INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

SN 81/173 A

THESE

Présentée à l'Institut National Polytechnique de Lorraine
en vue d'obtenir le Grade de

DOCTEUR D'ÉTAT MENTION SCIENCES

(Informatique)

par

Eugène CHOURAQUI

UNIVERSITE NANCY BIBLIOTHEQUE

CONTRIBUTION A L'ÉTUDE THÉORIQUE DE LA REPRÉSENTATION DES CONNAISSANCES

LE SYSTÈME SYMBOLIQUE ARCHES

Service Commun de la Documentation
INPL
Nancy-Brabois

Soutenue le 16 Oct. 1981 devant la commission d'examen

D 136 036492 8

Membres du Jury

Président :	**C. PAIR**
Examinateurs :	**M. BORILLO**
	D. COULON
	J.-B. GRIZE
	J.-L. LAURIÈRE
	A. LENTIN
Invité :	**M. GRIFFITHS**

136036492 8

INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

(T) 1981 CHOURAQUI, E

THESE

Présentée à l'Institut National Polytechnique de Lorraine
en vue d'obtenir le Grade de

DOCTEUR D'ÉTAT MENTION SCIENCES

(Informatique)

par

Eugène CHOURAQUI

CONTRIBUTION A L'ÉTUDE THÉORIQUE DE LA REPRÉSENTATION DES CONNAISSANCES

LE SYSTÈME SYMBOLIQUE ARCHES

Soutenue le 16 Oct. 1981 devant la commission d'examen

Membres du Jury

Président : C. PAIR
Examinateurs : M. BORILLO
D. COULON
J.-B. GRIZE
J.-L. LAURIÈRE
A. LENTIN
Invité : M. GRIFFITHS

à tous les miens,

Je tiens à exprimer ma très grande reconnaissance à Monsieur Claude PAIR pour l'intérêt qu'il a porté à ce travail en lui donnant une orientation décisive, et dont la confiance et les critiques constructives m'ont été indispensables pour le mener à bien. Je le remercie de l'honneur qu'il me fait en acceptant la présidence du jury.

Je voudrais témoigner ma profonde gratitude à Monsieur Mario BORILLO qui n'a cessé de prodiguer à mon endroit conseils et encouragements dans le cadre des voies de recherches qu'il a tracées sur la formalisation des raisonnements.

Mes remerciements vont également à Monsieur Daniel COULON dont les travaux en représentation des connaissances et compréhension du langage naturel m'ont toujours stimulé ; je suis très heureux de sa présence dans ce jury.

Les travaux de Monsieur Jean-Baptiste GRIZE sur la logique naturelle et l'analogie ont influencé mes propres recherches dans ce secteur d'activité ; qu'il trouve ici l'expression de ma profonde reconnaissance.

Je remercie très vivement Monsieur Jean Louis LAURIERE pour la confiance qu'il m'a témoignée en me faisant l'honneur d'examiner ce travail ; ses recherches en intelligence artificielle et celles de son équipe m'ont beaucoup appris.

Je suis particulièrement honoré de la présence dans ce jury de Monsieur André LENTIN dont les travaux ont contribué à nourrir ma réflexion scientifique.

Monsieur Michael Griffiths, nommé récemment directeur du L.I.S.H., me fait l'honneur de participer à ce jury ; je le remercie très vivement de la confiance qu'il me témoigne et de l'intérêt qu'il porte à mes travaux de recherche.

Enfin, un grand merci à tous mes amis et collègues du LISH à qui je dois beaucoup ; et plus particulièrement à Messieurs Louis BOURRELLY et Luis FARIÑAS DEL CERRO pour toutes les discussions très enrichissantes que nous avons eues depuis fort longtemps déjà.

Mesdames Christiane GINOUX et Simone GIRIER ont dactylographié ces pages et assuré la réalisation matérielle de cet ouvrage avec beaucoup de patience et de soin. Qu'elles trouvent ici l'expression de toute ma reconnaissance.

AVANT-PROPOS

Les travaux que nous exposons dans cet ouvrage ont été développés et réalisés dans le cadre des recherches conduites au Laboratoire d'Informatique pour les Sciences de l'Homme (L.I.S.H.) à propos de la représentation informatique des connaissances et la formalisation des raisonnements. Cet ouvrage, composé essentiellement de neuf chapitres, présente la description formelle d'un système symbolique particulier de représentation et de traitement de connaissances, le système ARCHES.

La conception de ce système procède non seulement de l'examen de plusieurs études de cas relatives à des domaines de connaissances réelles (comme les sciences de l'homme par exemple), mais aussi de l'analyse de travaux conduits en intelligence artificielle sur la représentation des connaissances (réseaux sémantiques, frames, systèmes experts, etc.).
Plus précisément ARCHES, comme tout système formel, est formé de deux composantes interdépendantes. La première composante est relative aux modalités de représentation de connaissances qui sont déterminées par le langage objet du système ARCHES et l'organisation algébrique de ses éléments (concepts, individus, graphes de résolution, termes descriptifs, connecteurs, descriptions, etc.). Ainsi ce système permet de représenter tout ensemble de faits réels (objets de la culture matérielle, assertions factuelles, évènements, etc.) dont la description et l'organisation sont adéquates à son architecture. La deuxième composante est relative à l'activité inférentielle de ARCHES dont l'objectif est d'obtenir de nouvelles connaissances - par résolution de problèmes - à partir des faits qui ont été enregistrés dans la base de connaissances correspondante . Elle permet de mettre en oeuvre deux types de raisonnement : le raisonnement déductif pour lequel un principe de résolution a été défini ; et le raisonnement analogique fondé sur un modèle analogique particulier qui rend satisfiable le système ARCHES.

Par ailleurs nous utilisons dans cet ouvrage un ensemble de conventions qui sont décrites ci-après :
Les mots ou groupes de mots, qui véhiculent une information essentielle pa rapport au contexte où ils se trouvent, sont représentés par des chaînes de caractères italiques.
Dans le texte les mots ou groupes de mots peuvent faire référence à un ou plusieurs paragraphes dont les numéros sont mis entre parenthèses ; quand le chapitre n'est pas spécifié les paragraphes appartiennent au même chapitre que les mots ou groupes de mots.
Enfin les renvois bibliographiques sont représentés dans le texte par des numéros mis entre accolades, ces numéros repérant dans la bibliographie les auteurs classés par ordre alphabétique .

CHAPITRE PREMIER.

PRESENTATION DU CADRE DE NOS RECHERCHES

I. INTRODUCTION.

Nos travaux de recherche ont été développés dans un contexte que la question suivante permet de cerner : Quels sont et comment mettre en oeuvre les dispositifs d'analyse, de formalisation, de traitement et d'instrumentation des données qui contribuent à résoudre des problèmes que peuvent se poser des spécialistes à propos de domaines de connaissances particuliers ? Cette question les situe dans un nouveau secteur d'activités que d'aucuns appellent les *sciences cognitives* ; en étroite interdépendance, les champs d'investigation les plus importants, qui composent ce secteur scientifique et technique, sont ceux qui ont trait à la représentation des connaissances, à la compréhension du langage naturel, aux systèmes intelligents de questions-réponses, à la simulation du raisonnement humain, et enfin à la résolution de problèmes. Les sciences cognitives se trouvent ainsi à la confluence de disciplines diverses telles que la science des ordinateurs, la logique, les mathématiques, la linguistique, l'épistémologie, les domaines d'application, etc. Cependant les équipes, le plus souvent pluridisciplinaires, qui conduisent des recherches dans ces différents champs, les abordent sous des points de vue et des éclairages particuliers qui définissent leur cadre de référence.

Notre recherche s'articule principalement autour de deux champs d'investigation : la représentation des connaissances et la résolution de problèmes. Elle a pour objet de concevoir un *système symbolique particulier de représentation et de traitement de connaissances* - le système ARCHES - capable de véhiculer les données relevant de domaines empiriques réels et de résoudre une classe spécifique de problèmes formulés à leur propos (voir § V.). Les modalités de construction de ce système puisent ses sources non seulement dans les recherches conduites au LISH sur les problèmes méthodologiques soulevés par l'introduction de l'informatique dans certaines disciplines des sciences de l'homme, mais aussi dans les travaux entrepris en intelligence artificielle à propos de la représentation des connaissances et de la résolution de problèmes. Un examen approfondi de ces différentes études nous permet de formuler quelques questions essentielles qui déterminent le cadre de référence de notre travail : Quelle est la nature des domaines de connaissances investigués ? Quelles sont les hypothèses qui fondent les méthodes formelles mises en oeuvre pour représenter et traiter ces connaissances ? En d'autres termes quels domaines symboliques vise-t-on ? Enfin quels sont les types de rapports qui existent entre les domaines de connaissances et les domaines symboliques ?

Dans ce qui suit nous situons notre travail par rapport à ces différentes interrogations.

II. NATURE DES DOMAINES DE CONNAISSANCES.

Nous analysons ici certaines des caractéristiques des domaines de connaissances qui ont orienté et influencé notre étude. Plus précisément elles concernent les types de domaines investigués, la manière dont les informations sont manipulées et véhiculées par les spécialistes de ces domaines, et enfin le cadre méthodologique sur lequel prennent appui ces mêmes spécialistes pour résoudre les problèmes qu'ils se posent.

II.1. Type de domaines visés.

Notre travail se situe sur l'axe de recherche qui a pour objet essentiel la représentation et le traitement de connaissances indispensables à la résolution de classes de problèmes relevant de *domaines empiriques et réels* qui s'inscrivent dans un contexte caractérisé par l'absence évidente de théorie ; mais pour lesquels ont été obtenus des résultats intéressants sur le plan méthodologique, comme le montre {15} à propos de certaines disciplines relevant des sciences humaines et sociales (considérées comme des domaines empiriques particuliers). Nos travaux antérieurs, qui sont à la source de notre recherche actuelle, ont précisément porté sur des études de cas relatives à de tels domaines, comme par exemple celles exposées dans {20}, {36}, {37} et {38}. Ces domaines possèdent deux caractéristiques principales. D'une part ils sont dotés d'un *savoir empirique* représenté par un ensemble de connaissances produites par les spécialistes des domaines concernés (on ne saurait décrire un quelconque phénomène sans se référer au moins implicitement à un état donné des connaissances). Ce savoir constitue un réservoir d'informations dont l'utilisation contribue à résoudre tout nouveau problème. D'autre part il existe un *savoir faire empirique* défini par un ensemble de méthodes mises en oeuvre, le plus souvent de manière intuitive et/ou implicite, par ces mêmes spécialistes à propos de la formulation et de la résolution des problèmes qu'ils se posent. Savoir et savoir faire empiriques leur permettent ainsi non seulement de formuler de nouveaux problèmes - producteurs de connaissances -, mais aussi d'élaborer des corps d'hypothèses comme solutions possibles de ces problèmes (voir par exemple l'utilisation raisonnée de ces deux modes de connaissances dans {18} à propos d'une étude sur le commerce maritime antique en méditerranée occidentale).

Cette situation de recherche, qui consiste à opérer sur des domaines réels pour résoudre de "vrais" problèmes (comme par exemple en physique ou en chimie, mais il s'agit ici des sciences de l'homme), a la vertu de montrer clairement la richesse et la complexité des informations qu'il est nécessaire d'analyser et de formaliser pour constituer les bases de connaissances aptes à répondre aux objectifs fixés. C'est dans cette perspective que se situe notre travail. Cette situation est sensiblement différente de celle rencontrée dans la plupart des travaux développés dans le secteur de l'intelligence artificielle. Généralement ces travaux ne prennent en compte que des univers empiriques très réduits, et le plus souvent arbitraires, qui conduisent à la formulation de "Toy-problems", comme le soulignent {8} et {27}. Néanmoins il y a de remarquables exceptions, comme par exemple le système MYCIN qui est un système expert d'aide au diagnostic médical (pour une analyse détaillée de ce système voir {124} ; pour une présentation plus générale des systèmes de production se reporter à {47} et {92}, voir également § III.2.5.).

II.2. Notion de langages naturels spécialisés.

Toute théorie formelle est une construction méthodique et organisée qui permet de représenter, de décrire et de traiter ses objets au moyen d'un langage conventionnel particulier. En l'absence de théorie bien établie décrivant et expliquant les domaines empiriques visés, les phénomènes complexes - objets réels de ces univers - ne peuvent pas être manipulés par de tels langages. Cependant la pratique de ces domaines montre que la manipulation de ces objets s'effectue à travers une utilisation spécialisée du langage naturel. Les praticiens ont progressivement élaboré des systèmes de communication fondés sur des utilisations particulières de sous-ensembles du langage naturel, dont les analyses font apparaître que les rapports formes/sens de la terminologie et de la phraséologie employées tendent vers une relative stabilité au fur et à mesure du

développement des connaissances (voir par exemple dans {78} l'analyse d'un vocabulaire spécialisé relatif à la description d'objets particuliers). Le langage naturel peut ainsi être considéré comme le langage du savant qui, dans son domaine, a précisé le vocabulaire en éliminant un grand nombre d'ambiguïté et/ou en redéfinissant de nouveaux termes ; et qui a déterminé les principales formes de réthorique utilisées pour enchaîner les propositions et développer l'argumentation. Sous cet angle, le langage du savant est un *langage naturel spécialisé* qui s'apparente très étroitement au langage naturel dont il est issu en ce sens qu'ils ont même support syntaxique (les phrases du langage spécialisé sont réconnues par au moins une grammaire du langage naturel). Cependant, il reste difficilement compréhensible par les non spécialistes du domaine concerné.

Si pour des champs construits et bien formalisés comme la physique ou la chimie ces langages existent et ont donné naissance à des langages artificiels, la littérature montre que de nombreuses autres disciplines - comme par exemple l'économie, la gestion des entreprises, les sciences juridiques, l'archéologie - se trouvent dans une situation où des langages conventionnels commencent à être élaborés ({19}, {34}, {63}, {84}). Cependant il serait vain de prétendre et de conclure que de tels langages possèdent les caractères de régularité et de reproductibilité inhérents aux langages artificiels. Au sein d'une même discipline et pour les mêmes objectifs, des divergences demeurent et sur l'axe paradigmatique et sur l'axe syntagmatique. Elles peuvent exprimer des éléments de sens non explicités, des modifications contextuelles de la signification des objets manipulés, des points de vue théoriques différents, des écoles de pensée opposées, etc. Cette conjoncture est d'autant plus marquée que les débats scientifiques sont plus vifs et/ou les niveaux d'élaboration des méthodes développées plus faibles.

Ainsi selon notre point de vue, deux composantes interdépendantes essentielles caractérisent les langages spécialisés : /1/ la *sémantique* dont le degré de spécialisation et de régularité dépend des domaines étudiés ; c'est la compréhension des messages émis qui devient essentielle à la place de la syntaxe, et elle est fonction des domaines particuliers de connaissances (pour des études de cas voir par exemple {17}, {36} et {38} ; pour des points de vue comparables voir par exemple {121}) ;

/2/ les *formes d'argumentation* qui déterminent les modes de raisonnement discursifs élémentaires utilisés par les praticiens de tels domaines ; c'est la composition de ces modes de raisonnement qui contribue à résoudre les problèmes qu'ils se posent (voir par exemple {16}, {18}, {22} {37} et {41}). Notre objectif est d'intégrer et de formaliser dans notre travail certains aspects de ces deux composantes. Des hypothèses précises sur leur choix permettront de délimiter la classe des faits et des problèmes qui sont adéquats à notre système de représentation et de traitement de connaissances (voir chapitre deuxième).

II.3. Nécessité de l'expérimentation.

A la différence des sciences exactes qui construisent entièrement leurs objets, les problèmes posés à propos des domaines empiriques portent sur des entités réelles qui leurs sont extérieures. De ce point de vue, nous nous trouvons dans une situation comparable à celle des sciences d'observation où la notion d'expérimentation est essentielle, comme le souligne {35}. Il serait vain, en effet, d'imaginer de sérieux progrès cognitifs dans les recherches portant sur des domaines empiriques si l'expérience, qui permet une observation et une analyse finalisées de la réalité, ne constitue pas l'élément moteur de la démarche scientifique mise en oeuvre dans un tel contexte. La construction de systèmes hypothético-déductifs, constitutive de cette démarche, définit le cadre formel pour la formulation et la résolution de problèmes particuliers.
Ces formulations et résolutions de problèmes sont déterminées, au cours des processus expérimentaux, par des chaînes argumentatives composées d'ensembles organisés et cohérents d'étapes où des raisonnements de nature très diverse sont développés : justification de l'intérêt cognitif des problèmes posés vis à vis des domaines empiriques concernés, nature et choix des hypothèses visant à les résoudre, choix des méthodes de résolution et analyse de leur adéquation par rapport aux phénomènes étudiés, définition des critères de sélection des données sur lesquelles doivent être effectués les traitements et constitution des bases de connaissances correspondantes, exécution des opérations organisant les traitements, interprétation des résultats, validation des hypothèses, etc. (pour des études particulières ou systématiques consacrées à ces questions, voir par exemple {15}, {18} et {37}).

Ainsi la production de connaissances nouvelles mobilise également des compétences très diversifiées. Dans ce contexte expérimental, notre recherche est essentiellement concernée par les aspects opératoires des raisonnements. Plus précisément, elle s'attache à construire des représentations formelles de mécanismes élémentaires d'obtention d'informations nouvelles, non enregistrées à priori dans les bases de connaissances, mais qui sont évidemment contenues potentiellement dans l'information enregistrée (voir par exemple à ce sujet {10} et {120}).

II.4. Caractéristiques fondamentales des domaines de connaissances visés.

Cette analyse sommaire, relative aux types de phénomènes que nous souhaitons représenter et traiter dans notre système, montre que cinq critères principaux caractérisent les domaines de connaissances à partir desquels notre recherche s'est développée :

α) Les domaines de connaissances sont des domaines empiriques réels, non théorisés ;

β) ils sont munis d'un savoir et d'un savoir-faire empiriques ;

γ) ils sont véhiculés par des langages naturels spécialisés ;

δ) ils sont le siège de problèmes réels, producteurs de connaissances nouvelles ;

ε) les modalités de leur investigation sont déterminées dans le cadre d'un schéma hypothético-déductif.

III. NATURE DES DOMAINES SYMBOLIQUES.

III.1. L'influence du recours à l'informatique.

Notre situation de recherche tend à articuler et à composer l'utilisation et les méthodes de l'ordinateur avec d'autres méthodes scientifiques et techniques (issues de disciplines connexes, voir § I.), dont l'ensemble vise à définir un cadre méthodologique cohérent et précis apte à décrire des systèmes de résolution de problèmes formulés à propos de domaines de connaissances empiriques et réels (voir § II.). Dans ces conditions comment s'insèrent les méthodes informatiques et leur instrumentation sur des matériels déterminés dans l'ensemble des moyens mis à la disposition des praticiens souhaitant exercer leurs activités dans ce contexte ? La réponse à cette question dépend des points de vue sous lesquels on se place pour analyser le rôle et l'influence des ordinateurs dans leur contribution à découvrir des phénomènes nouveaux. Selon nous, deux aspects fondamentaux et complémentaires de la science des ordinateurs ont concouru à clarifier et à modifier sensiblement la manière de représenter et de traiter les phénomènes auxquels sont confrontés de tels praticiens.

D'abord l'informatique est une *discipline empirique*. Tout système de programmes n'a pas de réalité physique en dehors des classes de phénomènes et de problèmes pour lesquelles il a été construit. C'est sa mise en oeuvre effective dans des conditions déterminées (données qui lui sont soumis, résultats qu'il peut fournir, influence de l'environnement sur son comportement, etc.) qui permet de tester expérimentalement ses qualités et ainsi de valider les hypothèses qui ont présidé à sa conception et à son implémentation. Ce premier aspect de la science des ordinateurs renforce notre situation de recherche dans laquelle il est précisément nécessaire de conduire selon une démarche expérimentale les travaux développés dans les domaines empiriques (voir § II.3.).

Mais l'informatique est aussi la *science de la manipulation formelle de symboles* ; science qui puise ses sources en premier lieu dans l'expérimentation (un point de vue comparable est développé dans {102}). Si les traitements numériques sont à l'origine des développements de l'informatique, les traitements non numériques ont transformé de manière radicale l'art de la programmation (la liste des publications dans ce secteur de la recherche informatique est très longue, citons par exemple {69}, {90} et {104}). Cependant la conception et la réalisation de LISP - Langage spécifique de manipulation de symboles ({95} et {113}) - ont introduit un niveau d'abstraction supplémentaire à partir duquel se sont développés des travaux très diversifiés totalement détachés des machines concrètes. Ces travaux concernent principalement la définition de nouveaux langages symboliques de haut niveau ({8}, {11}, {115}, {116}, et {129}). Ils ont contribué, en particulier, à développer en intelligence artificielle des structures formelles de représentation et de traitement de symboles associés à des réalités physiques déterminées, comme par exemple celles issues de domaines de connaissances ayant les caractéristiques décrites dans le paragraphe précédent (voir § II.).

De telles structures seront appelées *Systèmes symboliques*.

Ainsi, notre situation de recherche montre que l'introduction des méthodes et des moyens informatiques dans la construction de systèmes de résolution de problèmes *confronte* deux univers, l'un *réel* qui correspond à l'objet de l'étude (voir § II.) et l'autre *symbolique* incarné par des complexes "méthodes-outils" dont l'interrogation au cours du processus expérimental contribue à découvrir des connaissances nouvelles (voir § IV.,et aussi pour des positions comparables {9} et {102}). La construction des systèmes symboliques conduit nécessairement à l'examen de deux catégories de questions : d'une part l'élaboration de *systèmes de représentation* des objets réels étudiés, auxquels correspondent les *bases de connaissances* ; d'autre part la mise en oeuvre, sur ces bases, de *systèmes de traitement* définis à partir des *formes de raisonnement élémentaires* (déduction, procédures, inférences analogiques, statistiques inférentielles, etc.) conformes aux classes de problèmes à résoudre. C'est la diversité des modes d'inférence définis dans les systèmes de représentation et de traitement de connaissances qui justifie, selon nous, le terme de "système symbolique" (les systèmes formels n'autorisent qu'un seul mode de raisonnement : la déduction, voir à ce sujet {102} et {109}).

Les expériences menées dans ce secteur d'activités sont nombreuses. Nos productions antérieures sur ces questions ont fait l'objet de plusieurs publications, comme par exemple {21}, {37}, et {38}. Dans ce qui suit nous examinons les travaux les plus significatifs développés en intelligence artificielle (en particulier {10}et{31} présentent certains aspects de ces recherches). Nous les situerons, dans le paragraphe V.,par rapport à notre recherche actuelle.

III.2. Les systèmes de représentation.

Les recherches menées dans le domaine de la représentation des connaissances sont apparues très tôt, dès la fin des années 60. Les méthodes qui ont été élaborées, et qui sont aujourd'hui les plus connues, dépendent des champs d'application et des objectifs d'utilisation : compréhension du langage naturel au moyen de la représentation du sens des phrases ; analyse et représentation de faits, d'actions, d'évènements, de propriétés, de relations ; catégorisation des objets en individus, classes, ensembles, concepts, super-concepts ; etc. Parmi ces méthodes nous distinguons les réseaux sémantiques et leur évolution, les frames, les systèmes de production et enfin le calcul des prédicats.

III.2.1. Les premiers réseaux sémantiques.

L'idée de réseaux sémantiques a été introduite pour la première fois par QUILLIAN à propos de ses travaux sur l'élaboration d'un formalisme organisant l'information qui permet de représenter la "mémoire sémantique" de l'être humain;{110}. Il s'agit de représenter le sens des mots à l'aide d'un graphe composé de noeuds repérant des concepts reliés entre eux par différents types de liens (homonymie, qualification/modification, coordination, etc.). Tout concept est défini dans une structure particulière, appelée *plan*, dont les sommets sont des mots pointant vers les plans définissant les concepts véhiculés par ces mêmes mots.

Ensuite QUILLIAN a amélioré son modèle dans le cadre du projet "Teachable Language Comprehender, TLC" en définissant de manière plus précise les éléments qui composent son réseau initial (voir {111}) ; et en collaboration avec COLLINS,a entrepris une série de tests psychologiques pour tenter de valider par un modèle hiérarchique la mémoire sémantique

de l'humain (voir à ce sujet {44}). La figure 1.1. montre un exemple de ce type d'organisation : chaque noeud de l'arborescence représente un concept, les concepts étant reliés entre eux par une relation de générique à spécifique (tout à fait comparable à celle organisant les thésaurus dans les systèmes documentaires évolués, voir par exemple {20} ; dans le même ordre d'idées, se reporter également au très interessant article de SMITH {126} qui établit une analogie entre les systèmes de recherche d'information et les méthodes de l'intelligence artificielle). De plus, à chaque noeud est attaché un ensemble de propriétés définissant le concept correspondant. Signalons enfin que ce type d'organisation est à l'origine de la conception de très récents systèmes de représentation de connaissaices (voir en particulier § III.2.3. et § III.2.4.).

Un autre projet significatif, issu directement des travaux de QUILLIAN, a permis à CARBONELL {25} d'utiliser pour la première fois la notion de réseaux sémantiques comme structures de données permettant d'organiser une base de connaissances. Le programme correspondant SCHOLAR d'aide à l'enseignement permet de manipuler et d'interroger en mode inter-actif une base relative à la géographie de l'Amérique du Sud, et décrite dans les termes d'un réseau sémantique particulier.

.2.2. Les réseaux sémantiques et les grammaires de cas.

Les travaux linguistiques de FILLMORE sur les grammaires de cas {57} sont à l'origine de l'élaboration d'un nouveau type de réseaux sémantiques. FILLMORE suppose que les grammaires des langues naturelles peuvent être construites à partir de la notion de *cas*, comme celle rencontrée en Allemand ou en Latin : L'élément central de la phrase devient le verbe, les cas définissant les différents types de liens qui existent entre le verbe et ses arguments. Par ailleurs, il admet qu'un nombre raisonnablement petit de cas est suffisant pour exprimer toutes les relations verbes-arguments, et qu'il suffit pour comprendre une phrase d'y reconnaitre tous les cas.

Ces idées ont été reprises, développées et intégrées dans la conception de réseaux sémantiques dont l'objectif est le traitement et la compréhension du langage naturel : les cas deviennent des propriétés qui, comme dans les premiers réseaux sémantiques, sont accrochées sur les différents noeuds composant les réseaux sémantiques.

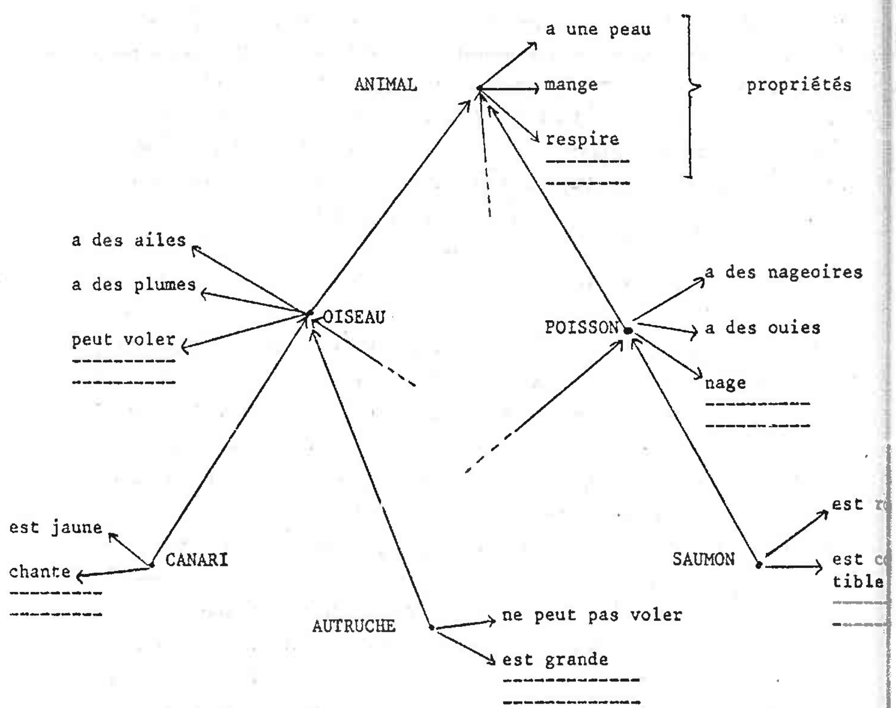


figure 1.1. - *Hiérarchie de type COLLINS/QUILLIAN.*

SIMMONS est le premier a avoir emprunté à FILLMORE la notion de cas pour construire un réseau sémantique utilisé pour comprendre et générer un ensemble particulier de phrases anglaise, {125}. Les noeuds correspondent aux concepts rencontrés dans les phrases, les concepts étant eux-mêmes regroupés dans des paradigmes selon la nature des relations sémantiques et syntaxiques (i.e. les cas) où ils peuvent intervenir. Ils sont reliés entre eux par des liens étiquetés par les relations mises en jeu effectivement dans les phrases analysées. Ce type de réseau ne porte aucune attention à la représentation de la "Connaissance du Monde" associée aux phrases à comprendre et à générer (comme c'est le cas dans les premiers réseaux) ; seule l'existence d'un lexique - extérieur au réseau - permet de définir les concepts en termes de paradigmes.

RUMELHART et NORMAN, {117}, supposent comme QUILLIAN que l'information dans la mémoire de l'être humain peut être représentée par un réseau sémantique ; mais la conception de ce réseau intègre des éléments permettant d'exprimer les structures de cas de FILLMORE. Par ailleurs leurs travaux sur la compréhension du langage naturel étant essentiellement d'ordre psycho-linguistique, ce type de réseau véhicule une certaine connaissance du monde (ce qui le rend beaucoup plus riche que celui élaboré par SIMMONS). Les verbes du langage naturel sont analysés et représentés par des structures sous-jacentes dans le réseau à l'aide de quatre classes de prédicats primitifs : les primitives d'états, de changement d'états, d'évènements et d'actions auxquelles correspondent quatre types de noeuds. Ces noeuds sont reliés aux valeurs prises par les arguments par des liens étiquetés par les noms de cas qui précisent la signification de ces arguments (AGENT, OBJET, INSTRUMENT, TEMPS, QUALIFICATION, etc.). Les noms et adjectifs (i.e. valeurs des arguments) sont représentés dans le réseau par des conjonctions de concepts qui les définissent au niveau sous-jacent. Ces concepts, qui déterminent un autre type de noeud, sont eux-mêmes organisés par des structures prédicatives appelées *prototypes* ; ces prototypes représentent la connaissance du monde étudié. Toutes les propriétés d'un concept défini par un prototype peuvent alors être héritées par d'autres noeuds reliés à ce concept par le lien ISA. Enfin, le lien particulier ISWHEN permet d'établir la relation qui existe entre un terme du langage naturel et sa représentation sous-jacente dans le réseau. La figure 1.2. représente le prototype HUMAN dans lequel la variable X indique que si un objet X est un humain alors il a toutes les propriétés du concept HUMAN (quantification universelle) ; la figure 1.3. représente la proposition "PETER BUT THE PACKAGE ON THE TABLE" (exemples extraits de {117}, ouvrage dans lequel sont décrits de manière très détaillée tous les éléments qui composent ce type de réseaux).

Enfin une autre recherche importante que nous rattachons à ce type de réseaux a été développée par SCHANK dans le cadre du projet MARGIE sur la compréhension du langage naturel, {121}. La conceptualisation est définie par un ensemble de onze actions primitives (*TRANS*, *INGEST*, *DO* ,etc.) à partir desquelles peuvent être représentées les phrases du langage naturel. Les actions primitives sont reliées à leurs arguments par des flèches de formes différentes définissant les relations dans lesquelles les concepts dépendent les uns des autres. Ces relations de dépendance sont telles que le système MARGIE peut évoquer à partir d'un concept donné tous les concepts qui en dépendent ; c'est là le principe de la *dépendance conceptuelle*. La figure 1.4. est un exemple de représentation conceptuelle pour la phrase "JOHN GIVES MARY A BOOK".

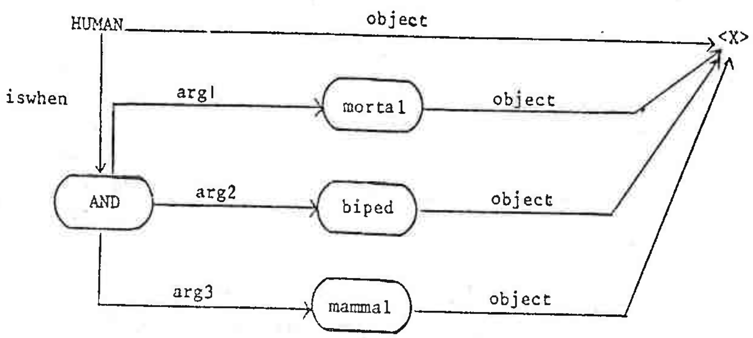
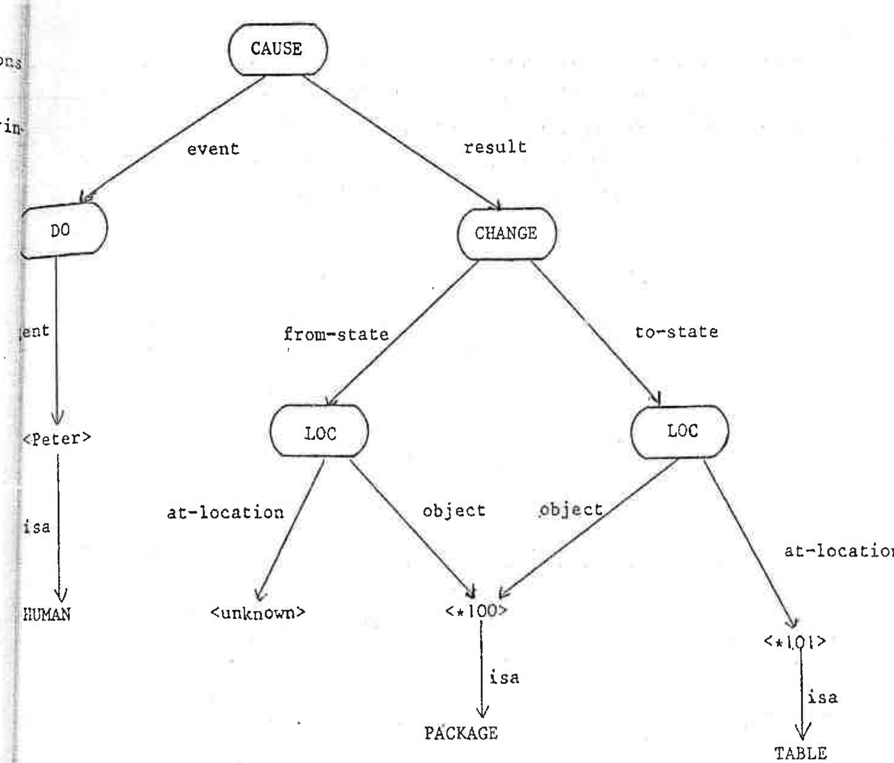


figure 1.2. - Exemple de prototype : Prototype HUMAN, Selon RUMELHART



(cette représentation indique que Peter est l'agent de l'action ; le résultat de cette action est le changement de localisation du colis, d'un endroit inconnu vers la table).

figure 1.3. - Représentation de l'énoncé "PETER PUT THE PACKAGE ON THE TABLE".

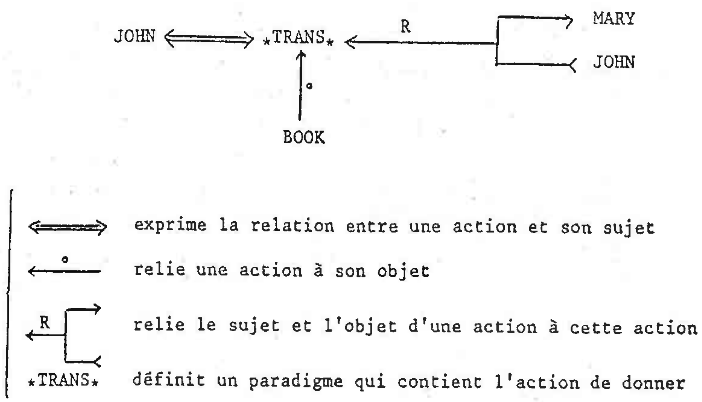


figure 1.4. - *Exemple de Dépendance Conceptuelle dans le système MARGIE.*

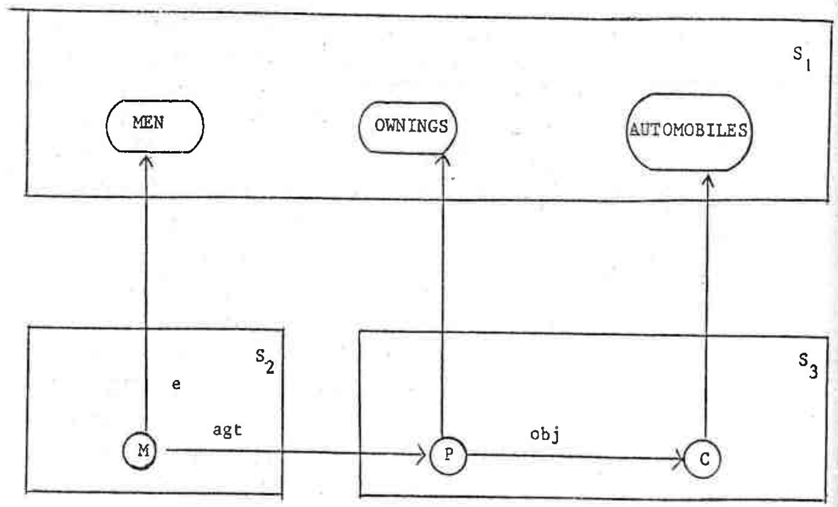
III.2.3. Les réseaux sémantiques et leurs fondements.

Les réseaux sémantiques (et ceux de la même génération) que nous avons sommairement décrits dans les deux paragraphes précédents présentent, pour la plupart, deux catégories de défauts : d'une part des lacunes conceptuelles liées à une absence d'explicitation d'un cadre méthodologique déterminant des principes généraux de représentation de connaissances ; d'autre part un certain nombre de confusions dans la définition des éléments composant les réseaux. La première catégorie de défauts a conduit essentiellement à élaborer des représentations qui ont un faible degré de généralisation. Les réseaux, qui ont été construits à propos d'études de cas particuliers, ne peuvent pas être directement utilisés dans le cadre d'autres applications ; il faut le plus souvent ajouter de nouveaux types de liens et/ou créer de nouvelles entités conceptuelles basiques (c'est par exemple le cas pour des systèmes comme TLC, SCHOLAR ou celui de SIMMONS). La deuxième catégorie de défauts concerne essentiellement la définition des noeuds et des concepts qui y sont attachés, ainsi que la nature des liens qui relient les concepts entre-eux ou à leurs propriétés. Par exemple les noeuds-concepts servent d'abord à représenter une classe (i.e. un ensemble d'individus).

Ils peuvent également devenir membre de la classe et représenter le concept lui-même ; la quantification (universelle/existentielle) des éléments manipulés est alors ambiguë, et sur le plan théorique cette confusion introduit un paradoxe. Enfin le noeud peut faire *référence* à un concept (distinction TYPE/TOKEN introduite par QUILLIAN, CARBONELL et RUMELHART). Dans ces conditions des liens comme ISA, MEMBER, SUPERSET, ISWHEN sont équivoques. Ils peuvent exprimer des rapports d'inclusion entre concepts, des rapports d'appartenance entre concepts et instances de concepts, et enfin des rapports TYPE/TOKEN entre concepts et références de concepts (voir par exemple le système SCHOLAR, et l'étude de RUMELHART et NORMAN sur la compréhension du langage naturel). Par ailleurs les notions de concepts et de liens sont intuitives et le plus souvent mal définies (QUILLIAN et SIMMONS) ; il n'y a pas de typologie organisant les liens, et souvent ils ne sont pas interprétés par les systèmes qui les manipulent (par exemple comme dans TLC).

Cependant les travaux récents sur la conception des réseaux sémantiques ont tenté de remédier aux types de faiblesses que nous avons signalées ci-dessus en s'appuyant en particulier sur la théorie des ensembles et le calcul des prédicats.

HENDRIX ({76} et dans {58} le deuxième chapitre, part I, 51-92) a introduit dans son système de représentation de connaissances un ensemble de liens permettant de supprimer les ambiguïtés dues à l'inclusion, l'appartenance et l'instanciation. Mais l'intérêt de ce système réside dans le fait qu'il est pourvu d'un mécanisme de quantification des concepts ; ce mécanisme est obtenu à partir de la notion de *réseaux partitionnés*. Une partition est un groupement formel de noeuds-concepts et d'arcs les reliant, appelé *"space"* (espace). La figure 1.5. illustre l'utilisation des espaces (indiqués sur la figure par des rectangles) pour représenter la phrase "SOME MAN M OWNS A CAR C". L'espace S_1 code l'information conceptuelle de base (au sujet des hommes, des propriétaires et des automobiles) ; S_2 indique qu'il s'agit d'un homme quelconque (lien e) et S_3 représente l'énoncé "-possède un car C" ; enfin le lien casuel agt établit les rapports entre S_2 et S_3. En outre la composition d'espaces individuels, le plus souvent dans les termes d'une hiérarchie, appelée *vista*, fournit un *mécanisme de représentation contextuelle* des connaissances (une idée comparable est développée dans {127}). Ce



$$(\text{vista} \left\{ \begin{array}{l} V_1=(S_1,S_2) \\ V_2=(S_1,S_3) \end{array} \right.)$$

figure 1.5. – *Représentation de "SOME MAN M OWNS A CAR C" à l'aide de la notion d'espaces (HENDRIX).*

mécanisme permet de représenter les connaissances à des niveaux de généralité différents ; à l'intérieur d'un vista les connaissances appartenant à un espace-parent sont *globales* pour l'ensemble des espaces-fils ; des espaces situés au même niveau possèdent des connaissances *locales*, non partageables.

Dans le but d'augmenter la puissance expressive des réseaux sémantiques, CERCONE et SCHUBERT ({122}, et dans {58} le quatrième chapitre, part I, 121-175) utilisent pour la définition des éléments composant leur réseau les possibilités qu'offrent à la fois le calcul des prédicats et le lambda-calcul. En plus des noeuds représentant des individus, des ensembles, des concepts de prédicats et de fonctions, ils introduisent deux

noeuds particuliers – le *noeud propositionnel* et le *noeud fonctionnel* – qui représentent respectivement les propositions et les fonctions considérées comme des unités d'information ; les liens PRED et FUNC sont respectivement associés de manière spécifique à ces deux types de noeuds (figure 1.6.). De plus, pour coordonner les propositions ils définissent des modalités de représentation des connecteurs logiques et des quantificateurs (figure 1.7.). Enfin leur système fournit la possibilité de représenter le temps, les opérateurs modaux et une description abstraite des fonctions en termes de lambda-calcul, les λ-abstractions.
Par ailleurs ces deux auteurs analysent l'adéquation de la conception des réseaux par rapport à un cadre de référence particulier – le calcul des prédicats –, et montrent que la représentation de connaissances par un formalisme de type réseau semble plus naturel et plus compréhensible que celle obtenue par la logique pure. D'autres auteurs ont utilisé la logique dans la conception des réseaux sémantiques, citons en particulier {56}. Par ailleurs, on trouvera dans {49} un point de vue légèrement différent, le calcul des prédicats apparaissant comme un formalisme puissant pour représenter les connaissances ; enfin {61} est un exposé systématique de l'utilisation de la logique et des problèmes qu'elle pose dans l'élaboration des bases de connaissances.

Enfin FAHLMAN a développé un système qui, selon nous, est intéressant pour deux catégories de raisons : /1/ D'abord quelques idées nouvelles apparaissent dans la conception de son réseau ; celles qui nous semblent les plus significatives sont sommairement présentées ici (pour plus de détails, se reporter à {54}). Le lien classique ISA donne naissance à deux types de noeuds – les *noeuds réguliers* et les *noeuds exclusifs* – qui organisent les concepts dans les termes d'une structure appelée "hiérarchie enchevêtrée" (sur le plan formel un treillis). En plus du lien ISA, plusieurs liens nouveaux contribuent à augmenter la puissance expressive de son système. Les *liens d'exception* (←/──) permettent de caractériser localement des réalisations de concepts qui sont eux-mêmes décrits par des propriétés différentes appartenant au même univers sémantique (exemple : COULEUR). Les *liens immédiats* (⟵⟵) permettent de caractériser uniquement les noeuds auxquels ils s'appliquent. *Le lien* OF permet de relier un concept à un fait que le caractérise et *le lien* IN permet de préciser le contexte de représentation des faits ;

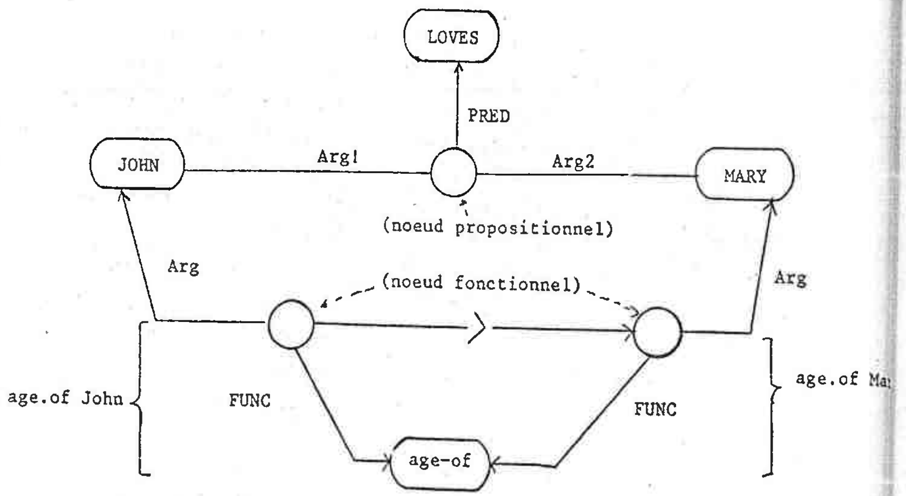


figure 1.6. - *Représentation de "JOHN LOVES MARY, JOHN IS OLDER THAN MARY" (d'après CERCONE et SCHUBERT).*

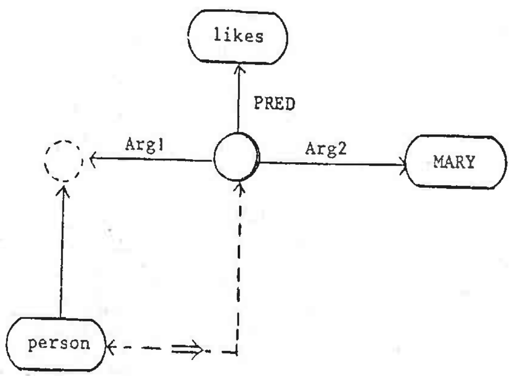


figure 1.7. - *Représentation de "EVERYONE LIKES MARY" (i.e. $\forall x[person(x) \Longrightarrow likes\ (x,Mary)]$).*

cependant il est clair que ces deux liens ont des significations qui se recouvrent, ce qui complique les modalités de leur utilisation. La figure 1.8. montre un exemple de représentation utilisant ces différentes notions. Par ailleurs les contextes sont eux-mêmes organisés à l'aide du lien ISA en sous-contextes (organisation comparable à celle des frames, voir § III.2.4.) ; ils déterminent les connaissances qui sont valides à un instant donné et sur lesquelles le système peut opérer./2/ En second lieu, dans le but de rendre plus efficace les mécanismes d'inférence mis en oeuvre dans son système (voir § III.3.), il propose (et à notre connaissance il est le premier) un *modèle Hardware de la conception de son réseau fondé sur un schéma de mémoire parallèle*. Il introduit ainsi des noeuds et des liens physiques. Chaque noeud est repéré par un numéro de série et caractérisé par une composition booléenne de marqueurs qui déterminent son statut instantané. Ces différentes unités sont connectées à la CPU par l'intermédiaire d'un bus général. La propagation des marqueurs le long de la mémoire parallèle permet de réaliser les commandes élémentaires de la CPU. En particulier cette propagation résoud très facilement et très rapidement l'association de descriptions globales et de descriptions locales caractérisant respectivement des concepts et des réalisations de concepts, et ceci simultanément pour des ensembles de faits grâce au parallélisme ; cette association correspond à ce que FAHLMAN appelle le *symbol-mapping problem*.

Les travaux que nous venons de présenter brièvement (et d'autres de la même génération) ont été développés dans un contexte où les fondements des réseaux sémantiques étaient examinés plus à fond à partir de réflexions d'ordre méthodologique et épistémologique sur les notions de liens, de concepts et de structures de données les organisant. Ces recherches débutent avec l'article de WOODS sur "What's in a link" {134}. Partant du constat que les techniques utilisées dans les réseaux sémantiques sont inadéquates pour représenter en général les connaissances (il n'y a donc pas de théorie de la représentation des connaissances, voir à ce sujet {75}), il examine ce que représente la notion de sémantique dans différents champs comme la philosophie, la linguistique, les langages de programmation et les réseaux sémantiques. Il aboutit à l'idée que la notion de relation, et de lien qui lui est attachée, est une caractéristique essentielle de la conception des réseaux sémantiques; et à partir d'exemples précis, il analyse les différents aspects que peut recouvrir cette notion (rapports attribut/valeur, relations n-aires et interprétation fonctionnelle, représentation prédicative des liens et ad quation avec la logique, structures de cas, etc.). BRACHMAN poursuit ce de recherche ; et analysant les confusions communément rencontrées dans réseaux sémantiques (voir notre analyse au début de ce paragraphe, § III ses investigations portent plus particulièrement sur la signification de noeuds et des concepts qui y sont attachés ({23} et {24}). Cette étude le conduit à distinguer très nettement trois axes de représentation : /1/ le premier axe définit les notions de classes et d'éléments. Les classes représentent des concepts ou des attributs génériques pouvant les caractériser ; les éléments représentent des instances de concepts ou des attributs spécifiques pouvant les caractériser. Le lien INSTANCE/OF relie un concept à ses instances, et le lien INSTANTIATES relie un attribut générique à ses attributs spécifiques. /2/ Le deuxième axe définit les rapports entre les concepts (ou les instances de concepts) et leurs éléments de description à l'aide du lien DATTRS (ou du lien ATTRS) /3/ Enfin le troisième axe définit les rapports entre éléments de description à l'aide d'un nombre fini de liens primitifs (ROLE, VAL, STRUCTURAL/CONDITION, etc.). La figure 1.9. illustre ces différents axes de représentation à l'aide de l'énoncé "THE EQUIPMENT OF FLIGHT AL26 IS DC9".

Les travaux d'ordre méthodologique et épistémologique sur la conception des réseaux sémantiques nous parait très importante dans la mesure où elle doit permettre non seulement d'aboutir à une meilleure représentation des connaissances relevant de domaines empiriques réels, mais aussi de définir un cadre théorique qui fonde les décisions prises pour construire ces représentations. C'est précisément l'approche que tente, par exemple, BRACHMAN ; approche que nous avons systématiquement mise en oeuvre pour concevoir le système symbolique ARCHES (voir § V. ; et chapitre deuxième).

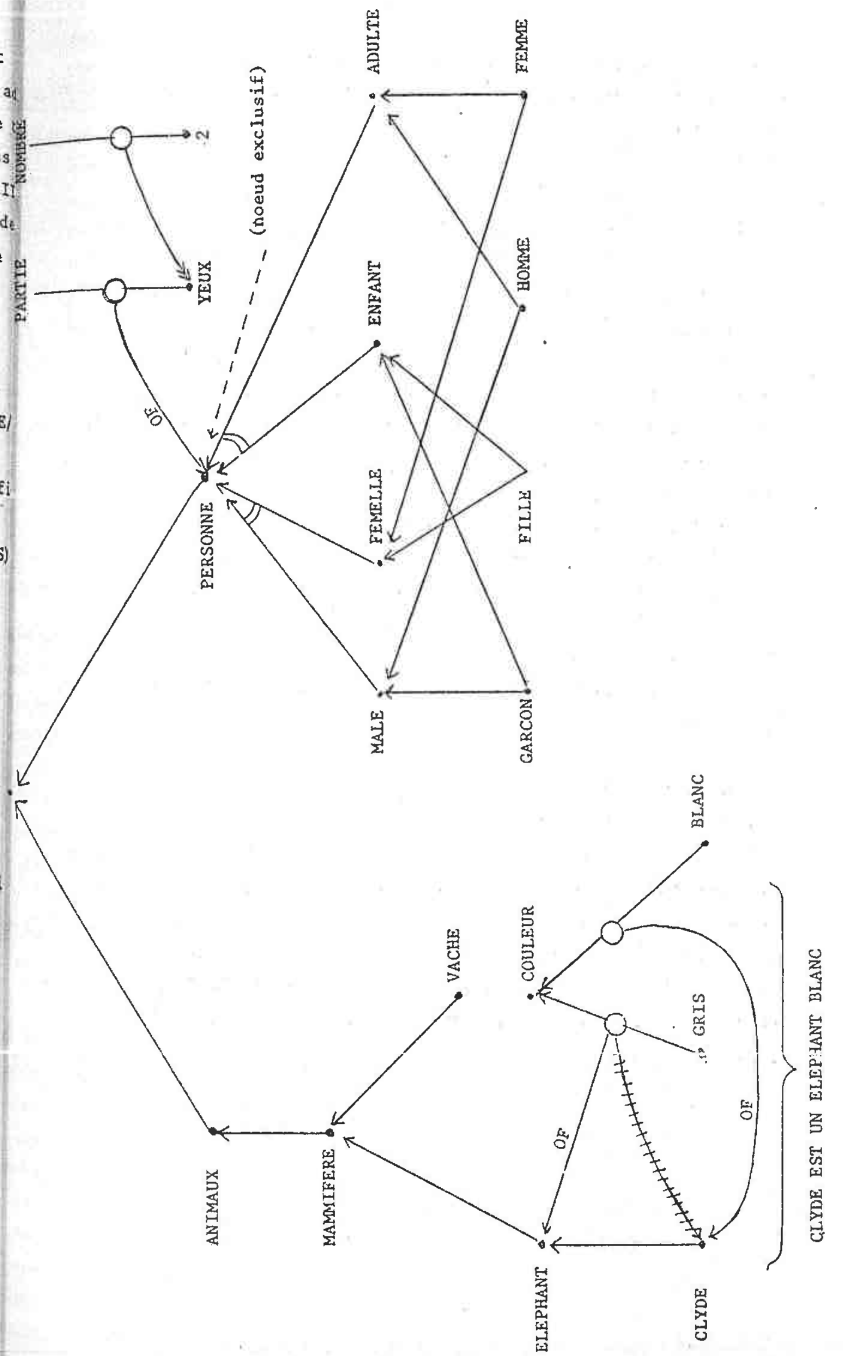


figure 1.8. - Représentation d'un extrait du monde "ETRE-VIVANT" dans un système de type FAHLMAN.

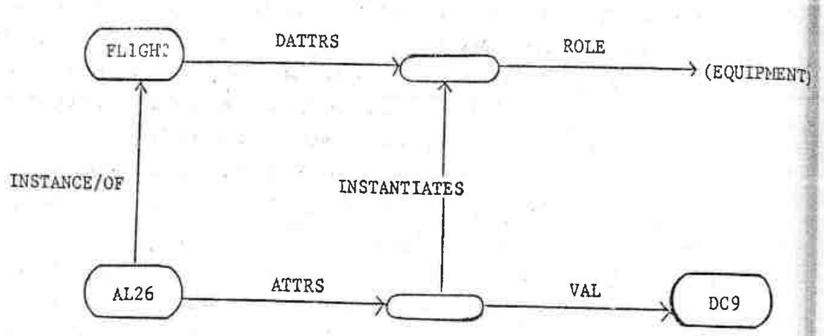


figure 1.9. - *Représentation de "THE EQUIPMENT OF FLIGHT AL26 IS DC9" (selon BRACHMAN).*

III.2.4. Les stéréotypes (ou "Frames").

En représentation des connaissances une autre difficulté importante relè du *problème des présuppositions*. Celui qui écrit (ou qui parle) sur un (d'un) sujet particulier suppose toujours que son lecteur (ou son auditeu sait beaucoup de "choses" qui n'ont pas à être précisées et/ou explicité Au contraire une machine doit posséder nécessairement toutes ces connais sances pour comprendre ce dont il est question. Ce problème fondamental est à l'origine de la mise en oeuvre de nouvelles modalités de représent tion de connaissances, que MINSKY a développées à l'aide de la notion de Frame, que nous traduirons par le terme de "Stéréotype" (voir {101} et dans {10} le sixième chapitre, 151-184 ; une notion comparable - les *Scri* a été définie par ABELSON dans {10}, chapitre dixième 273-309, et repris par SHANK dans {121}).

MINSKY définit un stéréotype comme une *structure de données représentant situation stéréotypée*, comme par exemple la description d'un salon ou d' scénario relatif à une personne allant au restaurant. Différentes sortes d'informations peuvent être attachées à un stéréotype : des informations relatives à la manière d'activer et d'utiliser les stéréotypes, des infor tions sur la prédiction d'évènements futurs, des commandes à effectuer si certaines hypothèses ne sont pas réalisées sous certaines conditions, de données représentant la connaissance implicite que la machine doit néces sairement posséder, etc. D'une manière générale les informations compos les stéréotypes peuvent être aussi bien de nature *déclarative* que *procéd*

Les stéréotypes décrivant et représentant tous les aspects d'une situation donnée sont reliés entre eux pour former un *réseau de stéréotypes* (le plus souvent une arborescence). Des procédures de remplacement ou d'héritage de stéréotypes sont attachées au réseau ; elles sont activées toutes les fois que les traitements sur les stéréotypes en cours d'examen conduisent à un échec (i.e. impossibilité de réaliser une opération d'unification, voir § III.3.). Cette méthode de représentation des connaissances semblait, au moins pour leurs auteurs, très prometteuse. Malheureusement, selon nous, elle pose de nombreux problèmes conceptuels et techniques qui n'ont pas été résolus : en premier lieu peut-on réduire des situations réelles relevant de domaines empiriques à des situations stéréotypées ? C'est là un problème théorique fondamental en l'absence d'un quelconque cadre méthodologique. Comment définir dans le réseau la notion de sous-stéréotype ? Il n'y a aucune règle de composition ou de décomposition des stéréotypes entre-eux. Comment comparer deux stéréotypes et quelle est la complexité des procédures d'appariement correspondantes ? etc. D'ailleurs, à notre connaissance, elle a été très peu utilisée.

I.2.5. Le système de production.

Les systèmes de production ont d'abord été utilisés pour définir les langages et les grammaires formels. De même on peut signaler leur adéquation aux *algorithmes de Markov* où deux types de règles (règles simples notées a $\longrightarrow$ b et règles conclusives a $\longrightarrow\cdot$ b, a et b $\in$ à un alphabet donné) ont permis de formaliser la notion d'algorithme selon des modalités déductives.

Depuis une dizaine d'années, ils ont été mis en oeuvre pour modéliser des bases de connaissances ; et ont donné naissance à des programmes qualifiés de *systèmes-experts*. Ces systèmes, spécialisés dans des domaines particuliers (médecine, psychologie, etc.), ont pour objectif d'aider les praticiens relevant de tels domaines à prendre rapidement des décisions dans des *univers de connaissances incomplets, c'est à dire imparfaitement connus*. La résolution de problèmes dans de tels systèmes est mis en oeuvre par des modes de raisonnement de type hypothético-déductif ; mais ces *raisonnements sont approchés* car ils supposent que l'information enregistrée est précisément incomplète. Parmi les nombreux systèmes-experts, MYCIN est un système de production - probablement l'un des plus achevés en intelligence artificielle - qui utilise de façon opérationnelle une importante base de connaissances médicales. Ce programme est conçu pour fournir de manière interactive une aide au médecin dans le diagnostic et le traitement des infections bactériennes du sang ({92}, {124}).

Un système de production est formé essentiellement de deux composants basiques : d'une part un ensemble de règles et son interpréteur, d'autre part la base de données à partir de laquelle opérent les règles (voir à ce sujet {47}).

α) <u>les règles</u>.

Les règles qui composent les systèmes de production sont appelées *règles de production* (ou parfois règles de réécriture, voir à ce sujet {7}). Ce sont des expressions de la forme :

$$PG \longrightarrow PD$$

dans laquelle la partie gauche PG de la règle exprime une certaine situation, et la partie droite PD l'action à effectuer quand la situation a été évaluée avec succès. Une telle règle peut être considérée comme un énoncé conditionnel (i.e. un énoncé de conséquence, voir § IV.3., chapitre deuxième) pour lequel la règle de détachement est le *modus ponens*. La complexité des règles dépend naturellement des domaines de connaissances étudiés. Comme les raisonnements qui sont développés dans les sytèmes de production sont des raisonnements approchés, les parties droites et gauches des règles peuvent être affectées d'un coefficient (en général compris entre 0 et 1) qui exprime le *degré de vraisemblance* de l'action à entreprendre. (figure 1.10., exemple extrait de {92}). L'interpréteur explore les parties gauches des règles jusqu'à ce qu'il trouve une règle dont la premisse peut être unifiée avec l'un des éléments de la base de données. Cet élément est alors remplacé par la partie droite de la règle sélectionnée, et l'exploration continue avec les règles suivantes. Au niveau basique, le mode de raisonnement mis en oeuvre par l'interpréteur est donc bien fondé sur une règle d'inférence de type modus-ponens. Et pour résoudre tout problème, l'interpréteur élabore une démonstration fondée d'une manière générale sur la construction d'une arbre ET/OU ({29} et {103}).

β) <u>la base de données</u>.

Les bases de données des systèmes experts contiennent les faits et les assertions relatifs aux mondes de connaissances investigués. Il n'y a à priori aucune contrainte sur leur complexité d'organisation. Cette organisation peut être représentée par une simple collection de symboles véhiculant l'état du monde analysé (structure vectorielle) ; mais elle peut aussi correspondre à des structures de graphes

<u>Si</u> p_1 et p_2 et... et p_n <u>ALORS</u> a_1 et a_2... et a_p

(format général d'une règle)

<u>Si</u> le site de la culture est le sang
ET l'organisme est à Gram négatif
ET l'organisme est de forme bâtonnet
ET le patient est un hôte à risque
<u>ALORS</u> Il est probable que l'organisme est le pseudomonia aeruginosa (valeur 0,6)

(exemple particulier, Règle 85)

figure 1.10. – *<u>Règles de production dans le système MYCIN</u>.*

très complexes. En fait la structure des bases de données dépend de celle des règles de production (en génral les faits sont représentés de la même manière que les parties gauches des règles de production). Par exemple le système MYCIN utilise des *quadruplets* pour représenter les connaissances : les trois premiers éléments véhiculent l'information à représenter en termes de PREDICAT, OBJET et ATTRIBUT ; le quatrième élément est une valeur qui indique le degré de vraisemblance du fait représenté :

<PREDICAT><OBJET><ATTRIBUT><VALEUR>

Ainsi ce sont les règles de production (complètes ou incomplètes) qui sont chargées de véhiculer la connaissance ; cette connaissance, donnée de façon modulaire, est aisément modifiable (ajout, suppression, remplacement) car elle n'est pas mêlée au corps des programmes.

Les sytèmes experts définissent des systèmes symboliques tout à fait satisfaisants pour des univers incomplètement décrits qui manipulent de grandes quantités de connaissances. Cependant, l'absence de règles générales déterminant des modes de représentation d'informations complexes (comme par exemple dans les réseaux sémantiques) limite selon nous certainement le champ d'efficacité de tels systèmes.

III.3. Les systèmes de traitement.

