'application qu'il veut en faire. On peut en effet considérer que le monde accessible est un monde logique. L'utilisateur peut alors indiquer au système les "règles du jeu", c'est-à-dire à quelle logique répond le monde qu'il veut manipuler, qui peut être différent du monde géométrique considéré au départ.

Ce pourra être le monde "réel" : dans ce cas, il faudra indiquer que par exemple, deux objets pleins ne peuvent pas avoir d'intersection commune, qu'on ne peut rien poser sur une pyramide, qu'aucun objet ne peut tenir sur sa pointe, ... Ce pourra aussi être un monde "imaginaire" quelconque, dont on indiquera des contraintes différentes. Toutes ces contraintes seront données comme des énoncés en français et transformées par le système en théorèmes qui formalisent le nouveau monde et seront utilisés par la suite comme les autres théorèmes.

On peut aussi utiliser des énoncés pour définir le sens de mots qui n'ont pas d'interprétation précise à priori, comme on l'avait vu au chapitre IV pour des mots comme "à droite de" ou "long". On a vu que, par exemple, les "grandes" dimensions exprimées par des mots comme "long" ou "haut" étaient représentées par une comparaison avec une valeur standard appelée "grande" dont la valeur était choisie par le système déductif. Il serait intéressant que l'on puisse définir soi-même cette valeur en disant par exemple : "une dimension est grande si elle dépasse 500 unités".

L'effet sera de remplacer la valeur standard (par exemple 300) par la valeur indiquée. On pourrait même raffiner cette définition en tenant compte du fait que les dimensions n'ont pas le même sens pour toutes les applications (un minuscule éléphant reste plus gros qu'une énorme souris). On pourrait dire "un cube est long si son arrête mesure plus de 300 unités" et "une pyramide est haute si sa hauteur dépasse 600 unités". Dans ce cas la valeur standard "grande" dépendra à la fois de l'objet référencé et de la dimension (longueur, largeur et hauteur).

Cette méthode permettant de définir la sémantique du langage ne pourra toutefois pas s'appliquer sur l'ensemble du vocabulaire dont l'essentiel a un sens précis et inaltérable : par exemple, le fait que "sous" définit la relation inverse de "sur" ou le fait que "détruire" et "supprimer" sont synonymes. D'une manière générale la sémantique intégrée au système d'analyse ne pourra pas être modifiée.

Enfin, un dernier point méritant l'attention serait la possibilité de créer en quelque sorte des procédures qu'on puisse ensuite appeler de la même façon que, dans un langage de programmation, on peut écrire une procédure opérant un certain calcul qu'on appelle au cours du traitement avec des paramètres différents. La création de procédures dans notre cas correspondrait à la création de théorèmes pour le système déductif.

L'expression en français de telles procédures se ferait au moyen de phrases déclaratives opérant une description de processus (si on veut utiliser de façon simples des actions complexes) ou une description de figure (si on veut utiliser des figures complexes par simple appel de nom).

Par exemple, si on veut utiliser des figures de la forme "église" (voir au chapitre II quelle suite d'ordres peut générer une telle figure), on peut utiliser les phrases suivantes :

Une nef est une figure formée d'un bloc plus long que large et d'un prisme aussi long et aussi large que le bloc et posé juste sur lui.

Une flèche est une pyramide à base carrée.

Un clocher est un objet formé d'un bloc plus haut que large et d'une flèche posé juste sur lui.

Une église est une figure formée d'une nef et d'un clocher placé derrière la nef et contre celle-ci, avec les faces droites du clocher et de la nef alignées.

On voit que les deux figures suivantes représentent alors des églises.

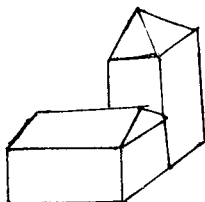


fig. 1

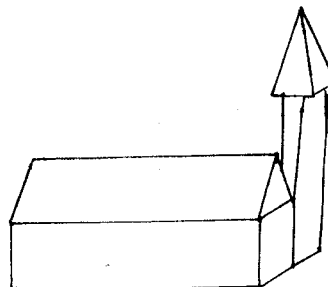


fig. 2

A partir de ces descriptions on pourrait créer directement des églises en spécifiant éventuellement certaines valeurs, de dimensions par exemple, jouant alors le rôle de paramètres de procédure.

Exemple : Créez une église dont le clocher ait une flèche très haute.

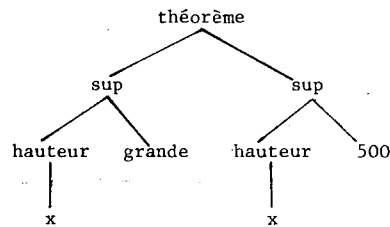
Il est évident qu'on ne pourra pas, dans un ordre, modifier des propriétés définies auparavant, par exemple dire : "Créez une église dont le clocher sera plus large que long", ce qui est contradictoire avec la définition d'un clocher, de même que dire "un cube plus long que large" est contradictoire.

3.2. REPRESENTATION DE L'INFORMATION

Les problèmes sont nombreux et nous ne ferons que les évoquer :

- il faudrait d'abord prévoir une structure profonde pour représenter les énoncés. Il est évident que la structure actuelle serait très insuffisante pour cela, puisque ces énoncés devraient être représentés sous la forme de théorèmes.

- il faudrait d'autre part arriver à "comprendre" les énoncés. Si on prend l'exemple "un objet est haut si sa hauteur dépasse 500 unités", on voit que cet énoncé est formé d'une partie nominative et d'une partie descriptive qu'il faudra transformer en les deux parties correspondantes d'un théorème, qu'on pourrait écrire :



- Enfin, pour les mots qu'on définit par des théorèmes et qui ne sont pas référencés dans les dictionnaires (comme église, flèche, clocher) le problème se poserait de leur analyse : pour calculer les variables qui les caractérisent et qui, pour les autres mots, sont obtenues lors de l'analyse morphologique grâce aux dictionnaires, on ne pourrait utiliser que le contexte même où ils sont définis : "une flèche est une pyramide ..." indique que flèche est du féminin et que ses variables sémantiques doivent être déduites de celle du mot "pyramide". Il faudrait donc écrire un système d'apprentissage capable de traiter les mots non encore définis et l'on sait tous les problèmes qui s'y attachent.