Les systèmes de représentation ont pour objet essentiel de véhiculer des ensembles de connaissances à partir desquels doivent être résolus les problèmes formulés à leur propos. Par exemple le modèle de mémoire sémantique de QUILLIAN détermine également, selon lui, un mode général de *représentation inférentielle des connaissances*. Les techniques d'inférence correspondantes sont fondées sur la notion de recherche croisée *(intersection search)* qui permet d'expliciter les relations potentielles qui existent entre les différents mots composant ce type de réseau. Etant donné deux mots et les plans qui les définissent, la procédure d'inférence consiste à rechercher dans le réseau deux chemins, issus de ces deux plans, qui ont un sommet commun ; si un tel sommet est trouvé, QUILLIAN infère qu'une certaine relation semantique existe entre ces deux mots. De même dans les hiérarchies de type COLLINS/QUILLIAN, on peut très facilement inférer en s'appuyant sur la relation "spécifique-générique" les énoncés suivants : "le canari est un oiseau", "le canari à des ailes", "le canari mange", etc. (figure 1.1.).
D'une manière plus générale, la résolution de problèmes met en oeuvre des mécanismes d'obtention d'informations nouvelles, déterminés par des calculs logico-mathématiques bien définis qui doivent s'appliquer sur les éléments composant les systèmes de représentation. Deux catégories de calculs peuvent être mobilisées. La première de nature strictement numérique fait appel à des théories mathématiques telles que les probabilités ou la statistique (comme par exemple dans {14} et {73}) ; la seconde de nature non numérique manipule des chaînes de symboles et repose plutôt sur la logique et/ou la théorie des fonctions (comme par exemple dans {9} et {16}). Les mécanismes "d'explicitation de l'information", qui correspondent à notre axe de recherche, sont mis en oeuvre plus particulièrement par la deuxième catégorie de calculs. Ils sont véhiculés par des systèmes de symboles organisés que nous appellerons *systèmes symboliques de traitement*. Leur fonction est d'explorer les systèmes symboliques de représentation à l'aide d'opérations logiques appropriéés dans le but de construire des espaces de solutions et d'y rechercher les solutions aux problèmes posés.

Les techniques de construction des systèmes de traitement dépendent d'abord de la définition formelle des figures de raisonnement élémentaires dont les compositions organisent les raisonnements contribuant à

résoudre les problèmes posés. Elles sont par nature très diverses : utilisation de méthodes axiomatiques comme dans {16}, mise en oeuvre de "procédures naturelles" de type GENTZEN comme dans {40}, exploitation formelle des propriétés sémantiques pour la dérivation d'information comme dans {21}, élaboration de réseaux de procédures fondés sur la notion de relativité des concepts comme dans {45} et {84}, etc. (pour un examen systématique de ces questions voir en particulier {7}). Ainsi il n'y a pas de méthodes générales qui permettent de définir les critères qui président à l'élaboration des systèmes de traitement : ces critères sont liés à la classe des phénomènes visés, et à celle des problèmes à résoudre.

Cependant les systèmes de traitement possèdent une particularité commune, celle d'opérer sur des systèmes de représentation bien déterminés. Ces deux types de systèmes entretiennent nécessairement de rapports spécifiques qui conditionnent leur réalisation respective. Plus précisément leur articulation est fondée essentiellement sur les opérations basiques qui permettent de manipuler les variables, intervenant dans la formulation et la résolution des problèmes, en accédant aux éléments composant l'un à partir des éléments organisant l'autre. Elle est donc réalisée par des modalités particulières d'accès aux structures de données, dont les définitions sont finalisées par les catégories de calculs qui doivent être appliqués sur ces structures. Les mécanismes de recherche de l'information ont conduit à construire des objets informatiques dont les modes d'accès sont déterminés non plus à partir de la définition (ou du calcul) de repères, mais à partir de la définition (ou du calcul) de certaines propriétés qui les caractérisent. En d'autres termes, les accès aux structures de données sont définis à partir de valeurs représentant tout ou partie de l'information véhiculée par ces structures (i.e. il s'agit du contenu et non plus du contenant). Ce type d'accès, appelé *accès associatif*, a fait l'objet de nombreux travaux, et en particulier dans le cadre qui nous préoccupe ici citons {20} et {22}. Il a été systématisé dans les langages de manipulation de chaînes et d'arbres, comme dans {69}, où la *reconnaissance de patrons*, qui permet d'accéder à des structures de données complexes, peut être considérée comme une première généralisation des mécanismes d'accès associatifs (par exemple, la conception et l'implémentation du système SYCIL mettent en oeuvre des mécanismes d'accès dirigés par reconnaissance de patrons, voir {36}). Une deuxième généralisation a été obtenue en introduisant dans la définition des patrons des variables pouvant être utilisées dans des calculs de nature diverse ; ce sont les *filtres*. Utilisés pour la résolution de problèmes en intelligence artificielle, notamment en démonstration automatique de théorèmes, en représentation de connaissances et en compréhension du langage naturel, ils ont donné naissance à une troisième généralisation des modes d'accès aux structures de données : C'est *l'opération d'unification* dans laquelle les deux opérandes peuvent véhiculer des variables ({79}, {80},{115}).

C'est dans ce contexte que se situe notre recherche : Le système symbolique ARCHES est muni d'un système de traitement dont les mécanismes de raisonnement élémentaires nécessitent des modalités d'accès aux structures de données l'organisant fondées en particulier sur des opérations spécifiques de filtrage et d'unification.

IV. RAPPORTS ENTRE DOMAINES DE CONNAISSANCES ET DOMAINES SYMBOLIQUES.

Le recours aux méthodes informatiques contribuant à mettre en oeuvre des systèmes de résolution de problèmes réels conduit - selon le cadre que nous nous sommes fixés - à définir les modalités de représentation de connaissances dans les termes de systèmes symboliques adéquats. Les opérations, qui permettent de construire les éléments symboliques à partir de ceux composant les univers réels étudiés, peuvent être considérées comme des *applications* qui transforment les aspects pertinents des domaines de connaissances en des systèmes symboliques (une idée comparable est développée dans {9}). Ces applications fixent les rapports formels qui existent entre les entités réelles investiguées et les entités symboliques les véhiculant. Leurs définitions, qui dépendent naturellement des recherches à effectuer (catégories d'objets réels à étudier, classes de problèmes à résoudre, types d'informations relatives au savoir et au savoir-faire empiriques à mobiliser au cours des investigations, examen de la validation des résultats, etc.), procèdent tout autant de méthodes sémiologiques ou linguistiques que de méthodes mathématiques ou logico-informatiques. Le caractère expérimental et scientifique de notre démarche leur confère cinq propriétés remarquables : /1/ Les applications sont finalisées ; /2/ elles sont réductrices ; /3/ elles sont régulières ; /4/ elles sont reproductibles ; /5/ et enfin elles sont évolutives.

Si le schéma proposé pour fixer les rapports entre domaines de connaissances et domaines symboliques est relativement simple (figure 1.11.), l'élaboration des applications à partir desquelles sont déterminées les systèmes symboliques pose des problèmes méthodologiques très complexes comme le soulignent par exemple {15}, {17} et {34}, et dont l'analyse systèmatique sort du cadre de notre recherche (pour la construction effective d'applications fondée sur des modalités raisonnées d'utilisation et de composition de théories d'emprunt comme la linguistique ou l'informatique se reporter par exemple à {12}, {18} et {36}). Cependant nous

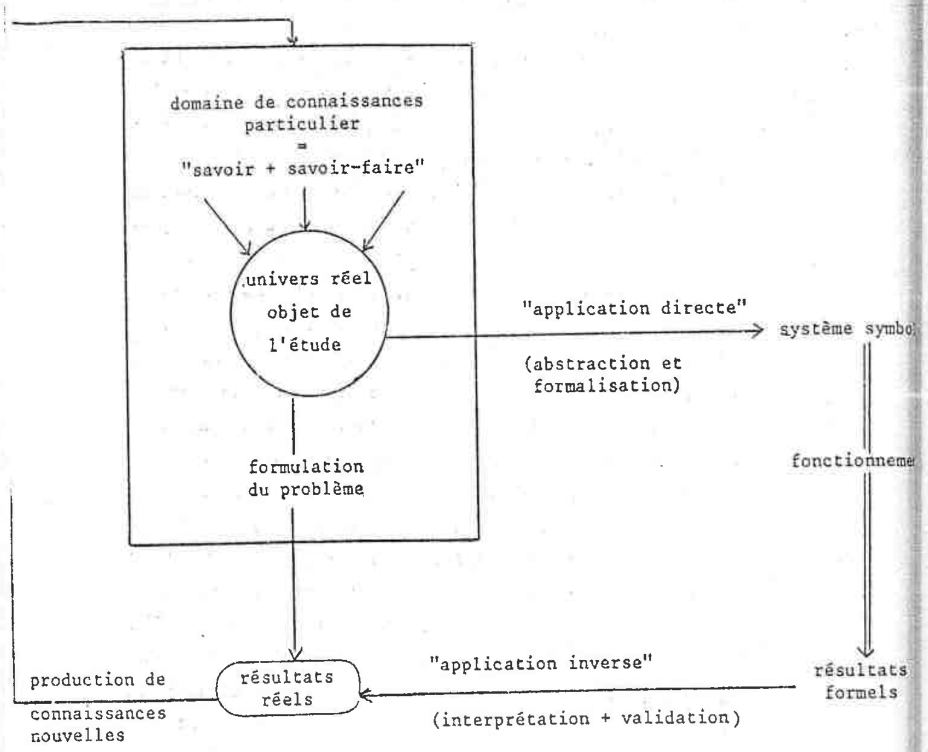


figure 1.11. - *Applications entre univers réels et représentations formelles.*

présentons sommairement quelques questions relatives à la *réduction de l'information* car elles interviennent, selon nous, à tous les niveaux d'élaboration des applications.

En l'absence de théorie formelle organisant les domaines de connaissances visés (voir § II.), les opérations qui appliquent les mondes réels sur les mondes symboliques dépendent des études particulières réalisées à leur propos. Elles sont déterminées à partir d'objectifs de recherche précis qui s'expriment d'abord en termes de formulation de problèmes et de corps d'hypothèses relatives à leur résolution. La justification des hypothèses par les spécialistes des domaines étudiés, dont l'argumentation prend appui sur le savoir et le savoir-faire empiriques caractérisant ces domaines, permet de déterminer les critères fixant la nature et l'étendue des informations nécessaires à la résolution des problèmes posés, et ceux établissant le choix des méthodes de résolution. Ces critères définissent en compréhension les univers réels (inclus dans les domaines de connaissances) qui représentent les ensembles-sources à partir desquels sont construits les applications. En conséquence l'observation scientifique des univers réels étudiés ne s'effectue pas dans toutes les directions, mais va être orientée par les objectifs visés. Les informations retenues - considérées comme pertinentes par rapport à ces objectifs - ne sont donc pas exhaustives. Des opérations de *sélection* et *d'extraction* de l'information permettent ainsi de définir des images réduites des objets réels étudiés. Elles utilisent pour leur mise en oeuvre des outils linguistiques, mathématiques, logiques et informatiques (voir par exemple {12},{20} et {36}) ; leur complexité dépendant naturellement de la complexité des domaines de connaissances. Les objets symboliques correspondants sont considérés - par rapport aux points de vue adoptés - comme les substituts de ces images, mais sont tenus pour équivalents aux objets réels ({37}). Enfin nous ne devons pas confondre ce type de réduction avec celui correspondant à l'information implicite contenue dans l'information retenue et effectivement enregistrée ; ce type de réduction, fondé sur les techniques de dérivation de l'information, permet en particulier un gain de place-mémoire appréciable (une analyse de ces questions est présentée dans {9} et {75}).

V. VUE D'ENSEMBLE DE L'ORGANISATION GENERALE DU SYSTEME SYMBOLIQUE ARCHES.

Ce paragraphe présente un aperçu de l'architecture et du fonctionnement du système symbolique ARCHES, architecture et fonctionnement qui sont décrits de manière systématique dans la suite de notre travail. Son objet et de situer notre recherche par rapport aux travaux qui se sont développés dans le même secteur d'activités, et de dégager ainsi les idées essentielles qui fondent la conception de ARCHES.

Notre recherche n'est pas liée à une application déterminée, mais est issue d'un examen détaillé d'un ensemble d'applications relatives à la représentation et aux traitements de connaissances (voir en particulier {35} et {37}). ARCHES est ainsi adéquat à une classe de phénomènes qui est circonscrite par un groupe précis d'hypothèses fondées sur la logique, la linguistique et la nature des univers de connaissances analysés (voir chapitre deuxième) : c'est un *méta-système* qui peut être utilisé pour générer des "systèmes d'information intelligents" dans différents domaines du discours scientifique. Remarquons que d'autres systèmes de représentation ont été réalisés dans une telle optique, et ce notamment depuis l'examen des fondements des réseaux sémantiques (voir § III.2.3.). Cependant nous pensons que les hypothèses qui fondent la conception de ARCHES, et qui sont développées dans le chapitre deuxième, déterminent un "véritable cahier des charges" permettant d'évaluer de manière précise son adéquation avec toute application.

La conception de ARCHES a été réalisée en distinguant très nettement quatre axes de représentation : /1/ le premier axe définit les notions de *concepts* et *d'individus* utilisées pour organiser et décrire les connaissances à représenter, {39}. Le lien SET permet de relier les concepts entre-eux, et donne ainsi naissance à une organisation formelle représentée par un ensemble fini de graphes disjoints particuliers, les *graphes de*

résolution ou *R-graphes*. Le lien INS permet de relier les concepts aux individus, parmi lesquels nous distinguons les instances de concepts et les termes référençant des concepts. Ces différents éléments permettent enfin d'organiser les connaissances dans les termes d'un réseau partitionné à l'aide des notions de *champs* et de *domaines* (voir chapitre troisième). /2/ Le deuxième axe définit les modalités de représentation des faits à partir de la notion de *termes descriptifs* qui permet de représenter les propriétés caractérisant les éléments décrits. Les termes descriptifs sont construits à l'aide de trois entités basiques : les *traits* qui expriment des qualités, des valeurs de propriétés, etc. ; les *classes* qui regroupent les traits de même nature sémantique et déterminent ainsi les relations d'état ; et enfin les *opérateurs* qui expriment les rapports entre classes et traits ({40}). Les *compositions* de termes descriptifs déterminent les objets *descriptions* ; elles sont définies par un ensemble fini de *connecteurs* dont certains permettent d'exprimer l'évolution des descriptions (ET-d'addition, OU-inclusif, négation, futur immédiat, futur médiat et enfin le ET-de-succession) (voir chapitre quatrième). /3/ Le troisième axe définit d'une part les rapports entre descriptions et individus qu'elles caractérisent, et d'autre part les rapports entre descriptions et d'autres descriptions qu'elles précisent localement. Ces rapports sont exprimés explicitement à l'aide de deux liens distincts, respectivement le lien ADP (attribution de description principale) et le lien ADL (attribution de description locale) (voir chapitre troisième et chapitre quatrième). /4/ Enfin le dernier axe définit les modalités de représentation des traitements élémentaires qui peuvent être effectués ; ces modalités sont définies à partir des notions générales de *règles de réécriture* et d'*énoncés de conséquences*. La composition formelle de ces règles et de ces énoncés permettent de mettre en oeuvre dans ARCHES deux types de raisonnements : le raisonnement déductif et le raisonnement analogique ({41}) (voir chapitre huitième et chapitre neuvième).

Il est évident que de nombreux travaux ont élucidé les notions de concepts et de liens à propos de l'organisation particulière de réseaux sémantiques (voir § III.2.3.), comme nous l'avons fait à propos de ARCHES (axes 1, 2 et 3 ci-dessus). Cependant l'originalité de notre travail est multiple. Au niveau de la représentation, des notions nouvelles sont introduites et permettent de clarifier les rapports entre caractérisés et caractérisations (comme par exemple les notions de classes et d'opérateurs, les liens ADP et ADL, la représentation de l'interprétatio[n] de la négation, ou enfin la représentation de l'évolution des descriptions). Par ailleurs ARCHES est, à notre connaissance, le premier système qui combine à la fois les avantages des réseaux sémantiques et des systèmes de production (axe 4 ci-dessus) (voir § III.2.5.). Enfin, notre système a été construit en s'appuyant sur la logique, ce qui a permis d'établir ses principales propriétés formelles (axiomatisation des éléments de représentation, règles d'inférence basiques, cohérence, complétude, etc.).

CHAPITRE DEUXIEME.

HYPOTHESES SOUS-JACENTES A LA CONCEPTION DU SYSTEME SYMBOLIQUE ARCHES

-=-=-=-=-=-=-=-=-=-

I. UN PRINCIPE FONDAMENTAL...

Le cadre méthodologique, qui a régi le développement de notre recherche et qui a déjà fait l'objet d'une étude particulière (voir chapitre premier), a pour fonction essentielle d'expliciter, et donc de préciser, les présuppositions qui fondent la conception du système symbolique ARCHES. Il ressort de cette étude un principe fondamental sur lequel prennent appui les hypothèses de conception de ARCHES : *"Il n'y a pas de théorie générale de la représentation des connaissances"* (voir en particulier {75 }). Comme nous l'avons observé, les travaux qui se sont développés en intelligence artificielle montrent que les études sur la représentation des connaissances n'ont pas de frontières bien délimitées. Quête essentiellement empirique, elles procèdent de disciplines et de domaines d'application aussi divers et différents que le langage naturel, la logique, la conception de langages de programmation, la robotique, la compréhension de récits d'enfants, la résolution de problèmes en sciences de l'homme, etc.. Si nous souhaitons conduire une recherche sur un schéma particulier de représentation de connaissances, il est donc nécessaire de formuler les hyptohèses - déterminées à partir de la nature empirique et cognitive des applications visées - qui fixent le cadre d'élaboration de ce schéma.

La description des phénomènes est exprimée le plus souvent à l'aide d'une utilisation spécialisée du langage naturel (voir § II.2., chapitre premier). Ce dernier apparaît comme le moyen de communication privilégié non seulement pour les véhiculer, mais aussi pour décrire et formuler les problèmes posés à leur propos. Ce constat montre que les langages naturels spécialisés entretiennent nécessairement des rapports très étroits avec les systèmes de représentation de connaissances. Ces relations concernent principalement les modalités de compréhension et d'analyse du langage naturel, que l'on vise ou non une acquisition automatique des phénomènes investigués. Les éléments composant de tels systèmes doivent être des représentations sous-jacentes de la signification des énoncés du langage naturel véhiculant ces phénomènes. En d'autres termes, l'élaboration des types de structures de données qui organisent les systèmes de représentation de connaissances est fondée sur un groupe d'hypothèses dont certaines correspondent précisément aux critères de définition de l'ensemble des énoncés en langage naturel pouvant véhiculer la classe des phénomènes visés.

La construction des structures de données basiques et composées conduit à un formalisme dont les caractéristiques logiques dépendent naturellement des qualités sémantiques et syntaxiques du sous-ensemble du langage naturel délimité par des hypothèses comme celles évoquées ci-dessous : Nature des représentations logiques des propriétés et des relations, combinaison formelle de ces représentations, rapports d'entraînement logique liés aux caractéristiques sémantiques des propriétés et/ou des relations, etc.. Si la résolution de problèmes peut utiliser les propriétés logiques liées à la représentation de ces structures de données, il va sans dire que dans le cas général elle met en oeuvre des raisonnements fondés sur les catégories d'opérations logico-mathématiques associées à ces structures. Là aussi la validité des résultats obtenus dépend des propriétés logiques des raisonnements et des catégories d'opérations qui les organisent. La cohérence intrinsèque de ces représentations et de ces traitements impose d'évaluer les rapports qu'entretiennent de tels formalismes avec les théories logiques et/ou algébriques. Avons-nous élaboré un système formel relevant strictement de la logique du calcul de prédicats ? Est-ce un système symbolique spécifique possédant des propriétés remarquables ? Quelles sont alors les relations qu'il possède avec les logiques classiques (ou non classiques) et/ou avec les structures algébriques ? Autant de questions nécessaires dont les éléments de réponses contribuent à déterminer les choix et les options à prendre pour construire les structures de données et les opérations associées.

S'il est élaboré à propos d'une étude de cas spécifique, le formalisme $\mathfrak{F}$ a en droit une seule interprétation qui correspond précisément à celle de l'univers réél objet de l'investigation. Dans le cas contraire, il est adéquat à une classe particulière de phénomènes qu'il est nécessaire de circonscrire : Le formalisme $\mathfrak{F}$ a alors plusieurs interprétations, chacune d'elles le faisant correspondre à un univers réél

déterminé issu de cette classe. Là encore, les rapports entre la réalité observée et les formalismes qui définissent ses représentations nécessitent d'explorer les voies qui précisent les hypothèses qui rendent adéquates tel formalisme à telle classe de phénomènes.

Ces quelques idées, suggérées par ce principe fondamental (dont le développement systématique sort du cadre de notre travail) posent à notre avis trois catégories de problèmes interdépendants et essentiels que tout concepteur de système de représentation de connaissances devrait élucider en fonction des objectifs qu'il poursuit : /1/ Quelles classes de faits doit-on représenter, et quelle est l'influence des critères de définition de ces classes sur les hypothèses qui fondent la conception des systèmes de représentation ? ; /2/ Quels rôles doit jouer le langage naturel - du double point de vue syntaxique et sémantique - dans les modalités de construction des structures de données organisant ces systèmes ? ; /3/ Enfin quelles sont les propriétés formelles que doivent posséder ces structures pour que les types de problèmes à résoudre puissent l'être effectivement ? ; en d'autres termes quels sont les rapports entre les systèmes de représentation et de traitement de connaissances et la logique ou les structures algébriques ? Ces différents problèmes ont été examinés à propos de la construction du système symbolique ARCHES, et les solutions locales que nous y avons apportées définissent les hypothèses justifiant les choix que nous avons faits dans l'élaboration des structures de données organisant ce système.

II. CLASSE DES PHENOMENES VISES.

II.1. Notion de procès.

Les phénomènes réels, qui constituent les matériaux à partir desquels se développent nos travaux sur la représentation des connaissances dans certains domaines empiriques - et en particulier dans les Sciences de l'Homme -, désignent des *procés* comme l'indiquent par exemple les études {18, 74, 78, 117, 119}. Comme il est d'usage dans certains travaux linguistiques ou informatiques {66, 91, 118, 128}, nous supposons que les procès se répartissent en deux catégories. D'une part, ceux qui expriment des *états* ; d'autre part, ceux qui déterminent des *actions*. Cette répartition, bien que marquée très nettement dans quelques langues pose des problèmes théoriques complexes car les unités linguistiques qui véhiculent les états et/ou les actions se recouvrent de manière plus ou moins continue sur le plan sémantique. Les difficultés théoriques pour élaborer des concepts linguistiques stables associés aux notions très générales de procès, d'état et d'action ne doivent pas cependant contrarier notre étude visant à construire un schéma particulier de représentation de connaissances. Les théories linguistiques - tout comme d'ailleurs la logique ou les mathématiques - doivent être considérées comme des théories d'emprunt dont les éléments, confrontés à la réalité des phénomènes à représenter, peuvent contribuer par filiation à l'élucidation et à la définition formelle de notions et/ou de concepts propres au modèle informatique que nous construisons. Ainsi le découpage Etat/Action, observé sur de nombreuses études de cas, nous a conduit à donner une *définition opératoire* de ces deux notions. Cette définition tient naturellement compte de nos objectifs qui sont de trois ordres : /a/ nous nous intéressons à une classe particulière de phénomènes dont la définition procède précisément de cette typologie ; /b/ nous visons à acquérir, à comprendre et à représenter des énoncés du langage naturel (langages spécialisés véhiculant les phénomènes étudiés) en élaborant des structures de données particulières dont l'intégration conduit

à un schéma de représentation adéquat aux phénomènes composant la classe ci-dessus ; /c/ enfin, l'organisation de ce schéma est telle qu'elle doit offrir la possibilité de réaliser certains types de traitements, et plus généralement de raisonnements producteurs de connaissances nouvelles. Ces objectifs sont suffisamment généraux pour naturellement donner naissance à plusieurs schémas, tous différents les uns des autres. Cette remarque montre que plus d'un choix peut présider à la clarification des notions d'etat et d'action. Celui que nous avons fait peut donc paraître arbitraire, même par rapport à ces objectifs. Cependant, il n'en est rien, il trouve sa justification dans une réalisation particulière de ces objectifs qui définit le cadre effectif d'élaboration de notre schéma.

II.2. Notion d'état.

Nous dirons que des phénomènes représentent un (ou des) état(s) s'ils expriment les manières d'être plus ou moins permanentes des entités les composant. Ces manières d'être sont véhiculées par des informations qui déterminent des *propriétés* ou des *qualités* (Exemples : "LA ROBE ROUGE EST EXCITANTE", "L'ELEPHANT A UN NEZ EN FORME DE TROMPE", etc.), des *situations* (Exemples : "JEAN DORT ALORS QUE NICOLE EST DEBOUT", "LE PRISONNIER EST ENFERME DANS UNE CELLULE", "L'AMPHORE N'EST PAS LOCALISEE DANS LE FOUR", etc.),et d'une façon plus générale des *marques caractéristiques quelconques* (Exemple : "LE VIN CONTENU DANS DES BOUTEILLES EST STOCKE POUR SA CONSERVATION DANS DES CAVES OU LA TEMPERATURE EST INFERIEURE A 10° ET LE DEGRE D'HUMIDITE COMPRIS ENTRE 20 ET 30 %", etc.). La valeur temporelle de ces caractéristiques dépend naturellement des phénomènes investigués. Elle exprime une durée qui est déterminée soit par une information qualitative (Exemple : "JEAN DORT DEPUIS LONGTEMPS", etc.), soit par une quantité mesurable (Exemples : "LES AMPHORES SONT RESTEES DANS L'EPAVE JUSQU'EN 1975", "LE VIN A ETE STOCKE DANS LA CAVE PENDANT TROIS MOIS", etc.). Elle peut également marquer un moment précis (Exemple : "PIERRE EST ARRIVE A TROIS HEURES", etc.). Nous devons noter que cette valeur temporelle est optionnelle ; son absence est ambiguë : elle peut indiquer la permanence de la caractéristique (Exemple : "LE CHAT A DES YEUX VERTS", etc.) ou une durée d'existence implicite et non déterminée (Exemple : "JACQUES EST ASSIS", etc.).

Les exemples cités ci-dessus montrent que les états sont exprimés par des énoncés plus ou moins complexes du langage naturel où des groupes verbaux, définissant des *relations d'état*, permettent de caractériser de manière temporelle des actants particuliers - les actants caractérisés - par d'autres actants, eux-mêmes pouvant jouer le rôle d'actants caractérisés dans d'autres relations d'état. La relation d'état la plus courante, du moins en français, est exprimée par le verbe ETRE suivi d'une locution verbale définissant sa sémantique (Exemples : "MARIE EST A GENOUX", "LE LIT EST SITUE ENTRE L'ARMOIRE ET LA COMMODE", etc.). De même l'auxiliaire AVOIR peut introduire une relation d'état à laquelle vient s'ajouter l'idée de possession (Exemples : "L'OISEAU A DES AILES", "DANS CE BAS-RELIEF, LA VIERGE A LES YEUX BLEUS-CLAIRS", etc.). D'une manière générale, un ensemble de verbes ou de groupes verbaux est reconnu, d'un point de vue purement pragmatique (et opératoire), comme pouvant introduire des relations d'état. Nous élargirons cette catégorie à des verbes qui permettent en fait de décrire des évènements particuliers (voir § II.3.), comme par exemple "MARIUS SE PROMENE". Nous dirons qu'un évènement est assimilé à un état quand la relation qui le sous-tend (en langage naturel, ce sont en général des verbes d'action) exprime une idée de continuité dans la réalisation de cette action par un agent, sans que sa cause soit explicitée et/ou nécessaire pour la compréhension de l'évènement. Un critère linguistique d'assimilation d'un évènement à un état pourrait être le suivant : Si une phrase P de la forme "N_1V_1" admet un équivalent sémantique de la forme "N_1 *est en train* de V_1(infinitif)" alors V_1 désigne un état dans P (voir § III.2.). Un tel critère montre de plus la dimension purement contextuelle (fonction de N_1, des circonstances, etc.) de la distribution entre état et action. Ainsi, l'énoncé "PAUL DORT PAISIBLEMENT" est considéré par rapport à cette analyse comme un phénomène représentant un état.

II.3. Notion d'action.

Nous dirons que des phénomènes représentent une (ou des) action(s) s'ils expriment une certaine activité exercée et/ou subie par les entités les composant. Plus précisément nous distinguons deux catégories d'action (en plus de celle que nous assimilons à la notion d'état). D'une part, les actions peuvent marquer le passage d'un état initial à un état final (exemple : "PAUL SE REVEILLE DOUCEMENT", etc.) et désignent en conséquence des *changements d'état*. Elles sont mises en oeuvre par des agents qui subissent leurs effets (exemples : "PIERRE SE TUE", "L'EMPLOYE QUITTE SON TRAVAIL A DIX-HUIT HEURES", etc.). D'autre part, les actions peuvent exprimer des *évènements*. Elles représentent alors de véritables "scénarios" où apparaissent les éléments qui concourent au déroulement de l'évènement : d'un côté l'agent et les moyens respectivement auteur et cause de l'action, de l'autre l'objet subissant les résultats de cette action (Exemples : "PAUL TUE PIERRE", "LE DRAGON A ATTAQUE HECTOR A L'AIDE DE SES FLAMMES", etc.). Nous devons remarquer que si l'énoncé "PAUL SE REVEILLE" exprime un changement d'état, il n'en est plus de même de l'énoncé "PIERRE REVEILLE PAUL" qui représente selon notre analyse un évènement. Ces deux exemples montrent la continuité sémantique qui marque le découpage des actions en changements d'état et évènements, et par là-même la difficulté théorique (au niveau linguistique) de justifer un tel découpage. En particulier, ils font apparaitre que les évènements peuvent également exprimer un changement d'état subi par l'objet de l'action.

Les actions, qui se déroulent bien évidemment dans le temps, sont repérées explicitement ou implicitement par une composante temporelle qui peut exprimer comme pour les états une date ponctuelle, une date d'origine, une date limite, une durée, etc.

Les exemples ci-dessus montrent que les actions sont véhiculées par des énoncés du langage naturel plus ou moins complexes. L'action élémentaire est gouvernée par une locution verbale, définissant une *relation d'action*, permettant de relier de manière temporelle un agent à un objet pour produire à partir d'un ensemble de moyens déterminés le résultat désiré par l'agent et subi par l'objet. Les différents éléments mis en relation dans une action élémentaire peuvent à leur tour participer à d'autres actions et/ou être actants de relations d'état. Les relations d'action sont le plus souvent exprimées par des locutions verbales dont le pivot est un verbe appartenant à une catégorie particulière que l'on qualifie d'un point de vue purement pragmatique de "verbes d'action" (ce sont en général des verbes transitifs).

II.4. Hypothèse dans ARCHES.

Par rapport à ce découpage opératoire des procès, nous nous intéressons à représenter une classe de phénomènes exprimant soit des états, soit des changements d'état. Ces phénomènes sont véhiculés en langage naturel par des énoncés plus ou moins complexes que nous supposons composés d'énoncés élémentaires mis en relation par des connexions linguistiques de nature diverse.

Un énoncé élémentaire est composé d'une relation d'état (au sens où cette notion a été définie précédemment) permettant de relier un actant - l'actant caractérisé - à une manière d'être particulière - l'actant caractérisant - permanente ou temporaire. Il définit un état élémentaire. Un état quelconque ou un changement d'état est défini comme une composition d'états élémentaires (nous verrons dans le paragraphe III les hypothèses de nature linguistique qui précisent les règles de composition des états).

C'est là notre première hypothèse sur laquelle s'appuie notre recherche.

Hypothèse 1. *Nous visons à représenter dans ARCHES des phénomènes factuels exprimant des états ou des changements d'état.*

III. RAPPORTS AVEC LE LANGAGE NATUREL.

III.1. Position du problème.

Les structures de données qui organisent le système symbolique ARCHES ont d'abord pour objet de représenter la signification des phénomènes visés. Elles définissent en quelque sorte la *représentation sémantique* des énoncés du langage naturel qui véhiculent ces phénomènes. Si l'acquisition et la compréhension automatiques du langage naturel étaient le but de notre travail, nous serions conduits à élaborer un système reliant de telles représentations sémantiques à des formes de surface par l'intermédiaire d'appareils transformationnels adéquats. A partir des critiques de la théorie "standard" des grammaires génératives (telle qu'on la trouve exposée dans CHOMSKY {33}), et à la suite des travaux de KATZ et FODOR sur l'élaboration d'une théorie sémantique {83}, de nombreuses recherches linguistiques ont tenté d'élaborer des méthodes et des modèles pour intégrer dans la description des phénomènes linguistiques la composante sémantique (citons par exemple {57}, {96} est une intéressante analyse dans {62}). Ces travaux reposent sur un postulat commun : Ils supposent que les informations linguistiques sont *décomposables* en unités de signification élémentaires, les *traits sémantiques*. De telles recherches théoriques en linguistique ont contribué à résoudre avec plus ou moins de succès le type de problème évoqué ci-dessus ; un grand nombre d'expériences en intelligence artificielle sur la compréhension du langage naturel ont développé, en les exploitant, un cadre d'analyse propre à l'informatique pour l'étude scientifique des langues naturelles ({119}, {121}, {132}, {133}). On trouve dans {13} et {52} deux analyses montrant en particulier les objectifs communs et les différences majeures des travaux entrepris par les informaticiens et les linguistes à propos de la représentation de la sémantique.

Cependant nos préoccupations de recherches actuelles ne sont pas centrées sur une approche informatique de la compréhension du langage naturel, bien que nous souhaitons représenter le contenu sémantique de la description des phénomènes visés. Notre objectif est de construire des structures de données permettant de représenter (voir § II.) : /1/ les relations d'état et les actants qui déterminent les énoncés élémentaires du langage naturel véhiculant les états élémentaires ; /2/ les connexions linguistiques qui composent les énoncés élémentaires et forment ainsi les énoncés véhiculant les états ou les changements d'état. Les choix qui fondent la représentation de ces différents éléments - relations d'état, actants, connexions linguistiques - définissent précisément les rapports qui existent entre le langage naturel et le système ARCHES.

III.2. Représentation des relations d'état.

La diversité sémantique des relations d'état est telle que leur nombre est très grand au niveau de la langue. Notre objectif est de concevoir dans ARCHES des structures de données qui permettent de les représenter par un nombre limité (et relativement restreint) d'éléments que nous considérons dans notre approche comme des items lexicaux profonds. Ce point de vue est commun à de nombreux travaux de nature expérimentale ; mais ses modalités d'application , différentes selon les recherches, dépendent des domaines empiriques investigués et des objectifs poursuivis (citons par exemple {4}, {67}, {78}). Il est évidemment fort différent, sinon opposé, de celui du linguiste dont l'objet d'étude est la description du fonctionnement de la langue sous tous ces aspects, c'est à dire phonologique, syntaxique, sémantique et pragmatique. Notre but est tout autre ; il vise à construire des représentations sémantiques en vue de leur manipulation dans un modèle informatique déterminé, et de leur insertion dans des modes de raisonnement mis en oeuvre par ce modèle.

Plus précisément pour déterminer le type de structure de données qui permet de représenter les relations d'état, nous faisons l'hypothèse que les énoncés élémentaires de surface peuvent être *paraphrasés* - de manière régulière et stable - de telle façon que le résultat de cette transformation soit à l'origine de l'élaboration de ce type de données. Nous entendons par paraphrase une transformation qui permet de passer d'une phrase-source P_s à une phrase cible P_c sans modifier le sens du

message véhiculé par P_s, c'est à dire que P_c a la même signification que P_s. (Pour une présentation introductive de l'étude linguistique de la paraphrase et de son rapport avec l'analyse de contenu voir {46} ; les relations qu'entretient la paraphrase avec la représentation sémantique et la signification sont particulièrement analysées dans {28}.).

Explicitons à l'aide de quelques exemples les mécanismes de cette transformation quand elle est appliquée à des énoncés décrivant des états ou des changements d'état.

Exemple 1.

"PIERRE DORT"

Cet énoncé paraphrasé de la manière suivante : "l'activité de Pierre est le sommeil" est tranformé symboliquement comme suit :

((ACTIVITE EST sommeil)DE Pierre)

En appliquant le même processus de transformation à l'énoncé plus complexe :

"PIERRE DORT PAISIBLEMENT"

nous obtenons le résultat :

((ACTIVITE EST (sommeil (QUALITE EST paisible))) DE Pierre)

Exemple 2.

"LE MANTEAU EST ROUGE CLAIR"

Cet énoncé a deux transformées du fait de l'ambiguïté du langage naturel.

α) Cet énoncé signifie : le manteau est rouge, le rouge est clair. Il se transforme comme suit :

((COULEUR EST(rouge(TEINTE EST clair))DE manteau)

β) Cet énoncé signifie : le manteau est rouge, le manteau est clair. Il se transforme comme suit :

(((COULEUR EST rouge)(TEINTE EST clair))DE manteau)

Exemple 3.

"L'AMPHORE EST SITUEE DANS LE FOUR"

Cet énoncé paraphrasé par la proposition "la localisation de l'amphore est dans le four" est transformé symboliquement comme suit :

((LOCALISATION DANS four)DE amphore)

Nous pouvons constater sur ces exemples que la transformation, que nous définissons, apparait comme une "opération de canonisation" des relations d'état. Ces dernières sont supposées être représentées par des entités sous-jacentes composées de deux termes : d'une part un "substantif d'état" qui définit à un niveau générique la *portée sémantique* des relations d'état (par exemple : dormir est une ACTIVITE, paisiblement est une QUALITE, être situé dans est une LOCALISATION, etc.) ; d'autre part un *relateur d'état* qui précise à un niveau spécifique les rapports qui existent entre ce substantif d'état et les actants-caractérisant (par exemple : relation d'appartenance exprimée par EST, nature de la localisation déterminée par DANS, etc.). L'actant caractérisant peut apparaitre explicitement dans l'énoncé de surface (exemple 2 et 3), ou être véhiculé implicitement par la relation d'état comme dans l'exemple 1. Dans ce dernier cas l'opération de canonisation a pour objet d'expliciter aussi la représentation de l'actant caractérisant. Compte tenu de cette hypothèse, la transformation d'un énoncé élémentaire de surface en un énoncé sous-jacent, adéquat aux structures de données que l'on se propose de construire, est définie par des règles formelles de réécriture représentées par la figure 2.1..

III.3. Représentation de la négation.

Les règles de transformation, définies ci-dessus, ont été élaborées à partir d'exemples représentant des descriptions d'état véhiculées par des énoncés affirmatifs. Or dans le cas général, les aspects descriptifs des phénomènes étudiés peuvent être exprimés aussi bien par des énoncés affirmatifs que par des énoncés négatifs. C'est là un critère important dont la prise en compte dans la conception du système ARCHES augmente dans des proportions intéressantes la puissance de sa représentation

α) Représentation des énoncés élémentaires de surface.

<ENONCE-ELEMENTAIRE> ::= <ACTANT-CARACTERISE><RELATION-D'ETAT>|
<ACTANT-CARACTERISE><RELATION-D'ETAT><ACTANT-CARACTERISANT>

β) Règle de transformation des éléments de surface en éléments sous-jacents.

<RELATION-D'ETAT> ::= <SUBSTANTIF-D'ETAT><RELATEUR-D'ETAT>|
<SUBSTANTIF-D'ETAT><RELATEUR-D'ETAT><ACTANT-CARACTERISANT>

γ) Représentation de la transformation.

<ACTANT-CARACTERISE><RELATION-D'ETAT>[<ACTANT-CARACTERISANT>] ::=

<SUBSTANTIF-D'ETAT>DE<ACTANT-CARACTERISE><RELATEUR-D'ETAT><ACTANT-CARACTERISANT> ::=

(<SUBSTANTIF-D'ETAT><RELATEUR-D'ETAT><ACTANT-CARACTERISANT>)DE<ACTANT-CARACTERISE>

figure 2.1. - Représentation sous-jacente d'un énoncé élémentaire.

et de son activité inférentielle. Dans ce cadre, nous n'examinons que des énoncés négatifs du type "L'AMPHORE N'A PAS D'ANSE ", "JEANNE N'EST PAS VETUE D'UNE ROBE", "LE MANTEAU N'EST PAS BLEU FONCE", etc.. Nous ne cherchons pas à représenter et à interpréter les négations qui portent sur les actants caractérisés. Par exemple si P1 est un prisonnier, décrit partiellement par l'énoncé "LE PRISONNIER P1 N'EST PAS ENFERME DANS LA CELLULE C10", nous ne représentons pas tous les énoncés négatifs du type "P1 N'EST PAS UNE AMPHORE", "P1 N'EST PAS UN ANIMAL-MARIN", etc. En d'autres termes nous considérons que la représentation de la négation d'objets par rapport à des univers conceptuels auxquels ils n'appartiennent pas n'est pas pertinente eu égard aux hypothèses qui fondent notre recherche (voir § III.5., IV.2.).

Nous souhaitons représenter et traiter dans ARCHES la négation des relations d'état.

En prenant appui sur des travaux linguistiques et psycholinguistiques relatifs au fonctionnement et à la compréhension de la négation ({3}, {81}, {86}), nous nous intéressons plus précisément à la *représentation* de la négation et aux *interprétations* que possèdent cette représentation quand les relations d'état sont décomposées au niveau sous-jacent en substantifs et relateurs d'état. Ainsi comment interpréter l'énoncé "LE CUBE N'EST PAS A GAUCHE DU CYLINDRE ?". Selon la nature des relateurs d'état qui ont été définis, dans ARCHES nous pouvons par exemple inférer l'énoncé complexe "LE CUBE EST A DROITE DU CYLINDRE OU SUR CELUI-CI". Ces interprétations sont évaluées à partir de l'étude formelle de la *portée de la négation* - c'est à dire la partie de l'énoncé qui est interprétée comme étant niée - en distinguant les cas où celle-ci influence les substantifs d'état, les relateurs-d'état ou la représentation des actants caractérisants.

III.4. Représentation des autres connexions linguistiques.

Les connexions linguistiques, qui nous préoccupent ici, sont celles qui établissent des liaisons entre énoncés du langage naturel véhiculant des phénomènes représentant des états ou des changements d'état. Ces liaisons très diversifiées au niveau de la langue, permettent d'exprimer l'addition, la disjonction, la succession, la causalité, l'implication, etc.. Elles assurent certaines des articulations sémantiques et logiques des discours tenus à propos de ces phénomènes tant du point de vue descriptif qu'explicatif et prédictif.

L'énoncé "L'amphore a une panse arrondie et un col en forme de boudin" est obtenu à partir de la composition de deux énoncés élémentaires ("L'amphore a une panse arrondie", "L'amphore a un col en forme de boudin") au moyen de la conjonction ET qui exprime la simple addition de caractéristiques d'état attribuées au même actant ; cette interprétation du ET lui confère les propriétés formelles de commutativité et d'associativité. Par contre l'énoncé "Pierre quitte sa maison et se dirige vers son bureau" est le résultat de la composition de deux énoncés obtenue par l'intermédiaire de la même forme ET qui désigne dans ce cas la succession immédiate ; ce ET ne possède plus la propriété formelle de commutativité. Les deux énoncés "les mammifères ont des pattes ou des nageoires" et "théorie du système général ou théorie de la modélisation" sont composés à partir de la conjonction OU qui marque dans le premier cas une alternative exclusive et dans le second cas un rapport de quasi-équivalence. La conjonction causale CAR introduit la cause qui explique un effet comme le montre l'énoncé "Il gèle car la pression est égale à 76 cm de Hg et la température à 0°C.". Le même phénomène pourrait être décrit en termes d'implication à l'aide de la forme SI : "S'il gèle, c'est que la pression est égale à 76 cm de Hg et la température à 0°C"., ou même encore à l'aide de la forme ET : "Il gèle ET la pression est égale à 76 cm de Hg et la température à 0°C". (figure 2.2.). Ces différents énoncés illustrent quelques modes d'articulation fréquemment rencontrés dans l'analyse et la description des états. Ils montrent certaines des interprétations que peuvent posséder des formes de surface comme le ET, le OU, le CAR et le SI. A titre d'exemple, le tableau de la figure 2.2. montre la liste (non exhaustive) des valeurs logico-sémantiques de la conjonction ET. Ces valeurs désignent des significations profondes de la forme de surface ET. En particulier la valeur de succession, représentée par le connecteur ETS, est utilisée pour définir la représentation profonde de tout changement d'état qui exprime le passage immédiat d'un état initial à un état final :

<CHANGEMENT-D-ETAT> ::= <ETAT-i> ETS <ETAT-f>

Par exemple l'énoncé "PIERRE SE REVEILLE" peut être transformé,selon les règles définies dans le paragraphe précédent, en deux énoncés profonds reliés par le connecteur ETS :

(((ACTIVITE EST sommeil)DE Pierre) ETS ((ACTIVITE EST veille)DE Pierre))

Valeurs	Autres types de liaison	Exemples
d'addition		"Marie est belle et grande"
de succession	Puis Et puis	"Pierre a mangé et a bu"
d'enrichissement	Et encore Et aussi Et même	"L'enfant a reçu des cadeaux, et des plus beaux"
d'opposition	Et cependant Et toutefois Mais	"Le repas était bon et cher" (i.e. le repas était bon mais cher)
de conséquence	Et par conséquent Et par suite	"La température est inférieure à 0°C., et il fait froid".
prægnante		"Informatique et sciences de l'homme".
causalité	car puisque,...	"Il fait froid et il neige"
composition	En même temps que	"Ce pull est jaune et vert".
symétrie		"Pierre et Léa se marient".

figure 2.2. - Quelques valeurs de la conjonction ET.

De nombreux travaux linguistiques ont abordé, aussi bien d'un point de vue diachronique que d'un point de vue formel, l'étude du sens et des propriétés logiques de connexions exprimant ces types d'articulation ({2}, {46}, {53}, {128}). Ils montrent en particulier que la description sémantique et logique de ces connexions est éloignée de celle définie dans le cadre de la logique symbolique classique ; en conséquence cette logique ne permet pas dans le cas général de représenter et de traiter de telles connexions. Devant la relative faiblesse sémantique des opérateurs de la logique classique au regard des connexions linguistiques, nous faisons l'hypothèse que les règles de composition des structures de données organisant le système symbolique ARCHES sont fondées sur la définition formelle de connecteurs qui représentent le sens profond de certaines articulations logiques (figure 2.3.). Une même forme de surface peut donc être représentée par plusieurs connecteurs qui définissent ses différentes interprétations (supposées basiques et indépendantes les unes des autres) ; par exemple les valeurs d'addition et de succession de la forme de surface ET, sémantiquement indépendantes, conduisent à la définition de deux connecteurs. De même, une forme de surface peut être représentée par une combinaison logique de traits spécifiques et de connecteurs représentant d'autres formes de surfaces ; par exemple la forme MAIS peut être interprétée comme un ET, ayant valeur d'addition, coordonnant deux caractéristiques d'état repérées par des traits spécifiques qui indiquent que l'une exprime une notion négative et l'autre une notion positive. Cette démarche montre que les connecteurs ne sont plus définis comme des opérateurs construisant des fonctions de vérité, mais comme construisant des compositions de structures de données. Nous expliciterons dans le paragraphe IV les hypothèses logiques sur lesquelles s'appuie ce mode de construction. Dans le cadre de notre travail nous nous sommes essentiellement intéressés à la représentation d'articulations logiques correspondant à certaines conjonctions de coordination ; cette approche nous a permis d'introduire dans ARCHES des liaisons logico-sémantiques "plus proches" du langage naturel que celles définies dans les logiques symboliques classiques.

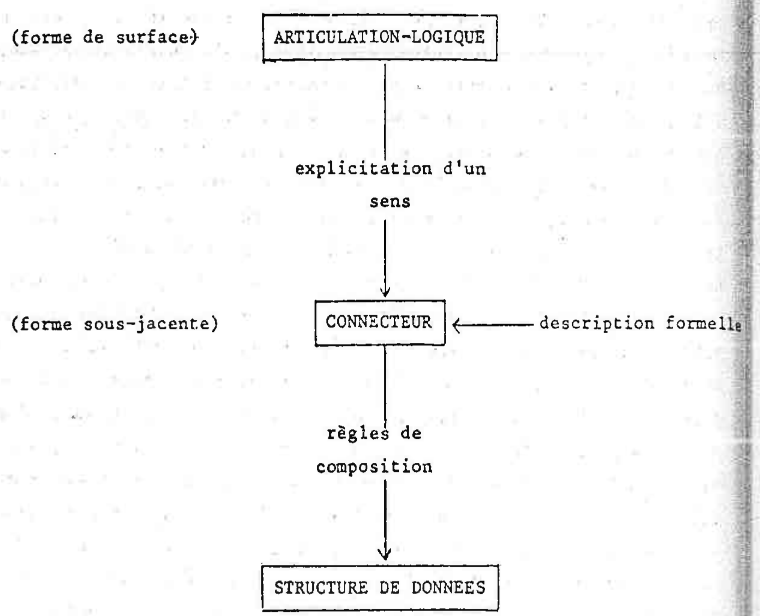


figure 2.3. - Modalités de représentation des articulations logiques.

III.5. Représentation des actants.

Les éléments de certaines théories du sens, qui ont été proposés aussi bien par des linguistes que par des logiciens (voir par exemple {26}, {94}), sont fondés sur des méthodes empiriques mettant en oeuvre deux principes irréductibles. L'élucidation du premier doit conduire à des méthodes permettant d'affecter une signification aux objets des langues naturelles, et d'une manière plus générale de comprendre les énoncés linguistiques où ils sont mis en jeu ; c'est *l'intention des objets*. Le second principe concerne l'acte de dénoter au moyen de la langue les objets que ceux-ci soient concrets - ils correspondent alors à des choses observables -, ou abstraits - ils correspondent alors à des idées élaborées indépendamment de toute réalité physique ; c'est *l'extension des objets*.

Les actants sont des objets particuliers dont la signification est déterminée par la nature des relations d'état qu'ils entretiennent entre eux. L'expression sous-jacente de ces relations en termes de substantifs et de relateurs d'état permet de représenter la sémantique de la description des actants, qui caractérise précisément leur nature intentionnelle. Dès lors se pose à propos de la représentation des actants une deuxième question : "Comment les référencer ?", ou en d'autres termes "Comment caractériser leur nature extensionnelle ?".
Les actants désignent des noms d'objets ou des idées générales. Dans les exemples "PIERRE DORT" et "CE PRISONNIER EST ENFERME DANS LA CELLULE C10", les objets "Pierre" et "Cellule C10" dénotent des noms propres, alors que "Ce prisonnier" désigne un nom commun faisant référence à un objet précis, soit P1, précédemment défini. Par contre dans l'exemple "L'HOMME EST MORTEL' nous sommes confrontés à un jugement émis à propos de l'objet universel "L'homme" (i.e. tous les hommes sont mortels). Nous supposerons donc avec {59} que les noms d'objets se réfèrent à des idées générales et abstraites qui elles-mêmes sont exprimées par l'intermédiaire de la notion de *concept*. Les objets appartiennent toujours à un concept ("PIERRE est un homme", "C10 est une cellule", "P1 est un prisonnier", etc.), les concepts pouvant eux-mêmes être reliés à d'autres concepts ("l'homme est un animal", etc.). Nous devons remarquer qu'un concept est toujours prédicatif, alors qu'un nom d'objet ne l'est jamais. Pour représenter les actants nous faisons donc l'hypothèse que les représentations de leur désignation véhiculent les concepts auxquels ils se réfèrent. Ce mode de représentation a l'avantage de référencer les actants par rapport à leurs univers conceptuels en les considérant comme éléments des ensembles d'individus qui représentent ces univers. Il permet ainsi d'établir entre les différents énoncés décrivant les états une relation de généralité-spécificité. Cette relation organise les concepts selon un graphe ne possédant pas de cycles (voir § I.3. et III.3., chapitre troisième) : un terme - sommet du graphe - peut être considéré comme contenant la même information sémantique que celle des termes auxquels il est relié, plus une information particulière (voir § IV.1., chapitre huitième). Du même coup la compréhension de chaque énoncé est d'autant plus forte que cet énoncé se trouve inséré dans un réseau conceptuel plus vaste. Ce réseau représente le contexte sémantique qui détermine les modalités d'interprétation de cet énoncé.

Cependant nous distinguons dans le système ARCHES une autre catégorie d'actants ; ceux qui ont une certaine compréhension, mais qui n'ont pas d'extension dans l'absolu. Nous qualifierons d'*informations sans extension* de tels actants. Par exemple ROUGE et AMABILITE EXCESSIVE n'ont aucune extension, mais "LA ROBE EST ROUGE" ainsi que "CES ENFANTS SONT D'UNE AMABILITE EXCESSIVE" représentent deux classes particulières de robes et d'enfants. Cette situation se rencontre essentiellement (et même exclusivement si on ne s'intéresse qu'à l'étude de phénomènes d'état relevant de domaines concrets) pour la représentation des actants caractérisants. En particulier dans certains énoncés de surface, ces derniers n'apparaissent pas explicitement ; ils sont contenus dans la relation d'état elle-même (voir § III.2.). Ainsi dans les exemples "PIERRE DORT" et "JACQUES SE REVEILLE" seuls les actants caractérisés PIERRE et JACQUES sont explicitement désignés dans les énoncés. C'est la transformation de la relation d'état en termes de substantif et de relateur d'état qui fait apparaître au niveau sous-jacent la représentation des actants-caractérisants. Dans le premier exemple, il s'agit de l'information sans extension sommeil ; dans le deuxième exemple, de la composition des informations sans extension sommeil et éveil. D'une manière plus générale, nous supposerons dans ce cas que les actants sont représentés par des unités d'information primitives ou par la composition des unités d'information primitives qui les définissent au niveau sous-jacent.

III.6. Hypothèses dans ARCHES.

L'analyse rapide, que nous venons de faire à propos des rapports que nous souhaitons établir entre le système ARCHES et certains aspects du langage naturel, nous conduit à formuler deux hypothèses sur la conception des structures de données organisant ARCHES.

Hypothèse 2. *Les structures de données sont construites à partir des éléments - substantif d'état, relateur d'état , interprétation de la négation et représentation des connexions linguistiques qui fondent la représentation sous-jacente des énoncés du langage naturel véhiculant les états et/ou les changements d'état.*

Hypothèse 3. *Les structures de données sont organisées autour de la notion de concept et/ou de construction de concepts, ce qui permet non seulement de référencer les actants - nature extensionnelle des concepts -, mais aussi de représenter la sémantique de leur description - nature intentionnelle des concepts -.*

IV. RAPPORTS AVEC LA LOGIQUE.

IV.1. Position du problème.

La conception de tout système de représentation de connaissances concerne deux problèmes solidaires : d'une part la représentation de la description du système ; d'autre part l'activité inférentielle que l'on souhaite mettre en oeuvre dans ce système. Les représentations sont des formalismes permettant de véhiculer la signification des phénomènes étudiés. La même connaissance peut conduire à des représentations différentes, mais la capacité du système à résoudre les problèmes pour lesquels il a été conçu est fortement influencée par le choix de la représentation. En d'autres termes quelle est la puissance logique de ces formalismes par rapport aux modes de raisonnement élémentaires dont sont dotés les systèmes de représentation de connaissances ? La problématique de la conception de ces systèmes abordée sous cet angle débouche tout naturellement sur l'examen des rapports qu'entretiennent les systèmes logiques généraux ou ad-hoc, classiques ou non classiques avec les éléments les organisant. Dans la perspective dans laquelle nous nous plaçons - à savoir définir un système informatique capable de réaliser les objectifs assignés au système ARCHES -, nous devons nous intéresser à ces rapports essentiellement sous l'angle du calcul et de la construction d'algorithmes, si nous ne voulons pas que les formalismes soient uniquement des systèmes de notation des connaissances. Les travaux logiques qui ont été développés dans cette direction concernent principalement deux types de méthodes : d'une part la méthode de résolution en logique du premier ordre ({30}, {115}) qui a conduit à des langages de programmation comme PROLOG ({116}, {129}) ; d'autre part la méthode des schémas de dérivation du "style GENTZEN" ({50}, {99}) qui a conduit très tôt à l'implémentation d'un système de preuves de théorèmes ({131}) et qui a contribué à développer des techniques de raisonnements formels ({7}).

Nous allons examiner successivement, dans le cadre de notre travail, ces deux points de vue que sont la représentation et le calcul, analyser leurs relations avec la logique, et montrer quelles sont les contraintes logiques que nous nous sommes imposées dans la conception du système ARCHES.

IV.2. Formulation logique des descriptions.

Logique et systèmes de représentations de connaissances sont des formalismes différents pour représenter les connaissances. Si des logiciens ont étudié d'un point de vue strictement logique des phénomènes langagiers ({112}), des informaticiens ont de leur côté emprunté à la logique certains de leurs éléments théoriques pour les intégrer dans la conception de tels systèmes ({49}, {56}, {122}). Examinons à partir de quelques exemples les problèmes que posent la formulation logique de la description des phénomènes que nous souhaitons représenter dans ARCHES.

Exemple 1.

Considérons un ensemble $\mathcal{E}$ d'individus décrits uniquement par des propriétés. Chaque propriété est représentée par un prédicat à une place d'argument dont la signification dépend de l'application ("ETRE-MORTEL", "ETRE-UN-OISEAU", "ETRE-PARESSEUX", etc.). L'ensemble $\mathcal{E}$ donne naissance à une base de données composée de formules atomiques du type $P^i_j(x_i)$ telle que :

$$\begin{cases} x_i \in \mathcal{E} \\ P^i_j \text{ est une propriété de } x_i \end{cases}$$

Supposons que sur cette base, extrêmement simple, nous définissions des opérations qui peuvent porter indifféremment sur les individus ou sur les propriétés. Ainsi les prédicats peuvent avoir le statut de variable comme le montre la question suivante : "Etant donné deux individus x1 et x2, ont-ils la même propriété P ?". Une telle question en langage naturel conduit à la représentation logique ci-après :

$$(\exists P)(P(x1) \wedge P(x2)) \ ?$$

dans laquelle la quantification porte sur le prédicat P. Nous sommes alors amenés au cours de la consultation de cette base à opérer sur des fonctions prédicatives du second ordre ; nous devons donc mettre en oeuvre une procédure de résolution relevant de la logique des prédicats étendue.

Exemple 2.

Soit l'énoncé "LA ROBE EST ROUGE CLAIR" ; nous souhaitons le représenter par une expression logique à partir des deux prédicats primitifs "ETRE-ROUGE" et "ETRE-CLAIR". Pour ce faire nous supposons que le prédicat ETRE-CLAIR est une propriété qui peut caractériser aussi bien le concept ROBE que l'information sans extension ROUGE. Nous pouvons donc interpréter cet énoncé de la manière suivante : "LA ROBE EST ROUGE, ET LE ROUGE EST CLAIR". L'expression

$$\mathrm{ROUGE}(x) \wedge \mathrm{CLAIR}(\mathrm{ROUGE}(x)) \qquad (1),$$

dans laquelle on remplace la variable individuelle x par la constante "LA ROBE", est la représentation logique de cet énoncé. Rappelons que les prédicats d'ordre 1, 2,...,ω sont classés en type, selon la nature de leurs places d'arguments, comme suit : /1/ toute variable individuelle est dite de type i ; /2/ si $P(x_1, x_1, \ldots, x_n)$ est une variable prédicative de n places d'arguments et si ses arguments sont de type $a_1, a_2, \ldots, a_n$, alors le type de $P(x_1, x_2, \ldots, x_n)$ est $(a_1, a_2, \ldots, a_n)$. Ainsi le prédicat ROUGE est de type (i), celui du prédicat CLAIR étant ((i)). CLAIR est donc un prédicat de prédicat. Nous sommes confrontés à une telle situation toutes les fois que nous nous interessons à représenter des propriétés de propriétés, et d'une manière plus générale des relations d'état de relations d'état. Par ailleurs cette représentation est relativement pauvre car elle ne permet pas de véhiculer toute la sémantique contenue dans ce type d'énoncé. En effet, comment inférer de (1) l'énoncé "LA ROBE EST CLAIR" ? Il faudrait introduire explicitement dans l'expression (1) les relations logiques qui existent entre les prédicats ROUGE et CLAIR ; relations de type nécessairement plus complexes qui compliqueront encore la représentation d'un énoncé aussi simple.

Exemple 3.

Donnons enfin une représentation logique des deux énoncés ci-après :

"LE NOMBRE D'AMPHORES EST CINQ" (1)

"L'AMPHORE A1 A DEUX ANSES" (2)

L'interprétation de ces énoncés conduit naturellement à affirmer qu'il est faux que chaque amphore (parmi les 5) soit caractérisée par le nombre 5, de même que chaque anse soit qualifiée par le nombre 2. Les nombres 5 et 2 sont des propriétés qui manifestement caractérisent les prédicats AMPHORE(x) et ANSE(y). Ils définissent des propriétés de prédicats. Tout nombre devra être représenté par une constante (ou une variable) prédicative d'ordre 2. Dans le cas présent les énoncés (1) et (2) conduisent aux représentations suivantes :

CINQ(AMPHORE)

DEUX(ANSES)

Ces formules sont synthétiques. Les prédicats d'ordre 2 définissant les nombres peuvent en fait être exprimés entièrement en termes de symboles logiques dont les arguments sont des variables individuelles.

L'énoncé (1) peut être paraphrasé comme suit : il y a cinq objets différents x1,x2,x3,x4,x5 qui sont des amphores et quel que soit z si z est une amphore alors z est l'un de ces objets. On a donc la définition suivante :

$$\mathrm{CINQ}(\mathrm{AMPHORE}) : \exists x1x2x3x4x5 \Big[\bigwedge_{\substack{i=1,4 \\ j=i+1,5}} \neg\equiv(xi,xj) \wedge \bigwedge_{i=1,5} \mathrm{AMPHORE}(xi) \wedge \forall z(\mathrm{AMPHORE}(z) \supset \bigvee_{i=1,5} \equiv(z,xi)) \Big]$$

De même l'énoncé (2) conduit à l'expression ci-après :

$$\mathrm{DEUX}(\mathrm{ANSES}) : \exists xy \Big[\neg\equiv(x,y) \wedge \mathrm{ANSE}(x) \wedge \mathrm{ANSE}(y) \wedge \forall z(\mathrm{ANSE}(z) \supset \equiv(z,x) \vee \equiv(z,y)) \Big]$$

Nous devons remarquer que la constante prédicative d'identité $\equiv$ est cependant elle-même définie à l'aide d'une variable prédicative d'ordre 1 : On dira que les individus x et y sont identiques si et seulement si n'importe quelle propriété P qui est vraie pour x, l'est aussi pour y :

$$x \equiv y : (\forall P)(P(x) \Longleftrightarrow P(y))$$

Les représentations logiques (au niveau analytique) des énoncés (1) et (2) sont donc déterminées par des expressions logiques complexes qui mettent encore en jeu des prédicats d'ordre supérieur à 1.

Nous pourrions à dessein multiplier les exemples de formulation logique des descriptions - en s'appuyant sur des énoncés aussi simples que ceux analysés ci-dessus - pour constater que : /1/ les représentations logiques des phénomènes, comme les états ou les changements d'état, conduisent à des expressions qui mettent en jeu des prédicats d'ordre supérieur à un ; cette situation se produit notamment (mais pas exclusivement, voir exemple 3) toutes les fois que les relations d'état ne portent pas sur les individus investigués, mais sur des relations qui les qualifient (exemple 2). /2/ La notation en termes de formules logiques classiques devient très vite complexe, surtout si on souhaite représenter toute l'information sémantique contenue dans les phénomènes étudiés en vue d'en déduire ultérieurement d'autres informations (exemple 3). Ces observations ne concernent pas uniquement la commodité de représentation et la facilité de manipulation des descriptions en termes de formules logiques classiques (i.e. expressions composées de formules atomiques). Elles sont plus fondamentales en ce sens qu'elles traduisent les rapports nécessaires qui existent entre les modes de représentation choisis et les catégories d'opérations qui peuvent être effectuées sur ces représentations (exemple 1). Dans ce cas nous savons depuis Gödel qu'il n'y a aucun système d'axiomes complet pour le calcul des prédicats étendu . En d'autres termes, quel que soit le système d'axiomes choisi, il y aura des formules valides qui ne pourront pas être dérivées ({89}). Si nous souhaitons bénéficier des travaux logiques qui ont un intérêt évident en informatique, nous devons nous intéresser précisément à ceux relatifs aux calculs algorithmiques (unification, généralisation de la notion d'affectation, résolution, etc.) qui ont été développés essentiellement dans le cadre de la logique du premier ordre (voir § IV.1.). Comme nous voulons rester dans ce cadre (pour les raisons évoquées précédemment)

nous concluons que le mode de représentation des phénomènes visés en termes d'expressions logiques (composition de formules atomiques définissant des propriétés ou des relations d'ordre $\geqslant 1$) n'est pas adéquat à ce cadre. Aussi notre hypothèse consiste à affirmer qu'il est nécessaire de concevoir un mode de représentation de la description des faits différent de celui exprimé par des formules logiques classiques. Ce mode de représentation sera déterminé dans le système ARCHES à partir de l'élaboration de structures de données basiques et composées qui permettent de décrire les actants-caractérisés. A chaque actant-caractérisé est ainsi associée une description définie en termes de structures de données ; l'attribution de ces descriptions aux actants caractérisés étant déterminée par des liens logiques spécifiques. En d'autres termes *les structures de données dans ARCHES sont considérées comme les arguments fonctionnels de formules atomiques particulières dont les symboles de prédicats (i.e. les liens logiques) sont des constantes*. La définition et le principe de résolution des règles de composition de ces structures permettront alors de représenter et de traiter les problèmes du système ARCHES dans le cadre de la logique du premier ordre (voir § IV.3.).
Par ailleurs, nous avons fait l'hypothèse que les actants caractérisés sont référencés par rapport aux univers conceptuels auxquels ils appartiennent. Là aussi leur représentation logique conduit à des formules d'ordre supérieur à un ; ceci se produit toutes les fois qu'un concept est relié à d'autres concepts (voir § III.5.). Pour rester dans le cadre que nous nous sommes fixés (i.e. logique du premier ordre), nous empruntons aux logiques à N champs leur méthodologie pour organiser le système ARCHES ({77}, {85}, {130}).

IV.3. Formulation logique des traitements.

Notre objectif est de pouvoir réaliser sur la classe des phénomènes réels, adéquate au système ARCHES, des modes de raisonnement conformes à cette réalité. Les méthodes de traitements que nous souhaitons mettre en oeuvre doivent donc rester proche, dans la mesure du possible, du "raisonnement réel" (c'est à dire celui qu'on utilise dans les sciences expérimentales) : *On part de suppositions ou d'hypothèses pour en tirer ensuite des conséquences que l'on rend indépendantes de hypothèses de départ par des règles appropriées.* Cette position de principe est tout à fait en accord avec les hypothèses que nous avons déjà faites à propos des modalités de construction des structures de données organisant ARCHES. Par exemple nous avons observé que les connecteurs représentant les connexions linguistiques seront définis de manière constructive (voir § III.4.), en s'appuyant précisément sur ce type de raisonnement. A partir d'hypothèses sur des types de structures de données, nous inférerons de nouvelles structures obtenues par composition ou décomposition des prémisses ; ces opérations étant réalisées respectivement par l'insertion ou l'élimitation de connecteurs. Le sens de ces connecteurs est entièrement déterminé par de tels schémas de dérivation. Ces derniers doivent être interprétés comme des conventions fixant les conditions d'emploi des connecteurs et, par là, déterminant les significations que les connecteurs peuvent recevoir dans le système ARCHES. Ce mode de définition des connecteurs - différent de celui rencontré dans les méthodes axiomatiques - permet d'exprimer de manière plus naturelle (et probablement de manière plus simple car les structures de données basiques ne sont pas de simples formules atomiques, voir § IV.2.) le sens profond des connexions linguistiques que nous souhaitons représenter dans ARCHES. Par ailleurs, une telle méthode a l'avantage de laisser le système "ouvert". En effet, nous pouvons ajouter dans le système en évolution des schémas définissant de nouveaux connecteurs. Naturellement ces extensions doivent être telles que les mécanismes de raisonnement mis en oeuvre dans le système soient toujours corrects. En d'autres termes nous devons nous assurer que d'un point de vue logique les formes élémentaires de raisonnement et leur organisation restent valides. Les méthodes axiomatiques poseraient des problèmes plus complexes en ce sens qu'il faudrait construire un autre système formel avec un nouveau système d'axiomes intégrant la définition des nouveaux connecteurs.

D'une manière plus générale les traitements que nous voulons mettre en oeuvre dans ARCHES sont fondés sur des *règles d'association formelles des structures de données* qui l'organisent (composition et décomposition définies également à partir d'autres éléments que les connecteurs, propriétés formelles des relations d'état, comparaison de sous-structures, etc. Déterminés par des catégories d'opérations logiques et leur organisation formelle, ils définissent les schémas de dérivation représentant les figures logiques incarnant les modes de raisonnement que nous visons. La formulation logique de ces schémas de dérivation s'inspirent des travaux de GENTZEN dans lesquels ce type de raisonnement, connu sous le nom général de *Méthodes de déduction naturelle*, est utilisé en particulier pour formaliser la déduction dans des systèmes logiques du premier ordre. Ces méthodes sont organisées autour de trois éléments essentiels : les énoncés de conséquence, les schémas d'axiomes et enfin les règles de dérivation ({7 }, {50}, {99}). Elles ont été généralisées par PORTE sous le nome de systèmes logistiques et systèmes déductionnels ({109}).

Ainsi *ARCHES est un système symbolique spécialisé* dont la conception repose sur une telle méthode. Il est composé d'un ensemble $\mathcal{T}$ de *thèses initiales* stockées dans une base de connaissances. Cet ensemble, qui décrit pour chaque application l'univers étudié, possède les propriétés suivantes : /1/ Toute thèse initiale est une formule bien formée du système ; /2/ L'ensemble $\mathcal{T}$ est consistant (ou non-contradictoire), c'est à dire que si $T_i \in \mathcal{T}$ sa négation n'appartient pas à $\mathcal{T}$ (nous supposons que la négation est un connecteur défini dans ARCHES). Le système ARCHES peut alors déduire des théorèmes à partir de cette base de connaissances grâce à des *règles d'inférence particulières* qui déterminent deux modes de raisonnement : le raisonnement déductif et le raisonnement analogique. Pour opérer ces déductions nous formalisons l'activité déductive de ARCHES en utilisant le calcul des prédicats qui sera donc considéré ici comme un méta-système. Le choix de ce méta-système est intéressant et commode car il permet l'emploi de méthodes et d'outils existants (unification, principe de résolution, langage de programmation du premier ordre comme PROLOG, etc.).

IV.4. Hypothèses dans ARCHES.

Nous avons montré dans les paragraphes précédents la nature des contraintes logiques que nous nous sommes imposés dans la conception du système ARCHES. Naturellement ces contraintes sont d'abord liées à des considérations théoriques, concernant en particulier les problèmes de complétude et de décision. Comme le formalisme que nous souhaitons élaborer n'est pas uniquement un simple procédé de notation, elles sont également liées à des problèmes de fonctionnalité et d'opérationnalité du système informatique. Nous entendons par là pouvoir bénéficier des résultats logiques conduisant à des instruments de calcul et de programmation. Dans ces conditions, le cadre logique sur lequel prend appui la construction formelle de ARCHES est déterminé par les hypothèses ci-après :

Hypothèse 4. *La construction des structures de données doit conduire à un formalisme restant dans les limites de la logique du premier ordre.*

Hypothèse 5. *L'extension des actants faisant référence à des catégories différentes d'objets (individus,* *concepts, super-concepts, etc.), nous empruntons aux logiques à N champs leur méthodologie pour organiser le système ARCHES.*

Hypothèse 6. *Des techniques comparables aux méthodes de la déduction naturelle sont mises en oeuvre pour formaliser les modes de raisonnement élémentaires (de type hypothético-déductif) autorisés dans le système ARCHES.*

V. PRESENTATION GENERALE DE NOTRE RECHERCHE.

Ce paragraphe a pour objet d'exposer les différents éléments qui organisent notre ouvrage, et de montrer les articulations qui les relient aux hypothèses sous-jacentes à la conception du système symbolique ARCHES.

Le chapitre troisième présente l'*architecture du système symbolique ARCHES* qui est déterminée à partir de la notion de *concepts* (voir § III.6. ; hypothèse 3). Les concepts, qui sont des ensembles, permettent de classifier les faits étudiés en différentes catégories d'objets appelés *individus*. Chaque individu peut être caractérisé par une *description* à l'aide du *lien ADP (attribution de description principale)* ; et chaque fait est représenté par une formule de la logique du premier ordre, appelée *structure*, donnant une description d'un individu (voir § IV.4. ; hypothèse 4). Les *liens INS* et *SET* permettent de structurer les ensembles de concepts et d'individus qui sont des données de ARCHES, et dépendent donc des applications considérées. En particulier la relation SET, qui définit sur l'ensemble des concepts un ensemble de graphes disjoints - *les graphes de résolution* -, permet d'organiser les faits enregistrés dans ARCHES en *champs* et en *domaines* (voir § IV.4. ; hypothèse 5). Un champ contient les structures qui décrivent les individus appartenant à un même concept ; et un domaine est formé de l'ensemble des champs associés aux concepts qui appartiennent à un même graphe de résolution. Enfin nous démontrons à propos de ces différentes notions quelques propriétés qui déterminent d'une part la cohérence intrinsèque du système symbolique ARCHES, et d'autre part les opérations de fusion de plusieurs systèmes ARCHES.

Le chapitre quatrième a pour objet essentiel de définir formellement la notion de descriptions (qui a été introduite dans le chapitre précédent). Plus précisément, il décrit les *caractéristiques du système formel* S_Δ qui expriment les deux aspects *Représentation* et *Déduction* des descriptions qualifiant les individus.

Une description est formée d'*éléments de description* ou *termes descripti*
reliés par des *connecteurs* qui sont définis par des règles comparables à celles mises en oeuvre par les techniques de déduction naturelle (voir § IV.4. ; hypothèse 6) : ce sont le ET d'addition (*), le OU non exclusif (+) et la négation (¬) (voir § III.6. ; hypothèse 2). Les term descriptifs permettent de représenter les propriétés et d'une manière pl générale les relations d'état qui caractérisent les individus (voir § II. hypothèse 1). Ils sont construits à partir de quatre entités basiques : le trait, la classe, l'opérateur et le symbole fonctionnel $. *Les traits* permettent de représenter les caractères distinctifs des individus. Ceux de même nature sémantique sont regroupés dans des ensembles nommés *class* Les symboles de classes expriment la portée sémantique des relations d'état qui sont attestées dans les descriptions élémentaires. Les rappor qui existent entre les classes et les traits sont exprimés par des symboles fonctionnels particuliers, en général n-aires, appelés *opérateurs*. Les opérateurs permettent de spécifier la nature sémantique des relatio d'état qui caractérisent les individus (voir hypothèse 2). Les trois entités traits, classes et opérateurs sont des données de ARCHES, et dépendent donc des applications considérées. Par ailleurs les propriétés et les relations d'état peuvent être *décrites localement* par l'intermédiaire du *lien ADL (attribution de description locale)* représenté par *le symbole fonctionnel* $.
La structure algébrique de l'ensemble Δ des descriptions est définie d'une part à partir des propriétés logiques des termes descriptifs, et d'autre part à partir des règles de formation des descriptions au moyen des connecteurs et des termes descriptifs. La représentation et les propriétés logiques des termes descriptifs permettent en effet de définir sur ces derniers trois types de règles de réécriture, notées par le symbole $\xrightarrow{\circ}$: *les règles de décomposition, les règles d'héritage et de transitivité* et enfin *les règles d'extension*. Ces règles expriment les pr tés sémantiques des classes. D'une manière générale, on se donne pour toute application plusieurs règles de réécriture du type précédent. Ces règles déterminent le système de règles de substitution-réduction du système ARCHES, à partir duquel sont produites les transformations des termes descriptifs. Ces transformations sont déterminées par la *relatio de réécriture* $\xrightarrow{*}$ définie sur l'ensemble des termes descriptifs comme la fermeture transitive et réflexive de la relation $\xrightarrow{\circ}$. Enfin on définit sur l'ensemble Δ *une relation de déduction, notée* $\Longrightarrow$, qui fixe les modalités de dérivation des descriptions.

$\Longrightarrow$ est la plus petite relation réflexive et transitive qui vérifie un ensemble de conditions permettant d'établir les propriétés logiques des connecteurs ainsi que les schémas de dérivation des descriptions (voir hypothèse 6) : règles d'insertion et d'élimination des connecteurs, rapports entre les relations $\xrightarrow{*}$ et $\Longrightarrow$, règles d'interprétation de la négation, etc. La définition et les propriétés de la relation de déduction $\Longrightarrow$ sont à la base de l'activité inférentielle du système symbolique ARCHES (voir chapitre septième).

Le chapitre cinquième étudie les propriétés du système formel S_Δ. Nous avons démontré que les résultats des opérations de transformation des termes descriptifs sont indépendants de l'ordre d'application des règles de réécriture. Ceci nous a permis de construire un *algorithme original de décidabilité pour la relation* $\xrightarrow{*}$. Cet algorithme est fondé d'une part sur une *procédure de recherche du terme descriptif irréductible* de tout terme descriptif V, et d'autre part sur une *fonction de discordance* de deux termes descriptifs V et W qui recherche le premier terme descriptif non décrit qui est différent dans V et W. Par ailleurs nous avons justifié formellement les schémas de dérivation des descriptions et montré que l'ensemble Δ, muni des trois connecteurs *, + et ¬, est un *treillis distributif complémenté*. Notons que l'intérêt essentiel de la négation réside dans son *interprétation* fondée sur le principe du "système de description clos" : tout élément de description caractérisant un individu quelconque ne peut pas être défini par la disjonction de tous les termes descriptifs construits sur la même classe. Et nous avons prouvé que l'ensemble Δ muni de cette interprétation est *consistant*. Enfin nous avons élaboré une *procédure de décision* permettant de résoudre le problème suivant : *Etant donné un couple de descriptions (H,C), déterminer s'il vérifie la formule* $H \Longrightarrow C$. Ce problème est évidemment essentiel pour la démonstration des théorèmes du système ARCHES (voir chapitre huitième et chapitre neuvième). La définition de cette procédure, qui s'appuie sur les propriétés formelles de la relation $\Longrightarrow$, utilise la méthodologie de résolution de problèmes par décomposition, et construction de graphes ET/OU correspondants. Plus précisément, cette procédure construit deux arbres ET/OU $\mathcal{A}_H$ et $\mathcal{A}_C$ associés respectivement à l'hypothèse H et à la conclusion C, les modalités de construction étant déterminées à partir des schémas de dérivation des descriptions et de leurs propriétés. Elle construit ensuite l'arbre ET/OU $\mathcal{A}_R$ en "accrochant" à chaque terminal de $\mathcal{A}_H$ l'arbre $\mathcal{A}_C$ sans sa racine ; et tente de valider la formule $H \Longrightarrow C$ en cherchant à valider

au moins un sous-arbre ET de $\mathcal{A}_R$ en utilisant en particulier l'algorit
de décidabilité de la relation $\longrightarrow^*$.

Le chapitre sixième a pour objet de définir formellement *les modalités d'évolution des descriptions* qualifiant les individus, et d'en étudier les propriétés logiques. Cette évolution est exprimée par l'intermédiai des connecteurs F et G ; le premier marquant les rapports d'évolution entre mondes de descriptions consécutifs - c'est le *"futur immédiat"* -, le second marquant les rapports d'évolution entre mondes de descriptio cessifs, c'est à dire non nécessairement consécutifs - c'est le *"futur* La définition formelle de ces deux connecteurs se fait dans le cadre d' *extension du système formel S_Δ*. Nous démontrons à partir de la caracté sation sémantique de l'évolution des descriptions que le système formel muni des connecteurs F et G reste consistant. Par ailleurs compte tenu de l'extension du système formel S_Δ, nous élaborons une *nouvelle procé dure de décision* pour la formule H $\Longrightarrow$ C. Cette procédure est déte minée non seulement à partir de la procédure de décision définie dans l chapitre cinquième, mais aussi à partir de propositions remarquables qu nous avons démontrées à propos des connecteurs F et G. Enfin les connec teurs $*$, $\neg$ et F permettent de définir le *ET de succession* (o) qui a une des valeurs de la conjonction PUIS du langage naturel. Il permet en par culier d'exprimer et de véhiculer des changements d'états (voir § II.4. hypothèse 1).

Le chapitre septième présente les caractéristiques formelles de représe tation et de manipulation du système symbolique ARCHES.
D'une part, ARCHES est muni d'un *langage des formules* - ou *structures* - noté $\mathcal{F}$, construit à partir d'un ensemble de base dont les éléments son issus du système formel S_Δ. Ses *thèses* forment un sous-ensemble partic $\mathcal{F}_T$ inclus dans $\mathcal{F}$ (i.e. ce sont des structures toujours vraies quel q soit les interprétations du système symbolique ARCHES) ; elles sont enr gistrées dans une *base de connaissances* et *décrivent en extension* l'en des individus étudiés.

D'autre part, l'activité inférentielle de ce système formel spécialisé est mise en oeuvre à partir d'une *relation d'inférence*, notée $\longmapsto$, dé sur $\mathcal{F}^n \times \mathcal{F}$. Cette relation d'inférence permet de représenter les for élémentaires de raisonnement au moyen de règles d'inférence particuli /1/ *Les règles d'inférence structurales*, qui dépendent de l'architectu générale du système ARCHES, mettent en oeuvre les raisonnements déduct et analogiques ; /2/ *Les règles d'inférence pragmatiques*, qui dépende des applications considérées, permettent de *décrire en intention* les

individus en définissant les lois générales qui les organisent. Ces règles permettent de *démontrer* les théorèmes du système symbolique ARCHES à partir de l'ensemble $\mathcal{F}_T$ des thèses. La formalisation du processus de démonstration de ces théorèmes nous conduit à utiliser un autre système formel permettant de manipuler non seulement les éléments spécifiques qui interviennent dans le déroulement des démonstrations, mais aussi les différentes composantes du système ARCHES. Compte tenu des propriétés formelles du système ARCHES (voir § IV.4. ; hypothèse 4), nous utilisons comme *méta-langage* la logique du premier ordre et plus particulièrement *les clauses de Horn*; ce qui permettra de plus l'utilisation d'outils existants, comme PROLOG par exemple.

Le chapitre huitième étudie *le raisonnement déductif*. Ce dernier est déterminé par deux règles d'inférence structurales : *la règle d'inférence intra-champ* qui est fondée sur la définition et les propriétés de la relation de déduction $\Longrightarrow$; et la *règle d'inférence inter-champs* qui est fondée sur la définition et les propriétés du prédicat SET. Pour prouver les théorèmes solidaires de ce mode de raisonnement, nous avons formalisé le processus de démonstration correspondant à partir d'une représentation par des clauses de Horn du système symbolique ARCHES. Cette représentation en logique du premier ordre a permis de définir *une règle de résolution pour le système ARCHES* équivalente à la règle d'inférence intra-champ. Cette règle utilise des *modalités particulières d'unification des descriptions* - la $\Longrightarrow$ - unification (lire flèche-unification) - pour laquelle nous définissons et prouvons un algorithme qui détermine l'ensemble des unificateurs de deux descriptions. Et nous avons démontré sa *complétude* à partir de la définition *d'arbres sémantiques* pour le système ARCHES. Par ailleurs l'application itérée de la règle d'inférence inter-champs détermine des *modalités spécifiques d'exploration des graphes de résolution* pour lesquelles nous établissons quelques propriétés et définissons un algorithme de cheminement. Enfin nous avons construit formellement et validé *le démonstrateur* qui met en oeuvre le raisonnement déductif dans ARCHES, démonstrateur qui est défini à partir de la règle de résolution et des modalités d'exploration des graphes de résolution.

Enfin *le raisonnement analogique* est étudié dans le chapitre neuvième. Il est déterminé à partir d'un *paradigme analogique* particulier exprimant la *ressemblance de rapports* entre descriptions dont la formulation la plus générale est : "La description A est à la description B ce que la description C est à la description D". Ce paradigme a une forme rigoureuse

(et limite) quand il représente une égalité de rapports définissant la proportion géométrique. Il permet ainsi de déduire au vue "de certaines ressemblances" une connaissance nouvelle (D) à partir de connaissances connues (A, B et C). Cependant la *validité logique* de cette inférence n'est pas postulée pour au moins deux raisons. D'abord la relation de ressemblance ne possède pas les propriétés logico-mathématiques de l'égalité, et donc le *principe d'identité* essentiel dans toute construction logique est mis en défaut. Ensuite cette inférence porte sur des univers de connaissances qui peuvent être *incomplets* sous le double point de vue de la caractérisation et de leur organisation (et dans ce cas le raisonnement déductif - et donc valide - n'est plus opératif). Aussi afin d'accroître *le degré de vraisemblance* du résultat obtenu, nous avons élaboré un ensemble d'hypothèses qui précisent notre paradigme analogique : non démonstrabilité de la description ¬D par le raisonnement déductif, méthode d'évaluation de la mesure de la notion de ressemblance entre deux descriptions quelconques, dérivation analogique par transfert d'éléments de description de B vers D, et enfin définition d'une relation de dépendance entre A et B d'une part et C et D d'autre part dont l'existence autorise ou interdit le déclenchement de l'inférence analogique. Ces hypothèses ont permis d'intégrer dans le système ARCHES *un modèle analogique défini comme une application particulière* qui fait correspondre les éléments (i.e. les opérateurs, les classes et les traits) qui composent A (respectivement B) à ceux de C (respectivement D). Cette application n'est définie que si les relations d'état (i.e. les doublets (classe, opérateur)) qui composent les descriptions B et D sont identiques, et que les descriptions A et C ont au moins en commun une relation d'état. Par ailleurs elle respecte la sémantique des traits, i.e. deux traits se correspondent si et seulement si ils appartiennent à la même classe. Si cette application est telle qu'elle représente localement *la fonction d'identité* pour certains des traits qui composent les descriptions A et B alors on affirme que "A est à B ce que C est à D". En d'autre termes le modèle permet d'exprimer une certaine ressemblance de rapports entre les couples (A,B) et (C,D). Si par ailleurs les éléments qui composent ces deux couples sont dans la même *relation de dépendance* alors on infère analogiquement D si on connait A,B et C.
Dans ce modèle la ressemblance est évaluée par la fonction d'identité ; en effet si cette application représente localement la fonction d'identité pour A (ou B) alors A et C (ou B et D) ont en commun des éléments de description. Cette ressemblance peut donc être évaluée de manière équiva-

à l'aide de la notion de filtre (ou de matching) bien connue en intelligence artificielle : nous l'avons exprimé dans ARCHES à l'aide de la relation de déduction $\Longrightarrow$, de la $\Longrightarrow$-unification et enfin de prédicats particuliers précisant la nature du filtrage. Ceci a permis de définir une *règle d'inférence analogique* dont les premisses ont pour objet essentiel d'une part d'évaluer *la ressemblance comme une relation de dérivation,* et d'autre part de *ne pas rendre inconsistant le système ARCHES.* L'application effective de cette règle dépend alors de l'existence de relations de dépendance entre les éléments des couples (A,B) et (C,D). Ces relations déterminent pour chaque interprétation du système symbolique ARCHES un *graphe de dépendance* qui conditionne l'utilisation de la règle d'inférence analogique. Ces modalités d'utilisation de la règle d'inférence analogique contribuent à produire des *solutions satisfiables,* i.e. des solutions qui sont vraies pour au moins une interprétation (celle qui a donné naissance au graphe de dépendance). Enfin nous avons construit formellement et validé le *démonstrateur* qui met en oeuvre le raisonnement analogique, démonstrateur qui est défini à partir de la règle d'inférence analogique et de ses modalités d'utilisation à l'aide des graphes de dépendance.
Pour démontrer par analogie un théorème de la forme T(y,D) le démonstrateur vérifie d'abord que la règle d'inférence analogique peut être appliquée en cherchant dans le graphe de dépendance s'il existe un sommet D' tel que D et D' soient $\Longrightarrow$ - unifiables. Puis il tente de construire, à partir des éléments qui sont reliés à D' dans le graphe de dépendance, les structures T(y,C) et P(x,A) telles que A et C se ressemblent au sens du modèle analogique. Enfin il cherche à prouver à l'aide du raisonnement déductif la structure P(x,B) en construisant B à partir de A et de D' de telle manière que B et A soient dans la même relation de dépendance que D et C, et que B et D se ressemblent au sens du modèle analogique.

CHAPITRE TROISIEME.

L'ARCHITECTURE DU SYSTEME SYMBOLIQUE ARCHES

-=-=-=-=-=-=-=-

I. NOTIONS DE CONCEPTS ET D'INDIVIDUS.

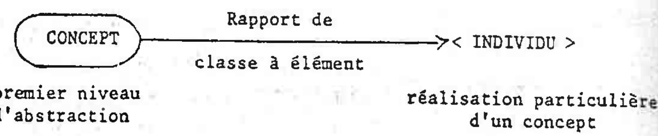
I.1. Représentation conceptuelle des phénomènes.

Les phénomènes dont la représentation est adéquate au système symbolique ARCHES portent sur la description d'objets (entités de la culture matérielle, assertions factuelles, évènements de types particuliers, etc. voir Hypothèse 1, § II.4., chapitre deuxième) qui ont une réalité sensible. Ces objets déterminent les actants caractérisés qui, d'après l'hypothèse 3 (§ III.6., chapitre deuxième), appartiennent à des univers sémantiques bien définis. Tout univers sémantique est repéré par une notion abstraite et générale. Ces notions définissent des *concepts* représentant des ensembles dont l'élaboration à partir des opérations d'abstraction et de généralisation explicite les aspects qui sont liés à leur *extension* et à leur *intention* : le premier aspect permet de référencer les objets, tandis que le second met l'accent sur les conditions logico-sémantiques que doit remplir la description de ces objets.

I.2. Représentation des individus.

I.2.1. Notion d'individus.

Les phénomènes qui sont véhiculés par le système ARCHES représentent les descriptions des éléments issus de l'extension des concepts. Ces éléments seront appelés *individus*. Par exemple, considérons un ensemble de N amphores qui sont l'objet d'une investigation particulière. L'extension du concept AMPHORE est représentée par un ensemble de cardinalité N dont les éléments définissent les individus de la base auxquels le concept AMPHORE s'applique. De même, dans le cadre d'une étude portant sur l'univers réel "REGIME-PENITENTIAIRE", supposons que l'un des objectifs soit l'examen, à des fins cognitives, de P prisonniers : qui les gardent ?, Dans quelles cellules sont-ils ?, Quels sont les motifs de leur incarcération ? , comment déterminer la nature de leurs peines ? etc. L'extension du concept PRISONNIER est également représentée par un ensemble de cardinalité P dont les éléments définissent les individus objet de l'investigation. Dès lors, se pose la question de la représentation et de la manipulation des individus. La notation choisie doit marquer d'une part les différences qui existent entre un concept et tout individu qu'il désigne, et d'autre part celles qui existent entre deux individus repérés par le même concept. Elle doit donc véhiculer explicitement le lien qui relie un ensemble (i.e. un concept) à un élément spécifique (i.e. un individu), comme l'indique le schéma ci-après :



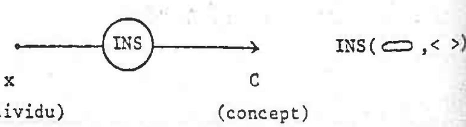
I.2.2. Définition du lien INS.

Si x est un individu appartenant à l'extension du concept C, on dira que x est une *instance* de C : INSTANCE représente le lien qui unit l'individu x au concept C. Ce lien, étiqueté INS, permet ainsi de dénoter dans ARCHES l' individu x par le terme fonctionnel INS(C,x).

Tout individu x peut appartenir simultanément à plusieurs concepts. Cette situation permet d'exprimer les différents points de vue sous lesquels les individus peuvent être analysés. Ainsi les termes fonctionnels INS(C_1,x) et INS(C_2,x) représentent le même individu x, mais considéré sous deux éclairages descriptifs différents déterminés respectivement par les concepts C_1 et C_2. Par exemple dans le règne animal une baleine peut être décrite d'abord comme un poisson, et ensuite comme un mammifère (voir § III.3. et III.4.). De même en Histoire de l'Art un même objet (i.e. un même individu) peut être considéré en même temps comme un mobilier, une sculpture et une peinture (voir à ce sujet {34} par exemple).

I.2.3. Représentation graphique.

Le terme fonctionnel INS(C,x) est représenté graphiquement par le schéma suivant :



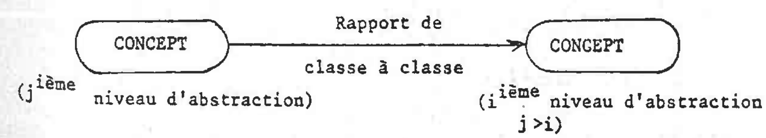
Remarque.

Il est évident qu'une telle notation permet de représenter et de manipuler sans ambiguïté les individus. Ainsi les deux expressions INS(CELLULE,C10) et INS(CELLULE,C21) véhiculent deux éléments différents appartenant à l'extension du concept CELLULE, le premier étant repéré par le nom propre C10, le second par le nom C21. Cependant, cette notation n'a pas une autonomie propre car les concepts désignés par la même graphie et repérant des univers sémantiques différents introduisent une ambiguïté quant à leur interprétation. Par exemple, les expressions INS(VOL,AF417) et INS(VOL,PERDRIX) identifient les individus AF417 et PERDRIX qui appartiennent aux extensions de deux concepts différents - c'est à dire définissant des univers sémantiques distincts -, mais repérés par la même graphie. Nous montrerons ultérieurement que la notion de domaine permet de lever cette ambiguïté (voir § III.4.).

I.3. Rapports inter-concepts.

I.3.1. Organisation des concepts.

Nous avons déjà observé que tout concept qui n'est pas "basique" est construit à partir d'une composition logique de notions définissant elles-mêmes d'autres concepts (basiques ou non) (voir § III.5., chapitre deuxième). Par exemple le concept MAMMIFERE a pour extension des éléments correspondant à des concepts comme VACHE, BALEINE, etc. ; de même le concept VERTEBRE a pour extension des éléments correspondant à MAMMIFERE, OISEAU, etc. Les concepts, qui représentent des ensembles, sont organisés dans ARCHES par une relation qui permet précisément d'exprimer leurs modalités de construction.



I.3.2. Définition de la relation SET.

Pour éviter toute confusion entre classe et élément et aboutir ainsi [à] un paradoxe (voir § III.2.3., chapitre premier), *nous supposons que [tout] concept peut être à son tour considéré comme un individu appartenant [à] un autre concept*. Par convention tout concept C_i sera alors noté $I(C_i)$ quand il a le statut d'individu ; il en résulte dans ce cas qu'il exi[ste] au moins un concept C_j tel que $I(C_i) \in C_j$.

$I(C_i)$ est un individu dont la description représente la description gl[o]bale du concept C_i ; il en résulte que $\forall x \in C_i$ l'individu x est décri[t] par cette *description générale*, lui-même étant décrit localement par [sa] *description spécifique* (voir § IV.2.2.a, chapitre septième). Ce sont ce[s] descriptions spécifiques qui permettent d'identifier tous les individ[us] appartenant à un même concept, et donc de les différencier.

Désignons par T l'ensemble fini des concepts qui organisent le systè[me] ARCHES ; nous définissons sur cet ensemble la relation SET comme suit :

Définition. On dit que deux concepts C_i et C_j sont reliés p[ar] la relation SET, notée $SET(C_i,C_j)$, si et seulem[ent] si $I(C_i) \in C_j$

$$C_i \xrightarrow{\text{SET}} C_j \qquad \text{SET}(\subset\!\supset , \subset\!\supset)$$

La relation SET exprime ainsi les modalités de construction du concept C_j en explicitant les aspects qui sont liés à son extension. Par exem[-]ple la relation SET(PERDRIX,OISEAU) montre que l'extension du concept OISEAU est formée d'individus correspondant à des concepts comme PERDRIX.

I.3.3. Axiomes.

La relation SET possède les deux propriétés axiomatiques suivantes :

Axiome 1. SET est antiréflexive :

$\forall X\ -SET(X,X)$

Axiome 2. SET est antisymétrique :

$\forall XY (SET(X,Y) \supset -SET(Y,X))$

Le premier axiome indique que $\forall A\ I(A) \notin A$; le second montre que $\forall A,B$ si $I(A) \in B$ alors $I(B) \notin A$ et réciproquement.

II. NOTION DE STRUCTURES.

II.1. Rapports des représentations des descriptions avec la logique à n champs.

Le système ARCHES manipule plusieurs catégories d'objets, chacune d'elles faisant référence à un univers sémantique particulier issu du découpage de l'univers réel investigué (Hypothèse 5, voir § IV.4., chapitre deuxième). Par exemple, on peut représenter simultanément dans ARCHES des objets de type AMPHORE et ceux de type FOUR ; de même, on peut décrire simultanément des individus issus respectivement des concepts BALEINE, VACHE, MAMMIFERE, etc. ARCHES se présente donc comme un système symbolique qui véhicule plusieurs types de variables ayant chacun un domaine de variation déterminé. Sa mission est de représenter, de formaliser et de traiter les descriptions des individus appartenant à ces différents domaines de variation.

Si la conception de ARCHES s'inspirait directement des systèmes logiques à n types de variables, tout élément de description serait représenté par un prédicat à k-places d'arguments, chaque place pouvant être remplie par un seul type de variable : $P(x_1,..,x_i,..,x_n)$, x_i étant une variable du ième type. Dans ce cas, on montre que l'on peut facilement se ramener à un système à un seul type de variable dont le domaine de variation est la réunion de tous les domainesde variation des n types de variables ; la façon la plus simple de résoudre ce problème est d'introduire n prédicats à une place d'argument $S_1,..,S_i,..,S_n$, {130}.
Une telle représentation nous conduit à faire deux remarques importantes : /a/ La traduction des descriptions dans un système logique à un seul type de variables (système classique) fournit des représentations véhiculées par des formules logiques relativement longues dont la manipulation serait complexe. Par exemple, le prédicat $LOC(x_a,x_f)$, représentant l'énoncé "x_a est localisé dans x_f" et ayant ses places d'arguments respectivement du type AMPHORE et du type FOUR, serait traduit par l'expression suivante : $(FOUR(y) \supset (AMPHORE(x) \supset LOC(x,y)))$. /b/ Le fait d'imposer à chaque place d'argument d'être remplie par un seul type de variables est une contrainte même pour des modalités de description très simplifiées. Par exemple on ne pourrait pas représenter à l'aide du prédicat $LOC(x_a,x_f)$ des énoncés du type "x_f est localisé dans x_r" dans lesquels x_r est une variable de type REGION. Lever cette contrainte conduirait naturellement à alourdir le formalisme.

Signalons enfin que cette représentation définit une structure formel[le] qui n'est pas adéquate au système ARCHES dont l'un des objectifs est [de] véhiculer des propriétés de propriétés ou plus généralement des relat[ions] de relations (Hypothèse 4, voir § IV.2. et IV.4., chapitre deuxième).

II.2. Le type "STRUCTURE" comme mode de représentation des descriptions.

II.2.1. La relation ADP.

Pour remédier à ces inconvénients, nous avons organisé le système de manière sensiblement différente. Notre premier problème est de repré-senter les descriptions des individus en distinguant les différents types de variables. Nous donnerons ultérieurement une définition pré-cise de la notion de description d'un individu en termes de structure[s] de données (voir § II.4., chapitre quatrième) ; désignons pour l'instan[t] par y une telle description. Soit par ailleurs x un individu apparte-nant à l'extension du concept C. La relation qui exprime la caractéri-sation de x par sa description y est définie dans ARCHES par un lien particulier qui précise la nature des rapports entre x et y. Ce lien, noté ADP *(Attribution de Description Principale)*, marque l'attributio[n] des descriptions (ou éléments de description) aux individus véhiculés par le système.

Ainsi la notation qui représente toute description y d'un individu x appartenant au concept C est définie par la relation ADP à deux place[s] d'arguments :

$$\mathrm{ADP}(\mathrm{INS}(C,x),y) \qquad (1)$$

II.2.2. Définition du type STRUCTURE.

Dans la suite nous supposons que la relation ADP(INS(C,x),y) sera no[tée] C(x,y).

Ceci se justifie du fait que d'une part les éléments ADP et INS sont invariants, et que d'autre part nous utilisons la méthodologie des l[o]giques à N champs pour organiser le système ARCHES (voir § III.).

Par ailleurs pour tout individu x, nous supposons que y est *unique* e[t] pourra donc être noté $\mathcal{D}(x)$ (car toute description est fonction des individus).

On appellera *structure* toute formule du type $C(x,\mathcal{D}(x))$.

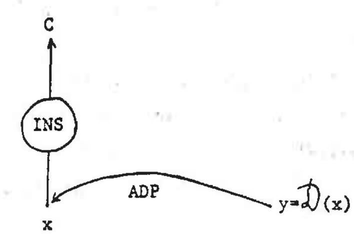
Les structures définissent dans ARCHES les modalités de représentati[on] des phénomènes dans les termes de formules donnant les descriptions

individus. Toute instance x_o du concept C conduit à la structure parti-culière $C(x_o,\mathcal{D}(x_o))$ (c'est donc une proposition) qui détermine la description effective $\mathcal{D}(x_o)$ de l'individu x_o.

Nous pouvons faire deux remarques à propos de ce mode de représentation de la description des individus : /a/ Le type des variables est véhicu-lé par la relation elle-même ; /b/ Les descriptions sont des termes fonctionnels dont la définition détermine leurs modalités de construc-tion (voir § II.4., chapitre quatrième). Ce mode de représentation per-met donc de décrire des individus de différents types, tout en manipu-lant des descriptions qui ne font pas sortir de la logique du premier ordre (Hypothèse 4) : il constitue l'ossature générale du système ARCHES.

[I]I.2.3. Représentation graphique.

Toute structure de la forme $C(x,\mathcal{D}(x))$ est représentée dans le système par le schéma suivant :



III. ORGANISATION GENERALE DU SYSTEME SYMBOLIQUE ARCHES.

III.1. Les deux modes d'organisation du système.

Le système ARCHES est un ensemble organisé de structures.

Les rapports entre les différentes structures sont définis de deux manières différentes. D'une part les liens existant entre les descriptions des individus représentés dans ARCHES déterminent un premier mode d'organisation que nous qualifierons de *dynamique* car elle est définie à posteriori par la nature des problèmes investigués. Les structures de données correspondantes fixent ainsi le cadre formel de ce mode d'organisation qui sera explicitée lors de leur élaboration (voir chapitre quatrième). D'autre part, les rapports existant entre les différents concepts déterminent un deuxième mode d'organisation que nous qualifierons de *statique* car elle est définie a priori par la relation SET (voir § I.3.). Ils fixent l'organisation générale du système ARCHES, que nous définissons ci-dessous à l'aide des notions de champs, de graphes de résolution et de domaines.

III.2. Notion de champs.

On appellera *champ* l'ensemble des structures qui représentent les descriptions de tous les individus appartenant à un même concept.

Le système ARCHES est donc composé d'autant de champs qu'il y a de concepts. Tout champ sera repéré par le nom qui désigne le concept associé aux individus qui le composent. En se référant aux exemples précédents, nous pouvons incarner ARCHES à l'aide des champs AMPHORE, FOUR, PRISONNIER, BALEINE, etc.

Les champs ont une organisation qui dépend de la nature des relations entre les descriptions des individus les composant ; il s'agit donc essentiellement d'une organisation dynamique. Ce même mode d'organisation peut naturellement mettre en rapport des champs à travers les liens qui existent entre leurs descriptions respectives (figure 3.1., elle montre un extrait de l'organisation dynamique des quatre champs AMPHORE, EPAVE, FOUR, REGION).

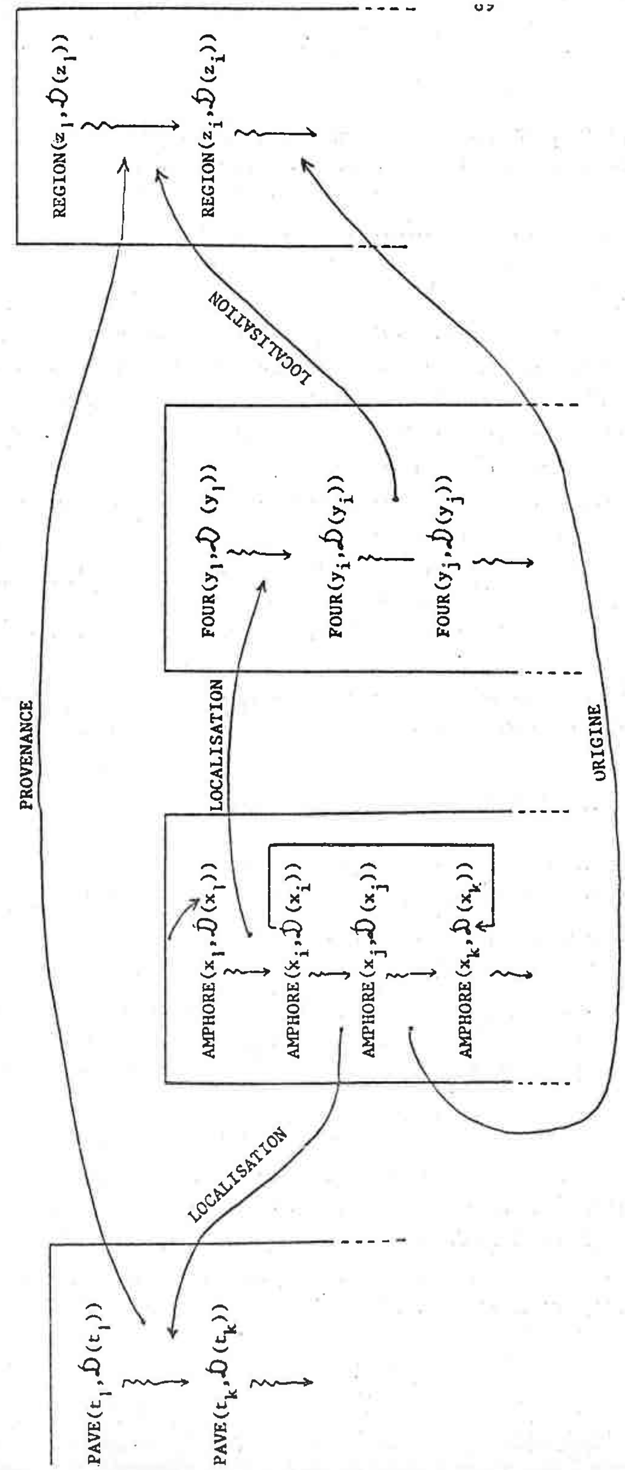


figure 3.1. - Schéma d'organisation dynamique des champs AMPHORE, EPAVE, FOUR et REGION.

III.3. Notion de graphes de résolution ou R-graphes.

Nous devons remarquer par ailleurs que les champs possèdent dans leur semble un autre mode d'organisation qui est précisément défini par la nature des rapports qui existent entre les différents concepts. Cette nisation est donc déterminée par le graphe de la relation SET, qu'on mera GARCHES (voir § I.3.2.). Par définition c'est la partie de TxT déf en compréhension comme suit :

$$\{(x,y) \mid SET(x,y) \text{ et } x,y \in T\}$$

Les sommets de GARCHES peuvent se répartir en quatre catégories selon valeur de leurs demi-degrés extérieurs ou intérieurs (on rappelle que demi-degré extérieur $d^+(x_i)$ du sommet x_i est égal au nombre d'arcs issu de x_i, tandis que le demi-degré intérieur $d^-(x_i)$ du sommet x_i est égal au nombre d'arcs aboutissant en x_i, {5}) :

α) $d^-(x_i)=d^+(x_i)=0$, le sommet x_i est un point isolé qui correspond à un concept basique n'entretenant aucune relation avec les autres concepts ;

β) $d^-(x_i)=0$ et $d^+(x_i)>0$, le sommet x_i correspond à un concept basique relié à d'autres concepts (non basiques) par le lien SET ;

γ) $d^-(x_i)>0$ et $d^+(x_i)>0$, le sommet x_i correspond à un concept non basique jouant à la fois le rôle de générique et de spécifique pour d'autres concepts ;

δ) $d^-(x_i)>0$, $d^+(x_i)=0$, le sommet x_i correspond à un concept tel que son degré de généralité ne permette pas de le considérer comme individu d'autres concepts.

Le graphe GARCHES possède les propriétés suivantes :

Propriété 1. GARCHES est un graphe sans circuit (du fait de l'antisymétrie et de l'antiréflexivité de la relation SET).

Il en résulte que GARCHES n'est ni symétrique, ni réflexif.

Propriété 2. GARCHES possède au moins un sommet x_i tel que $d^-(x_i)>o$ et $d^+(x_i)=o$ (car propriété 1 et cardinal de T fini).

Nous devons noter également que GARCHES peut posséder des cycles, et qu'en général il n'est pas connexe (il suffit pour cela d'être confronté à une application dont au moins un concept n'entretient aucune relation avec les autres concepts).

Les différentes composantes connexes de GARCHES, qui déterminent une partition de T, possèdent les propriétés 1 et 2 car SET est l'unique relation attestée dans le graphe ; de plus elles sont toujours connexes. Désormais nous supposerons que chaque composante connexe, désignée par $RG(T_a,SET)$, possède un seul sommet a tel que $d^-(a)>0$ et $d^+(a)=0$ (Nous ajouterons éventuellement dans T_a un sommet supplémentaire pour que cette condition soit remplie). Toute composante connexe $RG(T_a, SET)$ est donc supposée être un *graphe borné supérieurement*, le maximum de tous les sommets étant précisément le point a. Ce sommet correspond à un "concept-muet" représentant la notion la plus générale de T_a (i.e. la plus grande extension et la plus petite intention, voir § I.1.).

On appellera *Graphe de Résolution* ou R-graphe de ARCHES toute composante connexe de GARCHES (munie, éventuellement par construction, de cette nouvelle propriété).

Cette terminologie se justifie dans la mesure où les R-graphes définissent les *structures de contrôle* exploitées par ARCHES au cours des démonstrations de théorèmes nécessitant l'exploration de plusieurs champs(Résolution à n champs, voir chapitre huitième).

Tout R-graphe construit sur T est donc par définition un graphe de la forme $RG(T_a,SET)$ ayant les propriétés axiomatiques suivantes :

Propriété 1. $T_a \subset T$ et $T_a \neq \phi$; on dira que T_a *engendre* le R-graphe.

Propriété 2. $RG(T_a,SET)$ est un graphe connexe, et sans circuit.

Propriété 3. $RG(T_a,SET)$ est un graphe de borne supérieure a appelée *racine* du R-graphe ($d^-(a)>0$ et $d^+(a)=0$).

Propriété 4. Pour tout graphe de résolution $RG(T_b,SET)$ construit sur T et différent de $RG(T_a,SET)$, on a : $T_a \cap T_b = \phi$

Considérons deux systèmes ARCHES différents organisés à partir de deux ensembles de concepts distincts T_1 et T_2, mais ayant éventuellement une intersection non vide. On peut alors énoncer à leur propos la proposition très simple suivante :

Proposition. Si les R-graphes RG(T_a,SET) et RG(T_b,SET) constr respectivement sur T_1 et T_2 vérifient la relati $T_a \cap T_b \neq \phi$, alors les racines a et b ne peuvent p appartenir toutes les deux à $T_a \cap T_b$ (car la pro-priété 2 serait mise en défaut par la présence d'au moins un circuit).

Les figures 3.2. et 3.3. montrent des extraits de R-graphes particuliers

III.4. Notion de Domaines.

La définition des R-graphes permet d'introduire très naturellement la no-tion de domaines.

On appellera *Domaine* l'ensemble fini des champs associés aux concepts qu appartiennent au même R-graphe ; on dira que le domaine est réglé par so R-graphe.

Tout domaine représente l'univers conceptuel le plus général contenant l concepts le définissant.
Tout domaine sera repéré par le nom qui désigne la racine de son R-grap Par exemple, les domaines ANIMAL et RECIPIENT sont réglés par les R-grap schématisés par les figures 3.2. et 3.3.

Tout domaine Δ_a possède une organisation statique qui est précisément dé-finie par le R-graphe RG(T_a,SET) qui le règle. Cette organisation expri a priori les rapports sémantiques entre les différents champs composant Δ_a. $\forall C_{aj} \in \Delta_a$, le sens du champ C_{aj} dénoté par sa position dans RG(T_a,SET) est partiel dans la mesure où il est complété par la description des ind vidus le composant, et/ou les relations que ces individus peuvent entret nir avec les individus appartenant à d'autres champs (organisation dyna que). Nous montrerons ultérieurement comment cette organisation statique (i.e. les différents R-graphes et leurs propriétés) est mobilisée pour démontrer des théorèmes du système ARCHES qui mettent en jeu plusieurs champs (voir chapitre huitième).

Quels que soient les domaines distincts Δ_a et Δ_b réglés respectivement p RG(T_a,SET) et RG(T_b,SET) construits sur T, on a $\Delta_a \cap \Delta_b = \phi$ (car $T_a \cap T_b = \phi$). Deux champs appartenant à des domaines différents mais représentés par l

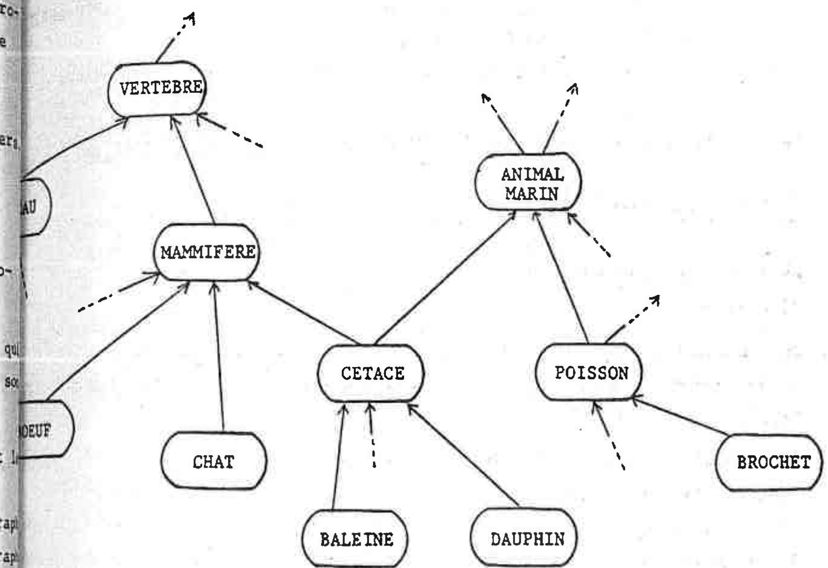


figure 3.2. - Extrait du R-graphe réglant le domaine ANIMAL.

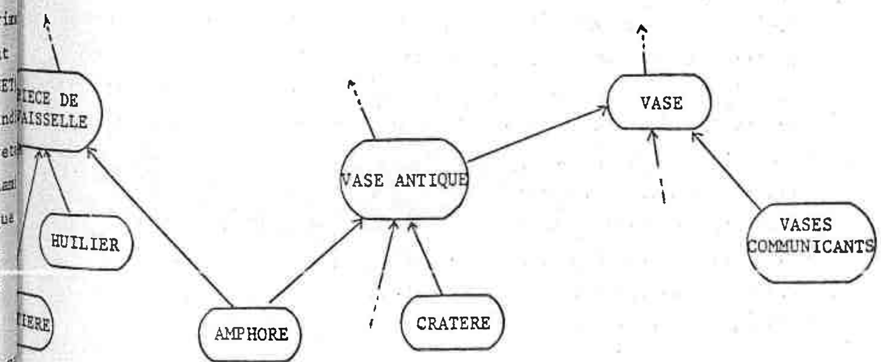


figure 3.3. - Extrait du R-graphe réglant le domaine RECIPIENT.

chaîne de caractères sont ainsi effectivement différenciés. Ils représentent deux "champs homonymes" dont les sens respectifs sont déterminés par les domaines qui les contiennent. Par exemple la notion VOL peut identifier deux champs homonymes appartenant respectivement aux domaines TRAFIC-AERIEN et MOUVEMENT d'ANIMAUX (voir § I.2.).
On appellera *famille de domaines de base T* l'ensemble fini de domaines réglés par les R-graphes construits sur T.

Si on désigne par Δ_a un domaine quelconque de la famille réglé par $RG(T_a, SET)$, nous rappelons que :

α) $\bigcup_a T_a = T$

β) $\forall ab \; T_a \cap T_b = \phi$ et $\Delta_a \cap \Delta_b = \phi$

Cette notion de famille détermine la texture de ARCHES :

Le système symbolique ARCHES est une famille de domaines de base T.

IV. OPERATION DE FUSION DES SYSTEMES ARCHES.

Dans le but de se donner les moyens de composer des fichiers relatifs à des systèmes symboliques ARCHES distincts, nous définissons dans ce qui suit une opération de fusion de tels systèmes, et nous énonçons les conditions nécessaires et suffisantes qu'ils doivent remplir pour que la fusion soit une loi de composition interne. Pour ce faire, nous définissons au préalable une nouvelle opération sur l'ensemble des graphes, et nous évaluons la fermeture de cette opération pour les sous-ensembles dont les éléments sont des R-graphes.

IV.1. Union de deux graphes.

IV.1.1. Définition.

Soit R_1 et R_2 deux relations définies respectivement sur les ensembles X_1 et X_2 ; et soit $G(X_1,R_1)$ et $G(X_2,R_2)$ les graphes correspondants.
Désignons par R la relation définie sur l'ensemble $X=X_1 \cup X_2$ de la manière suivante :

$$\forall x,y \in X \qquad R(x,y) \Leftrightarrow R_1(x,y) \text{ ou } R_2(x,y)$$

Soit $G(X,R)$ le graphe correspondant :
Ce graphe s'appelle *union* des graphes $G(X_1,R_1)$ et $G(X_2,R_2)$, et se note comme suit :

$$G(X,R) = G(X_1,R_1) \oplus G(X_2,R_2)$$

Nous devons remarquer que l'union de deux R-graphes construits respectivement sur T_1 et T_2 n'est pas dans le cas général un R-graphe ; en d'autres termes cette opération n'est pas une loi de composition interne pour l'ensemble des R-graphes. Cependant les graphes obtenus possèdent toujours les propriétés 1 et 2 du graphe GARCHES car SET est l'unique relation attestée dans ces graphes.

IV.1.2. Conditions de fermeture de l'union.

Le théorème ci-dessous détermine les conditions nécessaires et suffisantes pour que cette opération soit fermée vis à vis d'un ensemble particulier de R-graphes.

Théorème 1. L'union de deux R-graphes $RG(T_a,SET)$ et $RG(T_b,SET)$ construits respectivement sur T_1 et T_2 est un R-graphe si et seulement si :

$$b \in T_a \text{ ou } a \in T_b$$

démonstration. Si a=b, ce sommet est maximum pour le graphe union car il estmaximum pour les R-graphes $RG(T_a,SET)$ et $RG(T_b,SET)$; le graphe obtenu est donc un R-graphe. Si $a \neq b$ et $b \in T_a$, on en déduit que a est maximum de donc a est maximum pour le graphe union ; c'est donc un R-graphe. Réciproquement si le graphe union est un R-graphe, il existe un sommet c maximum de tous les sommets du graphe. Ce point c ne peut être que a ou b car a et b sont respectivement les racines de $RG(T_a,SET)$ et $RG(T_b,SET)$. Il en résulte ou bien que a et b sont confondus avec le point c ou bien que $a \neq b$ et c=a (par exemple). Dans ce cas on a a>b et il existe un chemin $\mu[b,a]$ entre b et a. Comme b n'est l'origine d'aucun chemin dans le R-graphe $RG(T_b,SET)$ car b est maximum dans ce graphe, on en déduit que $b \in T_a$ (c.q.f.d.).

VI.2. Fusion de deux systèmes ARCHES.

VI.2.1. Définition.

On dira que deux systèmes ARCHES $\mathcal{R}_1$ et $\mathcal{R}_2$ de bases respectives T_1 et T_2 possèdent la *propriété de fusionnement* si et seulement si pour tous les couples de R-graphes $RG_1(T_a,SET)$ et $RG_2(T_b,SET)$, réglant respectivement les domaines $\Delta_a^1 \in \mathcal{R}_1$ et $\Delta_b^2 \in \mathcal{R}_2$ et vérifiant la relation $T_a \cap T_b \neq \phi$, l'union $RG_1(T_a,SET) \oplus RG_2(T_b,SET)$ est un R-graphe.

Il résulte de cette définition que deux systèmes ARCHES, dont les bases respectives ont une intersection vide, possèdent toujours la propriété de fusionnement.

On appellera *fusion de deux systèmes* $\mathcal{R}_1$ et $\mathcal{R}_2$ de bases respectives T_1 et T_2 l'ensemble $\mathcal{F}$ de domaines défini par :

α) Si $\Delta_a^1 \in \mathcal{R}_1$ réglé par $RG_1(T_a,SET)$ vérifiant $T_a \cap T_2 = \phi$ alors $\Delta_a^1 \in \mathcal{F}$

β) Si $\Delta_b^2 \in \mathcal{R}_2$ réglé par $RG_2(T_b,SET)$ vérifiant $T_b \cap T_1 = \phi$ alors $\Delta_b^2 \in \mathcal{F}$

γ) Si $\Delta_a^1 \in \mathcal{R}_1$ réglé par $RG_1(T_a,SET)$ vérifiant $T_a \cap T_2 \neq \phi$ et si $\Delta_b^2 \in \mathcal{R}_2$ réglé par $RG_2(T_b,SET)$ vérifiant $T_b \cap T_a \neq \phi$ alors $\Delta_{ab} = \Delta_a^1 \cup \Delta_b^2 \in \mathcal{F}$

VI.2.2. Conditions de fusionnement de deux systèmes.

Théorème 2. La fusion de deux systèmes ARCHES $\mathcal{R}_1$ et $\mathcal{R}_2$ de bases respectives T_1 et T_2 est un système ARCHES de base $T_1 \cup T_2$ si et seulement si $\mathcal{R}_1$ et $\mathcal{R}_2$ possèdent la propriété de fusionnement (démonstration immédiate).

Nous devons noter que la fusion de deux systèmes dont les bases respectives T_1 et T_2 vérifient $T_1 \cap T_2 = \phi$ est toujours un système de base $T_1 \cup T_2$. Il en résulte que l'on peut toujours étendre un système donné en lui ajoutant de nouveaux domaines sans modifier l'organisation et le contenu des domaines du système initial ; il s'agit là du problème du rallongement toujours possible des fichiers associés aux systèmes ARCHES (pour une implémentation en PROLOG des conditions de fusionnement ainsi que des propriétés des R-graphes voir {114}).

CHAPITRE QUATRIEME.

LE SYSTEME FORMEL S_{Δ}

DE CARACTERISATION DES DESCRIPTIONS

-=-=-=-=-=-=-

I. COMPOSITION DU SYSTEME FORMEL S_Δ.

Les descriptions des individus véhiculés par le système symbolique ARCHES déterminent un système formel S_Δ composé de quatre éléments {109} qui expriment les deux aspects *"Représentation"* (voir § II.) et *"Déduction"* (voir § III. et § IV.), respectivement les définissant et les organisant :

$$S_\Delta = \{\mathcal{A}, \Delta, \Delta_T, \Longrightarrow\}$$

$\mathcal{A}$ est un alphabet à partir duquel est engendré l'ensemble Δ des descriptions ; $\Delta_T \subset \Delta$ représentant l'ensemble des thèses de S_Δ. Ces trois ensembles, qui ont pour objet essentiel de définir et de véhiculer la représentation des descriptions (voir § II.4., et § II.5.), sont construits et analysés à partir des notions basiques de traits, de classes et d'opérateurs (voir § II.1. et II.2.), ainsi qu'à partir de la structure de données primitive définissant les termes descriptifs (voir § II.3.).

$\Longrightarrow$ est une relation de déduction qui détermine l'activité inférentielle qui peut être développée sur les descriptions. Elle est déterminée d'une part à partir de relations de réécriture définies sur les termes descriptifs (voir § III.), et d'autre part à partir de conditions spécifiques qui la relient aux connecteurs d'addition, de disjonction et de négation (voir § IV.).

II. SYSTEME DE REPRESENTATION DES DESCRIPTIONS.

II.1. Notions de traits et de classes.

II.1.1. Définition.

Tout objet - concept ou individu - est décrit dans le système ARCHES par un ensemble organisé de *traits* qui expriment, par rapport aux finalités des problèmes à résoudre, ses caractères distinctifs : marques, qualités, propriétés, manières d'être de l'objet considéré dans ce qu'il a de plus ou moins permanent, expressions de ses relations d'état avec d'autres objets, etc. Les traits, ou leur composition, déterminent la représentation sous-jacente des actants-caractérisant après transformation par paraphrase des énoncés du langage naturel véhiculant les descriptions élémentaires (voir § III., chapitre deuxième). Par exemple, les énoncés "PRISONNIER P1 enfermé dans la CELLULE C21", "ELEPHANT ayant un NEZ en forme de TROMPE", "ROBE en LAINE ROUGE CLAIR" sont des aspects de la description des objets PRISONNIER, ELEPHANT et ROBE dont les traits caractéristiques sont représentés par des unités d'information telles que CELLULE, NEZ, TROMPE, etc. Nous devons remarquer que les traits caractérisant les objets peuvent être des *concepts* (exemple : TROMPE), des *instances de concepts* (exemple : CELLULE C21) ou des *informations sans extension* (exemple : ROUGE). Par ailleurs, ces exemples montrent que les traits ont un statut double par rapport aux informations qu'ils caractérisent : d'une part les traits peuvent caractériser les concepts ou les individus (exemple : "ELEPHANT ayant un NEZ") ; d'autre part ils peuvent caractériser les traits qui ne représentent pas des individus (exemples : "NEZ en forme de TROMPE", "LAINE ROUGE CLAIR").

Les traits sont véhiculés en langage naturel à l'aide d'énoncés élémentaires du type "x est ROUGE","y est localisé dans un FOUR","z possède un NEZ", "t a la forme d'une TROMPE", etc. Nous supposons que, pour toute application donnée, ces énoncés forment un ensemble fini E sur

lequel est définie une relation binaire $\mathcal{R}$ qui permet d'associer à tout "énoncé générique" - par exemple x a une COULEUR - un "énoncé spécifique" - par exemple x est ROUGE. Soit P un énoncé générique et $\mathcal{C}$ l'ensemble de tous les énoncés inclus dans E qui sont reliés à P par la relation $\mathcal{R}$. A cet ensemble $\mathcal{C}$ correspond un ensemble T qui regroupe tous les traits de même nature, c'est-à-dire faisant référence à un même champ sémantique, précisément celui déterminé par l'énoncé P. De tels ensembles, appelés *classes*, structurent selon des modalités particulières les traits retenus pour la description des objets. *La notion de classe permet ainsi d'exprimer la portée sémantique des relations d'état* qui sont attestées dans les descriptions élémentaires (voir § III.2., chapitre deuxième).

Le schéma de représentation des connaissances du système ARCHES est donc composé, en particulier, de deux ensembles organisés et distincts : d'une part l'ensemble des concepts qui permettent d'exprimer les découpages sémantiques et logiques de tout univers réel adéquat à cette représentation abstraite (voir § I., chapitre troisième) ; d'autre part l'ensemble des classes qui permettent d'exprimer leurs découpages relationnels. Ces deux ensembles n'entretiennent aucun rapport formel explicite, à l'exception du fait que les concepts et les individus peuvent avoir le statut de traits (voir § II.1.2., propriétés 3 et 4).

II.1.2. Propriétés.

Classes et traits possèdent les propriétés suivantes :

Propriété 1. Toute classe est une collection finie d'éléments primitifs appelés traits.

Propriété 2. Toute classe est repérée par un nom T_j qui exprime le type des traits qui la composent ; chacun de ces traits représente une constante de type T_j.

Propriété 3. Les traits qui composent une classe sont soit des informations sans extension, soit des concepts, soit des individus. Dans les deux premiers cas ils sont représentés par des chaînes de caractères, dans le troisième par des expressions de la forme INS(C,x).

Propriété 4. Les traits peuvent décrire les concepts, les individus ou les traits *qui ne représentent pas des individus*.

Propriété 5. La liste des traits de type T_j peut être introduite soit en extension par énumération des éléments de la classe, soit en intention à partir de procédures dont les arguments appartiennent à d'autres classes (voir § IV.2.3., chapitre septième).

II.2. Notion d'opérateurs.

II.2.1. Analyse de quelques exemples.

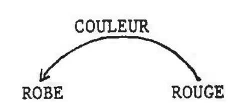
Nous avons vu que les traits caractérisant les concepts, les individus ou d'autres traits définissent des éléments de description qui sont véhiculés en langage naturel par des énoncés élémentaires, comme l'indique les exemples ci-après :

(1) La robe est rouge.
(2) L'insecte a une tête.
(3) Jean dort.
(4) Pierre pèse 70 kg.
(5) L'amphore est localisée dans le four.
(6) Marie a un poids supérieur à 50 kg.
(7) L'armoire est située entre le lit et la fenêtre.