En conséquence, on voit que la possibilité d'ajouter de l'information qui ouvrirait des perspectives séduisantes se heurterait à des problèmes méritant une étude approfondie.

BIBLIOGRAPHIE

- [1] C. BOITET
Sémantique et traduction automatique : quelques remarques
Document GETA - G-3000-A, décembre 1974
- [2] J.M. BRUN
Sur le programme EUCLID : création, manipulation et visualisation de formes tri-dimensionnelles dans un langage géométrique à génération dynamique.
Thèse de 3ème cycle - Paris, 1972
- [3] J. CHAUCHE, P. GUILLAUME, M. QUEZEL-AMBRUNAZ
Le système ATEF
Document GETA - G-2600-A, Octobre 1972
- [4] J. CHAUCHE
Transducteurs et arborescences
Thèse de doctorat d'état, Université de Grenoble I, décembre 1974
- [5] J. CHAUCHE
Présentation du système CETA
Document GETA - G-3100-A, janvier 1975
- [6] A. COLMERAUER
Les systèmes-Q ou un formalisme pour analyser ou synthétiser des phrases sur ordinateur
Projet TAUM, Université de Montréal, janvier 1971
- [7] A. COLMERAUER, H. KANOUI, P. ROUSSEL, R. PASERO
Un système de communication homme-machine en français
Rapport de recherche, Université d'Aix-Marseille, juin 1973
- [8] GEORGETOWN UNIVERSITY
Machine translation research project
General report 1952-1963, paper N° 30, juin 1963
- [9] O.S. KULAGINA
Machine translation from french into russian
Problems of Cybernetics, vol. 3, Pergamon Press, 1959
- [10] O.S. KULAGINA, G.V. VAKULOVSKAYER
Experimental translation from french into russian using a strela computer
Problems of Cybernetics, vol. 2, Pergamon Press, 1959
- [11] P. ROUSSEL
PROLOG, manuel d'utilisation
Rapport interne, Université d'Aix-Marseille, septembre 1975
- [12] R.R. SCHANK
A conceptual dependency representation for a computer oriented semantics
Technical report n° cs 130, Stanford University, mars 1969
- [13] R.F. SIMMONS
Semantic networks : their computation and use for understanding english sentences
University of Texas, Austin, mai 1972

- [14] R.F. SIMMONS, J. SLOCUM
Generating english discourse from semantic networks
CACM Vol. 15 n° 10, p. 891-905, octobre 1972
- [15] B. VAUQUOIS
La traduction automatique à Grenoble
Document de linguistique quantitative n°24, Dunod, 1975
- [16] D. WARREN
Warplan : a system for generating plans
Meme n° 76, School of Artificial Intelligence, University of Edinburgh, juin 1974
- [17] T. WINOGRAD
Procedures as a representation for data in a computer program for understanding natural language
Artificial Intelligence Laboratory, M.I.T., février 1971
- [18] W. WOODS
Semantics for a question-answering system
Report n° NSF-19, Harvard University, Cambridge, 1967
- [19] W. WOODS
Transition networks grammars for natural language analysis
A.C.M. vol. 13 n° 10, p. 591-606, octobre 1970
- [20] Y. LALOUM
Rapport de DEA
Grenoble, septembre 1975

ANNEXES I ET II :

Pour chaque exemple, on indique successivement le texte d'entrée et l'arborescence résultante sous forme linéaire et sous forme développée.

ANNEXE I

EXEMPLE D'EXECUTION DE TEXTE

PHASE MORPHOLOGIQUE

FILE: FILETEX1 1 PI

CAMBRIDGE MONITOR SYSTEM

PAGE 001

CREIEZ UN GROS CUBE A1 DROITE DE LA PLUS GRESSE PYRAMIDE.

.1.ULTXT (.2.ULFRA (.3.ULOCC (.4.CREER) , .5.ULOCC (.6.UN) , .7.ULOCC (.8.GROSSEUR) , .9.ULOCC (.10.CUBE) , .11.ULOCC (.12.ABSCISSE) , .13.ULOCC (.14.LA , .15.LA) , .16.ULOCC (.17.PLUS) , .18.ULOCC (.19.GROSSEUR) , .20.ULOCC (.21.ULMCP (.22.PYRAMIDE , .23.)))))

SOMMET 1 : UL(ULTXT).

SOMMET 2 : UL(ULFRA).

SOMMET 3 : UL(ULOCC).

SOMMET 4 CREIEZ: UL(CREER),SEG(B),CV(IDPR1,IDPR3,IDPR4,IMPR1,IMPR3),NBR(PLU),CAT(VB),MOTE(IDPR,IMPR),PRS(2),ARG2(N),AUX(A).

SOMMET 5 : UL(ULOCC).

SOMMET 6 UN: UL(UN),SEG(INVA),GNR(MAS),NBR(SIN),CAT(DET),SOUCAT(ARTI).

SOMMET 7 : UL(ULOCC).

SOMMET 8 GROS: UL(GROSSEUR),SEG(B),GNR(MAS),NBR(SIN,PLU),CAT(ADJ),SEM(TAILLE),UL1(GRANDE),STGNE(SUP).

SOMMET 9 : UL(ULOCC).

SOMMET 10 CUBE: UL(CUBE),SEG(B),GNR(MAS),NBR(SIN),CAT(NMC),OBB(1).

SOMMET 11 : UL(ULOCC).

SOMMET 12 DE: UL(ABSCISSE),SEG(INVA),CAT(PRE),SEM(PLACE),UL1(GRANDE1),STGNE(SUP).

SOMMET 13 : UL(ULOCC).

SOMMET 14 LA: UL(LA),SEG(INVA),GNR(FEM),NBR(SIN),CAT(PRP),PRS(3),ROLE(OBJ).

SOMMET 15 LA: UL(LA),SEG(INVA),GNR(FEM),NBR(SIN),CAT(DET),SOUCAT(ARTD).

SOMMET 16 : UL(ULOCC).

SOMMET 17 PLUS: UL(PLUS),SEG(INVA),CAT(ADV),OP(ADD,PROD),ARG(AJC),STGNE(SUP).

SOMMET 18 : UL(ULOCC).

SOMMET 19 GROSSE: UL(GROSSEUR),SEG(B),GNR(FEM),NBR(SIN),CAT(ADJ),SEM(TAILLE),UL1(GRANDE),STGNE(SUP).

SOMMET 20 : UL(ULOCC).

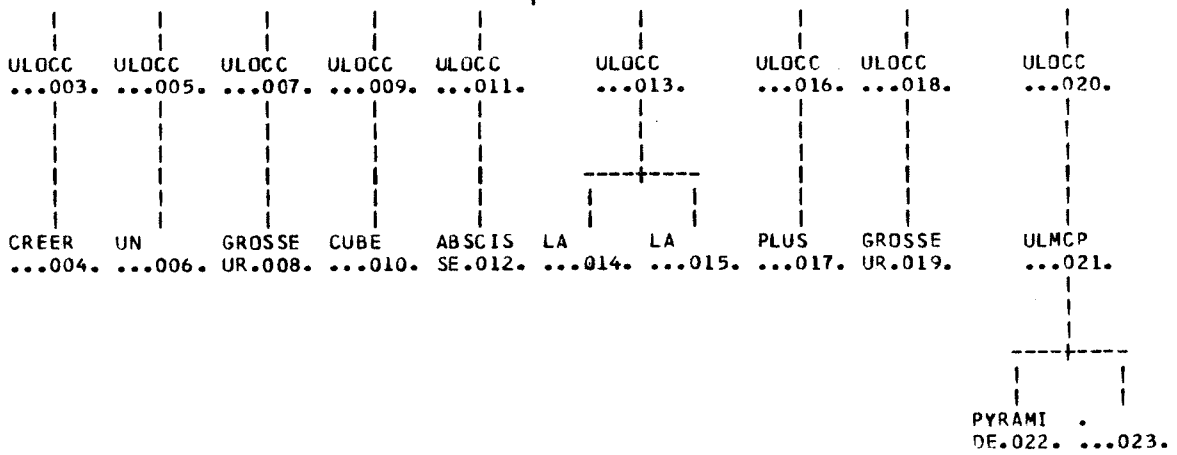
SOMMET 21 : UL(ULMCP).

SOMMET 22 PYRAMIDE.: UL(PYRAMIDE),SEG(B),GNR(FEM),NBR(SIN),CAT(NMC),OBB(1).

SOMMET 23 PYRAMIDE.: UL(.),CAT(PCT),CJ(ET),NC(1).

ULTXT
...001.

ULFRA
...002.



ANNEXE II

EXEMPLES D'EXECUTION DE TEXTES :

STRUCTURE PROFONDE

FILE: FILETEX1 34 P1
DETRUISEZ-LE.

CAMBRIDGE MONITOR SYSTEM

PAGE 001

.1.ULO (.2.ULO , .3.1 , .4.ULO (.5.ULO , .6.DERN , .7.ULO (.8.ULO , .9.ULO)) , .10.DESTR (.11.ULO)

SOMMET 1 : UL(ULO),ETQ(OD).

SOMMET 2 : UL(ULO),VF(+1).

SOMMET 3 : UL(1).

SOMMET 4 : UL(ULO),ETQ(REF).

SOMMET 5 : UL(ULO),VF(+1).

SOMMET 6 : UL(DERN).

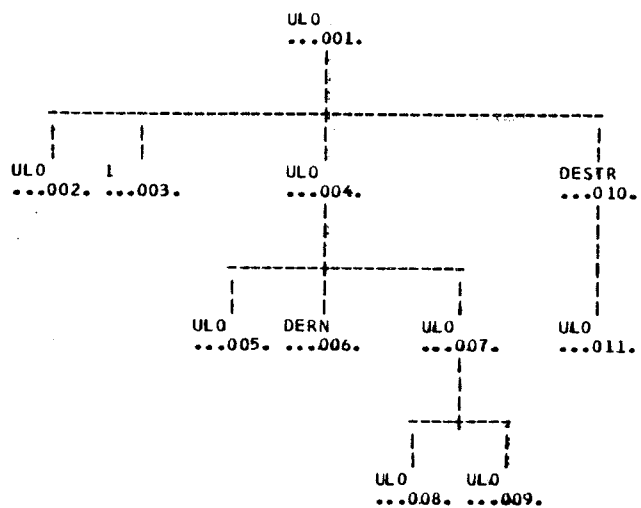
SOMMET 7 : UL(ULO),ETQ(GNRE).

SOMMET 8 : UL(ULO),VF(+1).

SOMMET 9 : UL(ULO),GNR(MAS).

SOMMET 10 : UL(DESTR).

SOMMET 11 : UL(ULO),VF(+1).



FILE: FILETEX1 I

P1

CAMBRIDGE MONITOR SYSTEM

PAGE 001

CREIEZ UN GROS CUBE A1 DROITE DE LA PLUS GROSSE PYRAMIDE.