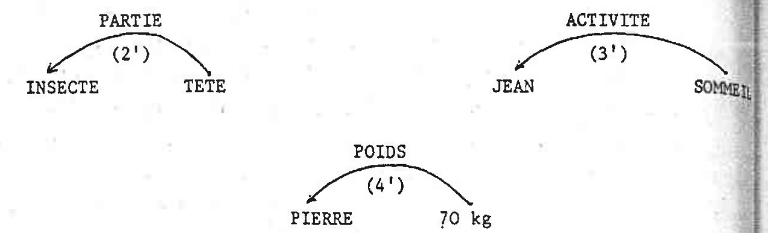
Ces énoncés peuvent être transformés en expressions sous-jacentes dans lesquelles apparaissent explicitement non seulement les traits et les classes auxquelles ils font référence, mais aussi les rapports qui existent entre ces derniers (voir § III.2., chapitre deuxième) :

(1') La COULEUR de la robe *est* ROUGE.
(2') Une PARTIE de l'insecte *est* la TETE.
(3') L'ACTIVITE de Jean *est* le SOMMEIL.
(4') Le POIDS de Pierre *est égal* à 70 kg.
(5') La LOCALISATION de l'amphore *est dans* le FOUR.
(6') Le POIDS de Marie *est supérieur* à 50 kg.
(7') La LOCALISATION de l'armoire *est entre* le LIT *et* la FENETRE.

L'expression (1') montre que le trait ROUGE est relié à la classe COULEUR par la simple relation d'appartenance ISA que l'on représente habituellement par le diagramme ci-après :



De même les expressions (2'), (3') et (4') conduisent à des diagrammes comparables :



Dans de telles représentations les rapports classes/traits n'apparaissent pas explicitement. De ce fait, leur interprétation peut être ambiguë comme le montre par exemple la représentation de (4'). Cette dernière peut être interprétée de cinq façons différentes : Pierre a un poids égal à 70 kg, > 70 kg., ⩾ 70 kg., < 70 kg., ou enfin ⩽ 70 kg.
Une représentation de ce type pour l'énoncé (5') conduit également à plusieurs interprétations :

l'amphore est dans le four
l'amphore est à l'extérieur du four
l'amphore est devant le four

----

quant aux énoncés (6') et (7'), ils ne peuvent plus être représentés selon un tel diagramme : le premier du fait de l'existence de l'opérateur de comparaison >, le second du fait que l'un des sommets correspond à un ensemble de traits.

Les insuffisances de cette représentation ont donc une double origine. D'une part, la non explicitation des rapports classes/traits conduit à une ambiguïté dans l'interprétation du sens véhiculé par le diagramme. D'autre part, la nature des noeuds n'étant pas précisée, le noeud source peut représenter une information élémentaire ou un ensemble d'informations ; elle dépend en particulier de l'arité de lien classe/trait et du type de classes mises en jeu. On trouvera dans {134} une analyse systématique de certaines des questions soulevées par l'explicitation et la représentation des relations dans la conception des réseaux sémantiques. Pour pallier à ces inconvénients, nous introduisons dans la conception du système ARCHES une entité primitive - l'opérateur - qui est mobilisé dans la construction des structures de données définissant la représentation des éléments de description (voir § II.3.).

II.2.2. Définition de l'opérateur.

On appellera *opérateur* tout lien qui exprime un rapport déterminé entre classe et traits.

D'une manière générale, les opérateurs sont n-aires car ils peuvent mettre en relation simultanément n traits avec la même classe. Le sens et l'arité des opérateurs dépendent essentiellement de la nature sémantique des classes. Par exemple, LOCALISATION est une classe qui peut impliquer des opérateurs unaires comme IN ou des opérateurs binaires comme BETWEEN, par contre la classe COULEUR n'est reliée aux traits qui la définissent que par l'opérateur unaire ISA.

Les traits reliés aux classes par l'intermédiaire d'opérateurs n-aires sont représentés par des *tuples* de degré n selon la notation $\#(t_1,\ldots,t_i,\ldots,t_n)$, les t_i étant des traits. Un tuple de degré 1 est représenté par le trait lui-même.

Exemples.

ROUGE
#(INS(LIT,L1), INS(FENETRE,F1))

Remarque.

Les opérateurs, comme représentation sous-jacente de certains aspects des relations d'état (voir § III.2., chapitre deuxième) sont toujours exprimés en anglais afin de les distinguer des mots du français qui peuvent les désigner en surface.

II.3. Termes descriptifs.

II.3.1. Termes descriptifs non décrits.

a) définition.

Les trois entités classe, trait et opérateur expriment le contenu logico-sémantique sous-jacent des éléments de description qui interviennent effectivement dans la description en langage naturel des concepts, des individus ou d'autres éléments de description. La notation formelle, choisie pour représenter de tels éléments, nommés termes descriptifs non décrits, doit donc véhiculer ces trois entités dans les termes d'une structure de données primitive ; plus précisément :

On appellera *terme descriptif non décrit* toute expression de la forme $op_n(T_j, \#(t_1,..,t_i,..,t_n))$, dans laquelle op_n est un opérateur n-aire, T_j une classe et enfin $\forall 1 \leqslant i \leqslant n$ t_i est un trait.

b) Représentation en réseau.

La figure 4.1. montre la représentation graphique des termes descriptifs non décrits. Le point d'application des relations potentielles que ce type de structure de données peut entretenir avec d'autres structures est représenté sur la figure 4.1. par le signe "." (point).
Certains des exemples précédents conduisent aux termes descriptifs schématisés dans la figure 4.2.

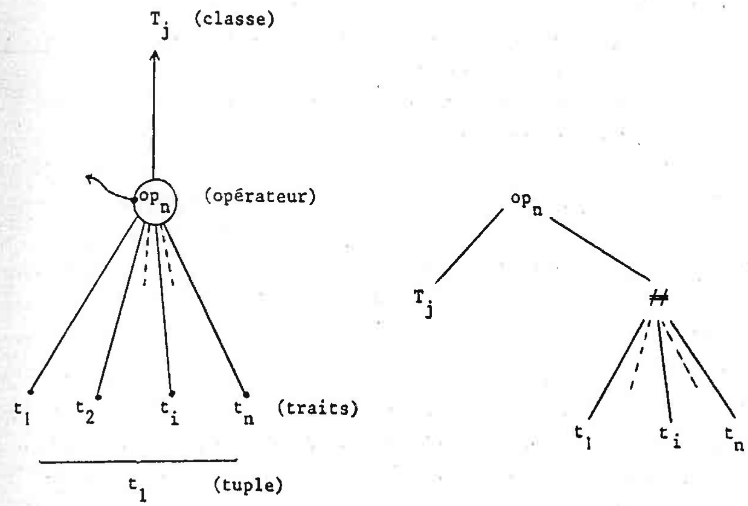


figure 4.1. - Représentations graphique et fonctionnelle d'un terme descriptif non décrit.

II.3.2. Termes descriptifs décrits.

a) Le lien ADL.

Nous avons vu que les traits peuvent être caractérisés par d'autres traits (propriété 4 des classes et des traits, voir § II.1.2.). Citons par exemple "PIERRE DORT PAISIBLEMENT", "ROBE en LAINE ROUGE CLAIR", "ARMOIRE en BOIS VERNIS", "INSECTE ayant une BOUCHE composée de MANDIBULES en CHITINE", etc. PAISIBLEMENT, CLAIR, VERNIS, etc. sont des caractérisations locales de DORT, ROUGE, BOIS, etc. Ces caractérisations sont véhiculées, au niveau de la représentation, par la structure de données primitive définie précédemment, à savoir les termes descriptifs non décrits (voir § II.3.1.). La relation qui exprime l'attribution d'un terme descriptif à un trait est exprimée par le lien ADL (*Attribution de Description Locale*), qu'on notera plus simplement par le signe "$" (dollar).

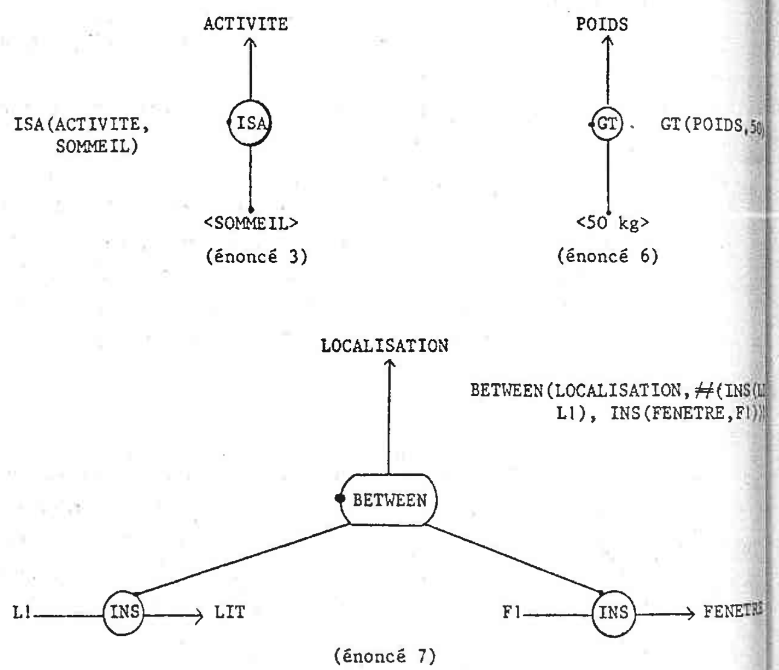


figure 4.2. - *Exemples de termes descriptifs non décrits.*

b) Définition.

On appellera *terme descriptif décrit* toute expression de la forme $op_n(T,t_1)$ dans laquelle t_1 est un tuple de degré n dont les éléments sont soit des traits, soit des expressions du type $\$(t_i,op_{ni}(T_i,t_{1i}))$.

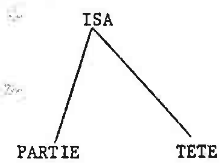
Cette définition conduit à un terme descriptif non décrit si le tuple n'est formé que de traits (voir § II.3.1.). Cependant, nous dirons que tout trait t_i qui n'est pas décrit localement par d'autres traits est caractérisé par la description vide, notée Λ (voir § II.4.). Ceci conduit à la représentation $\$(t_i,\Lambda)$ que nous supposerons par *convention d'écriture* équivalente à t_i ; la figure 4.3. en est un exemple. Dans la suite nous ne considérerons plus que cette convention pour représenter les traits non décrits. Ceci nous permettra désormais de ne manipuler dans ARCHES que des termes descriptifs décrits ; aussi nous n'utiliserons plus que l'expression "termes descriptifs" pour les désigner.

Remarque.

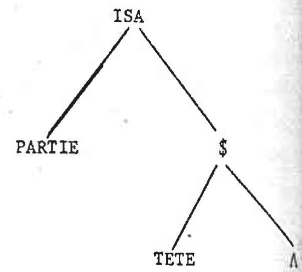
Rappelons que tout trait qui représente un individu x appartenant au concept C ne peut pas être décrit localement (voir § II.1.2. ; propriété 4). Par contre dans le champ C, l'individu x possède une description principale (voir § II.2.1. et § III.2., chapitre troisième). Nous donnons dans le chapitre sixième une grammaire complète des descriptions dans laquelle sont définies les règles de réécriture engendrant les termes descriptifs (voir § V., chapitre sixième).

c) Représentation en réseau.

La figure 4.4. montre la représentation graphique des traits qui sont caractérisés par une description locale.
Ainsi les deux premiers énoncés ci-dessus (voir § II.3.2. a) sont représentés par les structures de données schématisées par la figure 4.5.



ISA(PARTIE,TETE)



ISA(PARTIE,$(TETE,Λ))

figure 4.3. – *Représentations équivalentes d'un terme descriptif non decrit.*

$\$(ti, op_n(T, t_1))$

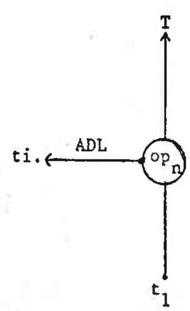
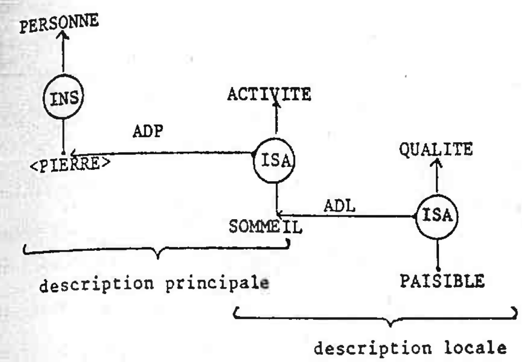
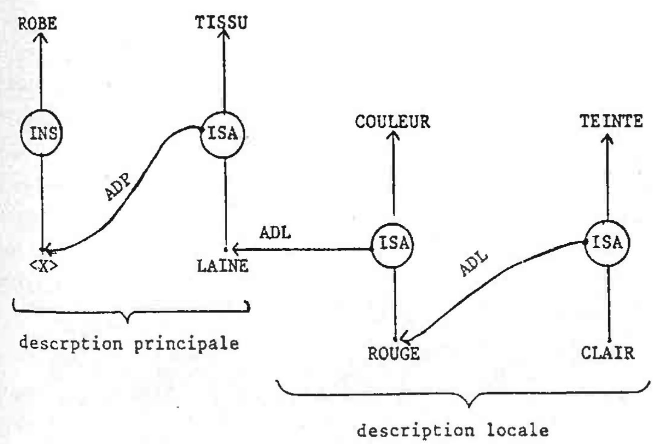


figure 4.4. – *Représentation graphique d'un trait décrit localement.*



Représentation de "PIERRE DORT PAISIBLEMENT" par la structure :
PERSONNE(PIERRE,ISA(ACTIVITE, $(SOMMEIL, ISA(QUALITE,PAISIBLE))))



Représentation de "ROBE en LAINE ROUGE CLAIR" par la structure :
ROBE(x,ISA(TISSU,$(LAINE,ISA(COULEUR,$(ROUGE,ISA(TEINTE,CLAIR))))))

figure 4.5. – *Exemples de termes descriptifs décrits.*

II.4. Les descriptions.

II.4.1. Hypothèses sur la construction des descriptions.

On appellera *description* d'un individu x appartenant à l'extension du concept C l'ensemble organisé des termes descriptifs qui le caractérise.

Plus précisément, les hypothèses qui fondent le cadre dans lequel s'inscrivent les modalités de construction des descriptions sont les suivantes.

Hypothèse 1 Tout terme descriptif est un élément de description.

Les termes descriptifs peuvent être niés selon des modalités d'interprétation bien définis (voir § III.3., chapitre deuxième). La représentation explicite de la négation a pour conséquence d'introduire dans ARCHES une définition métalinguistique de la contingence (voir chapitre neuvième).

Hypothèse 2 Toute description est représentée par une composition d'éléments de description.

Les compositions d'éléments de description sont déterminées à partir d'un nombre fini de connecteurs qui marquent les rapports logiques et sémantiques entre les différents termes descriptifs caractérisant les phénomènes étudiés (voir § III.4., chapitre deuxième). Ces connecteurs sont de deux types : d'une part les connecteurs statiques qui permettent d'exprimer les descriptions des individus dans un état donné (ET d'addition et OU non exclusif) ; d'autre part les connecteurs dynamiques qui permettent d'exprimer le caractère évolutif des descriptions (futur immédiat, futur médiat et comme conséquence le ET de succession, voir chapitre sixième).

Hypothèse 3 La chaîne vide Λ est une description.

Appelée *description vide*, Λ n'est donc formée d'aucune occurrence de termes descriptifs. Les informations qui sont caractérisées par Λ ne possèdent pas de descriptions (voir § II.3.2.). En particulier les individus, qui ont une réalité sensible, possèdent nécessairement une *description principale* $\mathcal{D}$(x) qui n'est pas Λ (condition d'existence des individus, voir § IV.1.2.). Naturellement cela n'empêche pas qu'un individu soit caractérisé par la description vide Λ là où il représente un trait (il s'agit en effet d'une *description locale* ; voir § II.1.2.; propriété 4).

II.4.2. Modalités de construction des descriptions.

a) L'alphabet.

Les descriptions sont engendrées à partir d'un alphabet $\mathcal{A}$ formé de quatre catégories de symboles : /1/ un ensemble dénombrable $\mathcal{L}$ de termes descriptifs : /2/ la description vide Λ ; /3/ les connecteurs $\neg$, $*$ et + nommés respectivement Négation, ET d'addition et OU non exclusif ; /4/ enfin les parenthèses (et).

$$\mathcal{A} = \mathcal{L} \cup \{\Lambda\} \cup \{\neg, *, +\} \cup \{(,)\}$$

b) Le langage des descriptions.

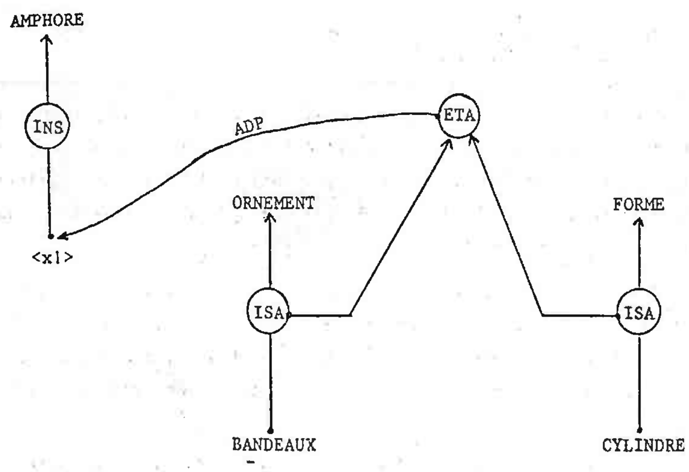
L'ensemble dénombrable Δ des descriptions est un langage formel construit sur $\mathcal{A}$. Les descriptions, qui en sont les formules bien formées, obéissent aux règles (R) de formation ci-après :

R1 Λ est une description ;
R2 Tout terme descriptif est une description ;
R3 Si D est une description, alors $\neg$D est une description ;
R4 Si D1 et D2 sont des descriptions, alors (D1$*$D2) est une description ;
R5 Si D1 et D2 sont des descriptions, alors (D1+D2) est une description ;
R6 Toute description est obtenue à partir des termes descriptifs en appliquant un certain nombre de fois les règles R2, R3, R4 et R5.

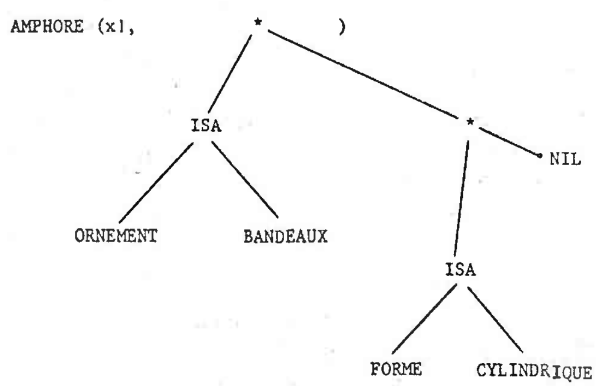
On dira que $\mathcal{L}$ est la *base* de Δ ($\mathcal{L} \subset \Delta \subset \mathcal{A}^*$).

II.4.3. Représentation en réseau.

Dans la représentation graphique des descriptions, les connecteurs * et + sont représentés respectivement par ETA et OUN. Ainsi les trois énoncés "AMPHORE à BANDEAUX, *et* ayant la forme CYLINDRIQUE", "MARIE est GRANDE *et* BELLE", et enfin "LES MAMMIFERES ont des PATTES *ou* des NAGEOIRES comme MEMBRES" sont représentés par les structures schématisées par les figures 4.6., 4.7., et 4.8. (dans le dernier énoncé, les traits "patte" et "nageoire" d'une part et "membre" d'autre part appartiennent respectivement aux classes PARTIE (de corps) et DEF(inition).).



a/ Représentation en réseau



b/ Représentation prédicative

figure 4.6. - *Structure représentant l'énoncé "Amphore x1 cylindrique et à bandeaux".*

II.5. Les thèses du système formel S_Δ.

Soit $\mathcal{J}$ l'ensemble fini des individus et $\mathcal{D}$ l'application qui permet d'associer à tout $x \in \mathcal{J}$ sa description $\mathcal{D}(x) \in \Delta$. Si l'individu x est une instance du concept C, nous avons vu que l'attribution $\mathcal{D}(x)$ à x est représentée par la structure $C(x, \mathcal{D}(x))$ (voir § II.2., chapitre troisième) : nous dirons que $\mathcal{D}(x)$ est une *thèse du système formel* S_Δ.

Soit $\Delta_T \subset \Delta$ l'ensemble des descriptions tel que l'application $\mathcal{D}$ soit une bijection de $\mathcal{J}$ sur Δ_T : les structures stockées dans le système ARCHES sont les formules toujours vraies du type $C(x, \mathcal{D}(x))$ (voir § III., chapitre septième) dans lesquelles x instance du concept C a pour description $\mathcal{D}(x) \in \Delta_T$.

Δ_T est l'ensemble des thèses du système formel S_Δ de caractérisation des descriptions.

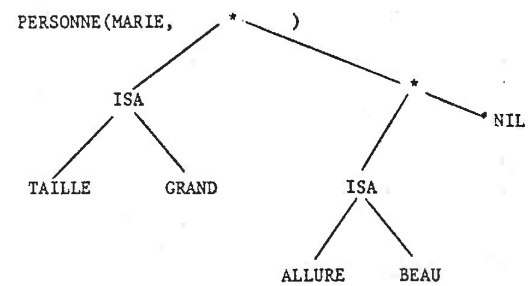
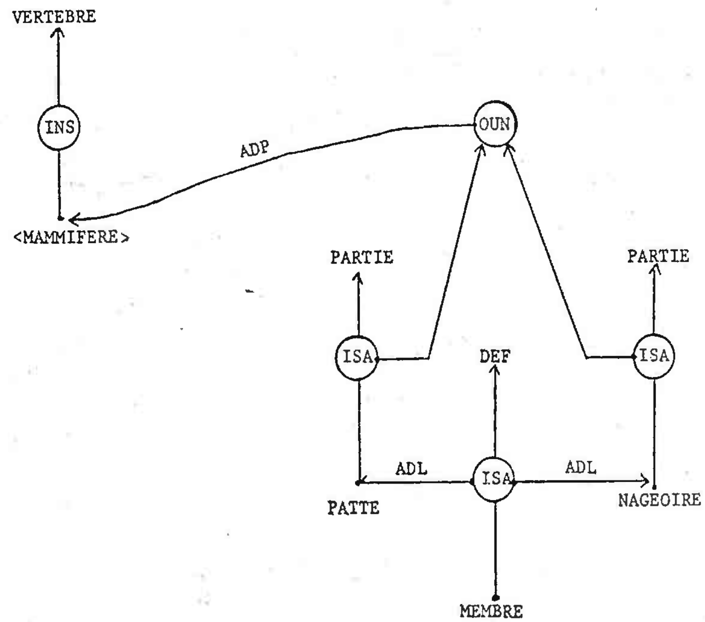
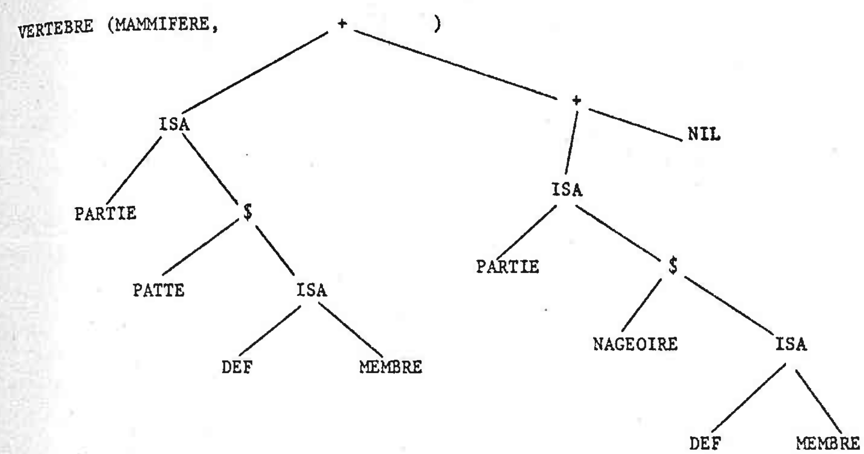


figure 4.7. - *Représentation de l'énoncé "Marie grande et belle".*



a/ Représentation en réseau.

figure 4.8. - *Suite page suivante.*



b/ Représentation prédicative.

(Dans cet exemple MAMMIFERE est une instance du concept VERTEBRE, la structure correspondante appartenant au champ VERTEBRE . MAMMIFERE ayant également le statut de champ, le passage du champ MAMMIFERE au champ VERTEBRE est contrôlé par le R-graphe réglant le domaine ANIMAL (voir § III., chapitre troisième ; voir également IV.2.2.a, chapitre septième).

figure 4.8. - *Exemple de représentation de OUN.*

III. SYSTEME DE REGLES ET DE RELATIONS DE REECRITURE DES TERMES DESCRIPTIFS.

Nous définissons dans ce paragraphe deux *relations de réécriture sur les termes descriptifs* (voir § III.3.2.) à partir d'un certain nombre de *règles de réécriture particulières* (voir § III.3.1.). Ces règles de réécriture sont fondées sur une représentation logique des termes descriptifs et des propriétés qui en découlent (voir § III.1.). Enfin ces relations de réécriture seront utilisées pour définir la *relation de déduction* entre les descriptions (voir § IV.), et donc seront mises en oeuvre pour la démonstration déductive des théorèmes du système ARCHES (voir chapitre cinquième et chapitre sixième).

III.1. Représentation logique des termes descriptifs.

III.1.1. Modalités de représentation.

Nous donnons dans ce paragraphe une représentation logique des termes descriptifs dont les propriétés formelles sont à l'origine de la définition des règles de réécriture des termes descriptifs.

Désignons par f un trait quelconque, et par T la classe à laquelle il fait référence. Un terme descriptif non décrit, construit sur f et T et caractérisant l'individu x, est représenté par un énoncé logique e(x) défini comme suit :

$$e(x) =_{df} \exists f\{f(x) \wedge T(f)\}$$

Par exemple l'expression "x est ROUGE" conduit à un énoncé e(x) dans lequel la variable prédicative f prend la valeur ROUGE, tandis que le prédicat T est représenté par la classe COULEUR.

Introduisons une ou plusieurs caractérisations de f à l'aide de la fonction complexe ρ(f). Le terme descriptif décrit correspondant est représenté par l'énoncé e'(x) défini comme suit :

$$e'(x) =_{df} \exists f\{f(x) \wedge T(f) \wedge \rho(f)\}$$

Supposons que ρ(f) dénote uniquement une propriété de propriété : "x est caractérisé par f, f étant décrit par g". La fonction ρ(f) s'écrit :

$$\rho(f) = \exists g\ \{g(f) \wedge T'(g)\}$$

d'où :

$$e'(x) = \exists f\ \{[f(x) \wedge T(f)] \wedge \exists g[g(f) \wedge T'(g)]\}$$

Nous ne devons pas confondre cet énoncé avec l'énoncé e"(x) qui exprime que f et g caractérisent x :

$$e''(x) = \exists f\{f(x) \wedge T(f)\} \wedge \exists g\{g(x) \wedge T'(g)\}$$

La figure 4.9. illustre cette situation à partir des deux expressions "ROBE ROUGE CLAIR" et "ROBE ROUGE ET CLAIRE" (voir § II.4.2.).

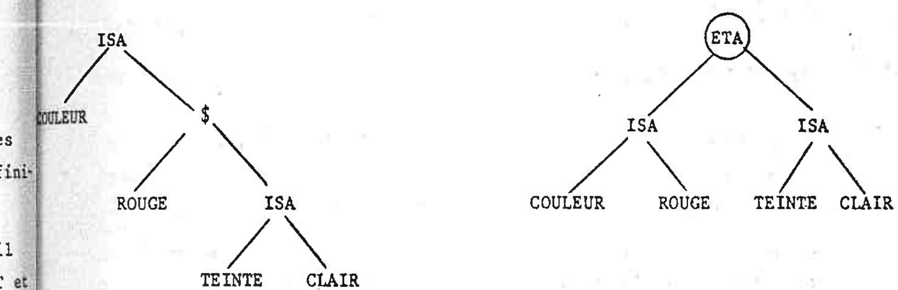


figure 4.9. - Représentation des énoncés e'(x) et e"(x) pour des valeurs particulières de f, T, g et T'.

III.1.2. Propositions remarquables.

a) Propositions générales.

Proposition 1. $\vdash \exists f\ \{f(x) \wedge T(f) \wedge \rho(f)\} \supset \exists f\{f(x) \wedge T(f)\}$

démonstration.

$\vdash \forall f\{\overline{f(x) \wedge T(f)} \vee (f(x) \wedge T(f))\}$

$\vdash \forall f\{\overline{f(x) \wedge T(f)} \vee (f(x) \wedge T(f)) \vee \overline{\rho(f)}\}$

$\vdash \forall f\{\overline{f(x) \wedge T(f) \wedge \rho(f)} \vee (f(x) \wedge T(f))\}$

$\vdash \forall f\{f(x) \wedge T(f) \wedge \rho(f) \supset f(x) \wedge T(f)\}$

$\vdash \exists f\{f(x) \wedge T(f) \wedge \rho(f)\} \supset \exists f\{f(x) \wedge T(f)\}$ (c.q.f.d.)

Proposition 2. $\vdash \exists f\{f(x) \wedge T(f) \wedge \rho(f)\} \supset \exists f\{f(x) \wedge \rho(f)\}$

(même démonstration que pour la proposition 1, car $T(f)$ et $\rho(f)$ jouent le même rôle).

Nous devons noter que les propositions 1 et 2 n'ont pas de réciproque.

Proposition 3. $\exists f\{f(x) \wedge T(f) \wedge \rho(f)\} \vdash \exists f\{f(x) \wedge T(f)\} \wedge \exists f\{f(x) \wedge \rho$

(conséquence immédiate des propositions 1 et 2).

Ces propositions se démontrent très facilement pour des énoncés véhiculant n traits $f_1, f_2, \ldots, f_{n-1}, g$ se caractérisant les uns les autres de manière récursive. Dans ce cas, la proposition 3 se généralise comme sui

Proposition 4. $\exists f_1 f_2 \ldots f_{n-1} g\{H(x,f_1,\ldots,f_{n-1}) \wedge g(f_{n-1}) \wedge T(g) \wedge \rho$

$\exists f_1 f_2 \ldots f_{n-1} g\{H(x,f_1,\ldots,f_{n-1}) \wedge g(f_{n-1}) \wedge T(g)\} \wedge$

$\exists f_1 f_2 \ldots f_{n-1} g\{H(x,f_1,\ldots,f_{n-1}) \wedge g(f_{n-1}) \wedge \rho(g)\}$

b) Propositions fondées sur des propriétés relatives aux classes.

Dans ce qui suit nous démontrons trois autres propositions fondées sur certaines propriétés que peuvent entretenir les classes entre elles. Pou ce faire, nous exprimons l'énoncé $e'(x)$ comme suit :

$$(1) \qquad e'(x) = \exists fg\{H_1(x,f) \wedge H_2(f,g)\} \quad \text{avec} \quad \begin{cases} H_1(x,f) = f(x) \wedge T(f) \\ H_2(f,g) = g(f) \wedge T'(g) \end{cases}$$

Premier cas. Nous supposons que les fonctions H_1 et H_2 vérifient la relation ci-après :

$$\forall xyz\{(H_1(x,y) \wedge H_2(y,z)) \supset H_2(x,z)\} \qquad (2)$$

Dans ce cas, nous dirons que le couple de classes (T_1, T_2) associées respectivement aux fonctions H_1 et H_2 possède la *propriété d'héritage*.

par existentialisation des universelles, (2) devient :

$$\exists xyz\{H_1(x,y) \wedge H_2(y,z)\} \supset \exists xzH_2(x,z) \qquad (2')$$

Par application de (2') à l'énoncé (1), on obtient :

$$\exists fg\{H_1(x,f) \wedge H_2(f,g)\} \supset \exists gH_2(x,g)$$

Comme par ailleurs, on a d'après la proposition 1 :

$$\exists fg\{H_1(x,f) \wedge H_2(f,g)\} \supset \exists fH_1(x,f)$$

on en déduit la proposition suivante :

Proposition 5. Si $\forall xyz\{(H_1(x,y) \wedge H_2(y,z)) \supset H_2(x,z)\}$

alors $\exists fg\{H_1(x,f) \wedge H_2(f,g)\} \vdash \exists f\{H_1(x,f)\} \wedge \exists g\{H_2(x,g)\}$

Remarque 1.

Si un couple de classes possède la propriété d'héritage, alors l'énoncé $e''(x)$ peut être déduit de l'énoncé $e'(x)$; la réciproque est toujours fausse (figure 4.9.).

Deuxième cas. Nous supposons que les fonctions H_1 et H_2 vérifient la relation ci-après :

$$\forall xyz\{(H_1(x,y) \wedge H_2(y,z)) \supset H_1(x,z)\}$$

Dans ce cas nous dirons que le couple de classes (T_1, T_2) associées respectivement aux fonctions H_1 et H_2 possède la *propriété d'extension*.

La même démarche que celle utilisée pour la démonstration de la proposition 5 permet de prouver la proposition suivante :

Proposition 6. Si $\forall xyz\{(H_1(x,y) \wedge H_2(y,z)) \supset H_1(x,z)\}$

alors $\exists fg\{H_1(x,f) \wedge H_2(f,g)\} \vdash \exists f\{H_1(x,f)\} \wedge \exists g\{H_1(x,g)\}$

Remarque 2.

La propriété d'extension impose que le trait g fasse référence à la fois à la classe T' et à la classe T ; on a donc $T \cap T' \neq \phi$. Enregistrer directement l'énoncé $\exists f\{H_1(x,f)\} \wedge \exists g\{H_1(x,g)\}$ dans ARCHES introduit dans ce cas une perte d'information car la proposition 6 n'a pas de réciproque.

Troisième cas. Nous supposons que les fonctions H_1 et H_2 vérifient la relation ci-après :

$$\begin{cases} H_1 \equiv H_2 \equiv H \\ \forall xyz\{(H(x,y) \wedge H(y,z)) \supset H(x,z)\} \end{cases}$$

Dans ce cas nous dirons que la classe T associée à la fonction H possède la *propriété de transitivité*.

On démontre facilement la proposition suivante :

Proposition 7. Si H est transitive, alors :

$$\exists fg\{H(x,f) \wedge H(f,g)\} \vdash \exists f H(x,f) \wedge \exists g H(x,g)$$

Remarque 3.

Nous devons noter que la propriété de transitivité est un cas particulier de la propriété d'héritage.

c) Propositions fondées sur des relations définies entre les propriétés des classes.

Enfin nous allons démontrer trois propositions qui déterminent certaines relations logiques entre les propriétés d'héritage et d'extension.

Premier cas. Nous supposons que les fonctions H_1, H_2 et H_3 vérifient les deux relations ci-après :

$$\begin{cases} \forall xyz\{(H_1(x,y) \wedge H_2(y,z)) \supset H_2(x,z)\} & (3) \\ \forall xyz\{(H_2(x,y) \wedge H_3(y,z)) \supset H_3(x,z)\} & (4) \end{cases}$$

Il en résulte que les couples de classes (T_1,T_2) et (T_2,T_3), associées respectivement d'une part aux fonctions H_1 et H_2 et d'autre part aux fonctions H_2 et H_3, possèdent la propriété d'héritage.

(3) devient :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_3(z,t)) \supset (H_2(x,z) \wedge H_3(z,t))\} \quad (3')$$

De (4) et (3'), il vient :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_3(z,t)) \supset H_3(x,t)\} \quad (5)$$

Enfin de (4) on déduit :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_3(z,t)) \supset (H_1(x,y) \wedge H_3(y,t))\} \quad (5')$$

de (5) et 5'), on tire :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_3(z,t)) \supset (H_1(x,y) \wedge H_3(y,t) \wedge H_3(x,t))\}$$

or : $$\forall xyt\{(H_1(x,y) \wedge H_3(y,t) \wedge H_3(x,t)) \supset ((H_1(x,y) \wedge H_3(y,t)) \supset H_3(x,$$

d'où la proposition suivante :

Proposition 8. Si les couples de classes (T_1,T_2) et (T_2,T_3), T_1, T_2 et T_3 étant associées respectivement aux fonctions H_1, H_2 et H_3, possèdent la propriété d'héritage alors :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_3(z,t)) \supset ((H_1(x,y) \wedge H_3(y,t)) \supset H_3(x,t))\}$$

Deuxième cas. Nous supposons que les fonctions H_1, H_2 et H_3 vérifient les deux relations ci-après :

$$\begin{cases} \forall xyz\{(H_1(x,y) \wedge H_2(y,z)) \supset H_1(x,z)\} & (6) \\ \forall xyz\{(H_2(x,y) \wedge H_3(y,z)) \supset H_2(x,z)\} & (7) \end{cases}$$

Il en résulte que les couples de classes (T_1,T_2) et (T_2,T_3), T_1, T_2 et T_3 étant associées respectivement aux fonctions H_1, H_2 et H_3, possèdent la propriété d'extension.

(6) devient :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_3(z,t)) \supset (H_1(x,z) \wedge H_3(z,t))\} \quad (8)$$

(7) devient :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_3(z,t)) \supset (H_1(x,y) \wedge H_2(y,t))\} \quad (7')$$

De (6) et (7') on tire :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_3(z,t)) \supset H_1(x,t)\} \quad (8')$$

De (8) et (8') on tire :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_3(z,t)) \supset (H_1(x,z) \wedge H_3(z,t) \wedge H_1(x,t))\}$$

or : $$\forall xzt\{(H_1(x,z) \wedge H_3(z,t) \wedge H_1(x,t)) \supset ((H_1(x,z) \wedge H_3(z,t)) \supset H_1(x,t))\}$$

d'où la proposition suivante :

Proposition 9. Si les couples de classes (T_1,T_2) et (T_2,T_3), T_1, T_2 et T_3 étant associées respectivement aux fonctions H_1, H_2 et H_3, possèdent la propriété d'extension alors :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_3(z,t)) \supset ((H_1(x,z) \wedge H_3(z,t)) \supset H_1(x,t))\}$$

Troisième cas. Nous supposons que les fonctions H_1, H_2 et H_3 vérifient les deux relations ci-après :

$$\begin{cases} \forall xyz\{(H_1(x,y) \wedge H_2(y,z)) \supset H_1(x,z)\} & (9) \\ \forall xyz\{(H_2(x,y) \wedge H_3(y,z)) \supset H_3(x,z)\} & (10) \end{cases}$$

Il en résulte que les couples de classes (T_1,T_2) et (T_2,T_3) possèdent respectivement la propriété d'extension et la propriété d'héritage.

(9) devient :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_3(z,t)) \supset (H_1(x,z) \wedge H_3(z,t))\} \quad (11)$$

(10) se réécrit :

$$\forall xzt\{(H_2(x,z) \wedge H_3(z,t)) \supset H_3(x,t)\} \quad (11')$$

De (11) et (11') on tire :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_2(x,z) \wedge H_3(z,t)) \supset (H_1(x,z) \wedge H_3(z,t) \wedge H_3(x,$$

d'où la proposition suivante :

Proposition 10. Si les couples de classes (T_1,T_2) et (T_2,T_3) possèdent respectivement la propriété d'extension et la propriété d'héritage, T_1, T_2 et T_3 étant associées respectivement aux fonctions H_1, H_2 et H_3, alors :

$$\forall xyzt\{(H_1(x,y) \wedge H_2(y,z) \wedge H_2(x,z) \wedge H_3(z,t)) \supset$$
$$((H_1(x,z) \wedge H_3(z,t)) \supset H_3(x,t))\}$$

III.2. Règles de transformation des termes descriptifs.

III.2.1. Opération de concaténation.

Désignons par $\mathcal{C}$ l'ensemble, non vide et fini, dont les éléments représentent les *termes descriptifs non décrits construits à partir d'opérateurs 1-aires* (cette restriction se justifie dans la mesure où les règles de transformation des termes descriptifs ne mettent en jeu que des couples de classes associées à des opérateurs unaires (voir § III.2.2.)).

$$\mathcal{C} = \{u_1, u_2, \ldots, u_p\}$$

Soit $\mathcal{C}^*$ l'ensemble formé de la description vide Λ et de tous les termes descriptifs construits à partir de l'ensemble $\mathcal{C}$. Si u_i et u_j appartenant à $\mathcal{C}$ sont respectivement de la forme ISA(PARTIE,\$(TETE,Λ)) et ISA(PARTIE,\$(NEZ,Λ)) alors le terme descriptif ISA(PARTIE,\$(TETE,ISA(PARTIE,\$(NEZ,Λ)))) appartient à $\mathcal{C}^*$; par convention il sera représenté par l'expression $u_i u_j$. D'une manière plus générale le terme descriptif U_i appartient à $\mathcal{C}^*$ et est représenté par la *séquence ordonnée* $u_{i1}u_{i2}\ldots u_{ij}u_{i(j+1)}\ldots u_{in}$ si et seulement si : /1/ $\forall 1 \leqslant j \leqslant n$ u_{ij} appartient à $\mathcal{C}^*$; /2/ $u_{i(j+1)}$ décrit localement u_{ij} ; /3/ $\forall kj \in [1,n]$ u_{ik} est différent de u_{ij}. Par convention d'écriture, on pose :

$$U_i = u_{i1}u_{i2}\ldots u_{ij}\ldots u_{in}$$

d'où :

$$\mathcal{C}^* = \{\Lambda, U_i \mid U_i = u_{i1}\ldots u_{ij}\ldots u_{in} \text{ et } u_{ij} \in \mathcal{C}\}$$

Le *degré* d'un terme descriptif U , noté $|U|$, est égal au nombre de termes descriptifs non décrits qui composent la séquence le définissant. En particulier $|\Lambda|=0$ (en effet, la description vide n'est composée d'aucune occurrence de termes descriptifs non décrits).

On appellera *concaténation de deux termes descriptifs* U et V , pris dans cet ordre, le terme descriptif W obtenu en remplaçant dans U la description vide Λ par V ; elle se note comme suit :

$$W = UV$$

Il est clair que la concaténation est une loi de composition, partout définie dans $\mathcal{C}^*$, qui est associative (mais non commutative), admettant pour élément neutre Λ.

On montre très facilement que $|W| = |U|+|V|$.

On appellera *sous-terme descriptif* de W tout terme descriptif U pour lequel il existe V tel que :

$$W = UV$$

En particulier V peut être égal à Λ ; il en résulte que tout terme descriptif possède au moins un sous-terme descriptif (qui est lui-même).

On dira que V est une *partie terminale du terme descriptif* W. Il est évident que tout terme descriptif a au moins une partie terminale (qui est égale à Λ).

On appellera *différence de deux termes descriptifs* W et V, le terme descriptif U défini par l'équation : W=UV ; elle se note comme suit :

$$U = W-V$$

Il est évident que cette opération n'est pas définie partout ; pour qu'elle le soit, il faut et il suffit que V représente une partie terminale du terme descriptif W. Dans ce cas le résultat de cette opération est unique (du fait de la définition de la concaténation), et il représente un sous-terme descriptif de W.

Il est clair que $|U|=|W|-|V|$.

Enfin dans ce qui suit, on utilise les fonctions Pr(X) et Dr(X) qui ont pour valeur les termes descriptifs représentés respectivement par le premier élément et le dernier élément du terme descritpif X. On a $|Pr(X)|=|Dr(X)|=1$.

III.2.2. Définition des règles.

Les règles de transformation des termes descriptifs sont définies à partir des propositions remarquables relatives aux énoncés du type e'(x) (voir § III.1.1.). Elles définissent certaines des opérations logiques (de type "implication") qui peuvent être mobilisées par le démonstrateur du système ARCHES pour valider ou invalider les propositions qui lui sont soumises (voir chapitre huitième).

a) Règle de décomposition.

De la vérité de la proposition 4, nous tirons la règle de décomposition des termes descriptifs :

> *Si un individu est caractérisé par un terme descriptif U, alors il est caractérisé par tout sous-terme descriptif de U.*

b) Règle d'héritage.

De la vérité de la proposition 5, nous tirons la règle d'héritage des termes descriptifs :

> *Si un individu est caractérisé par un terme descriptif de la forme $U_d\ u_1\ u_2\ U_f$,u_1 et u_2 étant deux termes descritifs non décrits tels que le couple de classes correspondantes possède la propriété d'héritage, alors cet individu est caractérisé par $U_d\ u_2\ U_f$.*

Si nous supposons que le couple de classes (PARTIE, MATERIAU) possède la propriété d'héritage, alors la figure 4.10. est un exemple d'utilisation de la règle d'héritage.

c) Règle d'extension.

Pour définir et manipuler plus aisément la règle d'extension, nous introduisons la notion de terme descriptif étendu (utilisé en particulier pour la démonstration du théorème 1, voir § I.1., chapitre cinquième). Soit deux termes descriptifs non décrits u_1 et u_2 définis comme suit :

$$\begin{cases} u_1 = op_{n1}(T_1,t_{11}) \\ u_2 = op_{n2}(T_2,t_{12}) \end{cases}$$

On appellera *terme descriptif étendu* u^*_1 déterminé à partir du couple (u_1,u_2) le terme $u^*_1 = op_{n1}(T_1,t_{12})$ obtenu en remplaçant dans u_1 le tuple t_{11} par le tuple t_{12}.
On note cette opération de la manière suivante :

$$u^*_1 = u_1 [\, t_{11}/t_{12} \,]$$

Il est clair que l'opération de substitution / est associative, i.e. $(t_{1i}/t_{1j})/t_{1k}=t_{1i}/(t_{1j}/t_{1k})$. Par contre cette opération n'est pas toujours possible ; en plus de la condition exigée par la remarque 2, il est nécessaire que l'arité de l'opérateur op_{n1} soit égale à celle de op_{n2}.
Dans ces conditions de la vérité de la proposition 6, nous tirons la règle d'extension des termes descriptifs :

Si un individu est caractérisé par un terme descriptif de la forme $U_d u_1 u_2 U_f$, u_1 et u_2 étant deux termes descriptifs non décrits de la forme $op_{n1}(T_1,t_1)$ et $op_{n2}(T_2,t_2)$ tels que le couple de classes (T_1,T_2) possède la propriété d'extension (i.e. $T_1 \cap T_2 \neq \emptyset$, op_{n1} et op_{n2} ont la même arité), alors cet individu est caractérisé par $U_d u_1^* U_f$, u_1^* étant le terme descriptif étendu déterminé à partir de u_1 et u_2, soit $u_1^* = u_1[t_1/t_2]$.

Si nous supposons que le couple de classes (PARTIE, DEF(inition)) possède la propriété d'extension, alors la figure 4.11. est un exemple d'utilisation de la règle d'extension.

d) Règle de transitivité.

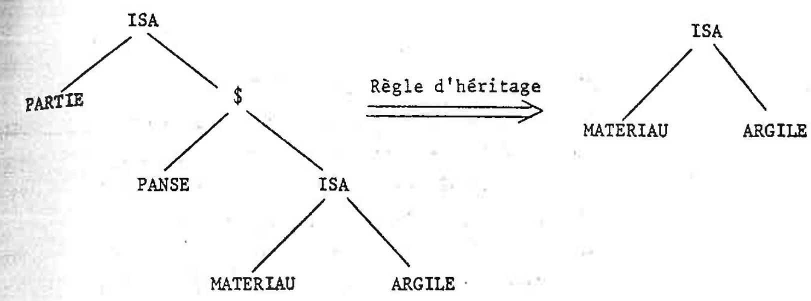
De la vérité de la proposition 7, nous tirons la règle de transitivité des termes descriptifs :

Si un individu est caractérisé par un terme descriptif de la forme $U_d\ u_1\ u_2\ U_f$, u_1 et u_2 étant deux termes descriptifs non décrits définis à partir de la même classe ayant la propriété de transitivité, alors cet individu est caractérisé par $U_d\ u_2\ U_f$.

La figure 4.12. illustre cette règle à travers la classe PARTIE qui est supposée être transitive.

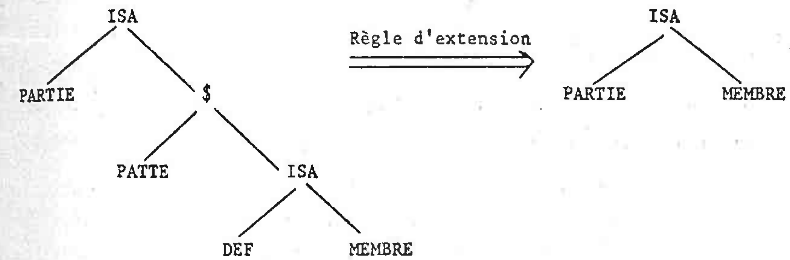
Remarque 4.

La règle de transitivité est un cas particulier de la règle d'héritage (voir à ce propos la remarque 3).



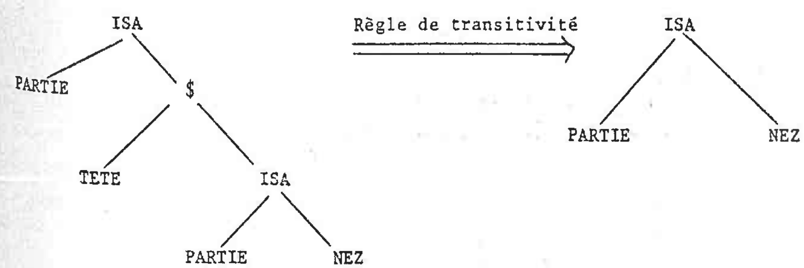
(dans cet exemple $U_d = U_f = \Lambda$)

figure 4.10. - Application de la règle d'héritage pour le couple (PARTIE, MATERIAU).



(dans cet exemple $U_d = U_f = \Lambda$)

figure 4.11. - Application de la règle d'extension pour le couple (PARTIE, DEF).



(dans cet exemple $U_d = U_f = \Lambda$)

figure 4.12. - Application de la règle de transitivité pour la classe PARTIE.

III.2.3. Propriétés axiomatiques des règles.

Nous supposons que les règles d'héritage et d'extension vérifient quatre propriétés axiomatiques particulières. Ces propriétés sont nécessaires pour démontrer que les résultats, obtenus à partir des calculs sur les termes descriptifs, ne dépendent pas de l'ordre d'application des règles de transformation (voir § I.1., chapitre cinquième). Les trois premières propriétés sont directement issues des propositions 8, 9 et 10 ; la quatrième se justifie par le fait que si le couple (T_1,T_2) possède la propriété d'héritage alors $T_1 \cap T_2 = \emptyset$.

Propriété 1. Si les couples de classes (T_1, T_2) et (T_2, T_3) possèdent la propriété d'héritage, alors le couple de classes (T_1, T_3) possède également la propriété d'héritage.

Propriété 2. Si les couples de classes (T_1, T_2) et (T_2, T_3) possèdent la propriété d'extension, alors le couple de classes (T_1, T_3) possède également la propriété d'extension.

Propriété 3. Si les couples de classes (T_1, T_2) et (T_2, T_3) possèdent respectivement la propriété d'extension et la propriété d'héritage, alors le couple (T_1, T_3) possède la propriété d'héritage.

Propriété 4. Tout couple de classes ne peut avoir simultanément la propriété d'héritage et la propriété d'extension.

III.3. Règles et relations de réécriture des termes descriptifs.

III.3.1. Les règles de substitution-réduction.

X étant un terme descriptif, nous dirons que X est une *partie* du terme descriptif A s'il existe deux termes descriptifs U et V tels que :

$$A = UXV$$

Soit X et Y deux termes descriptifs distincts tels que $|Y|<|X|$ et X est une partie du terme descriptif A. On fait correspondre à A le terme descriptif B obtenu en remplaçant l'occurrence de X dans A par une occurrence de Y.

$$A = UXV \xrightarrow{\text{transformation}} B = UYV$$

Cette transformation formelle est conforme aux règles de transformation des termes descriptifs (voir § III.2.2.) ; elle permet de définir un type particulier de règles de réécriture - *les règles de substitution-réduction* - qu'on note de la manière suivante :

$$X \xrightarrow{\circ} Y \qquad (\text{avec } |Y|<|X|)$$

Les règles de décomposition, d'héritage, de transitivité et d'extension déterminent ainsi les règles de substitution-réduction particulières ci-après :

α) $U \xrightarrow{\circ} \Lambda$, pour tout $U \in \mathcal{C}^*$ tel que U est toujours une partie terminale de terme descriptif.

β) $u_1u_2 \xrightarrow{\circ} u_2$, si les classes des termes descriptifs non décrits u_1 et u_2 possèdent la propriété d'héritage, ou si ces classes sont les mêmes et ont la propriété de transitivité.

γ) $u_1u_2 \xrightarrow{\circ} u_1^*$, si les classes des termes descriptifs non décrits u_1 et u_2 possèdent la propriété d'extension.

Exemple.

Le couple de classes (PARTIE, MATERIAU) a la propriété d'héritage ; d'où la règle de substitution-réduction suivante :

ISA(PARTIE,\$(x,ISA(MATERIAU,y))) $\xrightarrow{\circ}$ ISA(MATERIAU,y)

D'une manière générale, on se donne pour toute application le schéma de règle de type (α) (correspondant à la règle de décomposition) et plusieurs règles de type (β) ou de type (γ) ; l'ensemble de ces règles définit le *système de règles de substitution-réduction* du système ARCHES, à partir duquel sont produites les dérivations des termes descriptifs.

III.3.2. Les relations de réécriture.

a) La relation de réécriture $\longrightarrow$.

Etant donné un système de règles de substitution-réduction S, on dira que

deux termes descriptifs A et B sont *contigus* si B est la transformation de A par application d'une seule règle $R_i \in S$.
Cette relation de contiguïté est une relation de réécriture, notée $\longrightarrow$ qui est définie sur $\mathcal{C}^*$ comme suit :

$$A \longrightarrow B \text{ si et seulement si } \left| \begin{array}{l} X \overset{\circ}{\longrightarrow} Y \in S \text{ et } \exists U,V \in \mathcal{C}^* \\ \text{tels que : } A = UXV \text{ et} \\ \qquad\qquad B = UYV \end{array} \right.$$

On dira également que B *dérive directement* de A.

Il est évident que pour n'importe quelle règle $X \overset{\circ}{\longrightarrow} Y \in S$, la relation $X \longrightarrow Y$ est toujours vérifiée.

Remarque 5.

La relation de réécriture $\longrightarrow$ est toujours une opération de réduction. En effet si on désigne par $PG(\overset{\circ}{\longrightarrow})$ et $PD(\overset{\circ}{\longrightarrow})$ les parties gauche et droite de la règle $\overset{\circ}{\longrightarrow}$, on a toujours :

$$|PG(\overset{\circ}{\longrightarrow})| > |PD(\overset{\circ}{\longrightarrow})|$$

b) La relation de réécriture $\overset{*}{\longrightarrow}$.

A partir de la relation $\longrightarrow$, on définit la relation de réécriture $\overset{*}{\longrightarrow}$ sur $\mathcal{C}^*$ comme la fermeture transitive et réflexive de la relation $\longrightarrow$.

$$A_o \overset{*}{\longrightarrow} A_p \quad \text{ssi} \quad \left| \begin{array}{l} A_o = A_p \\ \text{ou} \\ \exists \text{ une suite finie } A_o, \ldots, A_{i-1}, A_i, \ldots \\ \text{telle que } \forall 1 \leqslant i \leqslant p \quad A_{i-1} \longrightarrow A_i \end{array} \right.$$

On dira que A_p *dérive* de A_o.
La figure 4.13. illustre un exemple de dérivation.

Si $A \overset{*}{\longrightarrow} B$ alors $|A| \geqslant |B|$ car les opérations de dérivation sont toujours réductrices (voir remarque 5). Soit $S' = S - \{U \overset{\circ}{\longrightarrow} \Lambda\}$ l'ensemble des règles de substitution-réduction à l'exception de la règle de décomposition.

Constituons à partir d'un terme descriptif A donné a priori l'ensemble des termes descriptifs obtenus par réductions successives à partir de l'application des règles de S'. L'opération de dérivation s'arrête sur tout terme descriptif intermédiaire sur lequel on ne peut plus appliquer la relation $\longrightarrow$; on appellera cet élément de description *terme descriptif irréductible* (on montre qu'il est unique, voir § I.1., chapitre cinquième).

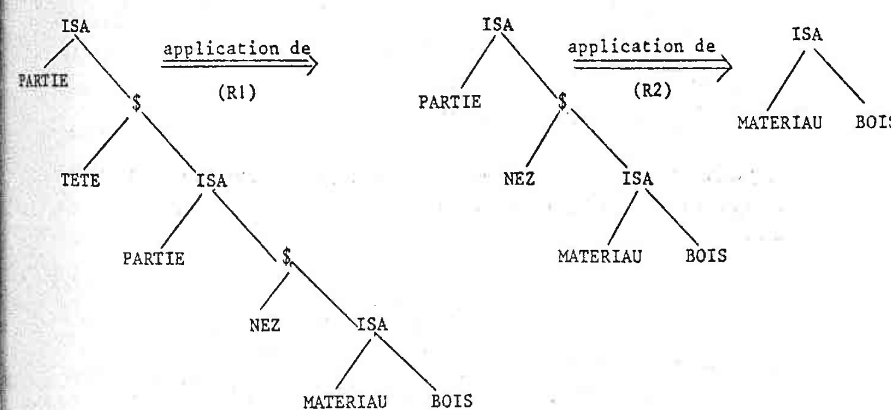
. Système de règles.

$$(S) \left\{ \begin{array}{l} R1 - ISA(PARTIE,\$(x,ISA(PARTIE,y))) \overset{o}{\longrightarrow} ISA(PARTIE,y) \\ R2 - ISA(PARTIE,\$(x,ISA(MATERIAU,y))) \overset{o}{\longrightarrow} ISA(MATERIAU,y) \end{array} \right.$$

. Terme descriptif à dériver.

A=ISA(PARTIE,$(TETE,ISA(PARTIE,$(NEZ,ISA(MATERIAU,BOIS)))))

. dérivation



(le résultat ne dépend pas de l'ordre d'application des règles)

figure 4.13. - Exemple de dérivation de A à partir du système S.

IV. L'ORGANISATION DEDUCTIVE DES DESCRIPTIONS.

Pour compléter la définition du système formel S_Δ de caractérisation des descriptions, nous devons définir les modalités de dérivation des descriptions, compte tenu d'une part de leurs règles de formation à partir des termes descriptifs et des connecteurs $*$, $+$ et $\neg$ (voir § II.4.2.), et d'autre part des relations de réécriture des termes descriptifs (voir § III.3.). Pour ce faire nous définissons, de manière analogue à celle que propose PORTE dans {109}, une relation de déduction entre les descriptions qui, comme toutes les relations de ce type, est réflexive et transitive (voir § IV.1.).Les conditions spécifiques auxquelles cette relation doit satisfaire permet de définir des schémas de dérivation qui déterminent les règles d'insertion et d'élimination des connecteurs $*$ et $+$ (voir § IV.2.). Enfin ces mêmes conditions permettent d'établir la nature déductive de cette relation (voir § IV.3.).

IV.1. Les modalités de déduction des descriptions.

IV.1.1. La relation de déduction $\Longrightarrow$.

Nous définissons sur l'ensemble Δ *une relation de déduction entre les descriptions, notée* $\Longrightarrow$, qui est la plus petite relation réflexive et transitive vérifiant les conditions suivantes :

C1 $a*b \Longrightarrow a$; $a*b \Longrightarrow b$

C2 si $a \Longrightarrow b$ alors $a*c \Longrightarrow b*c$

C3 si $a*\neg b \Longrightarrow b$ alors $a \Longrightarrow b$

C4 si $a \longrightarrow^{*} b$ alors $a \Longrightarrow b$

Les conditions C1 et C2, qui fixent les rapports logiques entre la relation $\Longrightarrow$ et le connecteur $*$, confèrent à la dérivation engendrée par $\Longrightarrow$ le statut de *déduction* (voir § IV.3.1.).
La condition C3 affirme à l'aide de la relation $\Longrightarrow$ que la définition de la négation est fondée sur le *raisonnement par l'absurde*. Cette définition sera complétée par la condition C11 (voir § IV.1.2.).
Enfin la condition C4 détermine les rapports déductifs (relations de spécifique à générique) entre termes descriptifs et descriptions. En particulier elle permet de montrer que tout terme descriptif décrit construit sur n termes descriptifs non décrits véhicule moins d'information que l'addition de ces n termes descriptifs non décrits (voir § IV.3.2.). Par ailleurs elle indique que,quel que soit la relation R réflexive et transitive qui vérifie les conditions précédentes, la formule $a R b$ est toujours vérifiée si la formule $a \longrightarrow^{*} b$ est vérifiée car cette dernière ne dépend pas de la relation $\Longrightarrow$.

IV.1.2. La relation d'égalité =.

On dira que $a=b$ (la description a *"est égale"* à la description b) si et seulement si $a \Longrightarrow b$ et $b \Longrightarrow a$.

On montre très facilement que la relation = est une relation d'équivalence.

La relation =, et donc également la relation $\Longrightarrow$, satisfait aux conditions suivantes :

C5	$a*b=b*a$	; $a+b=b+a$
C6	$a*(b*c)=(a*b)*c$	; $a+(b+c)=(a+b)+c$
C7	$a*a=a$	; $a+a=a$
C8	$a*\Lambda=\Lambda*a=\Lambda$	; $a+\Lambda=\Lambda+a=a$
C9	$a*(a+b)=a$	; $a+(a*b)=a$
C10	$a*(b+c)=(a*b)+(a*c)$	; $a+(b*c)=(a+b)*(a+c)$
C11	$a*\neg a=\neg a*a=\Lambda$	

Ces conditions déterminent les propriétés sémantiques des connecteurs $*$ (ET d'addition),$+$ (OU non exclusif), et $\neg$ (négation).
Le ET d'addition sert à coordonner des termes descriptifs qui ont la même fonction par rapport aux problèmes de caractérisation des individus. Ainsi dans les exemples des figures 4.6. et 4.7., il permet d'affirmer simultanément les énoncés élémentaires "MARIE est BELLE", "MARIE est GRANDE", "AMPHORE ayant pour ornement des BANDEAUX" et enfin "AMPHORE ayant la forme CYLINDRIQUE". Il est évident que ce connecteur est commutatif et associatif. De plus, nous supposons que la répétition d'un même

trait descriptif (exemple ; MARIE est BELLE, et BELLE") ne traduit aucun effet linguistique ; en d'autres termes ce connecteur possède la propriété d'idempotence. Enfin, la description vide Λ joue le rôle d'élément contradictoire car il est impossible de décrire au moyen de Λ tout individu ayant une existence logique (voir § II.4.1., hypothèse 3).

De même les propriétés d'associativité, de commutativité et d'idempotence caractérisent le OU non exclusif ; par contre la description vide Λ joue le rôle d'élément neutre pour cette opération.
Les conditions C9 se présentent comme une généralisation de la propriété d'idempotence (on le montre très facilement en posant b=a+c ou b=a*c). Elles renforcent l'idée qui consiste à ne pas considérer la répétition d'un même terme descriptif intervenant dans une opération * ou + comme une marque pertinente dans ARCHES. Si la condition C10 relative au connecteur * paraît naturel, celle relative au connecteur + l'est beaucoup moins dans la mesure où elle présuppose l'existence des conditions C9 (condition d'absorption) (cette condition se démontre facilement à partir des conditions C9 et C10 relatives au connecteur *).

La condition C11 donne une définition restreinte de la négation qui se présente, à ce stade, comme une négation de type intuitioniste. Mais les deux conditions C3 et C11 conduisent au principe du tiers exclu ; il en résulte que le quotient $\Delta/=$ est un *treillis distributif complémenté* (voir § II.2., chapitre cinquième ; proposition 10 et suivantes).

IV.2. Règles d'insertion et d'élimination des connecteurs * et +.

Les règles d'insertion et d'élimination des connecteurs * et + sont définies comme des schémas de dérivation à partir de la relation de déduction $\Longrightarrow$. Elles permettent d'associer à une suite finie de formules du type $a \Longrightarrow b$: $e_1, e_2, \ldots, e_n$ $(n \geq 1)$, une nouvelle formule : e ; elles sont définies formellement de la manière suivante :

$$\frac{e_1\ ;\ e_2\ ;\ldots;\ e_n}{e} \qquad (1)$$

La règle (1) s'interprète syntactiquement comme suit : si toutes les prémisses $e_1, \ldots, e_n$ sont vérifiées en même temps (i.e. interprétation métalinguistique du signe ";"), alors la conclusion e est également admise.