.1.CREER (.2.ULO , .3.1 , .4.ULO (.5.ULO (.6.ULO , .7.CUBE) , .8.ULO (.9.ULO (.10.GROSSEUR (.11.ULO) , .12.ULO (.13.ULO (.14.ULO , .15.1 , .16.ULO (.17.ULO (.18.ULO , .19.PYRAMIDE) , .20.ULO (.21.ULO , .22.TOJS , .23.ULO (.24.ULO , .25.PYRAMIDE) , .26.ULO (.27.GROSSEUR (.28.ULO) , .29.GROSSEUR (.30.ULO))) , .31.ULO (.32.ABSCISSE (.33.ULO) , .34.ABSCISSE (.35.ULO))))))

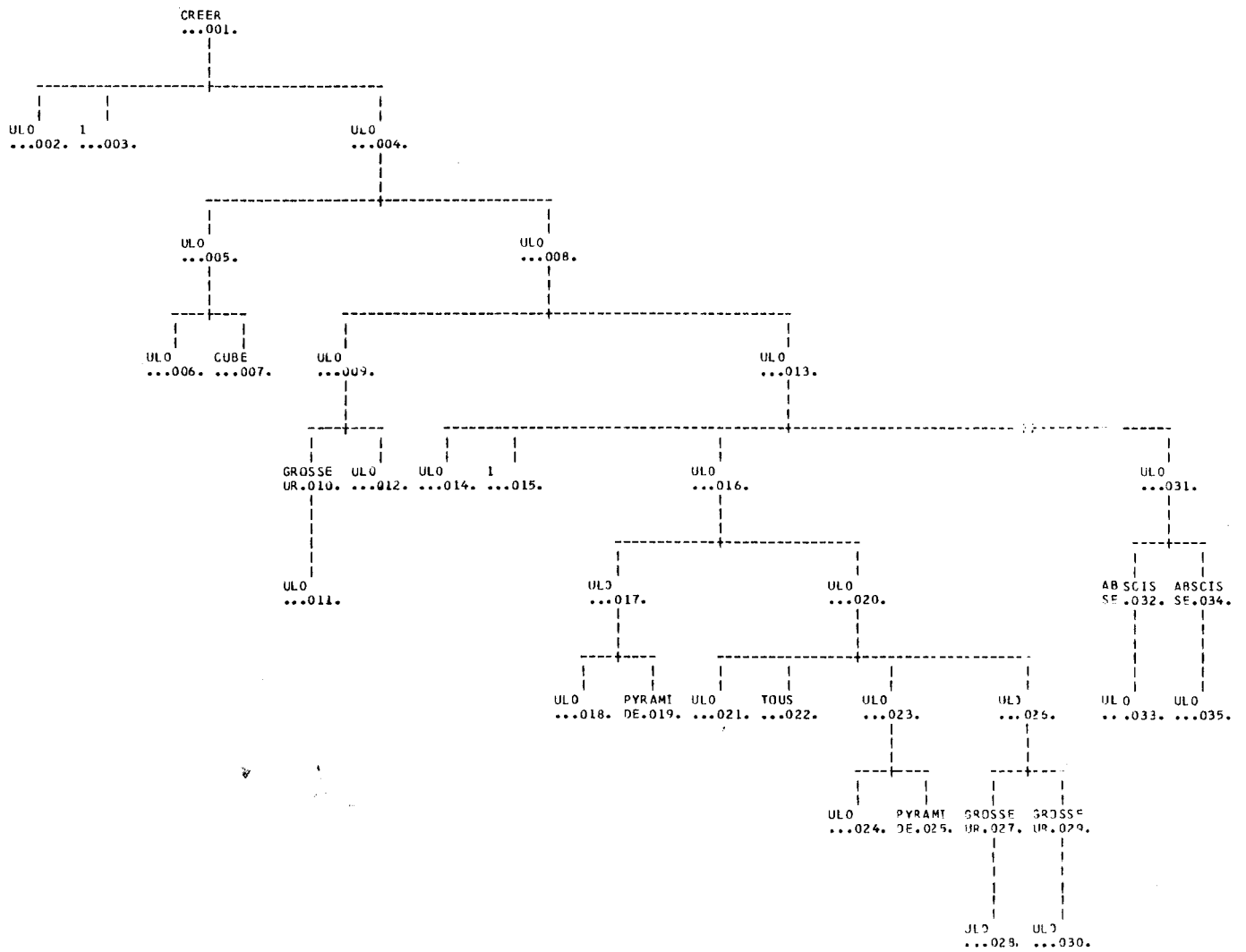
- SOMMET 1 : UL(CREER).
- SOMMET 2 : UL(ULO),VF(+3).
- SOMMET 3 : UL(1).
- SOMMET 4 : UL(ULO),CJ(ET).
- SOMMET 5 : UL(ULO),ETQ(NAT).
- SOMMET 6 : UL(ULO),VF(+3).
- SOMMET 7 : UL(CUBE).
- SOMMET 8 : UL(ULO),CJ(ET).
- SOMMET 9 : UL(ULO),SIGNE(SUR).
- SOMMET 10 : UL(GROSSEUR).
- SOMMET 11 : UL(ULO),VF(+3).
- SOMMET 12 : UL(ULO),UL1(GRANDE).
- SOMMET 13 : UL(ULO),ETQ(OD).
- SOMMET 14 : UL(ULO),VF(+2).
- SOMMET 15 : UL(1).
- SOMMET 16 : UL(ULO),CJ(ET).
- SOMMET 17 : UL(ULO),ETQ(NAT).
- SOMMET 18 : UL(ULO),VF(+2).
- SOMMET 19 : UL(PYRAMIDE).
- SOMMET 20 : UL(ULO),ETQ(OD).
- SOMMET 21 : UL(ULO),VF(+1).
- SOMMET 22 : UL(TOJS).
- SOMMET 23 : UL(ULO),ETQ(NAT).

RESULTAT DE L'EXECUTION , TEXTE : 1

SYSTEME C.E.T.A. 1

CODE LANGUE EX1

SOMMET 24 : UL(ULO),VF(+1).
SOMMET 25 : UL(PYRAMIDE).
SOMMET 26 : UL(ULO),SIGNE(SUP).
SOMMET 27 : UL(GROSSEUR).
SOMMET 28 : UL(ULO),VF(+2).
SOMMET 29 : UL(GROSSEUR).
SOMMET 30 : UL(ULO),VF(+1).
SOMMET 31 : UL(ULO),SIGNE(SUP).
SOMMET 32 : UL(ABSCISSE).
SOMMET 33 : UL(ULO),VF(+3).
SOMMET 34 : UL(ABSCISSE).
SOMMET 35 : UL(ULO),VF(+2).



FILE: FILETEX1 39

P1

CAMBRIDGE MONITOR SYSTEM

PAGE 001

DETRUISEZ LE CUBE QUI EST ENTRE LE BLOC ET LE PRISME QUI EST ENTRE "A" ET "B" .

.1.ULO (.2.ULO , .3.1 , .4.ULO (.5.ULO (.6.ULO , .7.CUBE) , .8.ULO (.9.ULO , .10.1 , .11.ULO (.12.)) (.13.ULO , .14.PRISME) , .15.ULO (.16.ULO , .17.1 , .18.ULO (.19.ULO , .20."B") , .21.ULO (.22.ULO , .23.1 , .24.ULO (.25.ULO , .26."A") , .27.ENTRE (.28.ULO , .29.ULO , .30.ULO))) , .31.ULO (.32.ULO , .33.1 , .34.ULO (.35.ULO , .36.DERN , .37.ULO (.38.ULO , .39.8LOC)) , .40.ENTRE (.41.ULO , .42.ULO , .43.ULO))) , .44.DESTR (.45.ULO))

- SOMMET 1 : UL(ULO),ETQ(UD).
- SOMMET 2 : UL(ULO),VF(+5).
- SOMMET 3 : UL(1).
- SOMMET 4 : UL(ULO),CJ(ET).
- SOMMET 5 : UL(ULO),ETQ(NAT).
- SOMMET 6 : UL(ULO),VF(+5).
- SOMMET 7 : UL(CUBE).
- SOMMET 8 : UL(ULO),ETQ(UD).
- SOMMET 9 : UL(ULO),VF(+3).
- SOMMET 10 : UL(1).
- SOMMET 11 : UL(ULO),CJ(ET).
- SOMMET 12 : UL(ULO),ETQ(NAT).
- SOMMET 13 : UL(ULO),VF(+3).
- SOMMET 14 : UL(PRISME).
- SOMMET 15 : UL(ULO),ETQ(UD).
- SOMMET 16 : UL(ULO),VF(+1).
- SOMMET 17 : UL(1).
- SOMMET 18 : UL(ULO),ETQ(NOM).
- SOMMET 19 : UL(ULO),VF(+1).
- SOMMET 20 : UL("B").
- SOMMET 21 : UL(ULO),ETQ(UD).
- SOMMET 22 : UL(ULO),VF(+2).
- SOMMET 23 : UL(1).

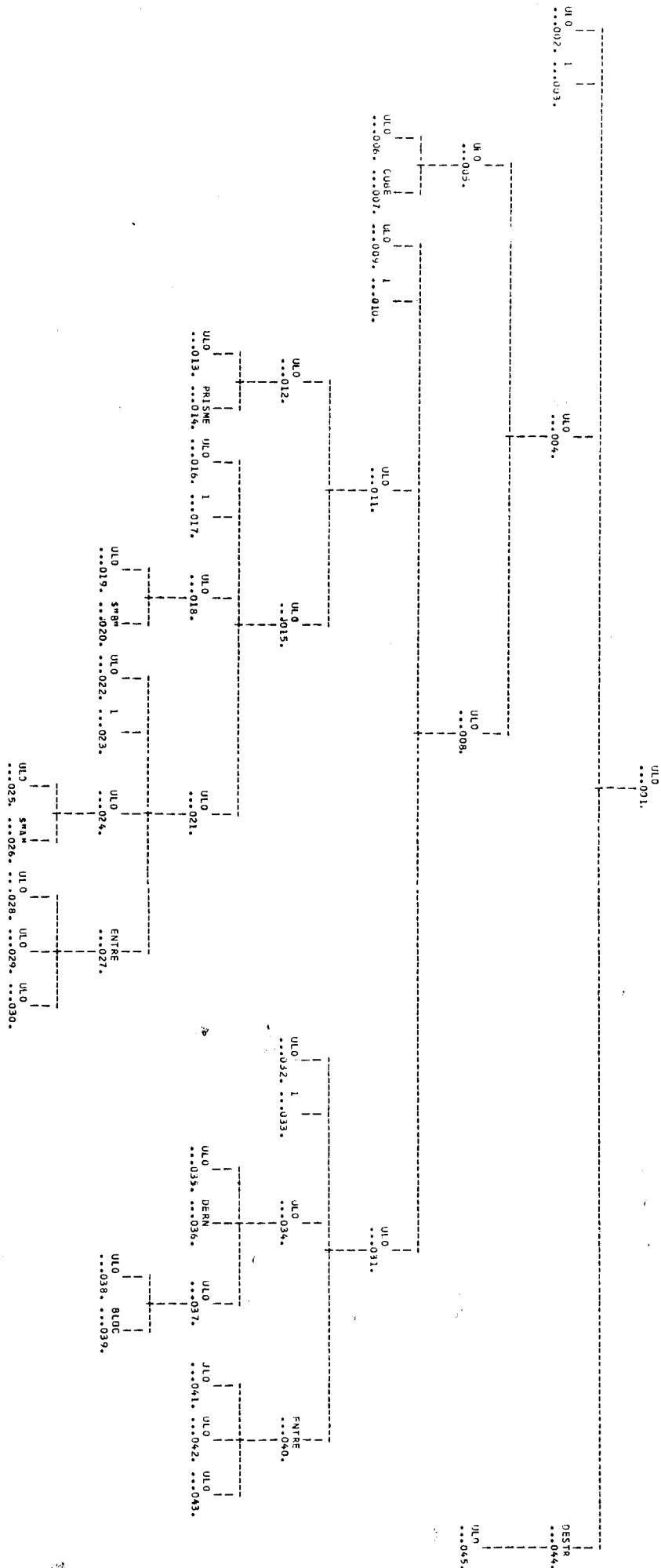
RESULTAT DE L'EXECUTION , TEXTE : 39

SYSTEME C.E.T.A. 1

PAGE 2

CODE LANGUE EX1

SOMMET 24 : UL(ULO),ETQ(NOM).
SOMMET 25 : UL(ULO),VF(+2).
SOMMET 26 : UL("A").
SOMMET 27 : UL(ENTRE).
SOMMET 28 : UL(ULO),VF(+3).
SOMMET 29 : UL(ULO),VF(+2).
SOMMET 30 : UL(ULO),VF(+1).
SOMMET 31 : UL(ULO),ETQ(OD).
SOMMET 32 : UL(ULO),VF(+4).
SOMMET 33 : UL(1).
SOMMET 34 : UL(ULO),ETQ(REF1).
SOMMET 35 : UL(ULO),VF(+4).
SOMMET 36 : UL(DERN).
SOMMET 37 : UL(ULO),ETQ(NAT1).
SOMMET 38 : UL(ULO),VF(+4).
SOMMET 39 : UL(BLOC).
SOMMET 40 : UL(ENTRE).
SOMMET 41 : UL(ULO),VF(+5).
SOMMET 42 : UL(ULO),VF(+4).
SOMMET 43 : UL(ULO),VF(+3).
SOMMET 44 : UL(DESTR).
SOMMET 45 : UL(ULO),VF(+5).