IV.2.1. Règles relatives au connecteur *.

R1 $\quad \dfrac{a \Longrightarrow b}{a*c \Longrightarrow b}$ $\qquad$ R2 $\quad \dfrac{a \Longrightarrow b\ ;\ a \Longrightarrow c}{a \Longrightarrow b*c}$

R3 $\quad \dfrac{a \Longrightarrow b*c}{a \Longrightarrow b}$ $\qquad$ et $\quad \dfrac{a \Longrightarrow b*c}{a \Longrightarrow c}$

IV.2.2. Règles relatives au connecteur +.

R4 $\quad \dfrac{a \Longrightarrow b}{a \Longrightarrow b+c}$ $\qquad$ R5 $\quad \dfrac{a \Longrightarrow b\ ;\ c \Longrightarrow b}{a+c \Longrightarrow b}$

R6 $\quad \dfrac{a+c \Longrightarrow b}{a \Longrightarrow b}$ $\qquad$ et $\quad \dfrac{a+c \Longrightarrow b}{c \Longrightarrow b}$

Ces six règles seront prouvées ultérieurement à partir des conditions C1 à C11 et du théorème 1 ci-après (voir § II.1., chapitre cinquième ; propositions 1, 2, 3 et 4).

IV.3. Nature déductive de la relation $\Longrightarrow$.

IV.3.1. Justification.

a) Théorème.

Théorème 1. La formule $a \Longrightarrow b$ est vérifiée si et seulement si la formule a*b=a est vérifiée.

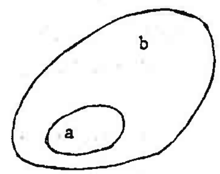
démonstration. Si $a \Longrightarrow b$ est vérifiée, alors $a*a \Longrightarrow b*a$ (d'après C2) ; or d'après C7 $a \Longrightarrow a*a$ d'ou $a \Longrightarrow b*a$ (transitivité). Par ailleurs $a*b \Longrightarrow a$ d'après C1 ; il en résulte que a*b=a. Réciproquement si a*b=a il vient $a \Longrightarrow a*b$ (définition de =) ; comme $a*b \Longrightarrow b$ d'après C1, il en résulte que $a \Longrightarrow b$ par transitivité (c.q.f.d.).

Remarque 6.

On déduit immédiatement de la relation a*b=a l'égalité a+b=b (d'après C9) et réciproquement.

b) Interprétation.

Le théorème précédent exprime que si la formule a $\Longrightarrow$ b est vérifiée, alors la description a véhicule une information plus *"spécifique"* que celle véhiculée par la description b ; comme peut le montrer l'interprétation ensembliste du connecteur * (l'intersection de a et de b est incluse dans b) :



$$\begin{cases} a*b=a \\ a+b=b \end{cases}$$

La relation $\Longrightarrow$ est une inférence qui transforme toujours une information en une information plus générale ; elle définit donc une opération de dérivation de type déductif.

c) Statut de la description vide.

Proposition 11. La formule $\Lambda \Longrightarrow a$ est toujours vérifiée $\forall a \in \Delta$

Cette proposition est évidente car $\Lambda*a=\Lambda$ et $\Lambda+a=a$ (condition C8). Comme $\Longrightarrow$ est une relation d'ordre, la description vide Λ représente le *minorant* de tous les éléments de Δ. D'un point de vue déductif, Λ peut être interprétée comme la description toujours "fausse" à partir de laquelle on peut dériver n'importe quelle description (voir § III.1., Y /3/ chapitre sixième ; point 7).

IV.3.2. Rapports entre la relation $\xrightarrow{*}$ et les connecteurs * et +.

Proposition 12. $\forall a,b \in \mathscr{C}^*$, la formule $a \xrightarrow{*} b$ est vérifiée si et seulement si $a*b=a$.

(démonstration immédiate à partir de la condition C4 et du théorème 1).

Exemple.

Si le couple de classes (COULEUR, TEINTE) possède la propriété d'héritage alors la relation suivante est vérifiée :

ISA(COULEUR,\$(BLEU,ISA(TEINTE,FONCE))) $\xrightarrow{*}$ ISA(TEINTE,FONCE)

On en déduit immédiatement d'après la proposition 12 :

ISA(COULEUR,\$(BLEU,ISA(TEINTE,FONCE)))*ISA(TEINTE,FONCE)= ISA(COULEUR,\$(BLEU,ISA(TEINTE,FONCE))).

Proposition 13. Si la formule $a_1*a_2*\ldots*a_n \Longrightarrow b$ est vérifiée et si b appartient à $\mathscr{C}^*$ ou est de la forme $\neg b'$ avec b' appartient à $\mathscr{C}^*$, alors l'une des formules $a_i \Longrightarrow b$ $(1 \leq i \leq n)$ est vérifiée.

démonstration. Supposons que la formule $a_1*a_2*\ldots*a_i*\ldots*a_n \Longrightarrow b$ soit vérifiée. Comme b appartient à $\mathscr{C}^*$ ou est de la forme $\neg b'$ avec b' appartenant à $\mathscr{C}^*$ il en résulte que b ne contient aucun connecteur de type * (énoncé inanalysé). Il est donc nécessaire au cours de la démonstration de cette formule d'éliminer en premier lieu le connecteur * de son antécédent car quel que soit la relation R réflexive et transitive qui vérifie les conditions C1 à C11 la seule formule toujours vérifiée est $a_1*a_2*\ldots*a_i*\ldots*a_n \Longrightarrow a_1*a_2*\ldots*a_i*\ldots*a_n$. Or il n'existe qu'une seule règle d'élimination de ce connecteur, à savoir la règle R3 (qui correspond en fait à la condition C1). Il existe donc une étape de sa démonstration où doit être nécessairement prouvée la formule $a_i \Longrightarrow b$ $(1 \leq i \leq n)$. En effet d'après la condition C4 la formule $a_i \Longrightarrow b$ ne peut être vérifiée que si après élimination des connecteurs *, + ou $\neg$ dans son antécédent, il existe un terme descriptif a appartenant à $\mathscr{C}^*$ tel que la formule $a \xrightarrow{*} b$ (ou $b' \xrightarrow{*} a$) soit vérifiée (méthode de démonstration "constructive" de type GENTZEN pour prouver la démonstrabilité d'un énoncé, {50}). Dans ce cas quel que soit la relation R réflexive et transitive qui vérifie les conditions C1 à C11, la formule aRb (ou b'Ra) est toujours vérifiée ; il en est donc de même de la formule $a \Longrightarrow b$ (ou $b' \Longrightarrow a$) car $\Longrightarrow$ est la plus petite relation réflexive et transitive qui vérifie les conditions C1 à C11. Il en résulte que la formule $a_i \Longrightarrow b$ $(1 \leq i \leq n)$ est vérifiée par transitivité (c.q.f.d.).

Corollaire. Si la formule $a_1*a_2*\dots*a_n \Longrightarrow b_1+\dots+b_j+\dots+b_p$ est vérifiée et si $\forall j \in [1,p]$ b_j appartient à $\mathcal{C}^*$ ou b_j est de la forme $\neg b'_j$ avec b'_j appartient à $\mathcal{C}^*$, alors l'une des formules $a_i \Longrightarrow b_1+\dots+b_j+\dots$ $(1 \leq i \leq n)$ est vérifiée.

(démonstration immédiate à partir de la proposition précédente).

Proposition 14. Si la formule $a \Longrightarrow b_1+\dots+b_j+\dots+b_p$ est vérifiée et si a appartient à $\mathcal{C}^*$, alors l'une des formules $a \Longrightarrow b_j$ $(1 \leq j \leq p)$ est vérifiée.

(démonstration analogue à la proposition précédente en remarquant qu'il n'existe qu'une seule règle d'insertion du connecteur + dans le conséquent d'une relation $\Longrightarrow$, à savoir la règle R4).

Exemple.

Soit le système de règles de réécriture suivant :

$$(S) \left| \begin{array}{ll} U \xrightarrow{\circ} \Lambda & \text{(règle de décomposition)} \\ ISA(COULEUR,\$(x,ISA(TEINTE,y))) \xrightarrow{\circ} ISA(TEINTE,y) & \\ & \text{(règle d'héritage)} \end{array} \right.$$

La condition C4 et la règle R2 permettent de prouver l'énoncé :

ISA(COULEUR,$(BLEU,ISA(TEINTE,CLAIR))) $\Longrightarrow$ ISA(COULEUR,BLEU)*ISA(TEINTE,C

Par contre la relation symétrique :

ISA(COULEUR,BLEU)*ISA(TEINTE,CLAIR) $\Longrightarrow$ ISA(COULEUR,$(BLEU,ISA(TEINTE,CLA

n'est pas démontrable d'après la proposition 13. Ceci montre bien que to terme descriptif décrit construit sur n termes descriptifs non décrits es plus spécifique que l'addition de ces n termes descriptifs non décrits.

CHAPITRE CINQUIEME.

LES PROPRIETES DU SYSTEME FORMEL S_Δ

I. ETUDE DES PROPRIETES DE LA RELATION $\xrightarrow{*}$.

I.1. Confluence de la relation $\xrightarrow{*}$.

Désignons par S le système de règles de substitution -réduction et par S' celui correspondant aux règles d'héritage, de transitivité et d'extension : $S' = S - \{ U \xrightarrow{\circ} \Lambda \}$ dans lequel $U \xrightarrow{\circ} \Lambda$ représente le schéma de règles de décomposition. Nous démontrons dans ce paragraphe un théorème de confluence qui montre que tout terme descriptif a un seul terme descriptif irréductible, quel que soit l'ordre d'application des règles appartenant à S' (voir § III.3.,chapitre quatrième). Pour ce faire, nous démontrons un lemme préliminaire qui montre que si la formule $A \xrightarrow{*} B$ est vérifiée par application des règles de S alors la règle de décomposition, si elle est utilisée, doit être appliquée la dernière.

Lemme. Si la formule $A \xrightarrow{*} B$ est vérifiée par application des règles de S, alors il existe un terme descriptif A' tel que d'une part la formule $A \xrightarrow{*} A'$ est vérifiée soit par application des règles de S' soit par reflexivité (i.e. $A' = A$), et que d'autre part la formule $A' \longrightarrow B$ est vérifiée par application de la seule règle de décomposition.

démonstration. Supposons que la formule $A \xrightarrow{*} B$ soit vérifiée, A et B étant respectivement de la forme $u_1 u_2 .. u_i .. u_n$ et de la forme $v_1 v_2 .. v_j .. v_p$, avec bien évidemment $p \leqslant n$. Si $p = n$ alors on a nécessairement $A = B$ et par application du schéma de règle de décomposition dans lequel U prend la valeur Λ , on tire $A \longrightarrow B$. Ce cas élémentaire étant examiné, supposons désormais que $p < n$. Remarquons tout d'abord que l'application de toute règle R_e appartenant à S' ne modifie pas l'ordre relatif d'occurrence des termes descriptifs u_i qui composent A.

La seule fonction des règles de S' est d'opérer un tassement avec suppression des termes descriptifs non décrits. Comme la formule $A \xrightarrow{*} B$ est vérifiée, il en résulte qu'il existe nécessairement dans A soit un $k \geqslant p$ tel que $v_p = u_k$ soit un doublet (k,k') avec $k' \geqslant p$ et $k' < k$ tel que $v_p = u^*_{k'}$, ($u^*_{k'}$, étant le terme descriptif étendu construit à partir de u_k et $u_{k'}$; voir § III.2.2.c, chapitre quatrième). Nous devons noter que u_k est unique dans A car $\forall i j \in [1,n]$ on a $u_i \neq u_j$ (voir § III.2.1., chapitre quatrième) ; et que la position relative de u_k dans A n'est pas modifiée par application des règles de S'. Dans ces conditions la règle de décomposition ne peut être appliquée sur A que si le schéma $U \xrightarrow{\circ} A$ se réalise de telle manière que U est une partie terminale de A vérifiant $|U| \leqslant n-k$. En effet dans le cas contraire (i.e. $|U| > n-k$), l'application de la règle de décomposition aurait pour conséquence de supprimer au moins u_k, et donc d'invalider la formule $A \xrightarrow{*} B$. Il en résulte que la formule $u_1 u_2 \dots u_k \xrightarrow{*} B$ est nécessairement vérifiée par application des règles appartenant à S' ; On en déduit que $A \xrightarrow{*} B u_{k+1} \dots u_n$. Il existe donc un terme descriptif A' (égal à $B U_{k+1} \dots U_n$) tel que $A \xrightarrow{*} A'$ par application des règles appartenant à S' et $A' \longrightarrow B$ par application du schéma $U \xrightarrow{\circ} A$ avec $U = u_{k+1} \dots u_n$ (c.q.f.d.).

Le théorème de confluence que nous démontrons ci-après est de même nature que ceux démontrés dans le cadre de l'étude formelle et générale des systèmes de réécriture ({79}, {80}).

Théorème 1. Etant donné un système de règles S', $\forall A \in \mathcal{C}^*$ il existe un et un seul terme descriptif irréductible B tel que la formule $A \xrightarrow{*} B$ soit vérifiée.

démonstration. Supposons que les règles $\xrightarrow{\circ}$ de S' soient totalement indépendantes les unes des autres, c'est à dire que si $u_i u_j \xrightarrow{\circ} u_k$ et $u'_i u'_j \xrightarrow{\circ} u'_k$ sont deux règles quelconques alors $\forall i j$ $u_j \neq u'_i$. Les transformations que peut subir A sont toutes indépendantes les unes des autres, chaque transformation ne produisant qu'un seul résultat. Quand l'opération de réduction s'arrête, on obtient un seul résultat qui est précisément le terme descriptif irréductible ; il est donc unique.

Supposons désormais que les règles de S' ne soient pas indépendantes et qu'il existe au moins deux règles telles que

$$\begin{cases} u_1 u_2 \xrightarrow{\circ} u' & (1) \\ u_2 u_3 \xrightarrow{\circ} u'' & (2) \end{cases}$$

Les résultats intermédiaires obtenus dépendent alors de l'ordre d'application des règles ainsi que du contexte dans lequel elles sont mises en oeuvre. Désignons par $Cu_1u_2u_3D$ le terme descriptif obtenu à la (k-1)$^{\text{ième}}$ réduction, la nouvelle dérivation pouvant utiliser les règles (1) et (2) :

$$\begin{tikzcd} & Cu_1u_2u_3D \arrow[dl, "(1)"'] \arrow[dr, "(2)"] & \\ Cu'u_3D & & Cu_1u''D \end{tikzcd} \qquad (k^{\text{ième}} \text{ dérivation})$$

Examinons toutes les possibilités de réduction liées à l'application de ces deux règles en fonction de leur nature (règles d'héritage, d'extension et de transitivité).

<u>Premier cas</u>. (1) et (2) définissent des règles d'héritage.

$$\begin{cases} u_1 u_2 \xrightarrow{\circ} u_2 \\ u_2 u_3 \xrightarrow{\circ} u_3 \end{cases}$$

Les couples de classes (T_1,T_2) et (T_2,T_3) possèdent la propriété d'héritage ; le couple (T_1,T_3) possède aussi cette propriété (propriété 1, voir § III.2.3. , chapitre quatrième). Dans ce cas $Cu_1u_2u_3D$ peut être réduit comme suit :

$$\begin{tikzcd} & Cu_1u_2u_3D \arrow[dl, "(1)"'] \arrow[dr, "(2)"] & \\ Cu_2u_3D \arrow[d, "(2)"] & & Cu_1u_3D \arrow[d] \\ Cu_3D & & Cu_3D \end{tikzcd} \qquad \text{propriété d'héritage}$$

Quel que soit l'ordre d'application des règles, on obtient un et un seul résultat intermédiaire.

<u>Deuxième cas</u>. (1) et (2) définissent des règles d'extension (voir § III.2.2.c, chapitre quatrième).

$$\begin{cases} u_1 u_2 \xrightarrow{\circ} u_1[t_1/t_2] \\ u_2 u_3 \xrightarrow{\circ} u_2[t_2/t_3] \end{cases}$$

D'après la propriété 2 (§ III.2.3, chapitre quatrième),le couple de classes (T , T) possède également la propriété d'extension

$$\begin{tikzcd}
 & Cu_1u_2u_3D \arrow[dl, dash, "(1)"'] \arrow[dr, dash, "(2)"] & \\
Cu_1[t_1/t_2]u_3D \arrow[d, dash, "\text{propriété d'extension}"'] & & Cu_1u_2[t_2/t_3]D \arrow[d, dash, "(1)"] \\
Cu_1[t_1/t_2/t_3]D & & Cu_1[t_1/t_2/t_3]D
\end{tikzcd}$$

On obtient encore dans ce cas un seul résultat intermédiaire.

Troisième cas. (1) définit une règle d'extension et (2) une règle d'héritage.

$$\begin{cases} u_1u_2 \xrightarrow{\circ} u_1[t_1/t_2] \\ u_2u_3 \xrightarrow{\circ} u_3 \end{cases}$$

D'après la propriété 3 (§ III. 2.3., chapitre quatrième) le couple de classes (T_1, T_3) possède la propriété d'héritage.

$$\begin{tikzcd}
 & Cu_1u_2u_3D \arrow[dl, dash, "(1)"'] \arrow[dr, dash, "(2)"] & \\
Cu_1[t_1/t_2]u_3D \arrow[d, dash] & & Cu_1u_3D \arrow[d, dash] \\
Cu_3D & & Cu_3D
\end{tikzcd}$$

Là aussi un seul résultat intermédiaire.

Quatrième cas. (1) définit une règle d'héritage et (2) une règle d'extension.

$$\begin{cases} u_1u_2 \xrightarrow{\circ} u_2 \\ u_2u_3 \xrightarrow{\circ} u_2[t_2/t_3] \end{cases}$$

$$\begin{tikzcd}
 & Cu_1u_2u_3D \arrow[dl, dash, "(1)"'] \arrow[dr, dash, "(2)"] & \\
Cu_2u_3D \arrow[d, dash, "(2)"'] & & Cu_1u_2[t_2/t_3]D \arrow[d, dash, "(1)"] \\
Cu_2[t_2/t_3]D & & Cu_2[t_2/t_3]D
\end{tikzcd}$$

Ce cas fournit également un seul résultat intermédiaire.

Nous n'examinons pas les cas qui font intervenir la règle de transitivité car il est évident qu'ils restituent un seul résultat intermédiaire quel que soit l'ordre d'application des règles, du fait précisément de la transitivité des classes correspondantes. De même nous ne considérons pas les règles qui ont même premier membre (i.e. $u_1u_2 \xrightarrow{\circ} u'$ et $u_1u_2 \xrightarrow{\circ} u''$) car tout couple de classes (T_1, T_2) ne peut avoir simultanément la propriété d'héritage (ou de transitivité) et la propriété d'extension (voir § III.2.3., chapitre quatrième ; propriété 4).

Nous constatons que quel que soit le type de règles mises en jeu et quelque soit l'ordre d'application des règles liées, on obtient au cours de deux opérations consécutives de réduction le même résultat intermédiaire. Nous sommes donc ramenés au cas précédent. Enfin la confluence locale (i.e. réduction d'expressions de la forme $Cu_1u_2u_3D$ conduisant à un résultat unique) permet de démontrer très facilement la confluence globale (i.e. réduction d'expressions de la forme $Cu_1u_2Eu_2u_3D$ conduisant à un résultat unique) (voir à ce sujet {79}). Quand l'opération de réduction s'arrête on obtient le résultat final qui ne dépend pas de l'ordre d'application des règles : c'est le terme descriptif irréductible, et il est unique (c.q.f.d.).

Remarque.

Le cas $u_1u_2 \xrightarrow{\circ} u'$ et $u_1u_2 \xrightarrow{\circ} u''$ n'a pas à être envisagé car tout couple de classes ne peut avoir simultanément la propriété d'héritage et la propriété d'extension (voir § III.2.3., chapitre quatrième ; propriété 4).

I.2. Conditions de décidabilité de la relation $\xrightarrow{*}$.

I.2.1. Définition générale.

On dira qu'un ensemble est *décidable vis à vis d'une relation* $\mathcal{R}$ s'il existe un *procédé effectif* permettant d'affirmer si une formule quelconque $a\mathcal{R}b$ est ou n'est pas vérifiée.

Par procédé effectif, nous entendons un procédé qui permet de valider ou d'invalider toute formule $a\mathcal{R}b$ en un nombre fini d'étapes avec des opérations prévues à l'avance.

Dans ce qui suit nous posons et résolvons le problème de décidabilité pour tout ensemble $\mathcal{C}^*$ de termes descriptifs muni de la relation $\xrightarrow{*}$: étant donné $A,B \in \mathcal{C}^*$ déterminer s'ils vérifient la formule $A \xrightarrow{*} B$.

I.2.2. Première méthode.

Théorème 2. Si deux termes descriptifs A et B vérifient la formule $A \xrightarrow{*} B$, alors il existe un terme descriptif B' tel que A et BB' ont le même terme descriptif irreductible.

démonstration. Si la formule $A \xrightarrow{*} B$ est vérifiée alors il existe un terme descriptif A' tel que $A \xrightarrow{*} A'$ est démontrée par application des règles appartenant à S' et $A' \longrightarrow B$ est vérifiée par application de la règle de décomposition (voir le lemme du § I.1.). Il existe donc un terme descriptif B' tel que A' = BB', d'où $A \xrightarrow{*} BB'$. Soit A_R et B_R les termes descriptifs irréductibles de A et BB' ; on a donc $A \xrightarrow{*} A_R$ et $BB' \xrightarrow{*} B_R$. Comme $A \xrightarrow{*} BB'$, il en résulte que $A \xrightarrow{*} B_R$ (transitivité de $\xrightarrow{*}$) ; on en déduit que $A_R = B_R$ car tout terme descriptif a un seul terme descriptif irréductible d'après le théorème 1 (c.q.f.d.).

Ce théorème est une condition nécessaire pour résoudre le problème posé ; il n'y a pas de condition suffisante.

Cependant nous donnons ci-après une procédure, notée $\mathcal{A}_c$, permettant de décider si la formule $A \xrightarrow{*} B$ est vérifiée. Elle s'appuie sur le lemme qui a été démontré dans le paragraphe précédent et qui affirme que si la formule $A \xrightarrow{*} B$ est vérifiée alors la règle de décomposition, si

elle est utilisée, doit être appliquée la dernière.

Cette procédure $\mathcal{A}_c$ génère systématiquement l'ensemble $\mathcal{C}_1$ de tous les termes descriptifs contigus à A (pas 1 de l'algorithme), puis l'ensemble $\mathcal{C}_2$ de tous les termes descriptifs contigus à chacun d'entre eux (pas 2 de l'algorithme), et ainsi de suite (figure 5.1.). L'algorithme s'arrête au pas p qui correspond à la génération de l'ensemble $\mathcal{C}_p$ de tous les termes descriptifs dont la longueur est égale à celle de B($r=n-p\geq 1$, avec $|A|=n$, $|B|=r$ et $n>r$). Si la formule $A \longrightarrow B$ est vérifiée par application de la règle de décomposition ou si $B \in \mathcal{C}_p$ ou si il existe $1\leq i<p$ tel que $A^i_j \in \mathcal{C}_i$ et la formule $A^i_j \longrightarrow B$ est vérifiée par application de la règle de décomposition, alors la formule $A \xrightarrow{*} B$ est vraie ; dans le cas contraire elle est fausse.

Les performances de cet algorithme sont en première approximation proportionnelles à la longueur cumulée Lmaxc de tous les termes descriptifs générés :

$$Lmaxc \leqslant (n-1)^2+(n-1)(n-2)^2+\ldots+(n-1)(n-2)\ldots(n-p)^2$$

Soit $t_i=(n-1)(n-2)\ldots(n-i)^2$ le terme général de cette somme. Le rapport $\frac{t_{i+1}}{t_i}$ est équivalent à n quand n augmente indéfiniment ; dans ces conditions la série géométrique $\sum_{i=1}^{p} n^{i+1}$ est une expression approchée de Lmaxc. Il en resulte que pour n grand Lmaxc varie comme la fonction kn^n, k étant une constante positive. Cette procédure, qui explose rapidement, reste cependant acceptable pour n petit ; ceci est le cas dans la pratique car définir des propriétés de propriétés ou des relations d'état de relations d'état sur une profondeur supérieure à cinq niveaux est une opération intellectuelle relativement complexe.

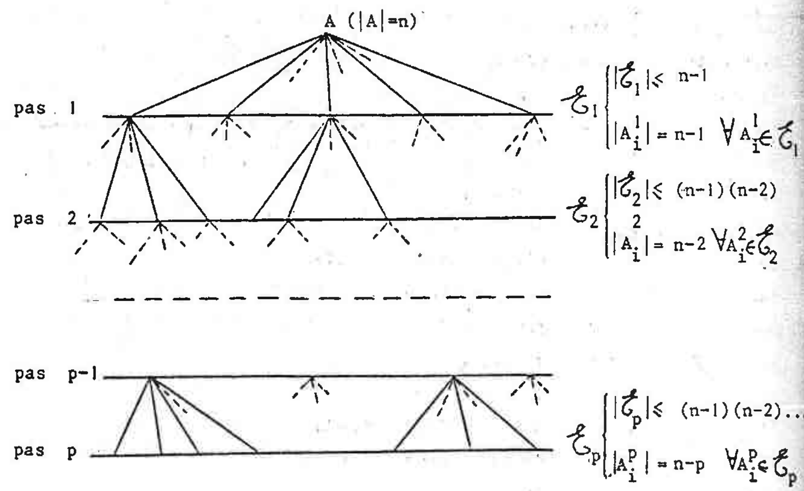


figure 5.1. – *Arbre de tous les termes descriptifs contigus générés à partir de A par l'algorithme* $\mathcal{A}_{\mathcal{C}}$.

I.2.3. Deuxième méthode.

Nous proposons dans ce qui suit un autre algorithme plus performant, noté $\mathcal{A}_R$, fondé sur une procédure de recherche du terme descriptif irréductible de tout terme descriptif A, notée $\mathcal{A}_I$, et sur la notion de *fonction de discordance* de deux termes descriptifs, notée D.

a) Algorithme $\mathcal{A}_I$ de recherche du terme descriptif irréductible.

Cet algorithme consiste à explorer le terme descriptif A, de longueur N, de la gauche vers la droite en cherchant à unifier à chaque position du curseur I tout doublet $u_i u_{i+1}$ avec la partie gauche d'une règle R_j appartenant au système de règles S' , soit $PG(R_j)$. Son remplacement (avec tassement) par la partie droite $PD(R_j)$ de cette règle fournit un terme contigu. L'exploration itérative de la chaine s'arrête quand on ne peut plus appliquer de règles $\xrightarrow{\circ}$: le terme descriptif correspondant A_I est le terme descriptif irréductible de A :

$$A_I = \mathcal{A}_I(A)$$

Définition de l'algorithme $\mathcal{A}_I$.

```
                |A|:=N
EXPLORATION     I:=1 ; LTD:=N
CONTIGU         si     u_i u_i+1 = PG(R_j)
                alors  Remplacer u_i u_i+1 par PD(R_j) dans A ; [avec tassement
                                                     i+1→i, i+2→i+1,... ]
                       N:=N-1 ;
                       si     I=N
                       alors  aller-à BOUCLE
                       sinon  aller-à CONTIGU
                       fsi
                sinon  I:=I+1 ;
                       si     I=N
                       alors  aller-à BOUCLE
                       sinon  aller-à CONTIGU
                       fsi
                fsi
BOUCLE          si     LTD=N
                alors  A_I:=A ; FIN
                sinon  aller-à EXPLORATION
                fsi
```

Théorème 3. Etant donné un terme descriptif quelconque A, A admet un seul terme descriptif irréductible qui a pour expression $\mathcal{A}_I(A)$.

démonstration. Pour prouver cette proposition, il suffit de vérifier que l'algorithme $\mathcal{A}_I$ ne boucle pas quand on lui soumet un terme descriptif quelconque A. A chaque pas de l'exploration, l'algorithme engendre un terme descriptif contigu à celui soumis à l'entrée :

$$A \longrightarrow A_2 \ldots \quad \ldots A_i \longrightarrow A_{i+1} \ldots \quad \ldots A_{n-1} \longrightarrow A_I (= \mathcal{A}_I(A)) \qquad (1)$$

Il en résulte que $\forall i \ |A_i| > |A_{i+1}|$ car A_{i+1} est contigu à A_i ; par suite $\forall ij (i \neq j)\ A_i \neq A_j$. L'algorithme n'engendre donc jamais un terme descriptif qui a déjà été généré. Il ne peut donc pas boucler ; il s'arrête quand aucune règle $\xrightarrow{\circ}$ ne peut être appliquée sur le terme descriptif intermédiaire qui lui est soumis. Ce terme A_I est donc le terme irréductible et l'on a d'après (1) $A \xrightarrow{*} \mathcal{T}_U(A)$ (c.q.f.d.).

Les performances de cet algorithme sont en première approximation proportionnelles à la longueur cumulée Lmaxi de tous les termes descriptifs engendrés. Supposons que l'algorithme s'arrête à la $p^{\text{ième}}$ exploration en engendrant le terme A_I de longueur $n-p>1$. On a donc :

$$\text{Lmaxi} = \sum_{1}^{p}(n-i) < \sum_{1}^{n-1}(n-i)$$

$$\text{Lmaxi} \leqslant \frac{n(n-1)}{2}$$

Pour n grand, l'exploration réalisée par cet algorithme est proportionnelle à $\frac{1}{2}n^2$.

b) Fonction de discordance D(A,B) de deux termes descriptifs A et B.

Nous appelons fonction de discordance de deux termes descriptifs A et B, tels que $|A| \geqslant |B|$, la procédure qui calcule les deux termes descriptifs A_F et B_D comme suit :

Soit U le plus grand terme descriptif, éventuellement égal à Λ, tel que les deux égalités $A=UA_1$ et $B=UB_1$ soient simultanément vérifiées ; alors par définition $A_F=A_1$ et $B_D=U\text{Pr}(B_1)$ (Rappelons que les fonctions Pr(X) et Dr(X) ont pour valeur les termes descriptifs représentés respectivement par le premier élément et le dernier élément du terme descriptif X ; voir § III 2.1., chapitre quatrième).

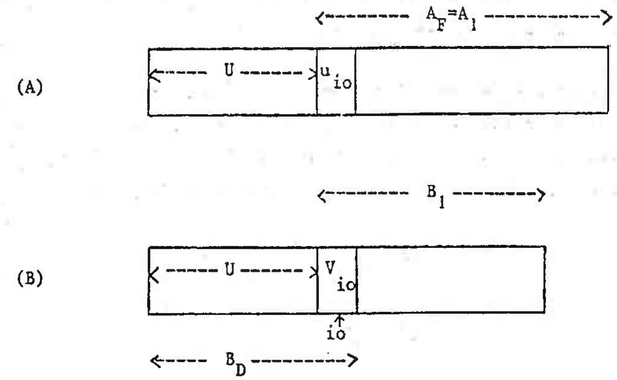
La notation de la fonction de discordance est : $\langle A_F, B_D\rangle = D(A,B)$.

Supposons que A et B soient de la forme :

$$\begin{cases} A=u_1u_2\ldots u_n \\ B=v_1v_2\ldots v_r \end{cases} \qquad \text{avec} \quad |A|=n \text{ et } |B|=r$$

La fonction D(A,B) montre que si i est la position du $i^{\text{ième}}$ terme descriptif non décrit dans B, alors il existe une valeur i_o de i telle que

$$\text{et} \quad \begin{cases} \forall i < i_o & u_i=v_i \\ i=i_o & u_{io} \neq v_{io} \end{cases}$$



Cette fonction a trois groupes de valeurs particulières :

Premier cas. $\langle A_F, B\rangle = D(BA_F, B)$

C'est la *valeur limite* de D qui correspond au cas où $B_1=\Lambda$; d'où $B=B_D=U$ et $A=BA_F$; avec comme cas particulier $\langle A, \Lambda\rangle = D(A,\Lambda)$.

Deuxième cas. $\langle \Lambda, A\rangle = D(A,A)$

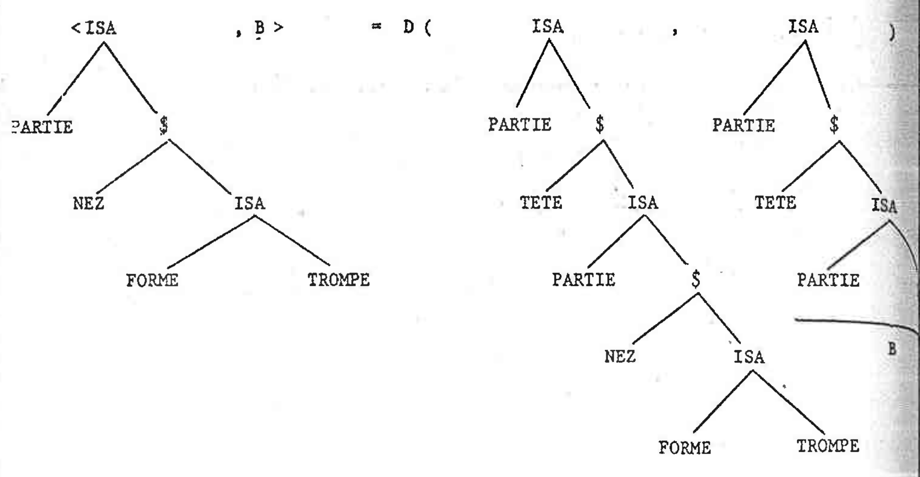
C'est la *discordance nulle* ; elle correspond au cas où $A_1=\Lambda$, d'où $A=U$ et $B=UB_1$. Or $|A| \geqslant |B|$, donc $B_1=\Lambda$; il en résulte que $A=B=B_D$ et $A_F=\Lambda$. (cas particulier : $D(\Lambda, \Lambda)$).

Troisième cas. $\langle A, \text{Pr}(B)\rangle = D(A,B)$

C'est la *discordance totale* qui correspond au cas où $U=\Lambda$; d'où $A=A_1$ et $B=B_1$; il en résulte que $A_F=A$ et $B_D=\text{Pr}(B)$.

Exemple

La fonction ci-après fournit la discordance des deux termes descriptifs A et B véhiculant respectivement les énoncés : "Tête possédant un nez en forme de trompe" et "Tête possédant des yeux".



c) Procédure de décision $\mathcal{A}_R$ pour la relation $\longrightarrow^*$.

Cet algorithme détermine la discordance $\langle A_F,B_D\rangle = D(A,B)$ des deux termes A et B. Si cette fonction n'est pas définie alors il y a échec. Par contre si $A = BA_F$ (valeur limite de D, premier cas) alors la formule $A \longrightarrow^* B$ est démontrée par application de la règle de décomposition ; de même si $A_F = \Lambda$ (discordance nulle, deuxième cas) cette formule est démontrée par reflexivité. L'algorithme se termine alors à l'étape SUCCES (étape 6). Si ces conditions ne sont pas vérifiées, $\mathcal{A}_R$ cherche dans A_F un sous-terme descriptif φ qui a pour terme descriptif irréductible $Dr(B_D)$. Si cette opération est impossible, alors il y a échec. Dans le cas contraire, on donne à A et B les valeurs A' (telle que $A_F = \varphi A'$) et $B-B_D$, et on recommence le processus. Cet algorithme s'appuie sur le lemme démontré dans le paragraphe I.1. en ce sens que la règle de décomposition est toujours appliquée la dernière.

Définition de l'algorithme $\mathcal{A}_R$.

```
DEBUT      si      D(A,B) n'est pas définie   (i.e. |A|<|B|)
           alors   ECHEC ; FIN
           sinon   <A_F,B_D>:= D(A,B)
           fsi
           si      A=BA_F ou A_F=Λ
ETAPE 6    alors   SUCCES ; FIN
           sinon   si φ ─▶* Dr(B_D) tel que A_F:=φA'    [A_I(φ):=Dr(B_D)]
                   alors  aller-à  CONTINUE
                   sinon  ECHEC ; FIN
                   fsi
           fsi
CONTINUE   A:=A'
           B:=B-B_D
           si      B=Λ
           alors   SUCCES ; FIN
           sinon   aller-à DEBUT
           fsi
```

Théorème 4. Si la formule $A \longrightarrow^* B$ est vérifiée, alors $\mathcal{A}_R(A,B) = \text{SUCCES}$.

démonstration. Examinons tout d'abord le cas élémentaire où l'on sort de $\mathcal{A}_R$ immédiatement après l'étape 6. Si la formule $A \longrightarrow^* B$ est vraie, alors l'une des deux possibilités suivantes est effectivement vérifiée : soit par application de la règle de décomposition (voir § III.2.2.a, chapitre quatrième) on obtient B à partir de $A=BA_F$ (valeur limite de D) ; soit $A=B$ ce qui correspond à la discordance nulle ($A_F=\Lambda$) (c.q.f.d.). Dans le cas général, si la formule $A \longrightarrow^* B$ est vérifiée, alors $\mathcal{A}_R$ doit engendrer une suite de termes descriptifs $A_1, A_2, \ldots A_{p-1}, A_p$ telle que :

$$A \longrightarrow^* A_1 \longrightarrow^* A_2 \ldots \quad \ldots A_i \longrightarrow^* A_{i+1} \ldots \longrightarrow^* A_{p-1} \longrightarrow^* A_p \longrightarrow B$$

A la première itération $\mathcal{A}_R$ engendre B_1 sous-terme de B et A'_1 tels que $A \longrightarrow^* B_1 A'_1$ car A est de la forme $[B_1 - Dr(B_1)]\varphi_1 A'_1$ et $\varphi_1 \longrightarrow^* Dr(B_1)$. A la seconde itération $\mathcal{A}_R$ engendre à partir de A'_1 et de $B-B_1$ les termes B_2 et A'_2 tels que $B_1A'_1 \xrightarrow{*} B_1B_2A'_2$ car A'_1 est de la forme $[B_2-Dr(B_2)]\varphi_2 A'_2$ et $\varphi_2 \xrightarrow{*} Dr(B_2)$; notons que B_1B_2 est un sous terme descriptif de B. Par récurrence, on a à la $(p-1)^{ième}$ itération $B_1B_2 \ldots B_{p-2}\ A'_{p-2} \xrightarrow{*} B_1B_2 \ldots B_{p-2}\ B_{p-1}A'_{p-1}$. Supposons que la $p^{ième}$ itération corresponde à la sortie SUCCES de $\mathcal{A}_R$: $\mathcal{A}_R$ génère à partir de $(B-B_1 \ldots -B_{p-1})$ et de A'_{p-1} les termes B_p et A'_p tels que :

$$B_1B_2 \ldots B_{p-1}\ A'_{p-1} \xrightarrow{*} B_1B_2 \ldots B_{p-1}\ \ B_p\ A'_p$$

car A'_{p-1} est de la forme $[B_p-Dr(B_p)]\varphi_p A'_p$ et $\varphi_p \xrightarrow{*} Dr\ (B_p)$. Comme on sort de l'algorithme on a : $B-B_1-B_2-\ldots-B_{p-1}-B_p=\Lambda$, d'où $B_1B_2 \ldots B_{p-1}B_p=B$. Il en résulte que $B_1B_2 \ldots B_{p-1}A'_{p-1} \xrightarrow{*} BA'_p$; et par application de la règle de décomposition $BA'_p \longrightarrow B$ (voir lemme du § I.1).

$\mathcal{A}_R$ engendre donc bien une suite de termes descriptifs telle que :

$$A \xrightarrow{*} B_1A'_1 \xrightarrow{*} B_1B_2\ A'_2 \ \ldots\ldots \quad \ldots \quad \ldots\ B_1B_2 \ldots B_{p-1}\ A'_{p-1} \xrightarrow{*} BA'_p$$

Montrons que cette suite est finie, c'est à dire que l'algorithme ne boucle pas. Pour cela il suffit de prouver qu'il existe une $p^{ième}$ itération (p fini) au cours de laquelle $B-B_1-\ldots-B_{p-1}-B_p=\Lambda$. Au cours de la première itération on a $|B-B_1|<|B|$ car B_1 est un sous terme de B ; à la deuxième itération on $|B-B_1-B_2|<|B-B_1|$ car B_2 est un sous terme de $B-B_1$, et ainsi de suite. On a donc

$$|B-B_1|=|B|-k_1 \qquad (k_1 \geq 1)$$

$$|B-B_1-B_2|=|B-B_1|-k_2 \qquad (k_2 \geq 1)$$

$$|B-B_1-\ldots-B_{i-1}|=|B-B_1\ldots-B_{i-2}|-k_{i-1} \qquad (k_{i-1} \geq 1)$$

$$|B-B_1-\ldots-B_i|=|B-B_1-\ldots B_{i-1}|-k_i \qquad (k_i \geq 1)$$

$$|B-B_1-\ldots-B_i| = |B|-\sum_{1}^{i} k_j$$

La longueur du terme descriptif $(B-B_1-\ldots-B_i)$ s'annule à la $p^{ième}$ itération, p étant défini par :

$$|B|= \sum_{1}^{p} k_j \geq p \qquad \text{car } \forall i\ k_i \geq 1$$

L'algorithme $\mathcal{A}_R$ ne boucle pas, le nombre d'itérations étant borné par la longueur du terme descriptif B. La suite définie par (1) est donc finie et par transitivité $A \xrightarrow{*} B$ (c.q.f.d.).

Supposons, en première approximation, que les performances de $\mathcal{A}_R$ soient proportionnelles à la longueur cumulée Lmaxr de tous les termes descriptifs engendrés. A chaque itération l'algorithme $\mathcal{A}_I$ appliqué à la chaîne φ_i de longueur n_i génère des termes descriptifs dont la longueur cumulée est $\leq \frac{1}{2} n_i(n_i-1)$ (voir théorème 3, algorithme $\mathcal{A}_I$ de recherche du terme descriptif irréductible) ; on a donc :

$$\text{Lmaxr} \leq \sum_{1}^{p} \frac{1}{2} n_i(n_i-1) \leq \frac{1}{2} \sum_{1}^{r} n_i(n_i-1) \quad \text{car } p \leq r=|B|$$

La longueur maximum de n_i correspond au cas où il y a un seul élément de discordance entre A et B ; d'où : $n_i \leq (n-r+1)$ avec $n=|A|$

Il en résulte que :

$$\text{Lmaxr} \leq \frac{1}{2}\, r(n-r)(n-r+1)$$

Les performances de l'algorithme $\mathcal{A}_R$ sont meilleures que celles de $\mathcal{A}_C$. Pour r donné, on montre facilement que Lmaxc est un infiniment grand d'ordre supérieur à Lmaxr, ce qui confirme le résultat souhaité au début de ce paragraphe.

II. ETUDE DES PROPRIETES DE LA RELATION $\Longrightarrow$.

II.1. Preuve des règles relatives aux connecteurs.

Dans ce paragraphe nous démontrons les propositions qui prouvent la validité des règles d'insertion et d'élimination des connecteurs (voir § IV.2. , chapitre quatrième).

Proposition 1. Si la relation $a \Longrightarrow b$ est vérifiée, alors $\forall c \in \Delta$ la relation $a*c \Longrightarrow b$ est aussi vérifiée (preuve de R1).

démonstration. Si $a \Longrightarrow b$ alors $a*b=a$ (théorème 1, § IV.3.1., chapitre quatrième) ; il en résulte que $\forall c \in \Delta$ $a*b*c=a*c$ (condition C2), soit $(a*c)*b=a*c$ (d'après C5,C6). On en déduit que $a*c \Longrightarrow b$ est vérifiée d'après le même théorème 1 (c.q.f.d.).

Proposition 2. Si les relations $a \Longrightarrow b$ et $a \Longrightarrow c$ sont vérifiées, alors la relation $a \Longrightarrow b*c$ est aussi vérifiée et réciproquement (preuve de R2 et R3).

démonstration. Si $a \Longrightarrow b$ et $a \Longrightarrow c$ alors $a*b=a$ et $a*c=c$, d'où $(a*b)*(a*c)=a*a$; soit après réduction $a*(b*c)=a$. On en déduit que la relation $a \Longrightarrow b*c$ est vérifiée. Réciproquement si $a \Longrightarrow b*c$, alors $b*c*a=a$. Or $a*b=a*(b+b*c)=a*b+a*b*c$; après substitution il vient $a*b=a*b+a=a$, la relation $a \Longrightarrow b$ est donc vérifiée. De même $a*c=a*(c+b*c)=a*c+a*b*c=a*c+a$ soit $a*c=a$; d'où $a \Longrightarrow c$ est vérifiée (c.q.f.d.).

Proposition 3. Si la relation $a \Longrightarrow b$ est vérifiée, alors $\forall c \in \Delta$ la relation $a \Longrightarrow b+c$ est aussi vérifiée (preuve de R4).

(démonstration analogue à la proposition 1.)

Proposition 4. Si les relations $a \Longrightarrow b$ et $c \Longrightarrow b$ sont vérifiées alors la relation $a+c \Longrightarrow b$ est également vérifiée, et réciproquement (preuve de R5 et R6).

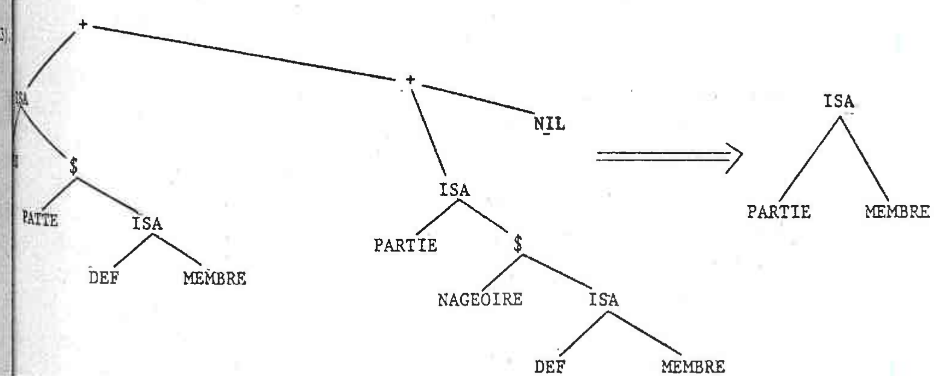
démonstration. Si $a \Longrightarrow b$ et $c \Longrightarrow b$, alors $a+b=b$ et $c+b=b$ (remarque du théorème 1, § IV.3.1. chapitre quatrième) ; d'où $a+c+b=b$. On en déduit que $a+c \Longrightarrow b$. Réciproquement si $a+c \Longrightarrow b$, alors $a+c+b=b$. Il en résulte $a*b=a*(a+c+b)=a$, d'où $a \Longrightarrow b$. De même $c*b=c*(a+c+b)=c$, d'où $c \Longrightarrow b$ (c.q.f.d.).

Cette proposition est essentielle car elle permet de transformer, moyennant la règle de réécriture $\Longrightarrow$, un énoncé disjonctif en un énoncé quelconque. Illustrons ce mode opératoire à travers l'exemple schématisé par la figure 4.8. (voir §II.4.3., chapitre quatrième). Si le couple de classes (PARTIE,DEF) possède la propriété d'extension (voir § III.2,2.c, chapitre quatrième, figure 4.11), alors les deux relations suivantes sont vérifiées :

ISA(PARTIE,\$(PATTE,ISA(DEF,MEMBRE))) $\Longrightarrow$ ISA(PARTIE, MEMBRE)

ISA(PARTIE,\$(NAGEOIRE,ISA(DEF,MEMBRE))) $\Longrightarrow$ ISA(PARTIE, MEMBRE)

La proposition 4 permet alors d'affirmer la relation ci-après qui illustre un exemple d'élimination du connecteur + :



II.2. Propositions remarquables.

Proposition 5. Si la relation $a \Longrightarrow b$ est vérifiée, alors $\forall a', b' \in \Delta$ la relation $a*a' \Longrightarrow b+b'$ est aussi vérifiée.

(démonstration immédiate à partir des propositions 1 et 3).

Proposition 6. Si la relation $a \Longrightarrow b$ est vérifiée, alors $a*\neg b=\Lambda$

démonstration. Si $a \Longrightarrow b$ alors $a*b=a$, d'où $a*b*\neg b=a*\neg b$. Il en résulte que $(a*\neg b)*(b*\neg b)=a*\neg b$, soit $(a*\neg b)*\Lambda=a*\neg b$. On en déduit que $a*\neg b \Longrightarrow \Lambda$, soit $a*\neg b=\Lambda$ d'après la proposition 11 (voir § IV.3.1.c.., chapitre quatrième) (c.q.f.d.).

Proposition 7. Si $a*b=\Lambda$, alors les relations $a \Longrightarrow \neg b$ et $b \Longrightarrow \neg a$ sont vérifiées.

démonstration. De $a*b=\Lambda$ et $b*\neg b=\Lambda$, on déduit que $(a*\neg b)*b=a*b$. Cette égalité est au moins vérifiée pour $a*\neg b=a$; il en résulte que $a \Longrightarrow \neg b$. En partant de $a*b=\Lambda$ et $a*\neg a=\Lambda$, on montrerait de même que $b \Longrightarrow \neg a$ est vérifiée (c.q.f.d.).

Cette proposition a une conséquence fondamentale en ce sens qu'elle fixe les rapports entre toute description et la négation de sa négation.

Proposition 8. La relation $a \Longrightarrow \neg\neg a$ est toujours vérifiée.

démonstration. La proposition 7 permet d'affirmer que les relations $\neg a \Longrightarrow \neg a$ et $a \Longrightarrow \neg\neg a$ sont toujours vérifiées car $a*\neg a=\Lambda$ (c.q.f.d.).

La première relation $\neg a \Longrightarrow \neg a$ exprime la réflexivité de la relation $\Longrightarrow$; elle n'est donc pas liée à la définition de la négation. La seconde relation $a \Longrightarrow \neg\neg a$ par contre dépend étroitement de la définition de la négation : la condition C11 introduit une négation du type intuitionniste.

Proposition 9 . La relation $\neg\neg a \Longrightarrow a$ est toujours vérifiée.

démonstration.

	$\Lambda \Longrightarrow a$	(proposition 11, voir § IV.3.1.c., chapitre quatrième)
	$\neg\neg a*\neg a \Longrightarrow a$	(car $\neg\neg a*\neg a=\Lambda$)
	$\neg\neg a \Longrightarrow a$	(condition C3, § IV.1.1., chapitre quatrième) (c.q.f.d.)

Les propositions 8 et 9 permettent d'énoncer, par définition de la relation $=$, la proposition suivante.

Proposition 10. $a=\neg\neg a$

Cette proposition montre que le quotient $\Delta/=$ est bien un treillis distributif complémenté (voir § IV.1., chapitre quatrième). Les propositions que nous démontrons ci-après sont classiques dans cette théorie ; en particulier le quotient $\Delta/=$ possède un majorant et un seul V qui représente la description universelle (proposition 13).

Proposition 11. L'égalité $(a+\neg a)=\neg\Lambda$ est toujours vérifiée $\forall a\in\Delta$.

démonstration.

	$a \Longrightarrow a$	(reflexivité)
	$b*a \Longrightarrow a+\neg a$	(proposition 5, avec $b\in\Delta-\{\Lambda\}$)
	$b*a*\neg(a+\neg a)=\Lambda$	(proposition 6)
	$b*\neg(a+\neg a) \Longrightarrow \neg a$	(proposition 7)
	$b*\neg(a+\neg a) \Longrightarrow \neg a+a$	(proposition 3)
	$b \Longrightarrow \neg a+a$	(condition C3)
	$b*\neg(\neg a+a)=\Lambda$	(proposition 6)
	$\neg(a+\neg a)=\Lambda$	(car $b\in\Delta-\{\Lambda\}$)
	$(a+\neg a)=\neg\Lambda$	(proposition 10) (c.q.f.d.).

Proposition 12. Toute description a a une et une seule négation, notée $\neg a$.

Pour démontrer cette proposition nous allons tout d'abord démontrer le lemme suivant :

Lemme. Si les expressions $a*b=a*c$ et $a+b=a+c$ sont vérifiées, alors $b=c$.

démonstration du lemme.

	$b=b+a*b$	(condition C9)
	$b=b+a*c$	(car $a*b=a*c$)
	$b=(b+a)*(b+c)$	(condition C10)
	$b=(a+c)*(b+c)$	(car $a+b=a+c$)
	$b=a*(b+c)+c*(b+c)$	(condition C10)
	$b=a*(b+c)+c$	(condition C9)
	$b=a*b+a*c+c$	(condition C10)
	$b=a*c+a*c+c$	(car $a*b=a*c$)
	$b=a*c+c$	(condition C7)
	$b=c$	(condition C9) (c.q.f.d.)

Remarque

L'équation $a*x=a*b$ admet au moins une solution pour $x=b$; d'après la proposition précédente cette solution est unique quand l'égalité $a+x=a+b$ est aussi vérifiée.

démonstration de la proposition 12. Supposons qu'il existe deux négations a' et a" ; il en résulte que :

$$\begin{cases} a*a'=\Lambda \\ a+a'=\neg\Lambda \end{cases} \quad \text{et} \quad \begin{cases} a*a''=\Lambda \\ a+a''=\neg\Lambda \end{cases}$$

d'où $a*a'=a*a''$ et $a+a'=a+a''$. Le lemme précédent permet de déduire l'égalité $a'=a''=a$ (c.q.f.d.).

Proposition 13. Le quotient $\Delta/=$ a un majorant et un seul V tel que $V=\neg\Lambda$; ce majorant sera appelé *description universelle.*

démonstration. de l'égalité $\Lambda*a=\Lambda$, on tire par application de la proposition 7 la relation $a \Longrightarrow \neg\Lambda$. Posons $V=\neg\Lambda$ car $\neg\Lambda$ est unique d'après la proposition précédente ; on a donc $\forall a\in\Delta \quad a \Longrightarrow V$. Comme $\Longrightarrow$ est une relation d'ordre, V est donc l'unique majorant de $\Delta/=$ (c.q.f.d.)

Proposition 14. Toute description a possède une seule négation $\neg a$ telle que :

$$\begin{cases} a*\neg a=\Lambda \\ a+\neg a=V \end{cases}$$

La démonstration est évidente.

Nous avons donc muni l'ensemble Δ d'un connecteur de négation dont les liens sémantiques qu'il entretient avec la relation $\Longrightarrow$ permettent d'obtenir le principe du tiers-exclu.
En particulier, on démontre très facilement à partir de la proposition précédente, les deux corollaires ci-après.

Corollaire 1. $\forall a,b\in\Delta$, on a toujours :

$$\begin{cases} \neg(a*b)=\neg a+\neg b \\ \neg(a+b)=\neg a*\neg b \end{cases}$$

Corollaire 2. Si la relation $a \Longrightarrow b$ est vérifiée, alors la relation $\neg b \Longrightarrow \neg a$ est également vérifiée, et réciproquement.

II.3. Interprétation de la négation.

II.3.1. Analyse du problème.

La négation est un connecteur unaire qui, appliqué à un terme descriptif, fournit une nouvelle description comme l'indique la figure 5.2. Comment interpréter cette description par rapport aux différents éléments qui composent le terme descriptif : opérateur, classe, trait et description de traits . Le sens du terme descriptif nié dépend de la portée de la négation. En d'autres termes, quel rôle logico-sémantique joue cette dernière par rapport aux éléments qui composent le terme descriptif ? La nature des liaisons, entre ce connecteur et les éléments sur lesquels il porte, est analysée à travers quelques exemples simples mais significatifs. Pour ce faire un foncteur à une place d'argument, noté NEG, permet d'exprimer explicitement les différentes interprétations de la négation : NEG(x) est une fonction qui définit l'interprétation de la négation quand sa portée intéresse l'entité x.
Cette analyse nous conduit à examiner quatre cas distincts.

Premier cas. La négation influence le trait dans un terme descriptif non décrit.
L'interprétation dans ARCHES de l'énoncé "La ROBE n'est pas ROUGE" est déterminée par l'énoncé "La ROBE a une couleur y telle que y n'est pas ROUGE". Elle exprime que la portée de la négation n'est pas la classe ou l'opérateur (respectivement COULEUR ou ISA) mais la valeur du trait, soit ROUGE. La figure 5.3. montre la représentation de l'énoncé précédent dans laquelle apparait explicitement la portée de la négation : NEG(ROUGE) définit le sens du terme descriptif nié ; c'est la valeur de la fonction NEG(t) pour t=ROUGE. Elle dépend de la classe T à laquelle appartient t ainsi que de ses rapports avec les autres classes. L'hypothèse d'un monde descriptif clos (voir § II.3.2.) permet d'attribuer à NEG(ROUGE) la valeur suivante :

NEG(ROUGE)=BLEU.OU.MARRON.OU.VERT... avec BLEU $\in$ COULEUR,...

L'énoncé précédent est ainsi interprété par la description ci-après :

$$\underset{i}{+}[ISA(COULEUR,t_i)] \quad \text{avec} \quad \forall i \begin{cases} t_i \in COULEUR \\ t_i \neq ROUGE \end{cases}$$

De la même manière, l'énoncé "L'AMPHORE n'est pas localisée DANS le four F1" conduit dans le cadre de cette interprétation à la structure de données schématisée par la figure 5.4. Dans ce cas la fonction NEG(F1) a par exemple pour valeur :

$$NEG(F1)=F2.OU.F3.OU.F4... \text{ avec } \forall i>1 \; F_i \in FOUR.$$

<u>Deuxième cas</u>. La négation influence la valeur de l'opérateur dans un terme descriptif non décrit.

Pour analyser ce deuxième cas, reprenons l'exemple "L'AMPHORE n'est pas localisée DANS le four F1". Une autre interprétation dans ARCHES de cet énoncé est définie comme suit :

"L'AMPHORE est localisée <OPERATEUR> y tel que y est le four F1"
<OPERATEUR> ≠ DANS.

Par exemple, il pourrait signifier que "L'AMPHORE est localisée A L'EXTERIEUR du four F1". Dans ce cas, il est clair que la négation affecte l'opérateur DANS auquel est appliqué la fonction NEG (figure 5.5.).

NEG(DANS)=A-L'EXTERIEUR.OU.A-COTE.OU.AU-DESSUS...

<u>Troisième cas</u>. La négation influence la caractérisation du trait dans un terme descriptif décrit.

Partons de l'exemple "La ROBE n'est pas rouge CLAIR". Nous faisons l'hypothèse que cet énoncé signifie que "la ROBE est ROUGE non CLAIR". En d'autres termes, nous l'interprétons en appliquant le foncteur NEG à la caractérisation du trait ROUGE, soit la nuance que le ROUGE a ou n'a pas (figure 5.6.).

D'une manière plus générale, nous faisons l'hypothèse suivante :

La portée de la négation d'un terme descriptif nié X intéresse l'élément Dr(X) (voir § III.2.1., chapitre quatrième). Ainsi dans l'énoncé "La ROBE n'est pas ROUGE TRES FONCE" le foncteur NEG intéresse la modalité TRES ; cet énoncé pourrait par exemple signifier que "La ROBE est ROUGE PEU FONCE".

L'analyse de ces trois cas détermine les options qui ont été faites dans ARCHES pour interpréter la négation des termes descriptifs quand sa portée n'intéresse pas la classe (voir quatrième cas). Quand cette condition est

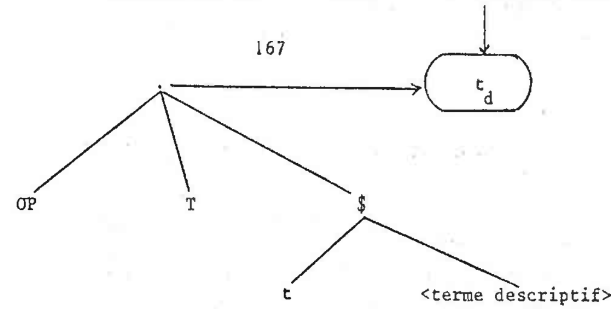


figure 5.2. - <u>*Terme descriptif nié.*</u>

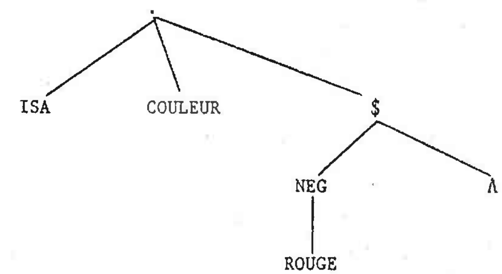


figure 5.3. <u>*Représentation de l'énoncé "- n'est pas rouge"*</u>.

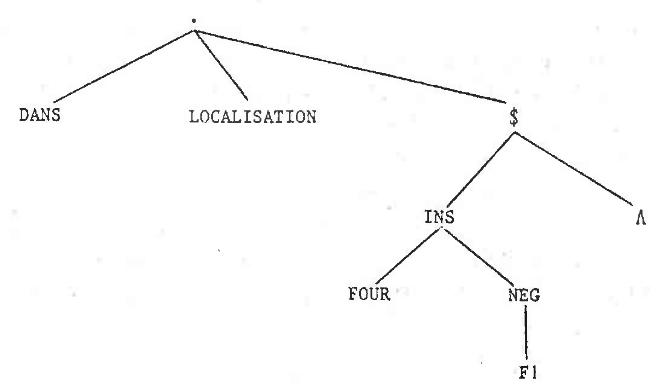


figure 5.4. - <u>*Une interprétation de l'énoncé "- n'est pas localisé dans le four F1"*</u>.

réalisée, nous faisons l'hypothèse que la négation de tout terme descriptif est interprétée comme une disjonction des possibilités offertes par l'analyse des trois cas précédents. Plus précisément, si on désigne par $I(\neg X)$ l'interprétation de la négation du terme descriptif X, ces choix sont définis de la manière suivante :

$$I(\neg\cdot(OP,T,\$(t,\Lambda))) = \cdot(NEG(OP),T,\$(t,\Lambda)) \mid \cdot(OP,T,\$(NEG(t),\Lambda))$$

$$I(\neg X) = [X-Dr(X)]\, I(\neg Dr(X))$$

La définition de la fonction NEG est déterminée logiquement par l'axiomatisation de l'interprétation de la négation relevant de l'étude de ces trois cas (voir § II.3.2.).

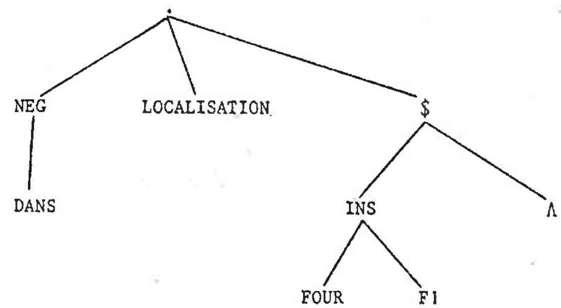


figure 5.5. Une autre interprétation de l'énoncé "– n'est pas localisé dans le four F1".

Quatrième cas. La négation influence la classe.

Pour illustrer cette dernière possibilité, prenons l'exemple "L'AMPHORE n'a pas d'ANSE". Cet énoncé signifie que "L'AMPHORE n'a pas la PARTIE y tel que y est une ANSE". Il ne peut naturellement pas signifier "L'AMPHORE a la partie y tel que y n'est pas une ANSE". Dans ce cas, la négation modifie clairement le sens de la classe PARTIE ; le foncteur NEG est donc appliqué à cette classe (figure 5.7.) : NEG(PARTIE) a pour valeur une classe qui exprime le "contraire" de PARTIE, soit par exemple la classe DESTRUCTION.

Nous faisons l'hypothèse que ce *cas est exclusif des trois précédents* : si la partie de la négation intéresse la classe, alors elle n'intéresse

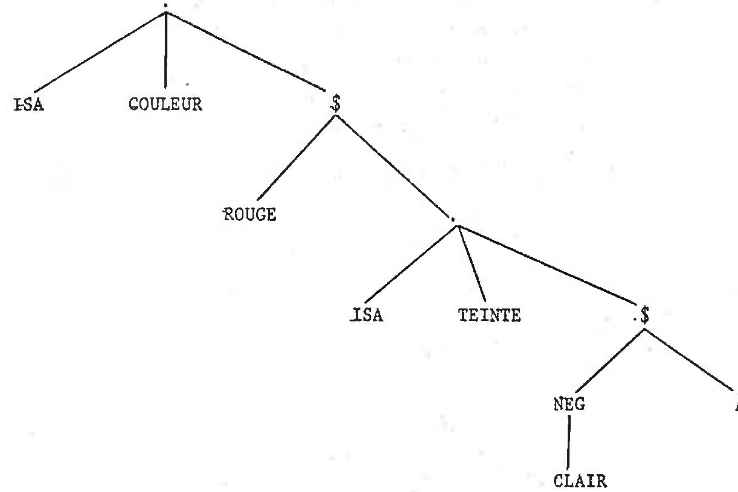


figure 5.6. – Représentation de l'énoncé "– n'est pas ROUGE CLAIR".