UL0 1
...002, ...003.

UL0 CUBE
...006, ...007, ...009, ...010.

UL0 1
...011.

UL0
...012.

UL0 PRISME
...013, ...014, ...016, ...017.

UL0 1
...018.

UL0
...015.

UL0
...008.

UL0
...001.

UL0 1
...032, ...033.

UL0
...034.

UL0 DEFN
...035, ...036.

UL0 1
...037.

UL0 BLOC
...038, ...039.

UL0
...040.

UL0 1
...041, ...042, ...043.

DEFN
...044.

UL0
...045.

FILE: FILETEX1 15 P1
INTERVERTISSEZ "A" ET "B" .

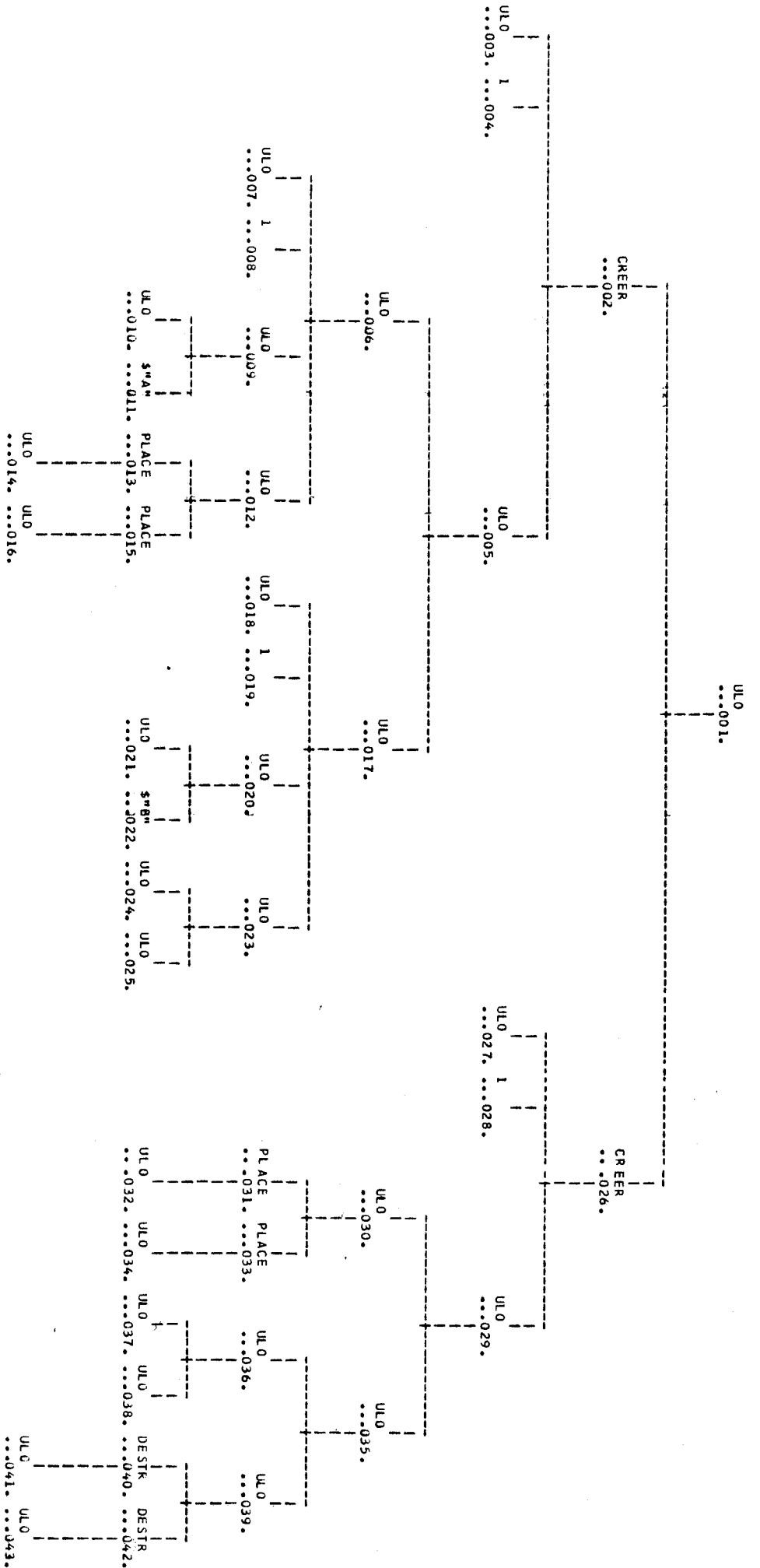
CAMBRIDGE MONITOR SYSTEM

PAGE 001

.1.ULO (.2.CREER (.3.ULO , .4.1 , .5.ULO (.6.ULO (.7.ULO , .8.1 , .9.ULO (.10.ULO , .11."A") , .12.ULO (.13. PLACE (.14.ULO) , .15.PLACE (.16.ULO))) , .17.ULO (.18.ULO , .19.1 , .20.ULO (.21.ULO , .22."B") , .23.ULO (.24. ULO , .25.ULO))) , .26.CREER (.27.ULO , .28.1 , .29.ULO (.30.ULO (.31.PLACE (.32.ULO) , .33.PLACE (.34.ULO) , .35.ULO (.36.ULO (.37.ULO , .38.ULO) , .39.ULO (.40.DESTR (.41.ULO) , .42.DESTR (.43.ULO))))))

- SOMMET 1 : UL(ULO),CJ(ET).
- SOMMET 2 : UL(CREER).
- SOMMET 3 : UL(ULO),VF(+4).
- SOMMET 4 : UL(1).
- SOMMET 5 : UL(ULO),CJ(ET).
- SOMMET 6 : UL(ULO),ETQ(OD).
- SOMMET 7 : UL(ULO),VF(+2).
- SOMMET 8 : UL(1).
- SOMMET 9 : UL(ULO),ETQ(NOM).
- SOMMET 10 : UL(ULO),VF(+2).
- SOMMET 11 : UL("A").
- SOMMET 12 : UL(ULO),SIGNE(EGAL).
- SOMMET 13 : UL(PLACE).
- SOMMET 14 : UL(ULO),VF(+4).
- SOMMET 15 : UL(PLACE).
- SOMMET 16 : UL(ULO),VF(+2).
- SOMMET 17 : UL(ULO),ETQ(OD).
- SOMMET 18 : UL(ULO),VF(+1).
- SOMMET 19 : UL(1).
- SOMMET 20 : UL(ULO),ETQ(NOM).
- SOMMET 21 : UL(ULO),VF(+1).
- SOMMET 22 : UL("B").
- SOMMET 23 : UL(ULO),ETQ(MEMOBJET).

SOMMET 24 : UL(ULO),VF(+4).
SOMMET 25 : UL(ULO),VF(+1).
SOMMET 26 : UL(CREER).
SOMMET 27 : UL(ULO),VF(+3).
SOMMET 28 : UL(1).
SOMMET 29 : UL(ULO),CJ(ET).
SOMMET 30 : UL(ULO),SIGNE(EGAL).
SOMMET 31 : UL(PLACE).
SOMMET 32 : UL(ULO),VF(+3).
SOMMET 33 : UL(PLACE).
SOMMET 34 : UL(ULO),VF(+1).
SOMMET 35 : UL(ULO),CJ(ET).
SOMMET 36 : UL(ULO),ETQ(MEMOBJET).
SOMMET 37 : UL(ULO),VF(+3).
SOMMET 38 : UL(ULO),VF(+2).
SOMMET 39 : UL(ULO),CJ(ET).
SOMMET 40 : UL(DESTRA).
SOMMET 41 : UL(ULO),VF(+2).
SOMMET 42 : UL(DESTRA).
SOMMET 43 : UL(ULO),VF(+1).



FILE: FILETEX1 33 : P1
ALLCNGER VERS LA DROITE "A" .

CAMBRIDGE MONITOR SYSTEM

PAGE 001

RESULTAT DE L'EXECUTION , TEXTE : 33

CODE LANGUE EX1

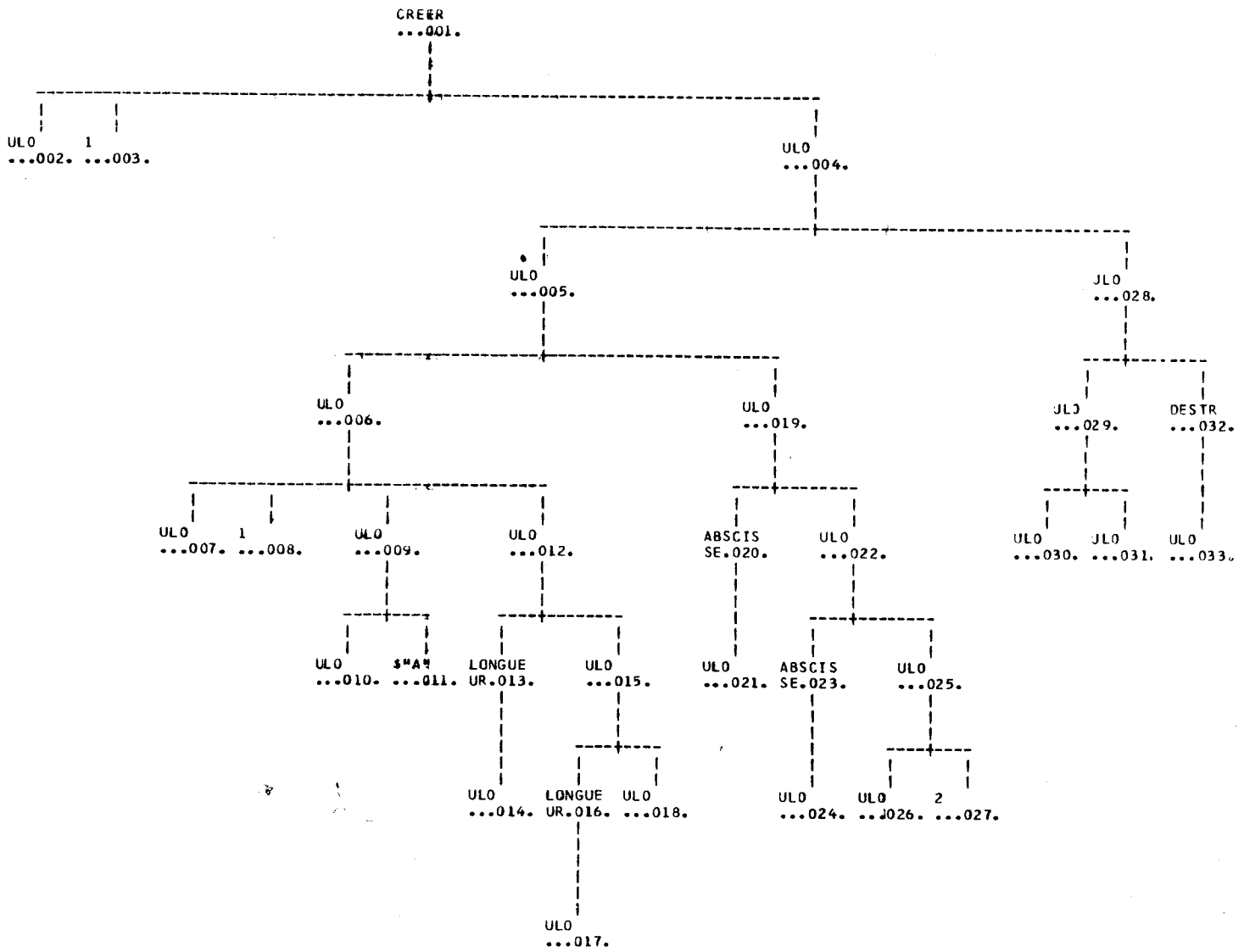
.1.CREER (.2.ULO , .3.1 , .4.ULO (.5.ULO (.6.ULO (.7.ULO , .8.1 , .9.ULO (.10.ULO , .11."A" , .12.ULO (.13.LONGUEUR (.14.ULO) , .15.ULO (.16.LONGUEUR (.17.ULO) , .18.ULO))) , .19.ULO (.20.ABSCISSE (.21.ULO) , .22.ULO (.23.ABSCISSE (.24.ULO) , .25.ULO (.26.ULO , .27.2))) , .28.ULO (.29.ULO (.30.ULO , .31.ULO) , .32.DESTR (.33.ULO)))