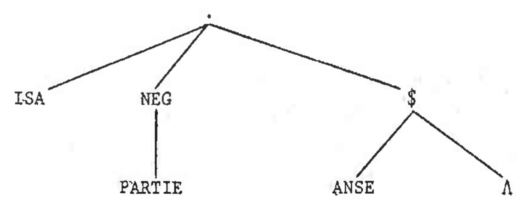


figure 5.7. – Représentation de l'énoncé "– n'a pas d'anse".

aucun autre composant du terme descriptif. Nous admettons ainsi qu'à toute classe T peut être associée une classe T' qui exprime son contraire. La classe T est dans la portée de la négation si et seulement si il existe une classe T' telle que :

$$NEG(T)=T' \quad \text{et} \quad NEG(T')=T$$

Dans ce cas l'interprétation de la négation est définie axiomatiquement par la relation :

$$\neg\cdot(OP,T,\$(t,\Lambda))=\cdot(OP,T',\$(t,\Lambda))$$

Cette interprétation présuppose la donnée du couple de classes (T,T').

II.3.2. Axiomatisation de l'interprétation.

Soit T une classe donnée n'ayant pas de contraire, et F l'ensemble fini des classes telles que pour tout T_i appartenant à F le couple de classes (T,Ti) possède la propriété d'extension.

On dit que deux termes descriptifs non décrits X_1 et X_2 sont reliés par la relation $\mathcal{C}$ si et seulement si ils sont construits à partir de la même classe T n'ayant pas de contraire, les traits correspondants t_1 et t_2 appartenant à l'ensemble $T-\bigcup_i(T\cap T_i)$, tel que $\forall i\ T_i \in F$.

$$X_1\mathcal{C}X_2 \Leftrightarrow \left|\begin{array}{l} X_1=OP1(T,t_1) \\ X_2=OP2(T,t_2) \\ t_1,t_2\in T-\bigcup_i(T\cap T_i),\forall i\ T_i\in F \end{array}\right.$$

Si la classe T n'est reliée à aucune classe par la propriété d'extension (i.e. $F=\phi$) alors $\bigcup_i(T\cap T_i)=\phi$.

Exemples.

ISA(COULEUR,ROUGE)$\mathcal{C}$ISA(COULEUR,BLEU)
IN(LOCALISATION,INS(FOUR,F1))$\mathcal{C}$IN(LOCALISATION,INS(FOUR,F2))
IN(LOCALISATION,INS(FOUR,F1))$\mathcal{C}$ABOVE(LOCALISATION,INS(FOUR,F3))
ISA(PARTIE,PATTE)$\mathcal{C}$ISA(PARTIE,NAGEOIRE)
ISA(PARTIE,PATTE)$\not\mathcal{C}$ISA(PARTIE,MEMBRE) car le couple de classes (PARTIE,DEF) possède la propriété d'extension et que MEMBRE $\in$ DEF

On montre très facilement que la relation $\mathcal{C}$ est reflexive, transitive et symétrique : *la relation $\mathcal{C}$ est donc une relation d'équivalence.*

Désignons par $\mathcal{C}_\Delta$ l'ensemble des termes descriptifs non décrits inclus dans

la base $\mathcal{L}$ de Δ (voir § II.4.2., chapitre quatrième). La relation $\mathcal{C}$ définit sur $\mathcal{C}_\Delta$ une partition de classes d'équivalence. Soit C_i l'une d'entre elles : C_i contient tous les termes descriptifs non décrits construits sur la même classe ; soit X_{ij} un élément quelconque de C_i. Nous supposons que $\forall i$ la condition suivante est vérifiée :

$$C12 \qquad X_{i1}+X_{i2}+\ldots+X_{ij}+\ldots+X_{in}=V \qquad \text{avec } n=|C_i|$$

On déduit de cette propriété la proposition suivante :

Proposition 15. $\neg X_{i1}*\neg X_{i2}*\ldots*\neg X_{ij}*\ldots*\neg X_{in}=\Lambda$

Cette condition (et la proposition 15 qui en découle) suppose que pour un état donné de toute application l'information descriptive est complète, c'est à dire que l'on se donne un *système de description clos*. Toute description formée par la disjonction de tous les termes descriptifs appartenant à une même classe d'équivalence est un majorant pour l'ensemble Δ (on trouve une hypothèse de nature comparable dans {43}). En d'autres termes, cette propriété exprime que tout élément de description caractérisant un individu quelconque ne peut pas être défini par la disjonction de tous les termes descriptifs construits sur la même classe. Ceci est une hypothèse tout à fait justifiable car de tels éléments de description n'apportent aucune valeur descriptive, en ce sens qu'ils représentent la description universelle.

On démontre facilement à partir de la condition C12 les deux égalités suivantes :

$$\left|\begin{array}{ll} \neg X_{ij}*(\overset{n-1}{\underset{j\neq k}{\overset{+}{1}}}\ X_{ik})=\neg X_{ij} & (1) \\ \neg X_{ij}\ \overset{n-1}{\underset{j\neq k}{\overset{+}{1}}}\ X_{ik}=\overset{n-1}{\underset{j\neq k}{\overset{+}{1}}}\ X_{ik} & (2) \end{array}\right.$$

Il en résulte les deux propositions suivantes :

Proposition 16. Tout terme descriptif nié est relié à la disjonction de tous les autres termes descriptifs appartenant à la même classe d'équivalence par la relation $\Rightarrow$.

$$\neg X_{ij} \Rightarrow (\overset{n-1}{\underset{j\neq k}{\overset{+}{1}}}\ X_{ik})$$

Proposition 17. La description, formée par l'addition des (n-1) termes descriptifs niés appartenant à la même classe d'équivalence de cardinal n, est reliée au $n^{ième}$ terme descriptif par la relation $\Longrightarrow$.

$$\overset{n-1}{\underset{j\neq k}{\ast}} (\neg X_{ik}) \Longrightarrow X_{ij}$$

Ces deux propositions, qui expriment l'axiomatisation de l'interprétation de la négation, déterminent formellement la fonction NEG.

Exemple.

Supposons que la classe COULEUR fasse référence aux trois traits BLEU, BLANC et ROUGE. La fermeture du système de description conduit à l'expression :

ISA(COULEUR,ROUGE)+ISA(COULEUR,BLEU)+ISA(COULEUR,BLANC)=V

On en déduit les interprétations suivantes :

$$\begin{cases} \neg ISA(COULEUR,ROUGE) \Longrightarrow ISA(COUELUR,BLEU)+ISA(COULEUR,BLANC) \\ \neg ISA(COULEUR,BLANC) \Longrightarrow ISA(COULEUR,BLEU)+ISA(COULEUR,ROUGE) \\ \neg ISA(COULEUR,BLEU) \Longrightarrow ISA(COULEUR,ROUGE)+ISA(COULEUR,BLANC) \end{cases}$$

$$\begin{cases} \neg ISA(COULEUR,ROUGE)*\neg ISA(COULEUR,BLEU) \Longrightarrow ISA(COULEUR,BLANC) \\ \neg ISA(COULEUR,ROUGE)*\neg ISA(COULEUR,BLANC) \Longrightarrow ISA(COULEUR,BLEU) \\ \neg ISA(COULEUR,BLEU)*\neg ISA(COULEUR,BLANC) \Longrightarrow ISA(COULEUR,ROUGE) \end{cases}$$

Dans ce qui suit nous précisons les propriétés de l'interprétation de la négation à partir d'une définition particulière de la notion de contraire. L'expression (1) admet plusieurs solutions pour la valeur de $\neg X_{ij}$, chaque solution définissant une interprétation de la négation (voir § II.2., lemme et remarque associée). Le choix de la solution que nous avons adoptée est fondé sur l'hypothèse suivante : le contraire de tout terme descriptif appartenant à une classe déterminée est défini par la disjonction de tous les autres termes descriptifs issus de la même classe.

$$\neg X_{ij} = \overset{n-1}{\underset{k\neq j}{\overset{+}{I}}} X_{ik}$$

On vérifie facilement que cette valeur de $\neg X_{ij}$ est bien solution de l'équation (1).

On déduit les propriétés suivantes :

$$\left| \begin{array}{ll} X_{ij} = \overset{n-1}{\underset{k\neq j}{\overset{*}{I}}} \neg X_{ik} & \\ X_{ij}+X_{ik}=V & \forall jk \\ X_{ij}*X_{ik}=\Lambda & \forall jk \end{array} \right.$$

Les deux dernières propriétés montrent que toute classe ne peut caractériser un individu quelconque que par l'intermédiaire d'au plus un seul terme descriptif construit sur cette classe (ceci est bien conforme à la notion de contraire que nous avons introduite). Si les classes dont le cardianl est égal à 2 vérifient bien cette propriété (par exemple la classe SEXE ={MALE,FEMELLE}), de nombreuses autres peuvent la vérifier comme par exemple la classe SEMAINE ={LUNDI,MARDI,MERCREDI,JEUDI,VENDREDI,SAMEDI, DIMANCHE} ou celle étudiée ci-dessous.

Exemple.

Supposons que la classe LOC(alisation) donne naissance aux quatre termes descriptifs ci-après :

IN(LOC,INS(FOUR,F1))	OUT(LOC,INS(FOUR,F1))
IN(LOC,INS(FOUR,F2))	OUT(LOC,INS(FOUR,F2))

La fermeture du système de description impose la condition suivante :

$$\overset{2}{\underset{1}{+}}[IN(LOC,INS(FOUR,F_i))+OUT(LOC,INS(FOUR,F_i))]=V$$

La négation de tout terme descriptif est ici interprétée de manière évidente à partir de la notion de contraire. On a par exemple :

¬IN(LOC,INS(FOUR,F1))=IN(LOC,INS(FOUR,F2))+OUT(LOC,INS(FOUR,F1))
+OUT(LOC,INS(FOUR,F2))

OUT(LOC,INS(FOUR,F2))=¬IN(LOC,INS(FOUR,F1))*¬IN(LOC,INS(FOUR,F2))
*¬OUT(LOC,INS(FOUR,F1))

Remarque.

Si cette propriété est vérifiée pour une classe donnée, alors tout individu ne peut être caractérisé que par un et un seul terme descriptif construit sur cette classe. Réciproquement, nous admettons qu'elle caractérise toute classe C_i pour laquelle la condition $X_{ij}+X_{ik}=V$ est vérifiée $\forall ij$. Cependant dans le cas général cette propriété n'est pas attestée pour n'importe quelle classe. Mais nous pouvons toujours nous ramener à ce cas en partitionnant la classe en un ensemble fini de sous-classes telles que pour chaque sous-classe la notion de contraire que nous avons introduite devient acceptable. Le cas limite correspond à une partition ou chaque sous-classe à un cardinal égal à 2 (i.e. chaque sous-classe est formée d'un terme descriptif et de son contraire).

II.4. Consistance de la relation $\Longrightarrow$.

II.4.1. Définition.

On dira que tout ensemble muni d'un opérateur de négation est *cohérent* ou *consistant vis à vis d'une relation* $\mathcal{R}$ si et seulement si il n'est pas possible de relier par la relation $\mathcal{R}$ tout élément a à la fois à un élément b et à sa négation.

II.4.2. Evaluation.

Proposition 18. Tout ensemble de descriptions Δ de base $\mathcal{L}$ muni de la relation $\Longrightarrow$ est consistant.

démonstration. Supposons que la relation $a \Longrightarrow b$ soit vérifiée ($a \in \Delta - \{\Lambda\}$ sinon si $a=\Lambda$ cette relation serait toujours vérifiée, voir § IV.3.1.c., chapitre quatrième). Montrons que la relation $a \Longrightarrow \neg b$ ne peut pas être simultanément vérifiée. Si les deux relations $a \Longrightarrow b$ et $a \Longrightarrow \neg b$ sont toutes les deux vérifiées, nous avons (voir § IV.3.1.a, chapitre quatrième, théorème 1) :

$$a*b=a \qquad \text{et} \qquad a*\neg b=a \qquad (1)$$

On en déduit que : $a*b*a*\neg b=a*a$

$$a*b*\neg b=a \qquad \text{d'où } a=\Lambda$$

ce qui est contraire à l'hypothèse. La relation $a \Longrightarrow \neg b$ n'est donc pas vérifiée en même temps que la relation $a \Longrightarrow b$ (c.q.f.d.). Nous devons remarquer que l'interprétation de

la négation n'introduit pas d'inconsistance. En effet la relation suivante est vérifiée d'après la proposition 16 :

$$\neg b \Longrightarrow (\overset{n-1}{\underset{1}{+}} X_{ik}) \qquad \text{avec} \quad X_{ik} \neq b \qquad (2)$$

Si la relation $a \Longrightarrow \neg b$ est vérifiée, nous avons par transitivité :

$$a \Longrightarrow (\overset{n-1}{\underset{1}{+}} X_{ik})$$

Il en résulte que $a*(\overset{n-1}{\underset{1}{+}} X_{ik})=a$

d'où

$$a*b*(\overset{n-1}{\underset{1}{+}} X_{ik})=a$$

$$a*b*\neg b*(\overset{n-1}{\underset{1}{+}} X_{ik})=a*\neg b=a \qquad \text{d'après (1)}$$

or $\neg b*(\overset{n-1}{\underset{1}{+}} X_{ik})=\neg b$ d'après (2) ; on en déduit que :

$$a*b*\neg b=a \qquad \text{d'où } a=\Lambda$$

Cette conclusion étant impossible, les deux relations $a \Longrightarrow b$ et $a \Longrightarrow \neg b$ ne peuvent pas être vérifiées simultanément même quand on interprète la négation par rapport au principe du système de description clos (c.q.f.d.).

II.5. Conditions de décidabilité de la relation $\Longrightarrow$.

II.5.1. Position du problème.

Dans ce qui suit nous posons et résolvons le problème de décidabilité pour tout ensemble de descriptions Δ de base $\mathcal{L}$ muni de la relation $\Longrightarrow$. Ce problème est évidemment essentiel pour la démonstration des théorèmes du système ARCHES (voir chapitre huitième). Etant donné dans un théorème une description C caractérisant l'individu x, peut-on déduire C de H (supposé vrai) à partir de la relation $\Longrightarrow$. En d'autres termes, *étant donné un couple de descriptions(H,C), déterminer s'il vérifie la formule* $H \Longrightarrow C$.

Pour résoudre ce problème nous supposons que toute description D est exprimée sous sa *forme canonique*, c'est-à-dire qu'elle est représentée par une addition de disjonctions de termes descriptifs affirmés ou niés (voir § II.2., proposition 14 et corollaires associés).

$$\begin{cases} D=D_1 * D_2 * \ldots * D_i * \ldots * D_n \\ D_i = D_{i1} + D_{i2} + \ldots + D_{ij} + \ldots + D_{ip} \end{cases}$$

Nous sommes donc ramenés à valider ou invalider toute formule de la forme :

$$H_1 * H_2 * \ldots * H_i * \ldots * H_n \Longrightarrow C_1 * \ldots * C_j * \ldots * C_m \qquad (1)$$

dans laquelle H_i et C_j sont des disjonctions de termes descriptifs affirmés ou niés.

La procédure de décision permettant de résoudre le problème posé est fondée sur la construction d'un arbre ET/OU, noté AEO, définissant *l'espace de validation* de toute formule $H \Longrightarrow C$. Plus précisément, l'arbre AEO est déterminé à partir de la composition de deux arbres ET/OU construits à partir de l'hypothèse H et de la conclusion C, qu'on note respectivement AEOH et AEOC. La définition de cette procédure, qui s'appuie sur les propriétés formelles de la relation $\Longrightarrow$ (voir § II.1., et II.2.), utilise la méthodologie de résolution de problèmes par décomposition, et construction de graphes ET/OU correspondants {29,103}.

Nous utiliserons dans la suite la notion de *sous-descriptions* de toute description D que nous définissons de la manière suivante : /1/ toute description D est une sous-description ; /2/ si D est de la forme $D_1 * D_2 * \ldots * D_i * \ldots * D_n$, alors D_i est une sous-description de D $\forall i \in [1,n]$; /3/ si D est de la forme $D_1 + D_2 + \ldots + D_j + \ldots + D_p$, alors D_j est une sous-description de D $\forall j \in [1,p]$; /4/ si D_j est une sous-description de D_i, et si D_i est une sous-description de D_k, alors D_j est une sous-description de D_k. Une conséquence immédiate de cette définition est que tous les termes descriptifs composant une description sont des sous-descriptions de cette dernière.

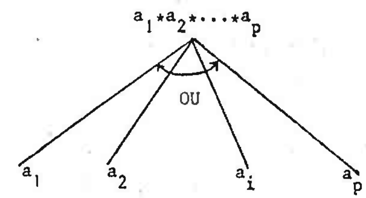
II.5.2. Définition de l'arbre AEOH.

Construit à partir de l'ensemble F_H formé des sous-descriptions issues de la description H exprimée sous forme d'addition de disjonctions de termes descriptifs affirmés ou niés, et d'un opérateur de succession h, l'arbre AEOH est caractérisé par les propriétés suivantes :

α) La racine de AEOH est la description $H=H_1 * H_2 * \ldots * H_i * \ldots * H_n$.

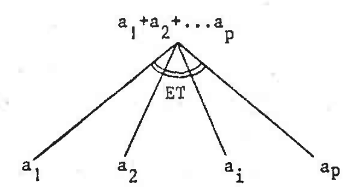
β) Toute description appartenant à F_H et différente d'un terme descriptif est un noeud non terminal de AEOH.

γ) Si a est une description de la forme $a_1 * a_2 * \ldots * a_p$, alors h(a) est une souche-OU de p noeuds, chacun de ces noeuds étant un des termes a_i exprimé sous forme de disjonction d'additions.



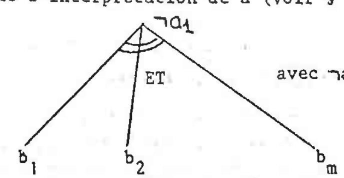
On dira que le noeud associé à $a_1 * a_2 * \ldots * a_p$ est un *noeud-OU*.

δ) Si a est une description de la forme $a_1 + a_2 + \ldots + a_p$, alors h(a) est une souche-ET de p noeuds, chacun de ces noeuds étant un des termes a_i exprimé sous forme d'addition de disjonctions.



On dira que le noeud associé à $a_1 + a_2 + \ldots + a_p$ est un *noeud-ET*.

ε) Si a est une description de la forme $\neg a_1$, a_1 étant un terme descriptif, alors a est un noeud-ET engendrant la souche-ET h(a) dont les noeuds représentent l'interprétation de a (voir § II.3.2.).



avec $\neg a_1 \Longrightarrow (b_1 + b_2 + \ldots + b_m)$

ζ) Tout terme descriptif $a \in F_H$ n'a pas de successeur : $h(a)=\phi$. C'est un noeud terminal de AEOH pouvant être considéré indifféremment comme un noeud-ET ou un noeud-OU.

Exemple.

Construction de l'arbre AEOH pour $H=(a+\neg b+c)_*(c+d)$ avec pour interprétation de $\neg b : I(\neg b)=b_1+b_2+b_3$ (figure 5.8.).

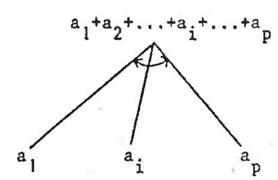
II.5.3. Définition de l'arbre AEOC.

Construit à partir de l'ensemble F_{C_j} formé des sous-descriptions issues de la description C_j exprimée sous forme d'une disjonction de termes descriptifs affirmés ou niés, et d'un opérateur de succession c, l'arbre AEOC est caractérisé par les propriétés suivantes :

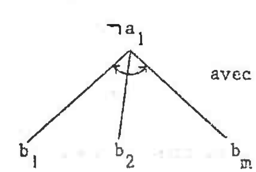
α) La racine de AEOC est la description $C_j=C_{j1}+..+C_{ji}+..+C_{jr}$.

β) Toute description appartenant à F_{C_j} et différente d'un terme descriptif est un noeud non terminal de AEOC.

γ) Si a est une description de la forme $a_1+a_2+...+a_p$, alors c(a) est une souche-OU formée de p noeuds, chacun de ces noeuds étant un des termes a_i représentant un terme descriptif affirmé ou nié.



δ) Si a est une description de la forme $\neg a_1$, a_1 étant un terme descriptif, alors c(a) est une souche-OU formée de noeuds représentant l'interprétation de a



avec $\neg a_1 \Longrightarrow (b_1+b_2+...+b_m)$

ε) Tout terme descriptif $a \in F_{C_j}$ n'a pas de successeur : $c(a)=\phi$. C'est un noeud terminal de AEOC pouvant être considéré indifféremment comme un noeud-ET ou un noeud-OU.

exemple.

A partir de la description $C=C_1*C_2$ telle que $C_1=\neg c$ et $C_2=(b+d)$, construction des arbres AEOC pour C_1 et C_2, avec pour interprétation de $\neg c : I(\neg c)=c_1+c_2+c_3+c_4$ (figure 5.9.).

II.5.4. Construction de l'arbre AEO.

L'arbre AEO est construit à partir de l'ensemble $F=F_H \cup F_{C_j}-\{C_j\}$ et d'un opérateur de succession s dépendant de h et de c. Les règles de construction sont les suivantes :

α) La racine de AEO est celle de AEOH ;

β) Si a est un noeud terminal de AEO, alors a est aussi terminal de AEOC ;

γ) Si $a \in F_H$ et $h(a) \neq \phi$, alors $s(a)=h(a)$;

δ) Si $a \in F_H$ et $h(a)=\phi$, alors $s(a)=c(C_j)$;

ε) Si $a \in F_{C_j}-\{C_j\}$, alors $s(a)=c(a)$.

Remarque.

La définition formelle de la construction de AEO indique que ce dernier s'obtient à partir de l'arbre AEOH dans lequel on "accroche" à chaque terminal l'arbre AEOC sans sa racine.

Exemple.

Construction de l'arbre AEO pour la formule $(a+\neg b)*(a+c) \Longrightarrow \neg c+d$, avec pour interprétation de $\neg b : I(\neg b)=b_1+b_2$ et pour celle de $\neg c : I(\neg c)=c_1+c_2+c_3$ (figure 5.10).

II.5.5. Arbre de validation.

On appellera arbre de validation Av tout sous-arbre de l'arbre AEO défini

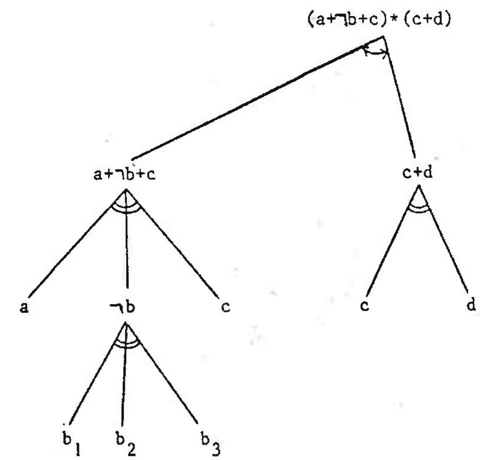


figure 5.8. - Exemple d'arbre AEOH.

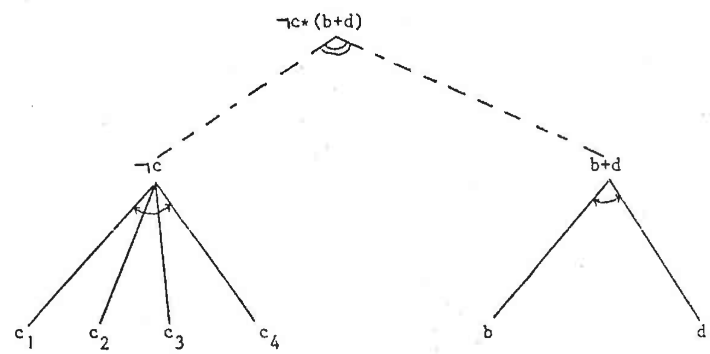


figure 5.9. - Exemple d'arbres AEOC.

deux arb
de type
AEOC

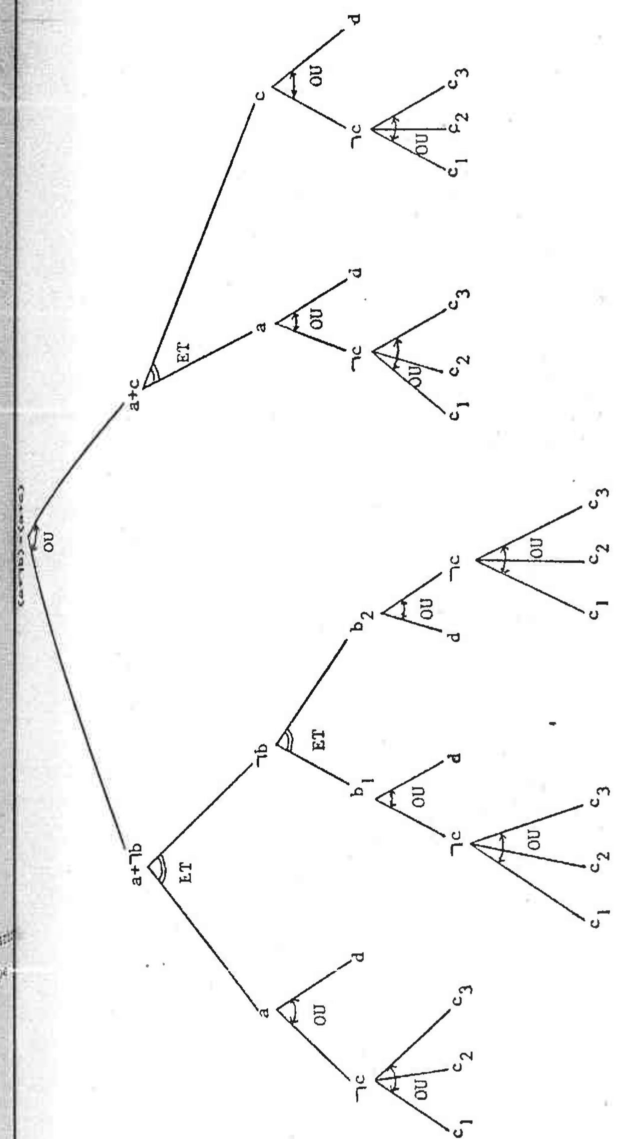


figure 5.10. - Arbre AEO pour la formule $(a+\neg b)*(a+c) \Longrightarrow \neg c+d$.

à partir de l'opérateur de succession v comme suit :

α) La racine de $A\nu$ est celle de AEO ;

β) Si a est un terminal de $A\nu$ alors a est aussi terminal de AEO ;

γ) Si $a \in A\nu$ et si a est un noeud-OU dans AEO, alors $v(a)=b$ tel que $b \in s(a)$;

δ) Si $a \in A\nu$ et si a est un noeud-ET dans AEO, alors $v(a)=s(a)$.

La figure 5.11. montre un arbre de validation construit à partir de l'arbre AEO associé à la formule $(a+\neg b)*(a+c) \Longrightarrow \neg c+d$ (voir figure 5.10.).

Nous devons noter que tout arbre de validation est un arbre ET (immédiat d'après la clause γ de la définition). De plus cette définition, qui détermine un procédé effectif de construction de tout arbre de validation, permet de démontrer la proposition ci-après.

Proposition 19. Le nombre total d'arbres de validation différents construits à partir du même arbre AEO est fini.

démonstration. Soit F l'ensemble de tous les noeuds de l'arbre AEO, supposés tous distingués les uns des autres (par exemple en indiçant de manière différente tous les sommets de même nom). D'après la clause γ de la définition des arbres de validation, tout noeud a de type OU dans l'arbre AEO peut donner naissance à $|s(a)|$ arbres de validation différents. Si N représente le nombre total d'arbres de validation et si $\rho(a)$ représente le nombre total de noeuds-OU dans AEO, alors :

$$N = \sum_{\rho(a)} |s(a)| \qquad \text{or} \qquad \begin{cases} \rho(a) \leqslant |F| \\ |s(a)| \leqslant |F| \end{cases}$$

d'où $N \leqslant |F|^2$; N étant borné supérieurement, et cette borne étant finie, on en déduit que N est fini (c.q.f.d.).

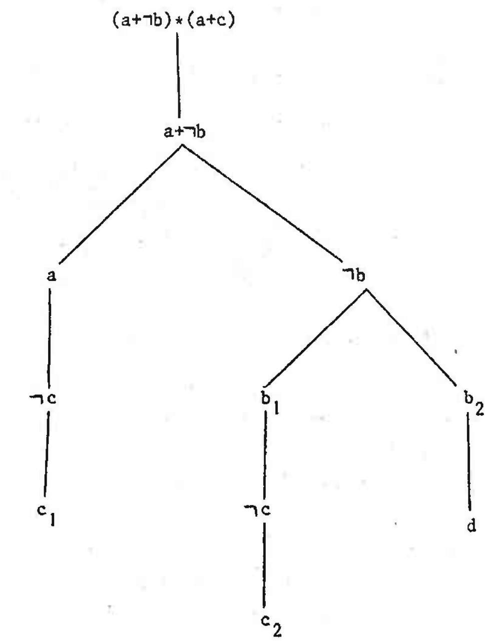


figure 5.11. - Arbre de validation construit à partir de la formule $(a+\neg b)*(a+c) \Longrightarrow \neg c+d$.

Nous désignerons le $i^{\text{ième}}$ arbre de validation construit à partir de l'arbre AEO par la notation $A\nu(i)$; ceci suppose naturellement que $\forall i \neq j$ $A\nu(i) \neq A\nu(j)$. Soit $\mu_i[H,t]$ un chemin quelconque de $A\nu(i)$ ayant pour origine la racine H de $A\nu(i)$ et pour extrêmité un des terminaux t de $A\nu(i)$; ce chemin appartient naturellement à l'arbre AEO car $A\nu(i)$ est un sous-arbre de AEO.

$$\mu_i[H,t] : H \longrightarrow a_2 \rightarrow \quad \dots \quad \rightarrow a_j \rightarrow \quad \dots \quad \rightarrow a_{n-1} \longrightarrow t$$

Soit a_{jo} le point de $\mu_i[H,t]$ qui vérifie les conditions ci-après :

$$\begin{cases} a_{jo} \in v(a_{jo-1}) \ ; \ a_{jo} \in h(a_{jo-1}) \\ v(a_{jo}) \in c(a_{jo}) \end{cases}$$

Ce point existe toujours et est unique par construction même de l'arbre AEO (clause (δ) de cette construction) : c'est un des terminaux de l'arbre AEOH. La procédure VALID(a_{jo},t), définie ci-dessous, établit la validité de la formule $a_{jo} \Longrightarrow a_j$ pour au moins un sommet a_j de $\mu_i[H,t]$ tel que $j>j_o$.

Définition de l'algorithme VALID.

	$\alpha := a_{jo}$
DEBUT	*si* $a_{jo} \Longrightarrow v(\alpha)$ est vérifiée $\Longleftrightarrow \begin{cases} a_{jo} \xrightarrow{*} v(\alpha) \\ \mathcal{A}_R(a_{jo}, v(\alpha)) = \text{SUCCES} \end{cases}$
	alors SUCCES ; FIN
	sinon $\alpha := v(\alpha)$;
	si $\alpha = t$
	alors ECHEC ; FIN
	sinon *aller-à* DEBUT
	fsi
	fsi

La procédure VALID(a_{jo},t) permet d'introduire les deux définitions suivantes :

On dira que *tout chemin* $\mu_i[H,t] = \{H, a_2, .., a_j, .., t\}$ *est valide* si et seulement si l'une des deux conditions suivantes est vérifiée :

ou
1) $\exists\, k<j_o$ et $l>j_o$ tels que $a'_l \xrightarrow{*} a'_k$ avec $a_k = \neg a'_k$ et $a_l = \neg$
2) VALID(a_{jo},t)=SUCCES.

On dira que tout *arbre de validation est valide* si et seulement si tous ses chemins sont valides.

II.5.6. Procédure de décision $\mathcal{A}_E$ pour la relation $\Longrightarrow$.

Posons $H = H_1 * H_2 * \ldots * H_i * \ldots * H_n$ et $C = C_1 * C_2 * \ldots * C_j * \ldots * C_m$.
L'algorithme $\mathcal{A}_E$(H,C) établit la validité de la formule $H \Longrightarrow C$ en cherchant à valider $\forall\, 1 \leq j \leq m$ les formules $H \Longrightarrow C_j$ en prouvant qu'il existe pour chacune d'elles au moins un arbre de validation qui soit valide.

Si $\mathcal{A}_E$(H,C)=SUCCES alors $H \Longrightarrow C$ est vérifiée
Si $\mathcal{A}_E$(H,C)=ECHEC alors $H \Longrightarrow C$ n'est pas vérifiée

Définition de l'algorithme $\mathcal{A}_E$.

	j:=0
DEBUT	j:=j+1
	Construction de l'arbre AEO associé à $H \Longrightarrow C_j$.
	i:=1
BOUCLE	Construction à partir de AEO de $A\nu$(i)
	si la construction est impossible
	alors ECHEC ; FIN
	sinon *si* $A\nu$(i) est valide
	alors *aller-à* SUITE
	sinon i:=i+1 ;
	aller-à BOUCLE
	fsi
	fsi
SUITE	*si* j=m
	alors SUCCES ; FIN
	sinon *aller-à* DEBUT
	fsi

Le théorème suivant exprime la *complétude* de l'ensemble de descriptions Δ de base $\mathcal{C}$ vis à vis de la relation $\Longrightarrow$; en d'autres termes il exprime que toute formule $a \Longrightarrow b$ (a et b $\in$ Δ) qui est vraie est démontrable.

Théorème 5. Si la relation $H \Longrightarrow C$ est vérifiée, alors $\mathcal{A}_E$(H,C)=SUCCES.

démonstration. Si la relation $H \Longrightarrow C$ est vraie, nous allons tenter de la prouver à partir de la définition et des propriétés de la relation $\Longrightarrow$ (voir § II.1., II.2.). Pour ce faire nous supposons que H et C sont sous forme canonique, soit

$$H_1 * \ldots * H_i * \ldots * H_n \Longrightarrow C_1 * \ldots * C_j * \ldots * C_m \qquad (1)$$

Si la relation (1) est vraie, alors la proposition 2 indique que pour tout $j \in [1,m]$ les relations $H \Longrightarrow C_j$ sont vraies, C_j étant une disjonction de termes descriptifs affirmés ou niés de la forme $C_{j1}+\ldots+C_{jl}+\ldots+C_{jr}$:

$$H \Longrightarrow C_{j1}+\ldots+C_{jl}+\ldots+C_{jr}$$

La proposition 14 et le corollaire associé (voir § IV.3.2., chapitre quatrième) indique qu'il existe au moins $i \in [1,m]$ tel que la relation $H_i \Longrightarrow C_{j1}+\ldots+C_{jl}+\ldots+C_{jr}$ est vérifiée. A ce stade de la démonstration nous avons construit le premier niveau de l'arbre AEO associé à la relation $H \Longrightarrow C_j$ (voir définition de l'arbre AEOH, clause γ) dans lequel nous avons retenu l'alternative H_i qui rend nécessairement vraie la relation $H \Longrightarrow C_j$ (figure 5.12.). La description H_i est une disjonction de termes descriptifs affirmés ou niés de la forme $H_{i1}+\ldots+H_{ik}+\ldots+H_{ip}$ car H est sous forme canonique ; d'où :

$$H_{i1}+\ldots+H_{ik}+\ldots+H_{ip} \Longrightarrow C_{j1}+\ldots+C_{jl}+\ldots+C_{jr} \quad (2)$$

D'après la proposition 4, la relation (2) est vérifiée si et seulement si $\forall 1 \leq k \leq p$ les relations $H_{ik} \Longrightarrow C_{j1}+\ldots+C_{jl}+\ldots+C_{jr}$ sont vraies. Cette proposition permet ainsi de construire dans la figure 5.12. la souche-ET engendrée par H_i, qui est un des éléments appartenant au deuxième niveau de l'arbre AEO (voir définition de l'arbre AEOH, clause δ). Désormais, nous sommes amenés à examiner deux cas car H_{ik} est soit un terme descriptif affirmé soit un terme descriptif nié.

Premier cas. *H_{ik} est un terme descriptif.* La proposition 15 (voir § IV.3.2., chapitre quatrième) indique qu'il existe au moin $l \in [1,r]$ tel que la relation $H_{ik} \Longrightarrow C_{jl}$ est vérifiée. On accroche ainsi à H_{ik} la souche-OU engendrée par C_j, appartenant au niveau 3 de l'arbre AEO, et dans laquelle nous avons retenu l'alternative C_{jl} qui rend nécessairement vraie la relation $H_{ik} \Longrightarrow C_j$ (voir définition de l'arbre AEOC, clause γ). Nous devons désormais examiner deux possibilités car C_{jl} est soit un terme descriptif affirmé, soit un terme descriptif nié : /1/ *C_{jl} est un terme descriptif affirmé :* dans ce cas la relation $H_{ik} \Longrightarrow C_{jl}$ est vraie si la relation $H_{ik} \xrightarrow{*} C_{jl}$ est vérifiée (voir § IV.1.1., chapitre quatrième ; condition C4), soit $\mathcal{A}_R(H_{ik},C_{jl})$=SUCCES (figure 5.12., premier cas, première possibilité). /2/ *C_{jl} est un terme descriptif nié de la forme $\neg C'_{jl}$ avec $C'_{jl} \in \mathcal{C}^*$* : dans ce cas on interprète la négation de C'_{jl} comme la disjonction de tous les termes descriptifs T_q construits à partir de la même classe que celle définissant C'_{jl}:$T_1+\ldots+T_q+\ldots+T_v$ (voir § II.3.2.).

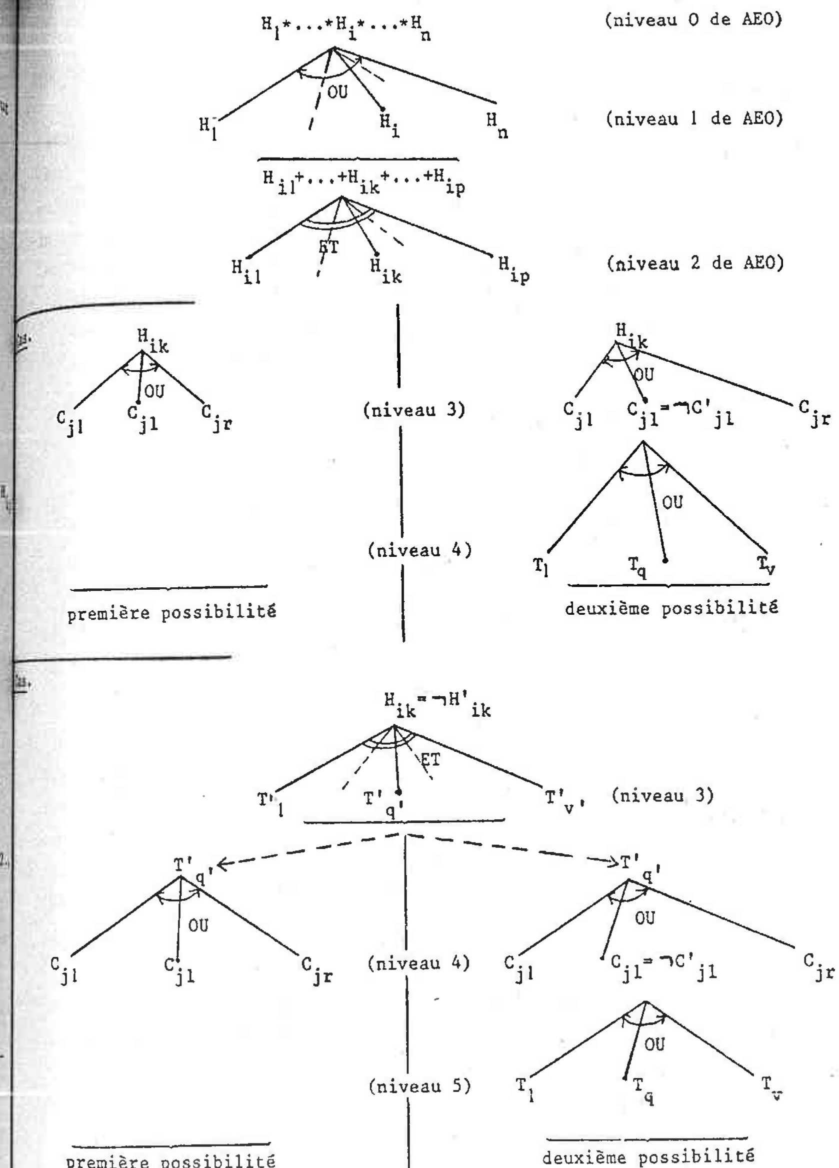


figure 5.12. - Arbre de validation de la preuve $H \Longrightarrow C_j$.

Si la relation $H_{ik} \Longrightarrow C_{jl}$ est vraie alors la relation $H_{ik} \Longrightarrow T_1+\ldots+T_q+\ldots+T_v$ est également nécessairement vérifiée. D'après la proposition 15 (voir § IV.3.2., chapitre quatrième), $\exists\ 1\leq q\leq v$ tel que la relation $H_{ik} \Longrightarrow T_q$ est vérifiée, T_q étant un terme descriptif affirmé. On accroche ainsi à C_{jl} une souche-OU qui appartient au quatrième niveau de l'arbre AEO (voir définition de AEOC, clause δ), et dans laquelle l'alternative T_q a été retenue (figure 5.12., premier cas, deuxième possibilité) ; on est alors ramené au stade précédent (première possibilité), soit $\mathcal{A}_R(H_{ik}, T_q)$=SUCCES.

Deuxième cas. *H_{ik} est un terme descriptif nié de la forme $\neg H'_{ik}$, avec $H'_{ik}\in \mathcal{C}^*$*. Soit $T'_1+\ldots+T'_q+\ldots+T'_{v'}$ l'interprétation de la négation de H'_{ik} (voir § II.3.2.). Si la relation $H_{ik} \Longrightarrow C_{j1}+\ldots+C_{jl}+\ldots+C_{jr}$ est vraie alors la relation $T'_1+\ldots+T'_{q'}+\ldots+T'_{v'} \Longrightarrow C_{j1}+\ldots+C_{jl}+\ldots+C_{jr}$ est nécessairement vérifiée. D'après la proposition 4, cette dernière relation est vérifiée si et seulement si $\forall\ 1\leq q'\leq v'$ les relations $T'_{q'} \Longrightarrow C_{j1}+\ldots+C_{jl}+\ldots+C_{jr}$ sont vraies. Ceci permet de construire la souche-ET engendrée par H_{ik} qui appartient au troisième niveau de l'arbre AEO (voir définition de AEOH, clause ε). Si la relation $T'_{q'} \Longrightarrow C_{j1}+\ldots+C_{jl}+\ldots+C_{jr}$ est vraie alors $\exists\ 1\leq l\leq r$ tel que la relation $T'_{q'} \Longrightarrow C_{jl}$ est vérifiée d'après la proposition 15. Il en résulte que $\forall\ 1\leq q'\leq v'$ on accroche à $T'_{q'}$ la souche-OU engendrée par C_j dans laquelle l'alternative C_{jl} a été retenue ; C_{jl} appartient ainsi au quatrième niveau de l'arbre AEO (voir définition de AEOC, clause γ). Nous devons désormais examiner deux possibilités car C_{jl} est soit un terme descriptif affirmé, soit un terme descriptif nié : /1/ *C_{jl} est un terme descriptif affirmé :* dans ce cas la relation $T'_{q'} \Longrightarrow C_{jl}$ est vraie si la relation $T'_{q'} \xrightarrow{*} C_{jl}$ est vérifiée, soit $\mathcal{A}_R(T'_{q'},C_{jl})$=SUCCES (figure 5.12., deuxième cas, première possibilité). /2/ *C_{jl} est un terme descriptif nié de la forme $\neg C'_{jl}$ avec $C'_{jl}\in\mathcal{C}^*$* : soit $T_1+\ldots+T_q+\ldots+T_v$ l'interprétation de la négation de C'_{jl}. Si la relation $T'_{q'} \Longrightarrow C_{jl}$ est vérifiée alors la relation $T'_{q'} \Longrightarrow T_1+\ldots+T_q+\ldots+T_v$ est nécessairement vérifiée. D'après la proposition 15 il existe $q\in[1,v]$ tel que la relation $T'_{q'} \Longrightarrow T_q$ est vérifiée. On accroche ainsi à C_{jl} une souche-OU qui appartient au cinquième niveau de l'arbre AEO (voir définition de AEOC, clause δ), et dans laquelle l'alternative T_q a été retenue (figure 5.12., deuxième cas, deuxième possibilité). Si la relation $T'_{q'} \Longrightarrow T_q$ est vérifiée alors il en est de même de la relation $T'_{q'} \xrightarrow{*} T_q$ (voir § IV.1.1., chapitre quatrième) d'où $\mathcal{A}_R(T'_{q'},T_q)$=SUCCES. Dans le cas

contraire (i.e. $\mathcal{A}_R(T'_{q'},T_q)$=ECHEC), la relation $\neg H'_{ij} \Longrightarrow \neg C'_{jl}$ doit être nécessairement vérifiée ; il en résulte que $C'_{jl} \Longrightarrow H'_{ik}$ d'après le corollaire 2. Comme C'_{jk} et $H'_{ik}\in\mathcal{C}^*$ il en résulte que la relation $C'_{jl} \xrightarrow{*} H'_{ik}$ est nécessairement vérifiée, soit $\mathcal{A}_R(C'_{jl},H'_{ik})$=SUCCES.

En conclusion, la relation $H\Longrightarrow C$ étant supposée vérifiée, nous avons élaboré une preuve de sa validité fondée sur les propriétés de $\Longrightarrow$. Nous avons montré que cette preuve construit pour tout $j\in[1,m]$ l'arbre AEO, associé à la relation $H \Longrightarrow C_j$, dans lequel nous avons isolé un sous-arbre *Av* qui vérifie les propriétés suivantes (figure 5.12.) : /1/ la racine de *Av* est celle de AEO ; /2/ si a est un terminal de *Av*, alors a est un terminal de AEO ; /3/ si a est un noeud-OU dans AEO et appartient également à *Av*, alors un seul élément de la souche engendrée par a appartient à *Av* ; /4/ si a est un noeud-ET dans AEO et appartient également à *Av*, alors toute la souche engendrée par a appartient à *Av* ; /5/ enfin si a est une négation appartenant à *Av*, elle engendre dans *Av* une souche-ET si a est une sous-description de H, et une souche-OU si a est une sous-description de C_j. Ces propriétés montrent que *Av* est un arbre de validation particulier associé à C_j, qui est construit par l'algorithme $\mathcal{A}_E$ en un nombre fini d'étapes. En effet le nombre total d'arbres de validation différents construits à partir du même arbre AEO est fini (voir § II.5.5., proposition 19) ; l'algorithme $\mathcal{A}_E$ ne boucle donc pas indéfiniment. Enfin nous avons prouvé que tous les chemins de *Av* sont valides. Il en résulte que $\mathcal{A}_E(H,C)$=SUCCES (c.q.f.d.).

CHAPITRE SIXIEME.

REPRESENTATION DE L'EVOLUTION DES DESCRIPTIONS : EXTENSION DU SYSTEME FORMEL S_Δ

-=-=-=-=-=-=-=-=-=-

I. POSITION DU PROBLEME.

Le système symbolique ARCHES, ayant pour objectif principal la production de connaissances à partir de connaissances déjà exprimées, intègre dans sa conception des opérations permettant de rendre compte du caractère évolutif des données, en rapport précisément avec sa finalité cognitive : acquisition de connaissances nouvelles, représentation d'évènements en termes de changements d'états, modification de la description de phénomènes étudiés à partir de calculs portant sur ces phénomènes et/ou sur des objets qui leur sont liés, etc. Cependant, les transformations que peuvent subir les données au cours des calculs où elles sont mobilisées, ne doivent pas rendre incohérent le système ARCHES (voir § II.4., chapitre cinquième). Par exemple, dans une réalisation particulière de ARCHES décrivant le comportement d'individus, nous devons pouvoir affirmer à un instant donné l'énoncé "JEAN MANGE", puis à un instant ultérieur l'énoncé "JEAN NE MANGE PAS" ; énoncés qui sont contradictoires dans un système logique classique où les propriétés et les relations sont affirmées ou niées de manière définitive, indépendamment du temps et de l'espace. De la même façon certains énoncés du langage naturel qui traduisent des changements d'états doivent être représentés dans ARCHES comme le passage d'un état descriptif actuel à un état descriptif ultérieur (voir § III. 4. , chapitre deuxième). Par exemple l'énoncé "MARIE SE REVEILLE" peut s'interpréter dans ARCHES comme un changement d'état entre le SOMMEIL et la VEILLE ; là encore les problèmes de cohérence sont présents en ce sens que les deux activités SOMMEIL et VEILLE ne peuvent pas être réalisées simultanément (voir § VII).

Ces quelques remarques montrent que l'évolution des traits qualifiant les phénomènes étudiés dans ARCHES présente deux caractéristiques principales : d'une part, elle exprime une suite de transformations dans le temps - donc toujours dans le même sens - de tout ou partie de l'ensemble des descriptions ; d'autre part, elle peut introduire des contradictions si l'on reste à l'intérieur d'un système formel classique. Cette évolution, contrôlée au niveau de la cohérence intrinsèque du système symbolique, définit ainsi une relation d'ordre projetée sur l'axe des temps entre les différents états associés aux transformations que subissent les phénomènes étudiés. Ces transformations peuvent être véhiculées en langage naturel, au moyen de mots outils (comme ET-PUIS, AVANT, ou APRES) qui permettent de relier des états descriptifs successifs ; cette succession indique une addition projetée dans le temps alors que le connecteur ET défini précédemment (voir § IV.1.2. , chapitre quatrième) introduit une opération d'addition indépendante du temps. L'extension du système formel S_Δ caractérisant l'ensemble Δ dans un sens tel que cette évolution puisse être marquée formellement permet ainsi de définir dans ARCHES une représentation sous-jacente du connecteur ET-PUIS.

II. REPRESENTATION FORMELLE DE L'EVOLUTION DES DESCRIPTIONS.

II.1. Définition des connecteurs F et G.

On appelle connecteur de *futur immédiat* - noté F -, et connecteur de *futur médiat* - noté G - deux nouveaux connecteurs qui permettent d'exprimer l'évolution des descriptions ; F et G appartiennent ainsi à l'alphabet $\mathcal{A}$ qui engendre les descriptions (voir § II.4.2., chapitre quatrième). Plus précisément, les deux règles suivantes complètent les règles de formation des descriptions :

R7 Si D est une description, alors FD est une description ;

R8 Si D est une description, alors GD est une description.

II.2. Rapports entre la relation $\Longrightarrow$ et les connecteurs F et G.

L'extension de l'activité inférentielle du système formel S_Δ sera établie dès lors que seront déterminés les rapports entre la relation de déduction $\Longrightarrow$ et les connecteurs F et G. Pour ce faire, nous supposons que la relation $\Longrightarrow$ satisfait également aux conditions ci-après (voir § IV.1., chapitre quatrième) :

C13	$F\neg a=\neg Fa$	
C14	$F(a*b)=Fa*Fb$	
C15	$GGa=Ga$	
C16	$a \Longrightarrow Ga$	$(\Leftrightarrow a*Ga=a)$
C17	$Fa \Longrightarrow Ga$	$(\Leftrightarrow Fa*Ga=Fa)$
C18	$FGa \Longrightarrow GFa$	$(\Leftrightarrow FGa*GFa=FGa)$
C19	$G(a+b) \Longrightarrow Ga+Gb$	
C20	$G\Lambda=\Lambda$	

C21 Si a $\Longrightarrow$ b alors Fa $\Longrightarrow$ Fb

C22 Si a $\Longrightarrow$ b alors Ga $\Longrightarrow$ Gb

La condition C13 exprime que l'évolution des descriptions est déterministe (les quantifications universelle et existentielle sont confondues, i.e. $\neg F \neg = F$). La condition C15 indique que le connecteur G est transitif. C16 et C17 expriment respectivement que le présent et le futur immédiat font partie du futur ; C18 détermine les rapports sémantiques entre futur immédiat et futur médiat. Par ailleurs C20 exprime la cohérence de l'évolution des descriptions, en ce sens que l'ensemble Δ possède un minimum et un seul (voir § IV.3.1.c, chapitre quatrième).

Enfin les conditions C21 et C22 indiquent que si la formule a $\Longrightarrow$ b est toujours vérifiée alors elle est également toujours vérifiée dans le futur.

Remarque 1.

Les conditions C1 et C5 à C20 sont de la forme a $\Longrightarrow$ b (ou a=b ce qui est équivalent à a$\Longrightarrow$b et b$\Longrightarrow$ a ; voir § IV.1., chapitre quatrième). Il en résulte que ces conditions sont les *axiomes* du système formel dont les formules sont de la forme a $\Longrightarrow$ b et qui formalise la relation de déduction $\Longrightarrow$(i.e. le symbole métalinguistique $\Longrightarrow$ devient un élément du langage objet de ce système formel). Par contre les conditions C2, C3, C4, C21 et C22 sont les *règles d'inférence* de ce même système formel car elles sont de la forme $\dfrac{a \Longrightarrow b}{c \Longrightarrow d}$.

III. CARACTERISATION SEMANTIQUE DU SYSTEME FORMEL S_Δ.

Dans ce paragraphe nous définissons l'interprétation du système formel S_Δ de caractérisation des descriptions. Cette définition détermine en particulier la sémantique de l'évolution des descriptions qui est exprimée sur le plan syntaxique par les connecteurs F et G. Elle a pour objet essentiel de prouver la cohérence intrinsèque des conditions auxquelles la relation $\Longrightarrow$ satisfait (voir § IV.1., chapitre quatrième ; et § II.).

III.1. Définition de l'interprétation de S_Δ.

Définition 1. On appellera interprétation du système formel $S_\Delta = \{\mathcal{A}, \Delta, \Delta_T, \Longrightarrow\}$ la structure $\mathcal{M} = \{D, \mathcal{L}\}$ dont les éléments sont définis comme suit :

α) Définition de D.

D est un ensemble non vide appelé *domaine d'interprétation de* S_Δ.

Pour chaque étude de cas conduite sous ARCHES il existe un tel domaine qui est inclus dans l'univers des connaissances réelles nécessaires pour résoudre les problèmes posés par cette étude.

L'ensemble D est formé de l'union de deux ensembles D_I et D_C : les éléments du premier correspondent aux symboles d'individus appartenant à l'ensemble $\mathfrak{J}$, et les éléments du deuxième correspondent aux symboles de concepts.

Ces ensembles possèdent deux propriétés déterminant les relations sémantiques qui expriment certaines des modalités d'interprétation des individus et des concepts :

/1/ Tout concept représente un ensemble d'individus (voir § I.1., chapitre troisième). Il doit donc être interprété comme une partie de l'ensemble D_I ; d'où la propriété suivante (dans ce qui suit on désignera l'ensemble des parties d'un ensemble quelconque E par $\mathcal{P}(E)$) :

$$D_C \subset \mathcal{P}(D_I)$$

/2/ Tout concept peut également être considéré comme un individu (voir § I.3.2., chapitre troisième). Il doit donc être interprété comme un individu ; d'où la propriété suivante :

$$D_C \subset D_I$$

β) Définition de $\mathcal{L}$.

$\mathcal{L}$ est une *fonction de correspondance* qui permet d'associer à chaque composante du système formel S_Δ son interprétation dans le domaine D.

/1/ *Interprétation des individus.*
A chaque individu x est associé son interprétation, notée $\mathcal{L}(x)$, qui appartient à l'ensemble D_I.

/2/ *Interprétation des concepts.*
L'interprétation de tout concept A, considéré comme un ensemble d'individus, est définie comme suit (voir α, /1/) :

$$\mathcal{L}(A) \in D_C \subset \mathcal{P}(D_I)$$

Le symbole fonctionnel I, qui permet de désigner par convention un concept ayant le statut d'individu (voir § I.3.2., chapitre troisième), est donc interprété comme la fonction d'identité (voir α, /2/) :

$$\mathcal{L}(I(A)) = \mathcal{L}(A)$$

/3/ *Interprétation des informations sans extension.*
Tout trait t représentant une information sans extension est interprété comme l'ensemble des interprétations des individus qui sont caractérisés par ce trait. En d'autres termes, toute information sans extension t est interprétée comme une partie de l'ensemble D_I :

$$\mathcal{L}(t) \in \mathcal{P}(D_I)$$

Comme tout trait est soit une information sans extension, soit un concept, soit un individu (voir § II.1.2., chapitre quatrième ; propriété 3), il en résulte que l'ensemble des traits est interprété par l'ensemble T tel que :

$$T = D_I \cup \mathcal{P}(D_I)$$

/4/ *Interprétation de la relation SET* (voir § I.3.2., chapitre troisième).
Le prédicat SET est interprété comme la relation d'appartenance définie sur $D_C \times D_C$:

$$\mathcal{L}(SET(A,B)) = \mathcal{L}(A) \in \mathcal{L}(B)$$

/5/ *Interprétation du lien INS* (voir § I.2.2., chapitre troisième).
Le symbole fonctionnel INS est interprété comme une application surjective de $D_C \times D_I$ dans D_I qui respecte le type des individus (voir § II.1., chapitre troisième) ; en d'autres termes $\mathcal{L}$ est définie de manière que :

$$\mathcal{L}(INS(A,x)) = \mathcal{L}(x) \text{ tel que } \mathcal{L}(x) \in \mathcal{L}(A), \text{avec } x \in \mathcal{J}$$

Cette interprétation montre en particulier qu'un individu peut appartenir à plusieurs concepts, ce qui est tout à fait conforme non seulement à la définition des individus, mais aussi aux propriétés formelles des R-graphes (voir respectivement § I.2.2., et § III.3., chapitre troisième).

/6/ *Interprétation du symbole fonctionnel* \$ (voir § II.3.2., chapitre quatrième).
Tout d'abord par convention d'écriture le terme fonctionnel $\$(t,\Lambda)$ est équivalent à t, et ceci quel que soit le type du trait t (voir § II.3.2. b, chapitre quatrième). Dans ce cas \$ est interprété comme la fonction d'identité :

$$\mathcal{L}(\$(t,\Lambda)) = \mathcal{L}(t)$$

Par ailleurs, quand la description locale n'est pas représentée par la description vide Λ, alors les traits qui sont décrits localement ne peuvent être qu'une information sans extension ou un concept (voir § II.1.2., chapitre quatrième ; propriété 4). Cependant l'interprétation du terme fonctionnel $\$(t, op_n(C,t_1))$ est la même quel que soit le type du trait t car il est toujours interprété comme un ensemble d'individus (voir β/2/ et β/3/).

Nous définissons l'interprétation du terme descriptif $op_n(C,t_1)$ comme l'ensemble des interprétations des individus caractérisés par ce terme descriptif.

En d'autres termes, l'interprétation de toute couple <classe, opérateur> de la forme $<C, op_n>$ détermine une application de T^n dans $\mathcal{P}(D_I)$:

$$\mathcal{L}(<C,op_n>): \quad T^n \longrightarrow \mathcal{P}(D_I)$$

avec $$\mathcal{L}(<C,op_n>)(\mathcal{L}(t_1)) \in \mathcal{P}(D_I)$$

Par suite le terme fonctionnel $\$(t,op_n(C,t_1))$ est interprété comme le sous ensemble des interprétations des individus, appartenant à l'ensemble $\mathcal{L}(t)$, qui sont caractérisés par le terme descriptif $op_n(C,t_1)$.

Plus précisément, *le symbole fonctionnel \$ est interprété comme l'opération d'intersection :*

$$\mathcal{L}(\$(t,op_n(C,t_1))) = \mathcal{L}(t) \cap \mathcal{L}(<C,op_n>)(\mathcal{L}(t_1))$$

Nous devons noter que si $\mathcal{L}(t)$ est toujours un élément appartenant à D_C quand t est un concept, il n'en est plus de même dans le cas général pour $\mathcal{L}(\$(t,op_n(C,t_1)))$. En d'autres termes, $\mathcal{L}(\$(t,op_n(C,t_1)))$ est un ensemble "homogène" d'individus en ce sens qu'il est inclus dans le concept $\mathcal{L}(t)$; mais cet ensemble ne forme pas un concept particulier représenté explicitement par un symbole dans le système ARCHES.

Remarque 2.

Enfin remarquons que l'interprétation du symbole fonctionnel \$ ne fait pas intervenir la notion d'évolution (i.e. l'expression du temps). En d'autres termes les descriptions locales permettant de caractériser les traits n'évoluent pas indépendamment des descriptions principales dans lesquelles elles interviennent. Ce sont les éléments de description de caractérisation des individus qui évoluent globalement (voir (β) /7/ et /8/), et ceci est tout à fait conforme à la définition syntaxique des connecteurs F et G (voir § II.).

/7/ *Interprétation des termes descriptifs formant la base $\mathcal{C}$ de Δ* (voir § II.4.2., chapitre quatrième).

L'ensemble dénombrable Δ des descriptions est construit, en particulier, à partir d'un ensemble dénombrable $\mathcal{C}$ de termes descriptifs appelé base de Δ. Tout élément appartenant à $\mathcal{C}$ est de la forme $op_n(C,\#(t_1,..,t_i,..,t_n))$ dans laquelle op_n est un opérateur n-aire, C une classe, et enfin $\forall 1 \leq i \leq n$ t_i est un trait décrit ou non décrit

(voir § II.3.2., chapitre quatrième). Nous nous proposons de définir l'interprétation des éléments appartenant à $\mathcal{C}$ compte-tenu des modalités d'évolution des individus que nous définissons ci-après.

Les *modalités d'évolution des individus* sont déterminées à partir de la définition d'un ensemble Ω isomorphe à $\mathbb{N}$. Tout élément de Ω sera appelé *état*. Les états marquent les différentes transformations que peuvent subir les descriptions caractérisant les individus. Chaque état, qui exprime la situation descriptive des individus à un moment donné, sera repéré par un entier naturel. Dans le cas général, un état quelconque donne naissance à de nouveaux états qui eux-mêmes évoluent à leur tour et ainsi de suite. Dans notre modèle d'interprétation nous supposons que l'*évolution est déterministe* car Ω est isomorphe à $\mathbb{N}$ (voir § II.2. ; condition C13) ; nous entendons par là le fait que chaque état possible ne peut évoluer que vers un état et un seul. Ainsi à partir d'un état initial E_o les interprétations des descriptions caractérisant les individus dans l'état E_o (voir également (β) /8/) évoluent de telle manière que les transformations qu'elles subissent engendrent une séquence linéaire d'états successifs $E_1,E_2,..,E_i,...$ organisés par la structure $\mathcal{M}$, organisation qui est de même nature que celle définie par Kripke pour les logiques modales, {55} et {123} (figure 6.1.).

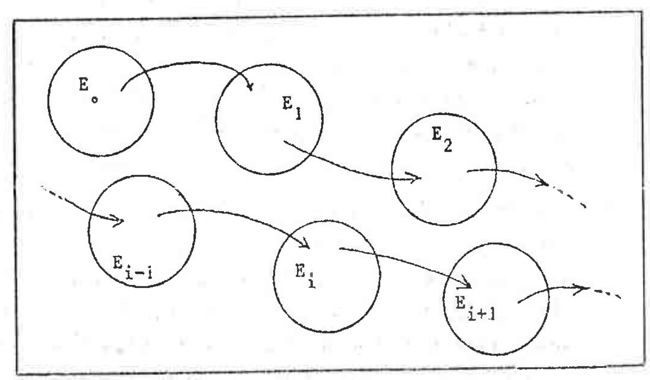


figure 6.1. - Modalités d'évolution des descriptions des individus.

L'interprétation d'un terme descriptif appartenant à $\mathcal{L}$ est définie comme l'ensemble des interprétations des individus qui sont caractérisés par ce terme descriptif ; et cette interprétation tient compte de l'évolution de ces individus qui sera exprimée à l'aide de l'ensemble $\mathbb{N}$ conformément à la définition de Ω.

En d'autres termes, l'interprétation de tout couple <classe,opérateur> de la forme $<C,op_n>$ détermine une application de T^n dans $(\mathbb{N} \longrightarrow \mathcal{P}(D_I))$:

$$\mathcal{L}(<C,op_n>) : \quad T^n \longrightarrow (\mathbb{N} \longrightarrow \mathcal{P}(D_I))$$

avec $\mathcal{L}(op_n(C,\#(t_1,\ldots,t_n)))(i)=\mathcal{L}(<C,op_n>)(\mathcal{L}(t_1),\ldots,\mathcal{L}(t_n))(i)\in \mathcal{P}(D_I)$

L'union des ensembles de la forme $\mathcal{L}(op_n(C,\#(t_1,\ldots,t_n)))(i)$ pour toutes les interprétations des couples <classe, opérateur> forme le ième état.

En particulier si pour l'interprétation du couple $<C,op_n>$ on a $\mathcal{L}(op_n(C,\#(t_1,\ldots,t_n)))(i)=\phi$ alors aucun individu n'est caractérisé par le terme descriptif $op_n(C,\#(t_1,\ldots,t_n))$ dans l'état i.

/8/ *Interprétation des descriptions* (voir § II.4.2., chapitre quatrième).

Toute description a est interprétée comme l'ensemble des interprétations des individus qui sont caractérisés par a ; plus précisément, elle est interprétée comme une application de $\mathbb{N}$ dans $\mathcal{P}(D_I)$ pour tenir compte de l'évolution des individus :

$$\Delta \longrightarrow (\mathbb{N} \longrightarrow \mathcal{P}(D_I))$$

avec $\qquad \mathcal{L}(a)(i) \in \mathcal{P}(D_I)$

$\mathcal{L}(a)(i)$ détermine l'ensemble des interprétations des individus qui sont caractérisés par la description a dans l'état i.

Nous désignerons par $\mathcal{L}_o$ l'application qui détermine l'interprétation des descriptions à l'état initial E_o :

$$\mathcal{L}_o(a)=\mathcal{L}(a)(0)$$

Les règles de formation des descriptions à partir des connecteurs *, +, ¬ , F et G ainsi que la définition de $\mathcal{L}_o$ contribuent à déterminer par récurrence la fonction de correspondance $\mathcal{L}$ à l'aide des règles ci-après (voir § II.4.2. b, chapitre quatrième) :

(1) $\mathcal{L}(a*b)(n) = \mathcal{L}(a)(n) \cap \mathcal{L}(b)(n)$

(2) $\mathcal{L}(a+b)(n) = \mathcal{L}(a)(n) \cup \mathcal{L}(b)(n)$

(3) $\mathcal{L}(\neg a)(n) = \complement \mathcal{L}(a)(n)$ $\qquad$ ($\complement$: complément dans D_I)

(4) $\mathcal{L}(Fa)(n) = \mathcal{L}(a)(n+1)$

(5) $\mathcal{L}(Ga)(n) = \bigcup_{p=o}^{\infty} \mathcal{L}(a)(n+p)$

(6) $\mathcal{L}(\Lambda)(n) = \phi$

Les trois premières règles correspondent à l'interprétation classique et intuitive des connecteurs *, + et ¬.

Les règles (4) et (5) déterminent la sémantique de l'évolution des descriptions : l'interprétation du connecteur F (règle (4)) montre que ce dernier permet d'exprimer l'évolution entre deux états consécutifs E_i et E_{i+1} (figure 6.2.) ; celle du connecteur G (règle (5)) montre que ce dernier exprime l'évolution entre deux états successifs E_i et E_j ($j \geq i$),c'est-à-dire non nécessairement consécutifs. La règle (5) permet de déduire que $\mathcal{L}(GGa)(n) = \mathcal{L}(Ga)(n)$,ce qui prouve la transitivité du connecteur G. Enfin remarquons que, contrairement au connecteur F, le connecteur G suppose que *le présent fait partie du futur* (car $j \geq i$).

Par ailleurs la règle (6) indique que, quel que soit l'état E_i, il n'existe aucun individu x qui soit caractérisé par la description vide Λ (voir § II.4., chapitre quatrième ; hypothèse 3).

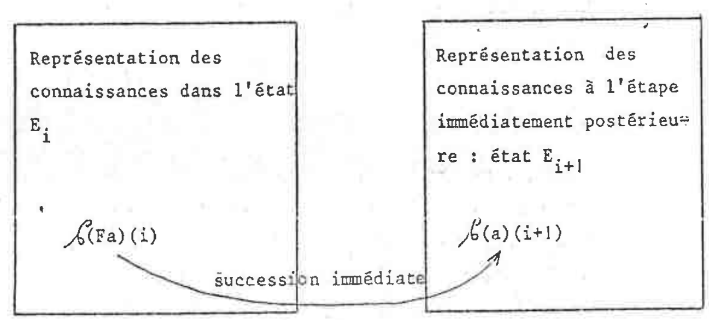


figure 6.2. - *Interprétation du connecteur F.*

Remarque 3.

L'introduction du connecteur G n'est pas nécessaire pour la définition du connecteur ET-PUIS. Cependant il est indispensable pour la dérivation des descriptions dans lesquelles la séquence des états n'est pas explicitée (on recherche au moins l'existence d'un état dans lequel un individu x a tel ou tel groupe de propriétés).

Remarque 4.

L'interprétation du système symbolique ARCHES concerne essentiellement l'interprétation des structures de la forme $C(x,\mathcal{D}(x))$ qui déterminent le langage des formules de ARCHES (voir § III.2., chapitre septième). Il est bien évident que cette interprétation utilise les modalités d'interprétation du système formel S_Δ ; elle sera définie ultérieurement (voir § III.4., chapitre septième).

Exemple.

On se propose de définir les éléments qui participent à la définition de l'interprétation de la description d= "Col rouge clair et revers rouge très foncé" pouvant caractériser des vêtements (manteau, veste,...).

. Représentation de la description d.

d=ISA(PARTIE,\$(COL,ISA(COULEUR,\$(ROUGE,ISA(TEINTE,CLAIR)))))*
ISA(PARTIE,\$(REVERS,ISA(COULEUR,\$(ROUGE,ISA(TEINTE,\$(FONCE,
ISA(INTENSITE,TRES)))))))

Cette description est composée des éléments formels suivants :

Classes : PARTIE, COULEUR, TEINTE, INTENSITE
opérateur : ISA
concepts : COL, REVERS
informations sans extension : ROUGE, CLAIR, FONCE, TRES

Posons :

a=ISA(PARTIE,\$(COL,ISA(COULEUR,\$(ROUGE,ISA(TEINTE,CLAIR)))))

b=ISA(PARTIE,\$(REVERS,ISA(COULEUR,\$(ROUGE,ISA(TEINTE,\$(FONCE,
ISA(INTENSITE,TRES)))))))

d'où :

d=a*b

. Modalités d'interprétation de d.

D'après (β)/8/, point (1) , on a :

$$\mathcal{L}(d)(i)=\mathcal{L}(a)(i)\cap\mathcal{L}(b)(i)\in\mathcal{P}(D_I)$$

Les termes descriptifs a et b sont alors interprétés de la manière suivante (voir (β) /7/) :

$$\mathcal{L}(a)=\mathcal{L}(\langle \mathrm{PARTIE,ISA}\rangle)(\mathcal{L}(\$(\mathrm{COL,ISA(COULEUR},\$(\mathrm{ROUGE,ISA(TEINTE,CLAIR)))))})$$

$$\mathcal{L}(b)=\mathcal{L}(\langle \mathrm{PARTIE,ISA}\rangle)(\mathcal{L}(\$(\mathrm{REVERS,ISA(COULEUR},\$(\mathrm{ROUGE,ISA(TEINTE},\$(\mathrm{FONCE,ISA(INTENSITE,TRES))))))))})$$

Les traits décrits COL et REVERS étant des concepts, il vient (voir (β) /6/) :

$$\mathcal{L}(a)=\mathcal{L}(\langle \mathrm{PARTIE,ISA}\rangle)(\mathcal{L}(\mathrm{COL})\cap\mathcal{L}(\langle \mathrm{COULEUR,ISA}\rangle)(\mathcal{L}(\$(\mathrm{ROUGE,ISA(TEINTE,CLAIR)))))})$$

$$\mathcal{L}(b)=\mathcal{L}(\langle \mathrm{PARTIE,ISA}\rangle)(\mathcal{L}(\mathrm{REVERS})\cap\mathcal{L}(\langle \mathrm{COULEUR,ISA}\rangle)(\mathcal{L}(\$(\mathrm{ROUGE,ISA(TEINTE},\$(\mathrm{FONCE,ISA(INTENSITE,TRES)))))))})$$

Finalement les informations sans extension ROUGE, CLAIR, FONCE et TRES conduisent aux interprétations ci-après (voir également (β) /6/) :

$$\mathcal{L}(a)=\mathcal{L}(\langle \mathrm{PARTIE,ISA}\rangle)(\mathcal{L}(\mathrm{COL})\cap\mathcal{L}(\langle \mathrm{COULEUR,ISA}\rangle)(\mathcal{L}(\mathrm{ROUGE})\cap\mathcal{L}(\langle \mathrm{TEINTE,ISA}\rangle)(\mathcal{L}(\mathrm{CLAIR}))))$$

$$\mathcal{L}(b)=\mathcal{L}(\langle \mathrm{PARTIE,ISA}\rangle)(\mathcal{L}(\mathrm{REVERS})\cap\mathcal{L}(\langle \mathrm{COULEUR,ISA}\rangle)(\mathcal{L}(\mathrm{ROUGE})\cap\mathcal{L}(\langle \mathrm{TEINTE,ISA}\rangle)(\mathcal{L}(\mathcal{L}(\mathrm{FONCE})\cap\mathcal{L}(\langle \mathrm{INTENSITE,ISA}\rangle)(\mathcal{L}(\mathrm{TRES})))))$$

III.2. Validité de la relation de déduction $\Longrightarrow$ par rapport à toute interprétation $\mathcal{M}$.

Définition 2. Etant donné deux descriptions quelconques a et b appartenant à Δ, on dira que *la formule a ⟹ b est valide* si pour toute interprétation $\mathcal{M}$ et pour tout i $\mathcal{L}(a)(i)$ est inclus dans $\mathcal{L}(b)(i)$:

$$\mathcal{L}(a)(i)\subset\mathcal{L}(b)(i)$$

De même on dira qu'une *condition de la forme "SI <Prémisses> ALORS <Conclusion >" est valide* si la validité des prémisses entraîne celle de la conclusion.

Définition 3. On dira que la *relation de déduction* $\Longrightarrow$ *est valide* si toutes les formules qui la composent sont valides.

Théorème. La relation de déduction $\Longrightarrow$ est valide.

démonstration. Pour démontrer que la relation de déduction $\Longrightarrow$ est valide il suffit de démontrer que toutes les conditions auxquelles cette relation satisfait sont valides.
La validité des conditions C1 à C12, qui déterminent les rapports entre d'une part les connecteurs $*$, $+$ et $\neg$ et d'autre part la relation $\Longrightarrow$, est immédiatement établie car le quotient $\Delta/=$ est un treillis distributif complémenté (voir § IV.1., chapitre quatrième et § II.2., chapitre cinquième).
Prouvons la validité des conditions C13 à C22 (voir § II.2.) :
Preuve de C13. Pour prouver la validité de la formule $F\neg a = \neg Fa$ il suffit de montrer que $\forall i\ \mathcal{L}(\neg Fa)(i) = \mathcal{L}(F\neg a)(i)$. Nous avons d'après la règle 4 $\mathcal{L}(F\neg a)(i) = \mathcal{L}(\neg a)(i+1)$; d'où $\mathcal{L}(\neg a)(i+1) = \complement\mathcal{L}(a)(i+1)$ d'après la règle 3. Il en résulte que $\mathcal{L}(F\neg a)(i) = \complement\mathcal{L}(Fa)(i)$ d'après la règle 4, d'où : $\mathcal{L}(F\neg a)(i) = \complement\complement\mathcal{L}(\neg Fa)(i)$ soit $\mathcal{L}(F\neg a)(i) = \mathcal{L}(\neg Fa)(i)$ (c.q.f.d.). Preuve de C14. $\mathcal{L}(F(a*b))(i) = \mathcal{L}(a*b)(i+1)$ d'après la règle 4 ; d'où $\mathcal{L}(F(a*b))(i) = \mathcal{L}(a)(i+1) \cap \mathcal{L}(b)(i+1)$ d'après la règle 1. Il en résulte que $\mathcal{L}(F(a*b))(i) = \mathcal{L}(Fa)(i) \cap \mathcal{L}(Fb)(i)$, soit $\mathcal{L}(F(a*b))(i) = \mathcal{L}(Fa*Fb)(i)$ (c.q.f.d.). Preuve de C15. Evident. Preuve de C16. $\mathcal{L}(Ga)(i) = \bigcup_{p=0}^{\infty} \mathcal{L}(a)(i+p)$ d'après la règle 5; d'où $\mathcal{L}(a)(i) \subset \mathcal{L}(Ga)(i)$, ce qui prouve la validité de la formule $a \Longrightarrow Ga$. Preuve de C17. $\mathcal{L}(Ga)(i) = \bigcup_{p=0}^{\infty} \mathcal{L}(a)(i+p)$; or $\mathcal{L}(Fa)(i) = \mathcal{L}(a)(i+1)$. Il en résulte que $\mathcal{L}(Fa)(i) \subset \mathcal{L}(Ga)(i)$, ce qui prouve la formule $Fa \Longrightarrow Ga$. Preuve de C18. On a simultanément d'une part $\mathcal{L}(GFa)(i) = \bigcup_{p=0}^{\infty} \mathcal{L}(Fa)(i+p) = \bigcup_{p=0}^{\infty} \mathcal{L}(a)(i+1+p)$, et d'autre part $\mathcal{L}(FGa)(i) = \mathcal{L}(Ga)(i+1) = \bigcup_{p=0}^{\infty} \mathcal{L}(a)(i+1+p)$. Il en résulte en particulier que la formule $FGa \Longrightarrow GFa$ est vérifiée. Preuve de C19. On a simultanément

d'une part : $\mathcal{L}(G(a+b))(i) = \bigcup_{p=0}^{\infty} \mathcal{L}(a+b)(i+p)$, et d'autre part $\mathcal{L}(Ga+Gb)(i) = \mathcal{L}(Ga)(i) \cup \mathcal{L}(Gb)(i) = \bigcup_{p=0}^{\infty} [\mathcal{L}(a)(i+p) \cup \mathcal{L}(b)(i+p)]$ $= \bigcup_{p=0}^{\infty} \mathcal{L}(a+b)(i+p)$. Il en résulte en particulier que la formule $G(a+b) \Longrightarrow Ga+Gb$ est vérifiée. Preuve de C20. Evident car $\forall i$ on a $\mathcal{L}(\Lambda)(i) = \phi$. Preuve de C21 et de C22. Evident d'après la définition 2 et les règles 4 et 5 (c.q.f.d.). Il en résulte que toutes les conditions auxquelles satisfait la relation $\Longrightarrow$ sont valides. On en déduit d'après la définition 3 que la relation $\Longrightarrow$ est valide (c.q.f.d.).

IV. PROPOSITIONS REMARQUABLES.

Proposition 1. $F(a+b) = Fa+Fb$

(immédiat à partir des conditions C13 et C14)

Proposition 2. $F(\Lambda)=\Lambda$ et $F(V)=V$

démonstration. $F(\Lambda)=F(a*\neg a)=F(a)*F(\neg a)=F(a)*\neg Fa=\Lambda$; $F(V)=F(\neg\Lambda)=\neg F(\Lambda)=\neg\Lambda=V$ (c.q.f.d.).

Cette proposition exprime que les descriptions vide et universelle (voir § II.4.1., chapitre quatrième, et § II.2., chapitre cinquième) - bornes inférieure et supérieure de Δ - n'évoluent pas. Nous en déduisons que l'évolution des descriptions est stable vis à vis du principe du système de description clos (voir § II.3.2., chapitre cinquième); en d'autres termes la condition C12, fixant ce principe, se généralise comme suit :

$$F(X_{i1}+\ldots+X_{in}) = X_{i1}+\ldots+X_{in} = V \quad \text{avec } n=|C_i|$$

Proposition 3. $\neg Ga \Longrightarrow F\neg a$; $\neg Ga \Longrightarrow G\neg a$

(De la condition C17 on tire $\neg Ga \Longrightarrow \neg Fa$, soit $\neg Ga \Longrightarrow F\neg a$; par transitivité de $\Longrightarrow$, on tire $\neg Ga \Longrightarrow G\neg a$).

A l'inverse de F, le connecteur G n'exprime plus l'unicité des états successifs ; en d'autres termes $G\neg a \Longrightarrow \neg Ga$ n'est jamais vérifiée.

Proposition 4. $GV=V$

(De la condition C20 et de la proposition précédente, on tire $V=\neg\Lambda=\neg G\Lambda$, soit $V \Longrightarrow \neg G\Lambda \Longrightarrow G\neg\Lambda \Longrightarrow GV$; il en résulte que $GV=V$ car V est la borne supérieure de Δ.)

Remarquons que l'on tire très facilement de cette proposition les deux égalités $\neg G\neg\Lambda=\Lambda$ et $\neg G\neg V=V$. Ceci montre que $\neg G\neg$ s'interprète comme la quantification universelle (interprétation analogue de $\exists$ et $\forall$ en logique classique : $\neg\exists\neg=\forall$) ; ce qui permet d'exprimer la permanence des descriptions quel que soit l'état des connaissances : si $\exists i\in\mathbb{N}$ tel que $x\in\mathscr{E}(\neg G\neg a)(i)$ alors $\forall j\ x\in\mathscr{E}(a)(j)$ et réciproquement (voir propositions 5, 11 et 13, ainsi que le tableau de la figure 6.4.).

Proposition 5. $\neg G\neg a \Longrightarrow a$;
$\neg G\neg a \Longrightarrow Fa$;
$\neg G\neg a \Longrightarrow Ga$

(immédiat à partir des conditions C16 et C17, compte tenu de la transitivité de la relation $\Longrightarrow$.)

Proposition 6. $G(a+b) = Ga+Gb$

démonstration. Des relations toujours vérifiées $a \Longrightarrow a$ et $b \Longrightarrow b$, on en déduit $a \Longrightarrow a+b$ et $b \Longrightarrow a+b$ (proposition 3, § II.1., chapitre cinquième). La condition C22 permet alors d'affirmer que les relations $Ga \Longrightarrow G(a+b)$ et $Gb \Longrightarrow G(a+b)$ sont toujours vérifiées ; il en résulte que la relation $Ga+Gb \Longrightarrow G(a+b)$ est toujours vérifiée (proposition 4, § II.1., chapitre cinquième). Par ailleurs la condition C19 montre que $G(a+b) \Longrightarrow Ga+Gb$. On en déduit que $G(a+b)=Ga+Gb$ (c.q.f.d.).

Proposition 7. $G(a*b) \Longrightarrow Ga*Gb$

démonstration. Des relations toujours vérifées $a \Longrightarrow a$ et $b \Longrightarrow b$, on en déduit $a*b \Longrightarrow a$ et $a*b \Longrightarrow b$ (proposition 1, § II.1., chapitre cinquième). La condition C22 permet alors d'affirmer que les relations $G(a*b) \Longrightarrow Ga$ et $G(a*b) \Longrightarrow Gb$ sont toujours vérifiées ; il en résulte que la relation $G(a*b) \Longrightarrow Ga*Gb$ est toujours vérifiée (proposition 2, § II.1., chapitre cinquième) (c.q.f.d.).
La réciproque n'est pas vérifiée ; on peut très facilement le montrer à partir de l'interprétation sémantique de la relation $Ga*Gb \Longrightarrow G(a*b)$ (voir § III.2.).

Proposition 8 . $\forall n \in \mathbb{N} \quad F^nGa \Longrightarrow Ga$

démonstration. de C17 et C15 on tire $FGa \Longrightarrow GGa \Longrightarrow Ga$, ce qui prouve la proposition pour n=1. Par application de la règle C21 à la relation $FGa \Longrightarrow Ga$, on obtient $F^2Ga \Longrightarrow FGa \Longrightarrow Ga$, ce qui prouve la proposition pour n=2 . Moyennant l'hypothèse d'induction, on démontre facilement que $\forall n \in \mathbb{N} \quad F^nGa \Longrightarrow Ga$ (c.q.f.d.).

Proposition 9 . $\forall n \in \mathbb{N} \quad F^na \Longrightarrow Ga$

démonstration. Cette proposition est vraie pour n=1 : $Fa \Longrightarrow Ga$ (condition C17). Par application de C21 et de la proposition 8 à la relation $Fa \Longrightarrow Ga$, on obtient : $F^2a \Longrightarrow FGa \Longrightarrow Ga$, ce qui prouve la proposition pour n=2. Moyennant l'hypothèse d'induction, on démontre facilement que $\forall n \in \mathbb{N} \quad F^na \Longrightarrow Ga$ (c.q.f.d.).

Proposition 10. $\forall n \in \mathbb{N} \quad F^nGa \Longrightarrow GFa$

démonstration. Cette proposition est vraie pour n=1 : $FGa \Longrightarrow GFa$ (condition C18) . Par application de C21,C17 et C15 à la relation $FGa \Longrightarrow GFa$, on obtient : $F^2Ga \Longrightarrow FGFa \Longrightarrow GGFa \Longrightarrow GFa$, ce qui prouve la proposition pour n=2. Moyennant l'hypothèse d'induction, on démontre que $\forall n \in \mathbb{N} \quad F^nGa \Longrightarrow GFa$ (c.q.f.d.).

Proposition 11. $\forall n \in \mathbb{N} \quad \neg G \neg a \Longrightarrow F^na$

démonstration. Cette proposition est vraie pour n=1 d'après la proposition 5. Par application de C21,de C13 et de la définition de la négation à $\neg G \neg a \Longrightarrow Fa$, on obtient : $\neg FG\neg a \Longrightarrow F^2a$ soit $\neg F^2a \Longrightarrow FG\neg a \Longrightarrow G\neg a$; d'où $\neg G \neg a \Longrightarrow F^2a$, ce qui prouve la proposition pour n=2. Moyennant l'hypothèse d'induction, on montre facilement que $\forall n \in \mathbb{N} \quad \neg G \neg a \Longrightarrow F^na$, (c.q.f.d.).

Proposition 12. $\forall n \in \mathbb{N} \quad GF^na \Longrightarrow Ga$

démonstration. Par application de C22 et de C15 à la proposition 9 , on obtient : $GF^na \Longrightarrow GGa \Longrightarrow Ga$ (c.q.f.d.).

Proposition 13. $\forall n \in \mathbb{N} \quad \neg G \neg a \Longrightarrow GF^na$

démonstration. De la proposition précédente, on tire $GF^n\neg a \Longrightarrow G\neg a$, soit $\neg G\neg a \Longrightarrow \neg GF^n\neg a$ soit $\neg G\neg a \Longrightarrow G\neg F^n\neg a \Longrightarrow GF^na$ (proposition 3 et condition C13) (c.q.f.d.).

V. GRAMMAIRE DES DESCRIPTIONS.

La proposition 7 montre que les descriptions du type G(a*b) et G¬(a+b) ne sont pas égales respectivement à Ga*Gb et à G¬a*G¬b car cette proposition n'a pas de réciproque (i.e. la formule Ga*Gb ⟹ G(a*b) n'est pas valide). Aussi nous supposerons que les descriptions de ce type ne sont pas des formules du système formel S_Δ. En d'autres termes Ga est une description si et seulement si a est un terme descriptif affirmé ou nié, ou alors a est la forme $F^n b$, b étant un terme descriptif affirmé ou nié. Cette hypothèse, qui ne consiste à prendre en compte dans ARCHES qu'un sous ensemble des descriptions construites à partir de G, est fondée sur deux critères. D'une part, au niveau de la représentation des connaissances, nous ne visons à décrire formellement que le connecteur ET-PUIS dont la définition n'utilise que le connecteur F (voir § VII.). D'autre part, l'intérêt du connecteur G est de pouvoir démontrer dans ARCHES des théorèmes mettant en jeu *l'existence* d'éléments de description indépendamment de tel ou tel état particulier . L'exclusion des descriptions du type G(a*b) et G¬(a+b) permet ainsi de représenter toute description D par une *forme canonique* C exprimée sous la forme d'une addition de disjonctions telle que D=C (i.e. D ⟹ C et C ⟹ D sont toujours vérifiées).

La figure 6.3. donne les règles de réécriture qui définissent la grammaire qui permet d'engendrer toute description exprimée sous sa forme canonique. Dans cette représentation les descriptions *atomiques* et *existentielles* sont des descriptions qui ne contiennent ni le connecteur d'addition (*), ni le connecteur de disjonction (+).


```
<DESCRIPTION-CANONIQUE> ::= <ADDITION-DESCRIPTION>

 <ADDITION-DESCRIPTION> ::= <DISJONCTION-DESCRIPTION>*
                            <ADDITION-DESCRIPTION>|<DISJONCTION-DESCRIPTION>

<DISJONCTION-DESCRIPTION> ::= <DESCRIPTION-TERMINAL>|
                              <DESCRIPTION-TERMINAL>+
                              <DISJONCTION-DESCRIPTION>

 <DESCRIPTION-TERMINAL> ::= <DESCRIPTION-ATOMIQUE>|
                            <DESCRIPTION-EXISTENTIEL>

 <DESCRIPTION-ATOMIQUE> ::= <TERME-DESCRIPTIF>|
                            ¬<TERME-DESCRIPTIF>|
                            F<DESCRIPTION-ATOMIQUE>

<DESCRIPTION-EXISTENTIEL> ::= G<DESCRIPTION-ATOMIQUE>|
                              ¬G<DESCRIPTION-ATOMIQUE>

     <TERME-DESCRIPTIF> ::= <OPERATEUR>(<CLASSE>,<TUPLE>)
                <TUPLE> ::= <ELEMENT>|
                            ##(<ELEMENT>{,<ELEMENT>}^n)

              <ELEMENT> ::= $(<TRAIT>,<DESCRIPTION-LOCALE>)|
                            $(INS(<CONCEPT>,<INDIVIDU>),Λ)

   <DESCRIPTION-LOCALE> ::= <TERME-DESCRIPTIF>|Λ
                <TRAIT> ::= <CONCEPT>|
                            <INFORMATION-SANS-EXTENSION>
```

figure 6.3. - *Grammaire des descriptions.*

VI. METHODE DE RESOLUTION DE LA FORMULE $H \Longrightarrow C$.

Compte-tenu de la nouvelle extension du système formel S_Δ, nous cherchons à valider où invalider toute formule $H \Longrightarrow C$ de la forme :

$$H_1 * H_2 * \ldots * H_i * \ldots * H_n \Longrightarrow C_1 * C_2 \ldots * C_j * \ldots * C_m$$

dans laquelle H_i et C_j sont des disjonctions de descriptions atomiques ou existentielles (voir § V.).
Dans ces conditions pour toute formule de ce type, on peut construire l'arbre AEO correspondant (voir § II.5., chapitre cinquième). Pour décider si la formule $H \Longrightarrow C$ est vérifiée ou non, il nous faut alors intégrer dans la procédure de validation VALID(a_{jo},t) les conditions de validation inhérentes à l'extension de la structure de S_Δ. Ces conditions de validation reposent sur les propriétés formelles de F et G ainsi que sur les propositions qui ont été démontrées à leur propos.
Le tableau de la figure 6.4. détermine les conditions de validation de toutes les formules basiques du type $\alpha \Longrightarrow \beta$ dans lesquelles α et β sont des descriptions atomiques et/ou existentielles. Dans toute case d'adresse [i,j] de ce tableau, on trouve la condition de validation de la formule $\alpha_i \Longrightarrow \beta_i$ correspondante, ainsi que les propriétés et/ou propositions qui la fondent. Les cases barrées indiquent que les relations correspondantes ne sont pas valides (car non démontrables à partir des conditions associées à F et G).
La procédure EXPLORE(α, β) permet ainsi, à partir de l'examen itéré de ce tableau, d'évaluer dans la procédure VALID(a_{jo},t) la validation de la formule $a_{jo} \Longrightarrow v(\alpha)$ (voir § II.5.5., chapitre cinquième). Il est évident qu'EXPLORE(α, β) est une procédure finie.

Définition du nouvel algorithme VALID.

```
       | α:=a_jo
DEBUT  | si      EXPLORE(a_jo, v(α))=SUCCES
       | alors   SUCCES ; FIN
       | sinon   α:=v(α)
       |         si      α=t
       |         alors   ECHEC ; FIN
       |         sinon   aller-à DEBUT
       |         fsi
       | fsi
```

Il en résulte que la procédure de décision pour la formule $H \Longrightarrow C$ est représentée par l'algorithme $\mathcal{A}_E$(H,C) dans lequel la validation de l'arbre de validation $A\nu$(i) est déterminée par la nouvelle procédure VALID définie ci-dessus. Le système formel S_Δ, ainsi étendu, est donc complet vis à vis de la relation de déduction $\Longrightarrow$ (voir § II.5.6., chapitre cinquième).

α_i \ β_i	b	$\neg b$	$F^p b$	$F^p \neg b$	Gb	$G\neg b$	$GF^p b$	$GF^p \neg b$	$\neg Gb$	$\neg G\neg b$	$\neg GF^p b$	$\neg GF^p \neg b$
a	$\mathcal{A}_R(a,b)$	§ II.5. chapitre cinquième			$Ga \Rightarrow Gb$ (C16)	$Ga \Rightarrow G\neg b$ (C16)						
$\neg a$	§ II.5. chapitre cinquième	§ II.5. chapitre cinquième			$G\neg a \Rightarrow Gb$ (C16)	$G\neg a \Rightarrow G\neg b$ (C16)						
$F^n a$			$a \Rightarrow b$ $n=p$ (C21)	$a \Rightarrow b$ $n=p$ (C21)	$Ga \Rightarrow Gb$ (P 9)	$Ga \Rightarrow G\neg b$ (P 9)	$F^n a \Rightarrow F^p b$ (C16,C22)	$F^n a \Rightarrow F^p \neg b$ (C16,C22)				
$F^n \neg a$			$\neg a \Rightarrow b$ $n=p$ (C21)	$\neg a \Rightarrow \neg b$ $n=p$ (C21)	$G\neg a \Rightarrow Gb$ (P 9)	$G\neg a \Rightarrow G\neg b$ (P 9)	$F^n_\neg a \Rightarrow F^p b$ (C16,C22)	$F^n_\neg a \Rightarrow F^p_\neg b$ (C16,C22)				
Ga					$a \Rightarrow b$ (C22)	$a \Rightarrow \neg b$ (C22)						
$GF^n a$					$Ga \Rightarrow Gb$ (P12,C22)	$Ga \Rightarrow G\neg b$ (P12,C22)	$F^n a \Rightarrow F^p b$ (C22)	$F^n a \Rightarrow F^p \neg b$ (C22)				
$GF^n \neg a$					$G\neg a \Rightarrow Gb$ (P12,C22)	$G\neg a \Rightarrow G\neg b$ (P12,C22)	$F^n_\neg a \Rightarrow F^p b$ (C22)	$F^n_\neg a \Rightarrow F^p_\neg b$ (C22)				
$\neg Ga$	$G\neg b \Rightarrow Ga$ (déf.*de $\neg$, C16)	$Gb \Rightarrow Ga$ (déf. de $\neg$, C16)	$G\neg b \Rightarrow Ga$ (déf. de $\neg$, P 9)	$Gb \Rightarrow Ga$ (déf. de $\neg$, P 9)	$G\neg a \Rightarrow Gb$ (P3)	$G\neg a \Rightarrow G\neg b$ (P3)	$G\neg b \Rightarrow G a$ (déf.de$\neg$, P3, P12)	$Gb \Rightarrow Ga$ (déf.de$\neg$, P3, P12)	$Gb \Rightarrow Ga$ (déf. de $\neg$)	$G\neg b \Rightarrow Ga$ (déf. de $\neg$)	$Gb \Rightarrow Ga$ (déf.de$\neg$, P12)	$G\neg b \Rightarrow Ga$ déf.de$\neg$, P12)
$\neg G\neg a$	$a \Rightarrow b$ (P5)	$a \Rightarrow \neg b$ (P5)	$F^p_a \Rightarrow F^p b$ (P11)	$F^p_a \Rightarrow F^p \neg b$ (P11)	$Ga \Rightarrow Gb$ (P5)	$Ga \Rightarrow G\neg b$ (P5)	$F^p a \Rightarrow F^p b$ (P13)	$F^p a \Rightarrow F^p \neg b$ (P13)	$Gb \Rightarrow G\neg a$ (déf. de $\neg$)	$G\neg b \Rightarrow G\neg a$ (déf. de $\neg$)	$Gb \Rightarrow G\neg a$ (déf. de $\neg$, P12)	$G\neg b \Rightarrow G\neg a$ (déf. de , P12)
$\neg GF^n a$			$F^n a \Rightarrow F^p b$ (déf. de $\neg$, C16,C22)	$F^n_\neg a \Rightarrow F^p \neg b$ (déf. de $\neg$, C16,C22)	$G\neg a \Rightarrow Gb$ (P3,P12)	$G\neg a \Rightarrow G\neg b$ (P3,P12)	$F^n_\neg a \Rightarrow F^p b$ (P3,C22)	$F^n_\neg a \Rightarrow F^p_\neg b$ (P3,C22)			$F^p b \Rightarrow F^n a$ (déf. de ($\neg$,C22)	$F^p_\neg b \Rightarrow F^n a$ (déf. de ($\neg$,C22)
$\neg GF^n \neg a$			$F^n a \Rightarrow F^p b$ (déf. de $\neg$, C16,C22)	$F^n a \Rightarrow F^p \neg b$ (déf. de $\neg$, C16,C22)	$Ga \Rightarrow Gb$ (P3,P12)	$Ga \Rightarrow G\neg b$ (P3,P12)	$F^n a \Rightarrow F^p b$ (P3, C22)	$F^n a \Rightarrow F^p_\neg b$ (P3, C22)			$F^p b \Rightarrow F^n_\neg a$ (déf. de $\neg$, C22)	$F^p_\neg b \Rightarrow F^n_\neg a$ (déf. de $\neg$,C22)

figure 6.4. - Conditions de validation de la formule $\alpha_i \Rightarrow \beta_i$.

VII. REPRESENTATION DU ET DE SUCCESSION : ETPUIS (OU $\circ$).

VII.1. Définition du connecteur $\circ$.

Le ET de succession - noté ETPUIS ou $\circ$ - a pour objet de suivre l'évolution des transformations, et plus généralement des changements d'états, que peut subir toute description a au cours des différents traitements où elle est mobilisée (voir § III.4., chapitre deuxième et § I.). Ainsi ce connecteur va permettre de représenter dans ARCHES des descriptions du type "PIERRE SE REVEILLE", "LE TRAIN QUITTE LA GARE", "JEAN PART DE CHEZ LUI ET VA A SON BUREAU", etc.

On dira que la description a évolué de manière immédiate vers la description b si et seulement si la condition suivante est vérifiée : $\forall i \in \mathbb{N}$ si $x \in \mathcal{G}(a)(i)$ alors $x \in \mathcal{G}(b)(i+1)$ tel que $\mathcal{G}(a)(i) \subset \mathcal{G}(b)(i+1)$ (voir § III.).

Par ailleurs nous supposons que toute description ne peut pas se transformer en elle-même, ce qui présuppose que $x \in \mathcal{G}(\neg a)(i+1)$ (voir figure 6.5.). Cette hypothèse exprime que toute description qui se transforme en elle-même n'évolue pas ! (voir des hypothèses analogues dans {1} pour la représentation sous-jacente de la conjonction THEN).

Plus précisément, le connecteur $\circ$ est une loi de composition interne partout définie sur Δ ; la règle ci-après le définit formellement :

$$a \circ b = a * F(\neg a * b)$$

Il est aisé de vérifier que la définition formelle du connecteur $\circ$ à partir des connecteurs $*$ et F est adéquate aux hypothèses ci-dessus qui fixent les caractéristiques sémantiques de l'évolution des descriptions (figure 6.5.).

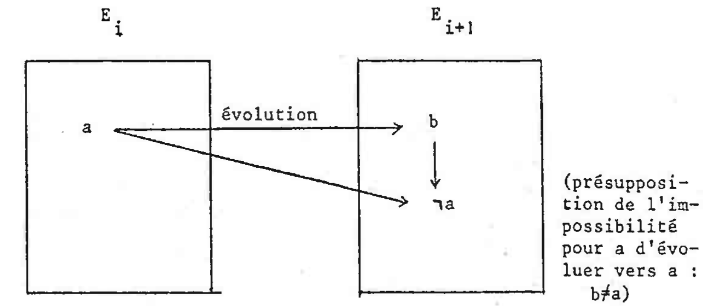


figure 6.5. - Interprétation sémantique de ETPUIS.

VII.2. Quelques propriétés remarquables du connecteur $\circ$.

Proposition 14. le connecteur $\circ$ n'est pas commutatif : $a \circ b \neq b \circ a$

Proposition 15. le connecteur $\circ$ n'est pas associatif : $(a \circ b) \circ c \neq a \circ (b \circ c)$

(on vérifie très facilement ces deux propriétés à partir de la définition de $\circ$.)

Dans ce qui suit, les descriptions mettant en jeu le connecteur $\circ$ sont représentées sans parenthèse ; nous supposons qu'elles sont équivalentes aux descriptions totalement parenthésées, dans lesquelles a été utilisé le parenthésage droite-gauche.

$$a \circ b \circ c \equiv a \circ (b \circ c) \qquad \text{(figure 6.6.)}$$

Proposition 16. le connecteur $\circ$ est distributif à gauche par rapport aux connecteurs $*$ et $+$:

$$\begin{cases} a \circ (b * c) = (a \circ b) * (a \circ c) \\ a \circ (b + c) = (a \circ b) + (a \circ c) \end{cases}$$

démonstration. $a \circ (b * c) = a * F(\neg a * b * c)$
$= a * F(\neg a * b) * a * F(\neg a * c)$
$= a \circ b * a \circ c$

La même démarche permet de prouver que $a\circ(b+c)=a\circ b+a\circ c$ (c.q.f.d.).

Proposition 17. $a\circ a=\Lambda$ (le connecteur $\circ$ n'est pas idempotent)

$(a\circ a=a*F(\neg a*a)=a*F\Lambda=a*\Lambda=\Lambda.)$

Proposition 18. $a\circ\Lambda=\Lambda\circ a=\Lambda$ (élément nul)

Proposition 19. Une description permanente ne peut pas évoluer :

$$\neg G\neg a\circ b=\Lambda$$

démonstration.
$$\begin{aligned}\neg G\neg a\circ b &= \neg G\neg a*F(G\neg a*b)\\ &= \neg G\neg a*FG\neg a*Fb && \text{(condition C14)}\\ &= \neg G\neg a*FG\neg a*G\neg a*Fb && \text{(proposition 8)}\\ &= \Lambda \text{ (c.q.f.d.).}\end{aligned}$$

Proposition 20. $\neg G\neg a\circ a=a\circ\neg G\neg a=\Lambda.$

(immédiat à partir de la proposition précédente et de la condition C16).

Proposition 21. $a\circ\neg a=a\circ V=a*F\neg a$

(immédiat à partir de la définition du connecteur $\circ$.)

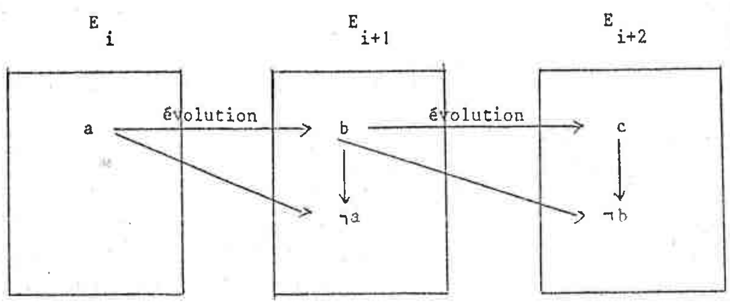


figure 6.6. - Modalités d'évolution de $a\circ b\circ c$.

VII.3. Exemples.

L'énoncé "PIERRE SE REVEILLE" est représenté par la structure schématisée par la figure 6.7. Il est paraphrasé de la manière suivante :

(((ACTIVITE ISA SOMMEIL)ETPUIS(ACTIVITE ISA EVEIL))DE PIERRE)

La figure 6.8. est la représentation de cette structure dans laquelle le connecteur ETPUIS est interprété à l'aide de * et de F.

La figure 6.9. représente l'énoncé "LE TRAIN QUITTE LA GARE". Ce dernier est paraphrasé de la manière suivante :

```
(((LOCALISATION IN GARE)ETPUIS((LOCALISATION OUT GARE)ETA
                                 (MOUVEMENT ISA DEPLACEMENT)))
                                  DU TRAIN)
```

La figure 6.10. représente l'énoncé "JEAN PART DE LA MAISON ET VA A SON BUREAU".

Enfin l'énoncé "l'AVION A105 DECOLLE DE LA PISTE P1" peut être représenté par la structure ci-après :

```
AVION(A105,∘(ISA(MOUVEMENT,$(ROULEMENT,
                    ON(LOCALISATION,INS(PISTE P1)))),
        ∘(ISA(MOUVEMENT,$(ELEVATION,
                    IN(LOCALISATION,INS(C-AERIEN,01)))),
                    NIL)))
```

Cette structure exprime le changement d'état suivant : l'avion A105 roule sur la piste P1 (état initial), puis s'élève dans les airs en prenant le couloir aérien 01 (état final). Il est évident que dans une telle structure les valeurs des traits associées à la classe LOCALISATION peuvent être indéterminées si elles ne sont pas connues à priori par l'expert qui analyse le décollage de l'avion A105. La figure 6.11. montre l'interprétation de cette structure à l'aide des connecteurs *, F et NEG (interprétation de $\neg$).

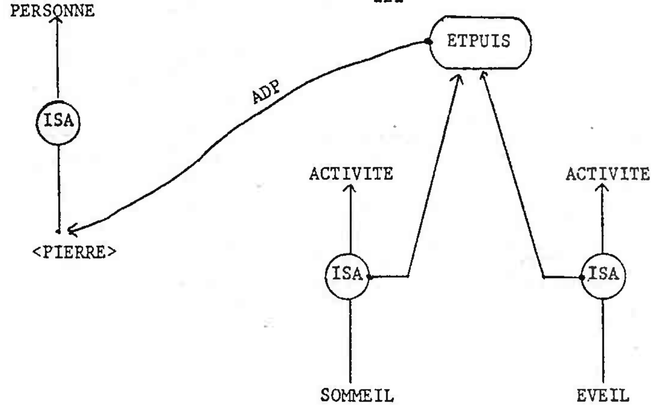


figure 6.7. - Structure représentant l'énoncé "PIERRE SE REVEILLE".

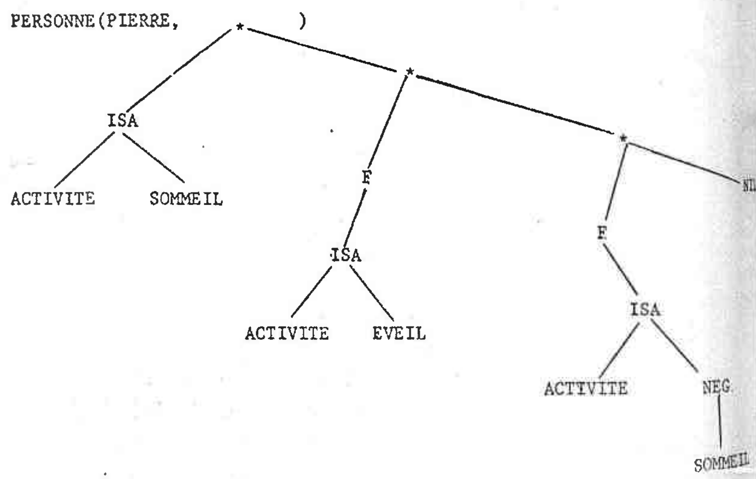


figure 6.8. - Interprétation de ETPUIS dans l'énoncé "PIERRE SE REVEILLE".

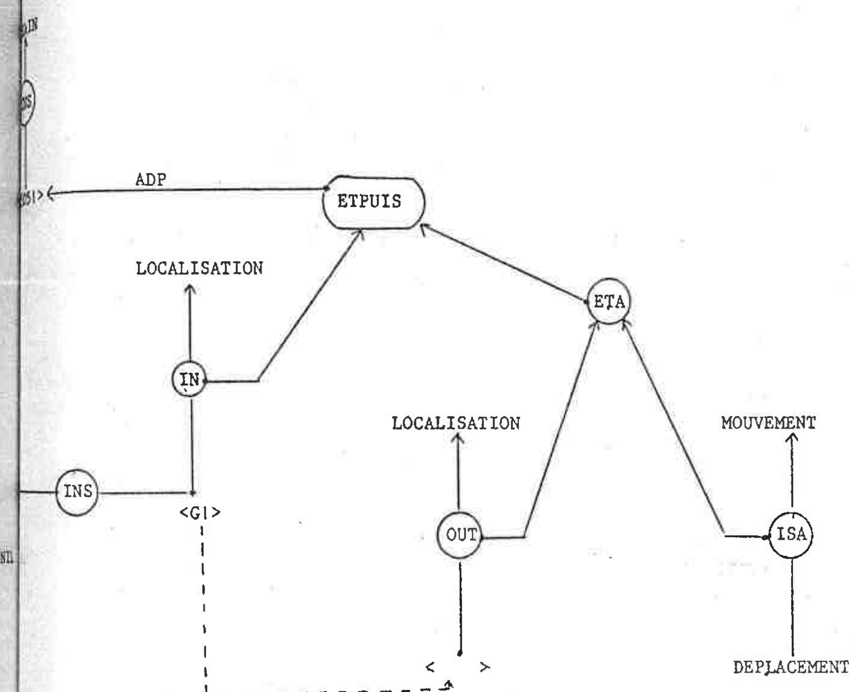


figure 6.9. - Représentation de l'énoncé "LE TRAIN N5051 QUITTE LA GARE G1".

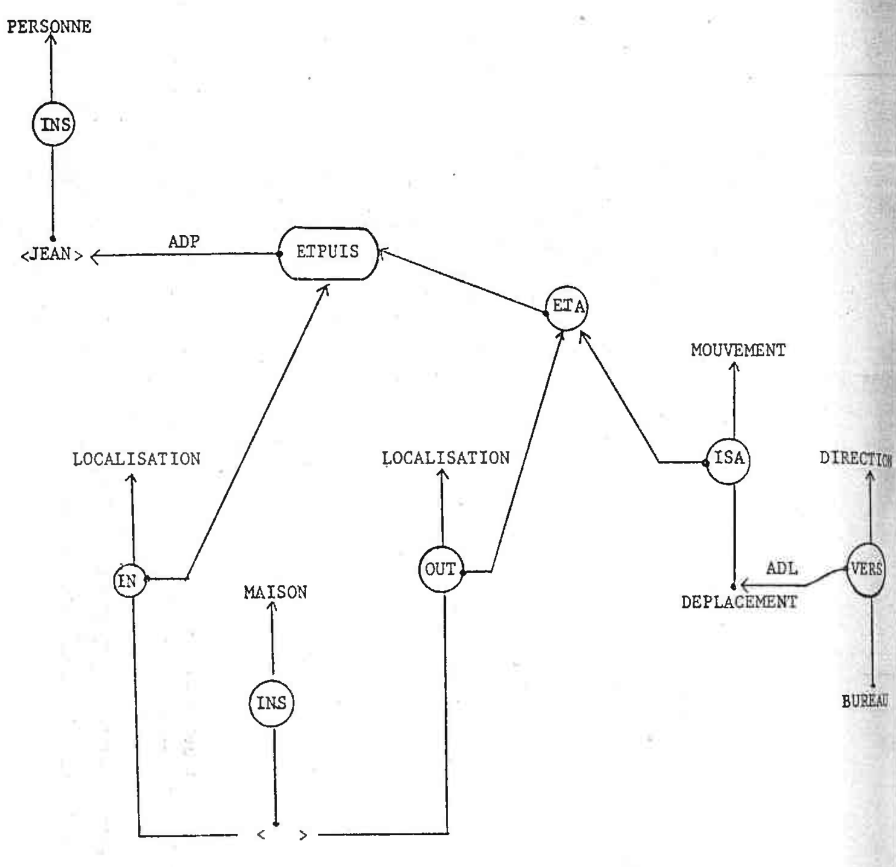


figure 6.10. - Structure représentant l'énoncé "JEAN PART DE LA MAISON ET VA A SON BUREAU".

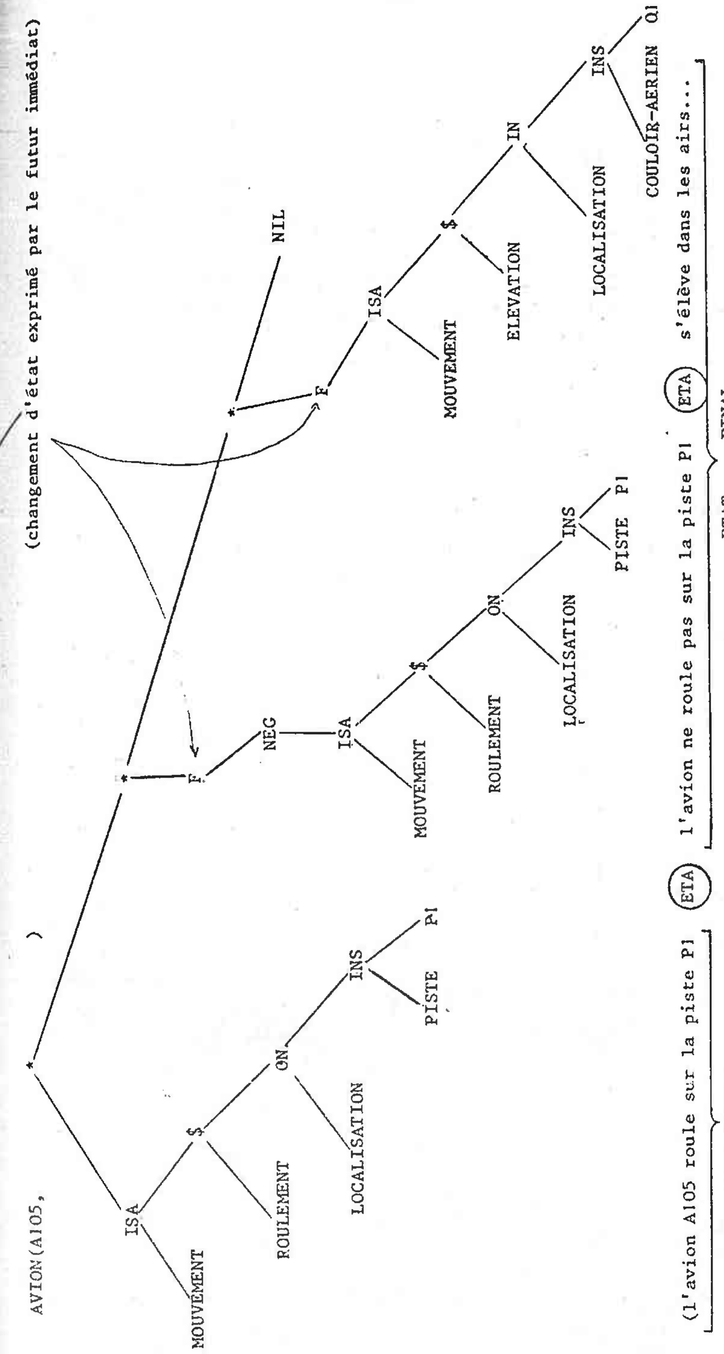


*figure 6.11. - Interprétation du connecteur ETPUIS à l'aide de *, F et NEG dans l'énoncé "L'AVION A105 DECOLLE DE LA PISTE P1".*

CHAPITRE SEPTIEME.

REPRESENTATION FORMELLE
DU SYSTEME SYMBOLIQUE ARCHES

-=-=-=-=-=-=-=-=-=-

I. COMPOSITION DU SYSTEME SYMBOLIQUE ARCHES.

Le système symbolique ARCHES est un *système formel spécialisé*, noté S_{ARCHES}, composé de quatre éléments qui déterminent des modalités particulières de *représentation* et de *dérivation* de connaissances (pour la distinction entre système symbolique et système formel voir § III.1., chapitre premier) :

$$S_{ARCHES} = \{\mathcal{P}, \mathcal{F}, \mathcal{F}_T, \longmapsto\}$$

α) $\mathcal{P}$ est l'ensemble de base du système symbolique ARCHES dont les éléments, appelés *éléments primitifs*, sont déterminés à partir du système formel S_Δ (voir § III.1.).

β) $\mathcal{F}$ est l'ensemble des *formules* - ou *structures* (voir § II., chapitre troisième) - construites à partir des éléments de $\mathcal{P}$ (voir § III.2.). Pour toute application sous ARCHES, le langage des formules permet de représenter un ensemble déterminé de faits. Chaque fait est véhiculé par une formule à deux places d'arguments donnant la description d'un individu : le premier argument est une variable ou une constante représentant un individu, le deuxième argument désigne sa description que nous considérons pour des raisons opératoires comme le résultat d'une certaine fonction de description (voir § II.).

γ) $\mathcal{F}_T$, inclus dans $\mathcal{F}$, définit l'ensemble des *thèses* du système symbolique ARCHES.

δ) $\longmapsto$ est la *relation d'inférence* qui détermine l'activité inférentielle du système symbolique ARCHES. Définie sur l'ensemble $\mathcal{F}$, elle permet de représenter les formes élémentaires de raisonnement au moyen de *règles d'inférence* particulières. Le système ARCHES est ainsi doté de deux modes de raisonnement : le raisonnement déductif et le raisonnement analogique (voir chapitre huitième et chapitre neuvième). La définition des règles d'inférence utilise d'une part les propriétés formelles de l'organisation des descriptions (i.e. utilisation des relations

SET et $\Longrightarrow$, voir chapitre troisième et chapitre quatrième), et d'autre part des caractéristiques spécifiques à chaque application (i.e. utilisation de prédicats utilitaires pour exprimer des relations conditionnelles)(voir § IV.). Enfin les règles d'inférence permettent de déduire des théorèmes à partir de l'ensemble $\mathcal{F}_T$ des thèses. On dira que la séquence des formules produites par les règles d'inférence pour prouver tout théorème est une *démonstration* de ce théorème. Pour réaliser les démonstrations il est donc nécessaire de formaliser l'activité inférentielle de ARCHES au moyen d'un autre système formel. Compte tenu des propriétés formelles de ARCHES nous utiliserons la logique du premier ordre, considérée ici comme un méta-système permettant de manipuler le système symbolique ARCHES (voir § V.).

II. REPRESENTATION FONCTIONNELLE DES DESCRIPTIONS.

II.1. Fonctions élémentaires de description.

Les termes descriptifs sont représentés, pour des raisons opératoires, par les résultats de fonctions particulières appelées *fonctions élémentaires de description*. Ces fonctions sont construites récursivement à partir d'un symbole fonctionnel constant, noté "." (point) ; leur sémantique est déterminée par les classes qui jouent le rôle de paramètre fonctionnel. Par exemple les termes descriptifs non décrits, dont les opérateurs sont unaires, sont représentés par les valeurs de certaines fonctions ayant la forme générale suivante :

$$\lambda xy.(x,T\ ,\$(y,\Lambda))$$

dans laquelle la variable libre T est un paramètre qui prend ses valeurs dans l'ensemble des classes. Les variables liées x et y prennent leurs valeurs respectivement dans l'ensemble des opérateurs et l'ensemble des traits (voir § II.1. et § II.2., chapitre quatrième). Ainsi la fonction $\lambda xy.(x,\mathrm{PARTIE},\$(y,\Lambda))$ permet de caractériser tout individu X par la relation d'état PARTIE (voir § III.2., chapitre deuxième). Elle a par exemple pour valeur $\cdot(\mathrm{ISA},\mathrm{PARTIE},\$(\mathrm{TETE},\Lambda))$. Nous devons noter que les fonctions élémentaires de description permettent de considérer les opérateurs comme des variables et non comme des symboles fonctionnels d'ordre >1 ; leur définition permet ainsi de rester dans le cadre des hypothèses que nous nous sommes fixés (hypothèse 4, voir § IV., chapitre deuxième) car elle est en particulier compatible avec les règles d'unification du principe de résolution en logique du premier ordre,{115}.

Plus précisément, les fonctions élémentaires de description sont engendrées de manière récursive par les deux règles de réécriture suivantes :

$$\left| \begin{array}{l} U ::= \ \cdot(x,T,\#(\$(y_1,U),\ldots,\$(y_n,U))) \\ U ::= \Lambda \end{array} \right.$$

Elles sont définies sur l'ensemble formé de l'union de l'ensemble des opérateurs et de l'ensemble des traits (concepts, individus, informations sans extension).
Nous utiliserons la notation usuelle $\lambda x_1 \ldots x_n f(x_1,\ldots,x_n)$ pour désigner toute fonction élémentaire de description à n arguments.

Exemples. $\lambda xy.(x,\mathrm{PARTIE},\$(y,\Lambda))$
$\lambda xyz.(x,\mathrm{COULEUR},\$(y,\cdot(x,\mathrm{TEINTE},\$(z,\Lambda))))$
$\lambda xy.(x,\mathrm{LOCALISATION},\$(\mathrm{INS}(\mathrm{FOUR},y),\Lambda))$

Les résultats des fonctions du type $\lambda x_1 \ldots x_n f(x_1,\ldots,x_n)$, pour n'importe quel ensemble de valeurs $[a_1,\ldots,a_n]$ des variables, représentent les termes descriptifs appartenant à $\mathcal{L} \subset \Delta$ (voir § II.4.2., chapitre quatrième), et se notent par convention $f(a_1,\ldots,a_n)$: *L'opération d'évaluation* permet d'obtenir les expressions du type $f(a_1,\ldots,a_n)$ en substituant dans la fonction $\lambda x_1 \ldots x_n f(x_1,\ldots,x_n)$ les variables $x_1,\ldots,x_n$ par les valeurs correspondantes $a_1,\ldots,a_n$ ({95}, {113}).

II.2. Fonctions de description.

Les fonctions de description sont construites à partir des fonctions élémentaires de description et des connecteurs $*$, $+$, $\neg$, F, G considérés comme symboles fonctionnels. Elles obéissent aux règles (F) de formation ci-après :

F1 Toute fonction élémentaire de description est une fonction de description ;

F2 Si $\lambda x_1 \ldots x_n H(x_1,\ldots,x_n)$ est une fonction de description alors $\lambda x_1 \ldots x_n \neg H(x_1,\ldots,x_n)$, $\lambda x_1 \ldots x_n FH(x_1,\ldots,x_n)$ et $\lambda x_1 \ldots x_n GH(x_1,\ldots,x_n)$ sont des fonctions de description ;

F3 Si $\lambda x_1 \ldots x_n H(x_1,\ldots,x_n)$ et $\lambda y_1 \ldots y_p K(y_1,\ldots,y_p)$ sont des fonctions de description, x_i et y_j étant non nécessairement distinctes, alors $\lambda x_1 \ldots x_n y_1 \ldots y_p [H(x_1,\ldots,x_n) * K(y_1,\ldots,y_p)]$ et $\lambda x_1 \ldots x_n y_1 \ldots y_p [H(x_1,\ldots,x_n) + K(y_1,\ldots,y_p)]$ sont des fonctions de description ;

F4 Toute fonction de description est obtenue à partir des fonctions élémentaires de description en appliquant un certain nombre de fois les règles F2 et F3.

Les fonctions de description sont définies également sur l'ensemble formé de l'union de l'ensemble des opérateurs et de l'ensemble des traits. Comme pour les fonctions élémentaires de description, leurs valeurs sont obtenues par application à ces fonctions de l'opération d'évaluation. Elles représentent les descriptions appartenant à l'ensemble Δ.

II.3. Définition des D-Termes.

Soit V un ensemble dénombrable de variables, et C un ensemble fini de constantes (définissant les opérateurs et les traits) tels que $V \cap C = \phi$; désignons par $\mathcal{A}$ l'ensemble $V \cup C$.
Par ailleurs soit $\mathcal{S}_F$ l'ensemble composé des symboles fonctionnels suivants : les symboles $\neg$, F et G d'arité 1 ; les symboles ., \$, INS, $*$ et $+$ d'arité 2 ; et enfin le symbole $\#$ d'arité n (n>1).
Soit enfin $\mathcal{T}$ l'algèbre libre de termes construite sur $\mathcal{A}$ et $\mathcal{S}_F$.
Les termes sont définis récursivement par les deux règles de réécriture ci-après :

$$\left| \begin{array}{l} \langle \mathrm{TERME} \rangle ::= \langle \mathrm{VARIABLE} \rangle \,|\, \langle \mathrm{CONSTANTE} \rangle \\ \langle \mathrm{TERME} \rangle ::= \langle \mathrm{SYMBOLE\text{-}FONCTIONNEL\text{-}D\text{-}ARITE}\ n \rangle \underbrace{(\langle \mathrm{TERME} \rangle,\ldots,\langle \mathrm{TERME} \rangle)}_{n\ \text{fois}} \end{array} \right.$$

Si $\lambda x_1 \ldots x_n H(x_1,\ldots,x_n)$ est une fonction de description et si $[a_1,\ldots,a_n]$ est un ensemble de n variables et/ou constantes, alors l'expression $H(a_1,\ldots,a_n)$ est un terme par construction même de la fonction H. Désignons par $\mathcal{T}_D$ le plus petit ensemble contenant les éléments obtenus par application de l'opération d'évaluation à toutes les fonctions de description, et fermé par cette opération ; il en résulte que $\mathcal{T}_D \subset \mathcal{T}$.

Les éléments de $\mathcal{T}_D$, appelés *D-termes*, se répartissent en deux groupes : d'une part ceux qui ne contiennent pas de variable et qui représentent les descriptions appartenant à l'ensemble Δ ; et d'autre part ceux qui ont des variables et qui représentent des schémas de description (voir § III.1., chapitre neuvième). En particulier aux descriptions atomiques et existentielles correspondent les D-termes atomiques et existentiels (voir § V., chapitre sixième). L'ensemble $\mathcal{T}_D$ possède donc la même structure algébrique que Δ, structure qui est déterminée par la relation de déduction $\Longrightarrow$. En effet les conditions auxquelles cette relation satisfait s'étendent facilement aux D-termes qui contiennent des *variables* pourvu qu'elles soient *quantifiées universellement* (voir § IV.1., chapitre quatrième).

III. LE LANGAGE OBJET DU SYSTEME SYMBOLIQUE ARCHES.

Le langage objet du système symbolique ARCHES est composé de trois ensembles: l'ensemble $\mathcal{P}$ contenant les éléments primitifs, l'ensemble $\mathcal{F}$ des formules construites à partir des éléments primitifs, et enfin l'ensemble $\mathcal{F}_T$ inclus dans $\mathcal{F}$ et contenant les thèses du système.

III.1. Les éléments primitifs.

L'ensemble $\mathcal{P}$ est formé des éléments primitifs suivants :

α) *Les symboles de constantes et de variables.* Ils permettent de représenter et de manipuler les opérateurs et les traits (concepts, individus, informations sans extension) : soit V l'ensemble dénombrable des variables qui seront notées par les dernières lettres de l'alphabet, éventuellement indicées (x, y, z, x_1, y_1, z_1,...) et C l'ensemble fini des constantes qui seront représentées par des chaines de caractères ; comme $\mathcal{J}$ désigne l'ensemble des symboles d'individus on a $\mathcal{J} \subset C$ (voir § III.1., chapitre sixième) ;

β) *Les symboles de classes et les symboles fonctionnels ., INS et* \$. Ils permettent de construire de manière récursive les fonctions élémentaires de description qui ont pour argument les symboles de constantes ou de variables ;

γ) *Les symboles fonctionnels* $*$, $+$, $\neg$, *F et G.* Ils permettent de construire les fonctions de description ;

δ) *Les D-termes.* Ils sont construits par application de l'opération d'évaluation aux fonctions de description ; ils forment l'ensemble $\mathcal{T}_D$ inclus dans l'algèbre libre $\mathcal{T}$ engendrée à partir de (α), (β) et (γ) ;

ε) *Les symboles de concepts.* Représentés par des chaines de caractères, ils définissent des symboles de prédicats à deux places d'arguments.

III.2. Les formules.

définition 1. Si A est un symbole de concept, x une variable ou une constante d'individu et y un D-terme (représentant une description ou un schéma de description), alors $A(x,y)$ est une formule du système symbolique ARCHES.