- SOMMET 1 : UL(CREER).
- SOMMET 2 : UL(ULO),VF(+2).
- SOMMET 3 : UL(1).
- SOMMET 4 : UL(ULO),CJ(ET).
- SOMMET 5 : UL(ULO),CJ(ET).
- SOMMET 6 : UL(ULO),ETQ(OD).
- SOMMET 7 : UL(ULO),VF(+1).
- SOMMET 8 : UL(1).
- SOMMET 9 : UL(ULO),ETQ(NOM)).
- SOMMET 10 : UL(ULO),VF(+1).
- SOMMET 11 : UL("A").
- SOMMET 12 : UL(ULO),SIGNE(EGAL).
- SOMMET 13 : UL(LONGUEUR).
- SOMMET 14 : UL(ULO),VF(+2).
- SOMMET 15 : UL(ULO),QP(ADD).
- SOMMET 16 : UL(LONGUEUR).
- SOMMET 17 : UL(ULO),VF(+1).
- SOMMET 18 : UL(ULO),ULI(MOYENNE).
- SOMMET 19 : UL(ULO),SIGNE(EGAL).
- SOMMET 20 : UL(ABSCISSE).
- SOMMET 21 : UL(ULO),VF(+2).
- SOMMET 22 : UL(ULO),QP(ADD).
- SOMMET 23 : UL(ABSCISSE).

RESULTAT DE L'EXECUTION , TEXTE : 33

CODE LANGUE EX1

- SOMMET 24 : UL(ULO),VF(+1).
- SOMMET 25 : UL(ULO),QP(QUOT).
- SOMMET 26 : UL(ULO),UL1(MOYENNE).
- SOMMET 27 : UL(2).
- SOMMET 28 : UL(ULO),CJ(ET).
- SOMMET 29 : UL(ULO),ETQ(MEMOBJET).
- SOMMET 30 : UL(ULO),VF(+2).
- SOMMET 31 : UL(ULO),VF(+1).
- SOMMET 32 : UL(DESTRA).
- SOMMET 33 : UL(ULO),VF(+1).



FILE: FILETEX1 18 P1

CAMBRIDGE MONITOR SYSTEM

PAGE 001

CREEZ UN PRISME 2 FOIS MOINS HAUT QUE LARGE, PAS LONG, D'ABSCISSE DU CENTRE
200 ET DE NOM "A" .

.1.CREER (.2.ULO , .3.1 , .4.ULO (.5.ULO (.6.ULO , .7.PRISME) , .8.ULO (.9.ULO (.10.HAUTEUR (.11.ULO) , .12.ULO (.13.LARGEUR (.14.ULO) , .15.21) , .16.ULO (.17.NON (.18.ULO (.19.LONGUEUR (.20.ULO) , .21.ULO) , .22.ULO (.23.ULO (.24.ABSCISSE (.25.ULO) , .26.200) , .27.ULO (.28.ULO , .29."A"))))))

SOMMET 1 : UL(CREER).
 SOMMET 2 : UL(ULO),VF(+1).
 SOMMET 3 : UL(1).
 SOMMET 4 : UL(ULO),CJ(ET).
 SOMMET 5 : UL(ULO),ETQ(NAT1).
 SOMMET 6 : UL(ULO),VF(+1).
 SOMMET 7 : UL(PRISME1).
 SOMMET 8 : UL(ULO),CJ(ET).
 SOMMET 9 : UL(ULO),SIGNE(EGAL).
 SOMMET 10 : UL(HAUTEUR).
 SOMMET 11 : UL(ULO),VF(+1).
 SOMMET 12 : UL(ULO),OP(QUOT1).
 SOMMET 13 : UL(LARGEUR).
 SOMMET 14 : UL(ULO),VF(+1).
 SOMMET 15 : UL(2).
 SOMMET 16 : UL(ULO),CJ(ET).
 SOMMET 17 : UL(NON).
 SOMMET 18 : UL(ULO),SIGNE(SUP).
 SOMMET 19 : UL(LONGUEUR).
 SOMMET 20 : UL(ULO),VF(+1).
 SOMMET 21 : UL(ULO),UL1(GRANDE).
 SOMMET 22 : UL(ULO),CJ(ET).
 SOMMET 23 : UL(ULO),SIGNE(EGAL).
 SOMMET 24 : UL(ABSCISSE).

RESULTAT DE L'EXECUTION , TEXTE : 18

SYSTEME C.E.T.A. 1

PAGE 2

CODE LANGUE EX1

SOMMET 25 : UL(ULO),VF(+1).
SOMMET 26 : UL(200).
SOMMET 27 : UL(ULO),ETQ(NGM).
SOMMET 28 : UL(ULO),VF(+1).
SOMMET 29 : UL("A").