Toute formule sera appelée *structure*.

définition 2. L'ensemble $\mathcal{F}$ des structures du système symbolique ARCHES est défini comme suit :

$$\mathcal{F}=\{A(x,y) \mid A=\text{symbole de concept}; x\in \mathcal{J}\cup V; y\in T_D\}$$

L'ensemble $\mathcal{F}$ détermine le langage des formules du sytème symbolique ARCHES.

III.3. Les thèses.

III.3.1. Définitions générales.

définition 3. On dira que la structure $A(a,D_a)$ est une *thèse* du système symbolique ARCHES si et seulement si a est un symbole d'individu, instance du concept A, qui est caractérisé par le D-terme D_a représentant une description déterminée appartenant à l'ensemble Δ.

Comme à chaque individu ne peut être associée qu'une seule description (voir § II.2.2., chapitre troisième), il en résulte qu'à *chaque individu a instance de A est associée une seule thèse $A(a,D_a)$.*
Par ailleurs par définition même de l'ensemble Δ_T, on a $D_a \in \Delta_T$ (voir § II.5., chapitre quatrième).

définition 4. L'ensemble $\mathcal{F}_T$ des thèses du système symbolique ARCHES est défini comme suit :

$$\mathcal{F}_T=\{A(a,D_a) \mid A=\text{symbole de concept}; a=\text{instance de } A\}$$

définition 5. On appellera *champ* l'ensemble S_A des thèses construites à partir du même concept A :

$$\mathcal{F}_T=\bigcup_A S_A$$

Les thèses sont enregistrées dans la base de connaissances correspondant au système symbolique ARCHES : *Elles décrivent en extension l'ensemble des individus appartenant à* $\mathcal{J}$.

Enfin rappelons que les champs sont organisés en *domaines* par l'intermédiaire de la relation SET (voir chapitre troisième).

III.3.2. Consistance des connaissances enregistrées.

Les relations implicites qui existent entre les thèses appartenant à l'ensemble $\mathcal{F}_T$, du fait de la structure algébrique de Δ_T (relation $\Longrightarrow$) et de l'organisation des champs en domaines (relation SET), ne doivent pas rendre *inconsistant* le système symbolique ARCHES. Pour ce faire les connaissances enregistrées - i.e. les thèses du système ARCHES - doivent vérifier les trois conditions ci-après qui expriment précisément la consistance de leur représentation.

Condition 1. Si $A(a,D_a)$ et $B(b,D_b)$ sont deux thèses appartenant à $\mathcal{F}_T$ alors aucune des deux formules $D_a \Longrightarrow D_b$ et $D_b \Longrightarrow D_a$ n'est vérifiée.

Condition 2. Si $A(x,D)$ et $A_1(I(A),D_1)$ sont deux thèses appartenant respectivement aux champs S_A et S_{A1} telles que la relation $SET(A,A_1)$ est vérifiée alors la formule $D*D_1 \Longrightarrow \Lambda$ ne doit jamais être vérifiée.

Condition 3. Si $A_1(I(A),D_1)$ et $A_2(I(A),D_2)$ sont deux thèses appartenant respectivement aux champs S_{A1} et S_{A2} telles que les relations $SET(A,A_1)$ et $SET(A,A_2)$ sont vérifiées alors la formule $D_1*D_2 \Longrightarrow \Lambda$ ne doit jamais être vérifiée.

La condition 1 indique que tous les individus ont des descriptions différentes ; en d'autres termes tous les individus enregistrés dans la base de connaissances associée au système ARCHES sont distincts. La condition 2 indique que si D_1 est la description dans le champ S_{A1} de l'individu I(A) représentant le concept A (voir § I.3.2., chapitre troisième) et si D est la description dans S_A d'un quelconque individu x alors la description $D*D_1$ doit être différente de Λ. Cette condition contribue à rendre valide la règle d'inférence inter-champs (voir § IV.2.2.,a) qui permet d'attribuer les "descriptions générales" des concepts aux individus qui les composent. Enfin la condition 3 exprime la consistance de l'organisation des champs appartenant à un même domaine (figure 7.1.).

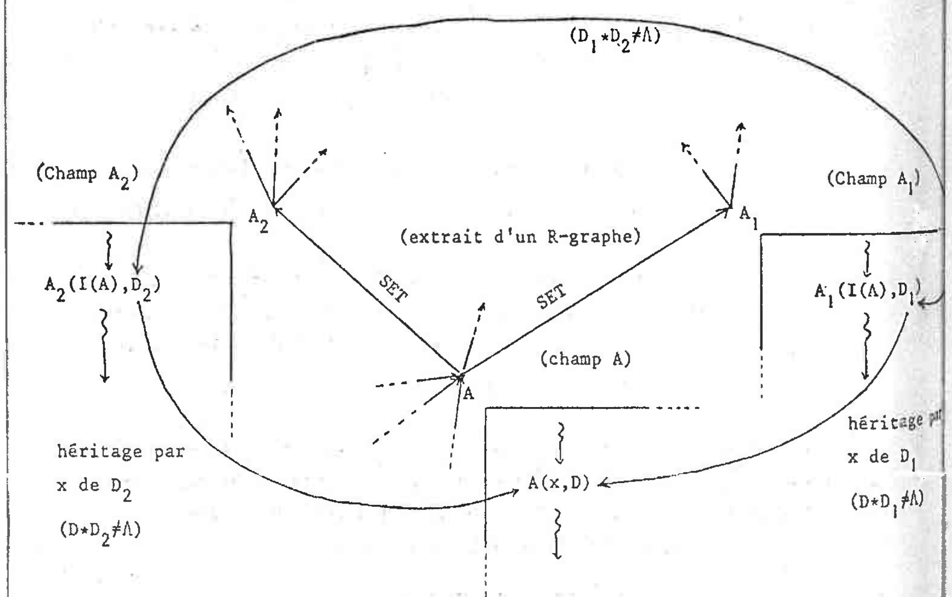


figure 7.1. - Consistance de l'organisation des champs à l'intérieur des domaines.

III.4. Interprétation du système symbolique ARCHES.

Ayant donné une interprétation du système formel S_Δ (voir § III., chapitre sixième), nous devons désormais définir l'interprétation du système symbolique ARCHES. Pour ce faire nous utilisons le même procédé qui a été mis en oeuvre pour la définition de l'interprétation des descriptions (voir § III.1., chapitre sixième ; point (γ)).

Tout d'abord par rapport au système formel S_Δ, le système symbolique ARCHES possède une autre catégorie de symboles, celle qui correspond aux variables (voir § III.1.).

Toute variable x appartenant à V est interprétée comme un trait appartenant à l'ensemble T (voir § III.1., chapitre sixième ; point (α), /3/) :

$$V \longrightarrow T$$

avec : $\mathcal{C}(x) \in T$

Toute structure A(x,y) appartenant à $\mathcal{F}$ est interprétée comme une relation d'appartenance définie sur $(D_C \times D_I) \times (\Delta \times \mathbb{N})$:

$$\lambda n\, \mathcal{C}(A(x,y))(n) = \mathcal{C}(INS(A,x)) \in \lambda n\, \mathcal{C}(y)(n)$$

Si on tient compte de l'interprétation des individus, il vient :

$$\lambda n\, \mathcal{C}(A(x,y))(n) = \mathcal{C}(x) \in \lambda n\, \mathcal{C}(y)(n) \text{ tel que } \mathcal{C}(x) \in \mathcal{C}(A)$$

On dira que la structure A(x,y) est vraie dans l'état E_i si et seulement si son interprétation est telle que l'individu x instance du concept A est caractérisé par la description y dans l'état E_i ; dans le cas contraire, on dira qu'elle est fausse.

En d'autres termes toute structure A(x,y) est interprétée comme un booléen permettant de lui attribuer une valeur de vérité (1=vrai ; 0=faux). Plus précisément, elle est interprétée comme une application de $\mathbb{N}$ dans B={0,1} pour tenir compte de l'évolution des individus :

$$\mathcal{F} \longrightarrow (\mathbb{N} \longrightarrow B)$$

avec : $\mathcal{C}(A(x,y))(i) \in B$

Cette application étant définie comme suit :

$$\begin{cases} \mathcal{C}(A(x,y))(i)=1 & \Leftrightarrow \quad \mathcal{C}(INS(A,x)) \in \mathcal{C}(y)(i) \\ \mathcal{C}(A(x,y))(i)=0 & \Leftrightarrow \quad \mathcal{C}(INS(A,x)) \notin \mathcal{C}(y)(i) \end{cases}$$

définition 6. On appellera *Modèle* du système symbolique ARCHES toute interprétation dans laquelle toutes les thèses sont vraies quel que soit l'état E_i.

Proposition 1. Si $A(a,D_a)$ est une thèse alors dans tout modèle la formule $D_a \Longrightarrow \Lambda$ n'est pas valide.

démonstration. dans tout modèle on a d'après la définition 6 $\forall i \in \mathbb{N}$ $\mathcal{C}(A(a,D_a))(i)=1$; il en résulte que $\forall i\, \mathcal{C}(INS(A,a)) \in \mathcal{C}(D_a)(i)$, d'où $\forall i\, \mathcal{C}(D_a)(i) \neq \phi$ (1). Supposons que la formule $D_a \Longrightarrow \Lambda$ soit valide ; on a donc $\forall i\, \mathcal{C}(D_a)(i) \subset \mathcal{C}(\Lambda)(i)$ (voir § III.2., chapitre sixième ; définition 2). Or $\mathcal{C}(\Lambda)(i)=\phi$; d'où $\mathcal{C}(D_a)(i)=\phi$, ce qui est contradictoire avec l'énoncé (1). La formule $D_a \Longrightarrow \Lambda$ n'est donc pas valide (c.q.f.d.)

IV. L'ACTIVITE INFERENTIELLE DU SYSTEME SYMBOLIQUE ARCHES.

IV.1. La relation d'inférence $\longmapsto$.

Désignons pas $\mathcal{F}^*$ l'ensemble formé de l'union des puissances successives de l'ensemble $\mathcal{F}$:

$$\mathcal{F} = \mathcal{F} \cup \mathcal{F}^2 \cup \ldots \cup \mathcal{F}^i \cup \ldots$$

Les éléments de $\mathcal{F}^*$ seront notés par les lettres grecques majuscules telles que ϕ, ψ, etc. ; chaque élément représentant une suite ordonnée de structures de la forme $<F_1,\ldots,F_j,\ldots,F_n>$ telles que $\forall j\ F_j \in \mathcal{F}$.

définition 7. On appelle *relation d'inférence* du système symbolique ARCHES une relation multi-aire, notée $\longmapsto$, définie sur $\mathcal{F}^* \times \mathcal{F}$ qui satisfait aux trois conditions ci-après :

(1) elle est réflexive :

$$F \longmapsto F \qquad \forall F \in \mathcal{F}$$

(2) elle est transitive :

si $\phi \longmapsto F_1$ et $F_1 \longmapsto F_2$
alors $\phi \longmapsto F_2$

(3) elle est monotone :

α) si $\phi \longmapsto F_1$ alors $\phi, F_2 \longmapsto F_1$
β) si $\phi, F_1, F_2, \psi \longmapsto F$ alors
$\phi, F_2, F_1, \psi \longmapsto F$

L'évaluation dans ARCHES de la validité des formules de la forme $\phi \longmapsto F$ est totalement déterminée par les règles d'inférence (voir § IV.2.).

définition 8. Si ϕ est une suite finie de structures appartenant à $\mathcal{F}^*$ et F une structure quelconque appartenant à $\mathcal{F}$ et si la formule $\phi \longmapsto F$ est vérifiée, alors on dira que F est *déductible* de ϕ dans le système ARCHES.

définition 9. On dira que la formule $\phi \longmapsto F$ est valide si toutes les fois que les structures qui composent ϕ sont vraies dans un modèle alors la structure F est également vraie dans le même modèle.

définition 10. On dira que la relation d'inférence $\longmapsto$ est valide si pour tout modèle les formules qui composent cette relation sont valides.

IV.2. Les règles d'inférence.

IV.2.1. Définition générale.

Les règles d'inférence du système ARCHES définissent des schémas de dérivation qui permettent d'affirmer qu'une relation d'inférence est vérifiée si certaines conditions (métalinguistiques) préalables sont reconnues valables. Ces conditions sont exprimées par des prédicats dont la définition dépend non seulement de la structure de l'ensemble Δ et de l'organisation des champs en domaines, mais aussi des applications réelles. Nous désignerons par C_1, $C_2, \ldots, C_i, \ldots$ de telles conditions.

Plus précisément les règles d'inférence du système ARCHES sont de la forme générale :

$$\frac{C_1;C_2;\ldots;C_p}{\phi \longmapsto F} \qquad (1)$$

Le schéma (1) s'interprète comme suit :

> *SI les conditions C_1, $C_2, \ldots,$ et C_p sont simultanément vérifiées, ALORS la formule $\phi \longmapsto F$ est également vérifiée.*

Nous définissons dans ARCHES deux types de règles d'inférence selon la nature des conditions qui déterminent leurs prémisses : les *règles d'inférence structurales* (voir § IV.2.2.) qui dépendent de l'architecture générale du système ARCHES et les *règles d'inférence pragmatiques* dont la définition effective dépend des applications susceptibles d'être réalisées sous ARCHES.

IV.2.2. Les règles d'inférence structurales.

Les règles d'inférence structurales sont fondées sur la structure et l'organisation des descriptions qui peuvent être précisément exprimées à l'aide de conditions métalinguistiques spécifiques. Nous distinguons deux catégories de règles d'inférence structurales : la première permet de mettre en oeuvre dans ARCHES le *raisonnement déductif* ; la seconde permet de mettre en oeuvre le *raisonnement analogique*.

a) Mise en oeuvre du raisonnement déductif.

Le raisonnement déductif est fondé sur les définitions et les propriétés de la relation de déduction $\Longrightarrow$ et du prédicat SET (voir § IV., chapitre quatrième et § I.3., chapitre troisième). Les deux modes d'organisation des connaissances, consécutifs à cette relation et à ce prédicat, permettent de déterminer les deux règles d'inférence basiques du système ARCHES qui expriment les formes élémentaires de ce mode de raisonnement : la *règle d'inférence intra-champ* et la *règle d'inférence inter-champs*.

définition 11. Etant donné un champ repéré par le concept A, SI la formule $D \Longrightarrow D'$ est vérifiée, ALORS la structure A(x,D') est déductible de la structure A(x,D) :

$$\frac{D \Longrightarrow D'}{A(x,D) \longmapsto A(x,D')}$$

Proposition 2. La règle d'inférence intra-champ est valide.

démonstration. Pour démontrer cette proposition il suffit de prouver que la formule $A(x,D) \longmapsto A(x,D')$ est valide quand la formule $D \Longrightarrow D'$ est valide (voir définition 9 et voir § III.2., chapitre sixième ;

définition 2). Si D $\Longrightarrow$ D' est valide alors $\forall i\ \mathcal{C}(D)(i) \subset \mathcal{C}(D')(i)$ (1).
Par ailleurs si A(x,D) est vraie dans un quelconque modèle, alors $\forall i$ $\mathcal{C}(INS(A,x)) \in \mathcal{C}(D)(i)$; il en résulte d'après (1) que $\mathcal{C}(INS(A,x)) \in \mathcal{C}(D')(i)$, d'où $\forall i\ \mathcal{C}(A(x,D'))(i)=1$. La structure A(x,D') étant vraie il en résulte que la formule A(x,D) $\longmapsto$ A(x,D') est valide ; donc la règle d'inférence intra-champ est valide (c.q.f.d.).

définition 12. Etant donné deux champs repérés par les concepts A et A_1 tels que la relation SET(A,A_1) est vérifiée, SI l'individu I(A) correspondant à A est décrit dans le champ A_1 par la description D' et SI x est un individu décrit par D dans le champ A, ALORS la structure A(x,D$*$D') est déductible de la suite <A(x,D),A_1(I(A),D')>:

$$\frac{SET(A,A_1)}{A(x,D),A_1(I(A),D') \longmapsto A(x,D*D')}$$

comme la formule D$*$D' $\Longrightarrow$ D' est toujours vérifiée (voir § IV.1., chapitre quatrième), on en déduit A(x,D$*$D') $\longmapsto$ A(x,D') d'après la règle intra-champ ; d'où A(x,D),A_1(I(A),D') $\longmapsto$ A(x,D') car la relation $\longmapsto$ est transitive. Il en résulte que *tout individu x appartenant à un concept A hérite de la description de ce concept.* Ainsi la règle d'inférence inter-champs permet d'affecter à tout individu des descriptions de plus en plus générales. Ces mécanismes de transfert des descriptions d'un champ vers un autre constituent le deuxième dispositif logique qui assure l'activité déductive dans ARCHES.

Proposition 3. La règle d'inférence inter-champs est valide.

démonstration. Pour prouver cette proposition on utilise la même démarche que celle mise en oeuvre pour la proposition 2. Si la structure A(x,D) est vraie dans un quelconque modèle il en résulte que $\forall i\ \mathcal{C}(x) \in \mathcal{C}(D)(i)$ tel que $\mathcal{C}(x) \in \mathcal{C}(A)$. De même si A_1(I(A),D') est vraie dans ce même modèle il en résulte que $\forall i\ \mathcal{C}(I(A)) \in \mathcal{C}(D')(i)$ tel que $\mathcal{C}(I(A)) \in \mathcal{C}(A_1)$ car la relation SET(A,A_1) est valide ; on en déduit que $\forall i\ \mathcal{C}(x) \in \mathcal{C}(D')(i)$ tel que $\mathcal{C}(x) \in \mathcal{C}(A)$. D'où $\forall i\ \mathcal{C}(x) \in \mathcal{C}(D)(i) \cap \mathcal{C}(D')(i)$, ce qui est équivalent à $\mathcal{C}(x) \in \mathcal{C}(D*D')(i)$. (Remarquons que d'après les conditions

2 et 3 la formule D$*$D' $\Longrightarrow$ Λ n'est jamais vérifiée, ce qui entraine nécessairement que $\mathcal{C}(D*D')(i) \neq \phi$, voir § III.3.2.). Il en résulte que la structure A(x,D$*$D') est également vraie dans le même modèle. La formule A(x,D),A_1(I(A),D') $\longmapsto$ A(x,D$*$D') est donc valide, ce qui entraine que la règle d'inférence inter-champs est valide (c.q.f.d.).

b) Mise en oeuvre du raisonnement analogique.

Le raisonnement analogique est fondé sur des règles métalinguistiques précises d'association de structures. Ces règles permettent d'exprimer aussi bien la "ressemblance" de structures que la "non-démonstrabilité". Elles sont à l'origine de la définition de la *règle d'inférence analogique* du système ARCHES dont les prémisses véhiculent précisément le contenu de ces règles. Leur définition repose sur un modèle analogique particulier que nous présenterons ultérieurement (voir § II., chapitre neuvième). Aussi pour des raisons de commodité et de compréhension nous définirons la règle d'inférence analogique dans le chapitre neuvième ; cette règle ayant naturellement la forme générale des règles d'inférence du système ARCHES (voir § IV.2.1.) : *Si certaines conditions analogiques sont vérifiées entre deux structures $P(x,\mathcal{D}(x))$ et $T(y,\mathcal{D}(y))$ telles que la formule $\mathcal{D}(x) \Longrightarrow \mathcal{E}_x$ est vérifiée, $\mathcal{E}_x$ "ressemblant" à $\mathcal{E}_y$, ALORS la formule $P(x,\mathcal{D}x)),T(y,\mathcal{D}(y)) \longmapsto T(y,\mathcal{E}_y)$ est vérifiée* (voir § III.2., chapitre neuvième).

IV.2.3. Les règles d'inférence pragmatiques.

Comme leur dénomination l'indique, les règles d'inférence pragmatiques sont fondées sur les propriétés qui décrivent de manière empirique l'organisation générale des univers de connaissances investigués. *Elles permettent ainsi de décrire les individus en intention en définissant les lois générales qui les organisent.* Les conditions métalinguistiques (quand elles existent), qui permettent d'inférer ces lois générales, sont représentées par des prédicats spécifiques, en général d'arité n (n>o), qu'on appellera *Prédicats utilitaires* ; il est évident que leur définition dépend des applications effectivement réalisées sous ARCHES (prédicats arithmétiques, de comparaison, etc.).

Si nous désignons ces prédicats par les symboles P_1, P_2,..., *la forme générale des règles d'inférence pragmatiques est la suivante :*

$$\frac{P_1 \; ; \; P_2 \; ; \ldots ; \; P_m}{F_1 \; , \; F_2 , \ldots , \; F_n \longmapsto F}$$

qui s'interprète comme suit :

La formule $F_1, F_2, \ldots, F_n \longmapsto F$ (i.e. une loi générale qui relie les descriptions des individus $x_1, x_2, \ldots, x_n$ et x représentées par les structures $F_1, \ldots, F_n$ et F) est vérifiée si les conditions utilitaires $P_1, P_2, \ldots, P_m$ sont simultanément vérifiées.

Il est possible que la liste des prédicats utilitaires soit vide ; dans ce cas la règle consiste à *affirmer de manière inconditionnelle* que la formule $F_1, F_2, \ldots, F_n \longmapsto F$ est toujours vérifiée.

Exemples.

Nous donnons, à titre d'illustration, deux règles d'inférence pragmatiques relatives à certaines connaissances associées au domaine VEHICULE, AUTOMOBILE étant l'un des champs qui composent ce domaine. La première règle utilise le prédicat utilitaire INF(y,z) (i.e. y est inférieur à z) ; la deuxième règle n'a pas de prémisse , c'est une affirmation inconditionnelle de la relation d'inférence.

Règle 1. SI y est inférieur à z ALORS pour toute automobile x ayant une capacité égale à z on déduit que sa capacité est supérieure à y :

$$\frac{\mathrm{INF(y,z)}}{\mathrm{AUTOMOBILE(x,EG(CAPACITE,z))} \longmapsto \mathrm{AUTOMOBILE\ (x, GT(CAPACITE,y))}}$$

Règle 2. Pour toute automobile x de type DIESEL on déduit que la nature de son carburant est le GAZOIL :

AUTOMOBILE(x,ISA(TYPE,DIESEL)) $\longmapsto$ AUTOMOBILE(x, ISA(CARBURANT,GAZOIL))

Remarque.

Les règles d'inférence pragmatiques jouent le même rôle que les règles qui composent les systèmes de production (voir § III.2.5., chapitre premier). Mais ces dernières sont des règles de réécriture alors que les règles d'inférence pragmatiques définissent des opérations de déduction élémentaires du fait de la définition du connecteur de négation $\neg$ (voir chapitre quatrième et chapitre cinquième). Elles sont donc plus complexes, et en particulier les structures qui les composent véhiculent des termes fonctionnels représentés par les D-termes qui appartiennent à l'ensemble $\mathcal{T}_D$ organisé par la relation de déduction $\Longrightarrow$ (voir § II.3.) (alors que le principe de résolution en logique du premier ordre suppose uniquement que les termes forment une algèbre libre {30, 115}).

IV.3. Notion de problèmes dans le système symbolique ARCHES.

Tout problème est défini par un énoncé de la forme $A(x,d)$ appartenant à l'ensemble $\mathcal{F}$ dans lequel x est une constante ou une variable, et d un D-terme sans variable représentant une description appartenant à l'ensemble Δ.

L'objet du système symbolique ARCHES est de résoudre tout problème à partir de ses thèses et au moyen des règles d'inférence. Si le problème ne possède pas de variable la solution est du type OUI/NON, c'est à dire l'individu x possède (ou ne possède pas) la description d ; par contre si x est une variable d'individu, il s'agit de trouver un (ou plusieurs) individu(s) appartenant au concept A caractérisé(s) par la description d. Nous donnerons dans le paragraphe suivant (voir § V.) une définition précise de la formalisation du processus de démonstration dans ARCHES.

V. LA LOGIQUE DU PREMIER ORDRE COMME LANGAGE DE MANIPULATION DU SYSTEME SYMBOLIQUE ARCHES.

V.1. Formalisation du processus de démonstration.

Pour réaliser une démonstration dans ARCHES nous devons utiliser au cours du processus de démonstration une règle de détachement du type modus-ponens. Or une telle règle n'existe pas dans le langage objet de ARCHES ; en conséquence pour formaliser ce processus nous devons manipuler le système symbolique ARCHES par un autre système - donc un méta-système - qui possède précisément une règle de détachement. Comme le langage objet du système ARCHES est du premier ordre (voir § III. ; et aussi § IV., chapitre deuxième), nous choisissons comme méta-système le calcul des prédicats qui possède naturellement une telle règle (modus-ponens) ; ce qui permettra par ailleurs l'utilisation d'outils existants comme le langage de programmation PROLOG et son démonstrateur fondé sur le principe de résolution de Robinson ({115}, {116}).

Dans ces conditions le calcul des prédicats doit contenir en particulier le langage objet du système symbolique ARCHES et les éléments qui formalisent sont activité inférentielle.
A ce sujet rappelons que si P est un symbole de prédicat à n places et si $t_1, t_2, \ldots, t_n$ sont n termes fonctionnels alors $P(t_1, t_2, \ldots, t_n)$ est une formule du calcul des prédicats. L'ensemble des termes fonctionnels détermine une algèbre libre qui contiendra en particulier les éléments de l'ensemble $\mathcal{T}$ et donc ceux de l'ensemble $\mathcal{T}_D$ (voir § II.3.). L'ensemble des formules définit le langage du calcul des prédicats. Il sera composé notamment de l'union des ensembles suivants : le langage des structures du système symbolique ARCHES (i.e. $\mathcal{F}$ et donc également $\mathcal{F}_T$; voir § III.2. et III.3.), l'ensemble des prédicats utilitaires intervenant dans la définition des règles d'inférence pragmatiques (voir § IV.2.3.), et enfin l'ensemble des relations intervenant dans la définition des règles d'inférence structurales ($\Longrightarrow$, SET, prédicats d'analogie).

Dans le calcul des prédicats la règle du modus-ponens est déterminée à partir de la relation de déduction $\vdash$ définie sur son langage des formules (voir en particulier {50}). Pour utiliser dans la formalisation du processus de démonstration ce modus-ponens, nous devons établir formellement les rapports qui existent entre les deux relations $\longmapsto$ et $\vdash$. Ces rapports doivent exprimer que les relations qui sont vraies dans ARCHES sont également vraies dans le calcul des prédicats, car les démonstrations sont formalisées et mises en oeuvre dans le calcul des prédicats. Comme par ailleurs la relation de déduction $\vdash$ est définie de la même manière que la relation d'inférence $\longmapsto$ (voir § IV.1. ; définition 7), et que $\mathcal{F}$ est inclus dans le langage des formules du calcul des prédicats, nous admettrons que *la relation d'inférence* $\longmapsto$ *est plus fine que la relation de déduction* $\vdash$.
Cette hypothèse conduit ainsi à définir dans le calcul des prédicats un schéma de dérivation qui permet de manipuler le système symbolique ARCHES, et de formaliser son activité inférentielle :

$$\frac{F_1, F_2, \ldots, F_n \longmapsto F}{F_1, F_2, \ldots, F_n \vdash F}$$

SI la formule $F_1, F_2, \ldots, F_n \longmapsto F$ est vérifiée dans le système symbolique ARCHES, ALORS l'énoncé de conséquence $F_1, F_2, \ldots, F_n \vdash F$ est vrai dans le calcul des prédicats.

Il en résulte que les éléments qui définissent le processus de résolution des problèmes dans ARCHES seront représentés dans le calcul des prédicats de la manière suivante :

α) *Représentation des thèses du système symbolique ARCHES.*

Toute thèse $A(a, D_a)$ du système symbolique ARCHES (i.e. appartenant à $\mathcal{F}_T$) sera représentée dans le calcul des prédicats par l'énoncé de conséquence $\vdash A(a, D_a)$.

L'énoncé de conséquence $\vdash A(a, D_a)$ indique que la structure $A(a, D_a)$ est affirmée inconditionnellement, quel que soit les hypothèses que l'on pourrait faire ; il signifie que $A(a, D_a)$ est toujours vraie (i.e. pour toutes les interprétations).

β) *Représentation des règles d'inférence du système symbolique ARCHES.*

Toute règle d'inférence du système symbolique ARCHES de la forme :

$$\frac{C_1 \; ; \; C_2 \; ; \ldots ; \; C_p}{\phi \longmapsto F}$$

sera représentée dans le calcul des prédicats par le schéma de dérivation ci-après :

$$\begin{array}{c} \vdash C_1 \\ \vdash C_2 \\ \text{==============} \\ \vdash C_p \\ \hline \phi \vdash F \end{array}$$

Comme nous sommes dans le cadre de la logique classique du premier ordre, ce schéma de dérivation peut être représenté par l'énoncé de conséquence ci-après (voir à ce sujet {123a}) :

$$C_1, C_2, \ldots, C_p, \phi \vdash F \qquad (1)$$

Cet énoncé affirme que si l'on admet tous les antécédents (i.e. les conditions métalinguistiques C_1, C_2,... et C_p et la suite de structures ϕ) alors le conséquent est également admis (i.e. la structure F). En d'autres termes, nous dirons que la relation (1) est *valide* si et seulement si toutes les fois que les C_i et les structures composant ϕ sont vraies alors la structure F est également vraie (dans la même interprétation).

γ) *Représentation des relations qui définissent l'architecture générale du système symbolique ARCHES.*

La relation de déduction $\Longrightarrow$ sera représentée par le prédicat $\vec{U}$ à deux places d'arguments dans le calcul des prédicats. Si l'énoncé de conséquence $\vdash \vec{U}(D,D')$ est vrai, D et D' étant deux D-termes, on dira que D et D' sont $\Longrightarrow$-unifiables (voir § II.2.2., chapitre huitième).

Les règles de substitution-réduction qui déterminent les propriétés sémantiques des classes seront représentées par le prédicat REGLE à deux places d'arguments. Si l'énoncé de conséquence $\vdash$REGLE(uv,w) est vrai dans le calcul des prédicats, alors la règle uv $\xrightarrow{\circ}$ w est admise dans le système ARCHES (voir § III.3.1., chapitre quatrième).

La relation SET qui organise les concepts entre eux donnera naissance dans le calcul des prédicats à un ensemble fini G d'énoncés de conséquence de la forme $\vdash$ SET(A_i,A_j) dans laquelle A_i et A_j sont deux symboles de concepts. Cet ensemble G définit les différents R-graphes organisant ARCHES (voir § III.3., chapitre troisième).

Enfin les prédicats qui évaluent la ressemblance dans la règle d'inférence analogique seront définis ultérieurement (voir § III.1., chapitre neuvième).

V.2. Représentation par des clauses de Horn du système symbolique ARCHES.

Nous nous proposons de définir les modalités de représentation du système symbolique ARCHES et de son architecture par des clauses de Horn. Ainsi le langage de représentation du système ARCHES - en abrégé *le langage ARCHES* - pourra bénéficier pour sa conception et son implémentation de toute la puissance du langage de programmation PROLOG dont il constituera la base {116}. Cette situation est tout à fait comparable à celle où des langages de programmation de type intelligence artificielle, comme PLANNER ou CONNIVER par exemple, ont été élaborés et développés à partir de LISP (voir à ce sujet {8}, {95}, {97}).

D'après l'hypothèse 4 sous-jacente à la conception du système ARCHES (voir § IV.4., chapitre deuxième) le théorème de la déduction naturelle permet de montrer que tout énoncé de conséquence de la forme $A_1,\ldots,A_i,\ldots,A_n \vdash B$ est équivalent à la clause de Horn $\{B,-A_i \mid 1 \leqslant i \leqslant n\}$, i.e. ont la même valeur de vérité dans la même interprétation. Rappelons que toute clause de Horn de la forme $\{B,-Ai \mid 1 \leqslant i \leqslant n\}$ est un ensemble de prédicats dans lequel la virgule est interprétée comme la disjonction logique, le signe "-" représentant la négation du calcul des prédicats. Conventionnellement les clauses de Horn seront représentées dans les termes du langage PROLOG :

$$B\text{-}A_1\text{-}\ldots\text{-}A_i\text{-}\ldots\text{-}A_n. \qquad (1)$$

L'expression (1) s'interprète comme suit : *La formule B est vraie si toutes les formules Ai sont vraies (dans la même interprétation).*

La formule B est appelée *Tête* de la clause, les A_i formant la *queue* de la clause. Si la queue est vide (i=0), alors on obtient une *clause unaire* qui affirme inconditionnellement B. Enfin la *clause vide*, notée $\square$, n'est formée d'aucune formule.
Si A est une formule, alors l'ensemble {A,-A} est appelée *paire complémentaire*. Nous devons remarquer à cet effet que toute clause qui contient une paire complémentaire est une tautologie.

Ces quelques considérations permettent d'établir les règles de représentation par des clauses de Horn du système symbolique ARCHES et de son architecture (figure 7.2.). Les graphes de résolution, les règles de substitution-réduction, les thèses et les règles d'inférence pragmatiques donnent naissance à quatre ensembles de clauses de Horn qu'on désignera respectivement par G, R, S et P. Ces ensembles dépendent essentiellement des applications qui sont effectivement réalisées sous ARCHES. Par contre l'ensemble des clauses de Horn correspondant aux règles d'inférence structurales est indépendant des applications ; il ne dépend que de l'architecture et du formalisme du système symbolique ARCHES.

définition 13. On appelle *base de connaissances* associée au système symbolique ARCHES l'ensemble des clauses de Horn formé de l'union des ensembles G, R, S et P.

Les ensembles G et R sont utilisés par les règles d'inférence structurales au cours du processus de démonstration ; l'ensemble S décrit en *extension* les individus et enfin l'ensemble P les décrit en *intention* en définissant à l'aide de clauses de Horn les lois générales des univers de connaissances étudiés.

La figure 7.3. donne un extrait d'une base de connaissances associées au domaine VEHICULE.

Ainsi le langage ARCHES permet de décrire et de programmer des classes de problèmes dans les termes de bases de connaissances ; et de les résoudre à partir de ces bases au moyen des règles d'inférence structurales. Ces règles d'inférence permettent de construire le démonstrateur associé au langage ARCHES. Ce démonstrateur est composé de deux parties. D'une part, il assure la mise en oeuvre du raisonnement déductif par

Nature des éléments		Système symbolique ARCHES	Clauses de Horn
Graphes de résolution (voir § III.3., chapitre troisième)		SET(A,B)	SET(A,B).
Règles de Substitution-Réduction (voir § III.3., chapitre quatrième)		$uv \xrightarrow{\circ} w \in S$	REGLE(uv,w).
Thèses (voir § III.3.)		$A(x,y) \in \mathfrak{F}_T$	A(x,y).
Règles d'inférence pragmatiques (voir § IV.2.3.)		$\dfrac{P_1;P_2;\dots;P_m}{F_1,F_2,\dots,F_n \longmapsto F}$	$F\text{-}F_1\text{-}\dots F_n\text{-}P_1\text{-}\dots\text{-}P_m$.
Règles d'inférence structurales	Règles intra-champ (voir § IV.2.2.a)	$\dfrac{y \Rightarrow z}{A(x,y) \longmapsto A(x,z)}$	$A(x,z)\text{-}A(x,y)\text{-}\vec{U}(y,z)$.
	Règles inter-champs (voir § IV.2.2.a)	$\dfrac{SET(A,B)}{A(x,y),B(I(A),z) \longmapsto A(x,y*z)}$	$A(x,y*z)\text{-}A(x,y)\text{-}B(I(A),z)\text{-}SET(A,B)$.
	Règles analogiques (voir § IV.2.2.b)	voir chapitre neuvième.	

figure 7.2. - Modalités de représentation du système symbolique ARCHES.

l'intermédiaire d'une règle de résolution de même nature que la règle de résolution de Robinson. Définie à partir des règles d'inférence intra-champ et inter-champs, cette règle de résolution utilise des modalités spécifiques d'unification des termes ("flèche-unification" exprimée à l'aide du prédicat $\vec{U}$; voir § II., chapitre huitième). Nous montrerons que ce mode de raisonnement est consistant et complet, et donnerons l'algorithme de construction de la partie correspondante du démonstrateur (voir chapitre huitième). D'autre part, il assure la mise en oeuvre du raisonnement analogique par l'intermédiaire de la règle d'inférence analogique fondée sur un modèle analogique particulier (voir § II., chapitre neuvième). Nous montrerons que ce mode de raisonnement est satisfiable, et donnerons l'algorithme de construction de la partie correspondante du démonstrateur (voir chapitre neuvième). Ceci montre que le langage ARCHES intègre dans sa conception à la fois la *sémantique descriptive* qui permet de décrire un problème à l'aide d'une base de connaissances et la *sémantique opératoire* qui permet de le résoudre grâce précisément à la représentation logique de ses éléments.
Dans ces conditions, comme nous l'avons déjà signalé au début de ce paragraphe, PROLOG constitue le noyau du langage ARCHES en ce sens que ce dernier bénéficie de son formalisme de représentation et de son démonstrateur fondé sur le principe de résolution de Robinson. Les éléments spécifiques du langage ARCHES sont pris en compte par des extensions qui sont greffées sur PROLOG : modalités d'enregistrement des informations dans la base de connaissances associée à ARCHES ; reconnaissance et analyse syntaxique des éléments composant cette base et contrôle de sa cohérence interne; représentation dans les termes de ce langage d'un problème ARCHES et décomposition automatique de ce problème en sous-problèmes ; génération automatique des règles d'inférence structurales ; programmation du démonstrateur ARCHES à partir de celui de PROLOG avec insertion de prédicats évaluables définissant la résolution de la relation $\Longrightarrow$ et les notions de ressemblance intervenant dans le modèle analogique, etc.


```
SET(HIPPOMOBILE ,VEHICULE).
SET(MOTEUR-AERIEN,VEHICULE).
SET(MOTEUR-A-REACTION,VEHICULE).
SET(MOTEUR-A-EXPLOSION,VEHICULE).
SET(CALECHE,HIPPOMOBILE).
SET(BATEAU-A-VOILE,MOTEUR-AERIEN).
SET(AVION,MOTEUR-A-REACTION).
SET(NAVIRE,MOTEUR-A-REACTION).
SET(NAVIRE,MOTEUR-A-EXPLOSION).
SET(AUTOMOBILE,MOTEUR-A-EXPLOSION).
SET(PETROLIER,NAVIRE).
SET(PAQUEBOT,NAVIRE).
------------------------------------
------------------------------------
------------------------------------
```

ensemble G

(Représentation d'un extrait du R-Graphe VEHICULE)

figure 7.3. - Suite page suivante.

```
AUTOMOBILE(V1,*(ISA(PROPRIETAIRE,INS(PERSONNE,PIERRE)),        Représentation     E
              *(ISA(MARQUE,R12),                               du champ           N
              *(ISA(COULEUR,$(BLEU,ISA(TEINTE,FONCE))),        AUTOMOBILE         S
              *(EG(CAPACITE,5),                                (extension)        E
              *(IN(LOCALISATION,INS(GARAGE,G)),NIL))))).                          M
--------------------------------------------------                                B
--------------------------------------------------                                L
                                                                                  E
--------------------------------------------------
--------------------------------------------------

AVION(A105,o(ISA(MOUVEMENT,$(ROULEMENT,ON(LOCALISATION,        Représentation
                             INS(PISTE,P1)))),                 du champ
          o(ISA(MOUVEMENT,$(ELEVATION,IN(LOCALISATION,         AVION
                             INS(C-AERIEN,O1)))),NIL)))        (extension)        S
--------------------------------------------------
--------------------------------------------------

--------------------------------------------------
--------------------------------------------------

AUTOMOBILE(x,GT(CAPACITE,y))-AUTOMOBILE(x,EG(CAPACITE,z))      description en     E
                             -INF(y,z).                        intention du       N
AUTOMOBILE(x,EG(POIDS,$(y,ISA(UNITE,KG))))-AUTOMOBILE(x,       champ              S
                             EG(POIDS,$(z,ISA(UNITE,T))))      AUTOMOBILE         E
                             -EGAL(z,1000Xy).                                     M
AUTOMOBILE(x,ISA(CARBURANT,GAZOIL))-AUTOMOBILE(x,                                 B
                                   ISA(TYPE,DIESEL)).                             L
--------------------------------------------------                                E
--------------------------------------------------
--------------------------------------------------                                P
```

figure 7.3. - Représentation de connaissances associées au domaine VEHICULE.

CHAPITRE HUITIEME.

RAISONNEMENT DEDUCTIF DANS ARCHES

-=-=-=-=-=-=-=-

I. INTRODUCTION.

Le raisonnement déductif dans le système symbolique ARCHES est mis en oeuvre par les règles d'inférence intra-champ et inter-champs. Nous nous proposons de formaliser ce mode de raisonnement dans le cadre du calcul des prédicats afin de pouvoir établir ses propriétés logiques et de construire le démonstrateur qui lui est associé.

Il est clair que ce démonstrateur peut être programmé dans les termes du langage PROLOG dans la mesure où nous avons donné une représentation en clauses de Horn des règles d'inférence structurales (voir § V.2., chapitre septième). Pour ce faire il faut en particulier représenter et définir par un ensemble de clauses de Horn le prédicat $\vec{U}(D,D')$ qui détermine les modalités de résolution de la relation $\Longrightarrow$. Dans ce cas, ce démonstrateur s'articule directement sur le démonstrateur PROLOG et utilise par conséquent la règle de résolution de Robinson. De plus il utilise nécessairement des prédicats évaluables du langage PROLOG qui permettent de manipuler dans le langage objet du calcul des prédicats des éléments de son métalangage (par exemple le prédicat évaluable "/" qui permet de représenter la négation, de manipuler certains aspects de la non-démonstrabilité, etc.).

Cependant nous présentons dans ce chapitre une *description algorithmique* du démonstrateur déductif associé au système symbolique ARCHES (indépendamment du démonstrateur PROLOG) pour deux catégories de raisons : d'une part pour des raisons d'efficacité, et d'autre part pour des raisons théoriques.

L'évaluation du prédicat $\vec{U}(D,D')$ doit utiliser bien évidemment les procédures de résolution des relations $\xrightarrow{*}$ et $\Longrightarrow$. Or nous avons construit deux algorithmes efficaces qui permettent précisément de déterminer les conditions de décidabilité des formules $a \xrightarrow{*} b$ et $a \Longrightarrow b$ (voir § I.2.3. et

§ II.5.6., chapitre cinquième). Nous entendons bénéficier de cette efficacité dans la méthode d'évaluation de ce prédicat. Par ailleurs les modalités d'exploration des R-graphes que nous définissons tiennent compte de manière explicite de la consistance des connaissances enregistrées dans la base de connaissances associée au système ARCHES (voir § III.3.2., chapitre septième). L'algorithme correspondant permet ainsi de limiter considérablement la longueur des chemins effectivement explorés. Par rapport à PROLOG dans lequel les modalités d'exploration ne sont définies que par la représentation en clauses de Horn des règles d'inférence inter-champs, cet algorithme permet de supprimer un très grand nombre de choix conduisant à coup sûr à un échec.

Au niveau théorique, nous souhaitons établir les propriétés logiques de ce démonstrateur directement à partir des caractéristiques formelles des éléments qui composent le système symbolique ARCHES, et notamment à partir de la structure algébrique qui organise les D-termes (voir § II.3., chapitre septième). Or la règle de résolution de Robinson qui est à la base du démonstrateur PROLOG suppose que les termes forment une algèbre libre. C'est pourquoi il nous a paru opportun de donner une construction formelle de ce démonstrateur qui soit indépendante du démonstrateur PROLOG. Pour ce faire, nous définissons des modalités d'unification des D-termes qui tiennent compte de l'organisation algébrique de l'ensemble Δ des descriptions (i.e. de la relation de déduction $\Longrightarrow$, voir § III. et § IV., chapitre quatrième). Ces modalités permettent de déterminer la résolution de la formule $D \Longrightarrow D'$ qu'on appellera $\Longrightarrow$ - Unification, et qui sera exprimée par la valeur de vérité du prédicat $\vec{U}(D,D')$ (voir § II.).La règle d'inférence intra-champ et le prédicat de $\Longrightarrow$ - unification $\vec{U}(D,D')$ nous permettent alors de définir *une règle de résolution pour le système symbolique ARCHES* dont on prouvera sa validité (voir § III.). Enfin nous déterminerons les *modalités d'exploration des R-graphes* compte tenu de la règle d'inférence inter-champs et des conditions de consistance des connaissances enregistrées (voir § IV.). Ainsi le démonstrateur qui met en oeuvre le raisonnement déductif dans ARCHES est défini à partir de cette règle de résolution et de ces modalités d'exploration des R-graphes. Nous donnerons sa construction algorithmique que nous validerons en démontrant en particulier la *complétude* de la règle de résolution (voir § V.).

II. MODALITES D'UNIFICATION DES D-TERMES.

II.1. Substitution et unification dans $\mathcal{T}$.

L'unification des D-termes utilise, en particulier, pour sa définition et l'étude de ses propriétés les caractéristiques de l'unification classique développée dans l'algèbre libre de termes $\mathcal{T}$. Ce paragraphe a pour objet de rappeler sommairement les éléments de cette unification qui sont nécessaires à notre étude (30).

II.1.1. Substitution.

définition 1. On appelle *composante de substitution* toute paire $<v_i,t_i>$, dans laquelle v_i est une variable et t_i un terme quelconque ne contenant aucune occurrence de v_i (donc en particulier différent de v_i).

définition 2. On appelle *substitution* un ensemble fini, éventuellement vide, de composantes de substitution $<v_i,t_i>$ telles que tous les v_i sont différents :

$$\theta=\{<v_1,t_1>,<v_2,t_2>,\ldots,<v_n,t_n>\}$$

La substitution vide est notée par ε ; les autres substitutions sont notées par les lettres grecques telles que σ, ρ, θ, η,...

définition 3. Si E est un terme et $\theta=\{<v_i,t_i>\mid 1\leqslant i\leqslant n\}$ une substitution, on appelle *instance* de E par θ l'expression, notée $E\theta$, obtenue en *remplaçant simultanément*, $\forall 1\leqslant i\leqslant n$, chaque occurrence de la variable v_i dans E par le terme t_i.

De cette définition, il résulte très clairement que :

α) $E\varepsilon=E$

β) Si $E\lambda=E\sigma$ alors $\lambda=\sigma$

Remarque 1.

La substitution de la variable v_i par le terme t_i dans l'expression E n'introduit jamais la variable v_i (d'après la définition 1). Par contre elle peut introduire d'autres variables $v_j \neq v_i$ contenues dans t_i ; mais celles-ci ne sont pas modifiées par la substitution θ du fait de la simultanéité imposée aux opérations de substitution définies dans θ.

définition 4. On appelle *composition de deux substitutions* θ et λ, telles que $\theta=\{<v_i,t_i>|1\leqslant i\leqslant n\}$ et $\lambda=\{<v'_j,t'_j>|1\leqslant j\leqslant m\}$, la substitution, notée $\theta\lambda$, telle que :

$$\theta\lambda=\{<v_i,t_i\lambda>|v_i\notin t_i\lambda,1\leqslant i\leqslant n\}\cup\{<v'_j,t'_j>|v'_j\neq v_i,1\leqslant j\leqslant m,1\leqslant i\leqslant n\}$$

On supprime dans $\theta\lambda$ d'une part toutes les composantes de substitution $<v_i,t_i>$ telles que $t_i\lambda$ contient au moins une occurrence de v_i (donc en particulier $v_i \equiv t_i\lambda$), et d'autre part les composantes de substitution $<v'_j,t'_j>$ telles que $v'_j=v_i$ pour au moins un $i\in[1,n]$.

On démontre très facilement les propriétés suivantes :

α) $\theta\varepsilon=\varepsilon\theta=\theta$ (la composition admet ε pour élément neutre)

β) $(\theta\lambda)\mu=\theta(\lambda\mu)=\theta\lambda\mu$ (la composition est associative)

γ) $(E\theta)\lambda=E(\theta\lambda)$

définition 5. On dit que la substitution σ *est plus générale que la substitution* θ, ou que θ *est une instance de* σ, s'il existe une substitution λ telle que :

$$\theta=\sigma\lambda$$

On montre très facilement que la relation "est plus générale que" (ou "est une instance de") est une relation d'ordre notée $\leqslant$:

$$\sigma\leqslant\theta \iff \exists\lambda \text{ tel que } \theta=\sigma\lambda$$

Exemple 1.

$\sigma=\{<x,\$(y,.(ISA,TEINTE,\$(z,\Lambda)))>\}$

$\theta=\{<x,\$(BLEU,.(ISA,TEINTE,\$(CLAIR,\Lambda)))>,<y,BLEU>,<z,CLAIR>\}$

$\sigma\leqslant\theta$ car $\exists\lambda=\{<y,BLEU>,<z,CLAIR>\}$ telle que : $\theta=\sigma\lambda$

II.1.2. Unification dans $\mathcal{T}$.

définition 6. On appelle *unificateur* de deux termes E et E' toute substitution σ telle que : $E\sigma=E'\sigma$.

Soit $\mathcal{U}$ l'ensemble des unificateurs de E et E' :

$$\mathcal{U}(E,E')=\{\sigma|E\sigma=E'\sigma\}$$

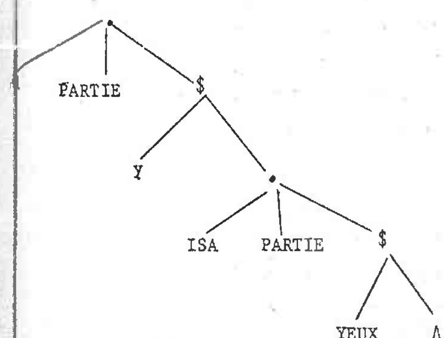
définition 7. On dit que les deux termes E et E' sont *unifiables* si et seulement si $\mathcal{U}(E,E')\neq\phi$.

Théorème 1. Si E et E' sont deux termes unifiables, alors il existe un *unificateur le plus général* (noté UPG) et un seul, au renommage près des variables.

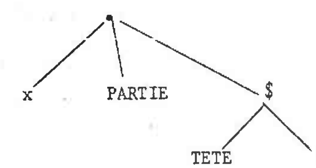
Désignons par σ_o l'UPG de E et E' : $E\sigma_o=E'\sigma_o$; d'après la définition 5, $\forall\rho\in\mathcal{U}(E,E')\ \exists\eta$ telle que $\rho=\sigma_o\eta$ soit $\sigma_o\leqslant\rho$. L'UPG est donc l'élément minimum de $\mathcal{U}(E,E')$.

L'UPG de deux termes unifiables E et E' est calculé par *l'algorithme d'unification de Robinson* {115}.

Exemple 2.



(E)



(E')

$$UPG(E,E')=\sigma_0=\{<x,ISA>,<y,TETE>,<z,.(ISA,PARTIE,\$(YEUX,\Lambda))>\}$$

$$E\sigma_0=E'\sigma_0=.(ISA,PARTIE,\$(TETE,.(ISA,PARTIE,\$(YEUX,\Lambda))))$$

II.2. $\Longrightarrow$ - Unification (Flèche-Unification).

II.2.1. Instances de D-termes.

Etant donné un D-terme D et une substitution σ, l'instance de D par σ n'est pas, dans le cas général, un D-terme : $D\sigma$ est un terme de l'algèbre libre $\mathcal{T}$ construite sur $\mathcal{A}$ et S_F (voir § II.3., chapitre septième). Désignons par Σ l'ensemble des substitutions tel que $\forall\theta\in\Sigma$ et $\forall D\in\mathcal{T}_D$, l'instance de D par θ est un D-terme, c'est-à-dire que $D\theta\in\mathcal{T}_D$ (remarquons que $\Sigma\neq\phi$ car il contient au moins la substitution vide ε). Il en résulte que si $\theta=\{<x_i,t_i>|1\leq i\leq n\}$, alors quel que soit i, t_i est une variable, une constante ou un D-terme atomique de la forme $\cdot(x,T,\$(y,U))$ (U étant un D-terme atomique), ou enfin un terme véhiculant une description locale (i.e. de la forme $\$(y,U)$).

Nous supposerons désormais que tout D-terme D est mis sous sa forme canonique, c'est-à-dire qu'il est représenté par une addition de disjonctions de D-termes atomiques et/ou existentiels (voir § II.3., chapitre septième) :

$$D=\mathop{*}_{i=1}^{n}\left[\mathop{+}_{j=1}^{m(i)} D_j^i\right]$$ (D_j^i étant un D-terme atomique ou existentiel)

Proposition 1. $\forall\theta\in\Sigma$ et $\forall D\in\mathcal{T}_D$ et exprimé sous forme canonique, l'instance de D par θ est obtenue en remplaçant dans D tous les D-termes atomiques et existentiels par leur instance par θ :

$$D\theta=\left[\mathop{*}_{i=1}^{n}\left[\mathop{+}_{j=1}^{m(i)} D_j^i\right]\right]\theta=\mathop{*}_{i=1}^{n}\left[\mathop{+}_{j=1}^{m(i)} (D_j^i\ \theta)\right]$$

(démonstration évidente à partir de la définition 3).

Par ailleurs nous admettons que toute instance d'une règle de substitution-réduction est une règle de substitution-réduction. Plus précisément les règles de réécriture de ARCHES verifient la propriété axiomatique suivante (voir § III.3.1., chapitre quatrième) :

Propriété. Si $\theta\in\Sigma$ et si $u_iu_j\overset{\circ}{\longrightarrow}u_k$ est une règle de réécriture appartenant au système de règles de substitution-réduction de ARCHES, alors $(u_iu_j)\theta\overset{\circ}{\longrightarrow}u_k\theta$ est une règle appartenant au même système de règles.

Proposition 2. $\forall\theta\in\Sigma$ et pour tout D-terme atomique de la forme $u_1u_2\ldots u_k\ldots u_p$, tel que $\forall k<p$ u_k représente un terme descriptif non décrit (ou éventuellement une variable pour k=p), l'instance de $u_1u_2\ldots u_k\ldots u_p$ par θ s'exprime (de manière évidente) comme suit :

$$(u_1u_2..u_k..u_p)\theta=(u_1\theta)(u_2\theta)..(u_k\theta)..(u_p\theta)$$

Proposition 3. Si $\theta\in\Sigma$ et si D_1 et D_2 sont deux D-termes atomiques représentant des termes descriptifs quelconques tels que la formule $D_1\overset{*}{\longrightarrow}D_2$ est vérifiée, alors la formule $D_1\theta\overset{*}{\longrightarrow}D_2\theta$ est également vérifiée.

(immédiat à partir de la propriété précédente et de la proposition 2).

Proposition 4. $\forall\theta\in\Sigma$ et $\forall D_1,D_2\in\mathcal{T}_D$ tels que $D_1\Longrightarrow D_2$, alors la formule $D_1\theta\Longrightarrow D_2\theta$ est également vérifiée.

démonstration. Si $D_1\Longrightarrow D_2$ est vérifiée, alors il existe au moins un arbre de validation $Av(i)$ valide (voir § II.5.5., chapitre cinquième et § VI., chapitre sixième). Il en résulte que tous les chemins de $Av(i)$ sont valides. Dans la procédure VALID(a_{j_0},t) toutes les formules de la forme $a_{j_0}\overset{*}{\longrightarrow}v(\alpha)$ sont vérifiées ; il en est donc de même des formules $a_{j_0}\theta\overset{*}{\longrightarrow}v(\alpha)\theta$ d'après la proposition 3. On en déduit que $D_1\theta\Longrightarrow D_2\theta$ (c.q.f.d.)

Proposition 5. Si la formule $D_1\Longrightarrow D_2$ n'est jamais vérifiée, avec D_1 et $D_2\in\mathcal{T}_D$, alors $\forall\theta\in\Sigma$ la formule $D_1\theta\Longrightarrow D_2\theta$ n'est jamais vérifiée.

(démonstration immédiate à partir de la propriété et des propositions précédentes).

II.2.2. Définition de la $\Longrightarrow$ - Unification.

définition 8. On dit que deux D-termes D et D' sont $\Longrightarrow$-unifiables (lire flèche -unifiables) si et seulement si il existe au moins un D-terme D_1 tel que $D \Longrightarrow D_1$, et D_1 et D' sont unifiables.

Désignons par $\vec{U}(D,D')$ le prédicat dont la vérité indique que D et D' sont $\Longrightarrow$ - unifiables :

$$\vdash \vec{U}(D,D') \Leftrightarrow \exists\, [D_1,\sigma_1] \text{ tel que } \left|\begin{array}{l} D \Longrightarrow D_1 \\ D_1\sigma_1 = D'\sigma_1 \end{array}\right.$$

Il est clair que si le prédicat $\vec{U}(D,D')$ est vrai, alors il existe en général plusieurs paires $[D_i,\sigma_i]$ qui le satisfont, sauf si D et D' sont des D-termes sans variable ; dans ce dernier cas il existe une seule paire qui a pour valeur $[D',\varepsilon]$.

Proposition 6. Si $\vdash \vec{U}(D,D')$, alors il existe au moins une substitution $\sigma \in \Sigma$ telle que $D\sigma \Longrightarrow D'\sigma$.

démonstration. Si $\vec{U}(D,D')$ est vrai, alors $\exists [D_1,\sigma]$ telle que $D \Longrightarrow D_1$ et $D_1\sigma = D'\sigma$. D'après la proposition 4, il en résulte que $D\sigma \Longrightarrow D_1\sigma$, soit $D\sigma \Longrightarrow D'\sigma$ (c.q.f.d.).

Cette proposition montre que si le prédicat $\vec{U}(D,D')$ est vrai, alors il existe un ensemble d'unificateurs, noté $\vec{\mathcal{U}}(D,D')$, qui le satisfont :

$$\vec{\mathcal{U}}(D,D') = \{\sigma_i \mid D\sigma_i \Longrightarrow D'\sigma_i\}$$

Proposition 7. Si D et D' sont deux D-termes tels que $\exists \sigma \in \Sigma$ pour laquelle la formule $D\sigma \Longrightarrow D'\sigma$ est toujours vérifiée, alors D et D' sont $\Longrightarrow$ - unifiables.

démonstration. Soit $\mathcal{E} = \{D_1, D_2, \ldots, D_i, \ldots\}$ l'ensemble de tous les D-termes tels que quel que soit i $D_i\sigma = D'\sigma$ ($\mathcal{E} \neq \phi$ car $D' \in \mathcal{E}$). L'ensemble $\mathcal{E}$ est donc unifiable et admet un unificateur le plus général μ défini par $\sigma = \mu\gamma$ (théorème 1). Comme la formule $D\sigma \Longrightarrow D'\sigma$ est toujours vérifiée, quelque soit i on a $D\sigma \Longrightarrow D_i\sigma$ (1). Supposons que pour tout i la formule $D \Longrightarrow D_i$ n'est jamais vérifiée ; d'après la proposition 5, on en déduit que la formule $D\sigma \Longrightarrow D_i\sigma$ n'est jamais vérifée, ce qui est contradictoire avec l'énoncé (1). Il existe donc au moins un D-terme $D'' \in \mathcal{E}$ (éventuellement identique à D') tel que $D \Longrightarrow D''$ et $D''\mu = D'\mu$ car μ est l'UPG de D' et D'' ; on en déduit que D et D' sont $\Longrightarrow$ - unifiables (c.q.f.d.)

Proposition 8. Si D et D' sont deux D-termes unifiables dans l'algèbre libre $\mathcal{T}$, alors ils sont $\Longrightarrow$- unifiables.

(démonstration immédiate du fait de la réflexivité de la relation de déduction $\Longrightarrow$; la réciproque n'est pas vérifiée dans le cas général).

II.2.3. Algorithme de $\Longrightarrow$ - unification.

L'algorithme de $\Longrightarrow$ - unification détermine l'ensemble des unificateurs $\vec{\mathcal{U}}(D,D')$ de deux D-termes D et D' $\Longrightarrow$ - unifiables ; il détecte également le fait que $\not\vdash \vec{U}(D,D')$ en donnant pour résultat $\vec{\mathcal{U}}(D,D') = \phi$. Sa construction s'appuie sur les propositions 4 et 6 qui se résument comme suit :

$$\text{et} \left|\begin{array}{l} \forall \theta \in \Sigma \text{ si } D_1 \Longrightarrow D_2 \text{ alors } D_1\theta \Longrightarrow D_2\theta \\ \vdash \vec{U}(D,D') \Leftrightarrow \vec{\mathcal{U}}(D,D') = \{\sigma_i \mid D\sigma_i \Longrightarrow D'\sigma_i\} \neq \phi \end{array}\right.$$

L'algorithme de $\Longrightarrow$ - unification engendre l'arbre AEO associé à la formule $D \Longrightarrow D'$ (voir § II.5.4. et § II.5.6., chapitre cinquième). Puis il calcule les substitutions ρ qui valident tous les arbres de validation $A\upsilon(i)$ contenus dans AEO, dont la racine est évidemment D et dont les terminaux sont représentés par les ensembles $T_i = \{t_k^i \mid 1 \leq k \leq p\}$; rappelons que cette validation consiste à valider tous les chemins $\mu[D\rho, t_k^i\rho]$ de $A\upsilon(i)$ (voir § II.5.5., chapitre cinquième). Pour ce faire, il construit, pour chaque arbre de validation, une arborescence de substitutions intermédiaires dont les compositions contribuent à fournir les différentes solutions. A partir de la substitution vide ε, il engendre le premier niveau de cette arborescence en déterminant toutes les substitutions $\sigma_1(j)$ qui valident le chemin $\mu[D\sigma_1(j), t_1^i\sigma_1(j)]$, et auxquelles correspondent les

D-termes D_j^1 tels que $D \Longrightarrow D_j^1$ et $D_j^1\sigma_1(j)=D'\sigma_1(j)$, $\sigma_1(j)$ étant l'UPG de D_j^1 et D'. Pour chaque sommet $\sigma_1(1)$, l'algorithme de $\Longrightarrow$- unification engendre le deuxième niveau composé de toutes les substitutions $\sigma_2(j)$ qui valident le chemin $\mu[D\sigma_1(1)\sigma_2(j), t_2^i\sigma_1(1)\sigma_2(j)]$, et ainsi de suite jusqu'au niveau p (figure 8.1.). S'il existe un niveau $k \leq p$ tel qu'aucune substitution ne valide le chemin correspondant au terminal t_k^i, alors $A\nu(i)$ n'est pas valide, et $\vec{\mathcal{U}}(D,D')$ reste inchangé. Dans le cas contraire toutes les substitutions de la forme $\sigma_1(l_1)\sigma_2(l_2)\ldots\sigma_p(l_p)$ sont des solutions. Ensuite cet algorithme examine dans les mêmes conditions l'arbre de validation suivant, et ainsi de suite jusqu'au dernier.

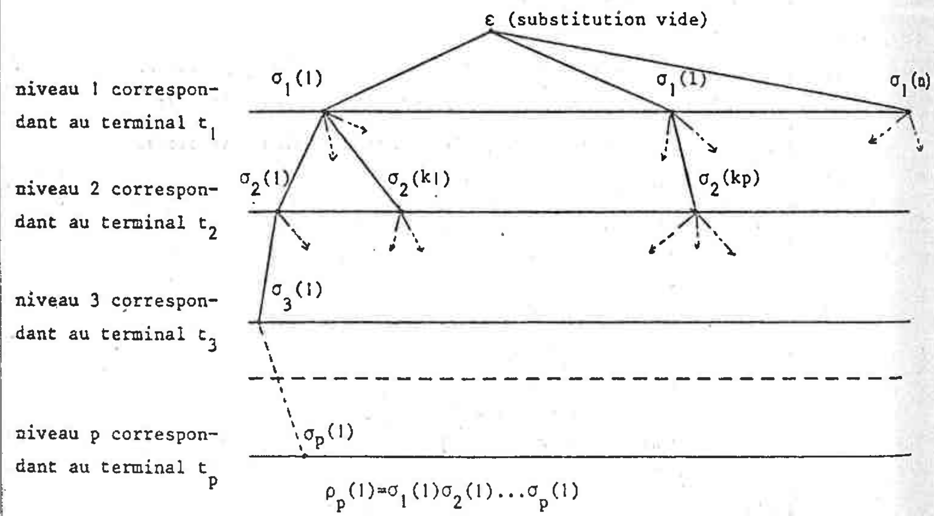


figure 8.1. - Arbre des substitutions intermédiaires pour l'ensemble $T=\{t_k \mid 1\leq k\leq p\}$ des terminaux de $A\nu(i)$

définition de l'algorithme de $\Longrightarrow$- unification.

(1) Construction de l'arbre AEO associé à la formule $D \Longrightarrow D'$.

(2) $\vec{\mathcal{U}}(D,D'):=\phi$; i:=1

(3) Construction à partir de AEO de $A\nu(i)$.

(4) *Si* la construction est impossible
alors *si* $\vec{\mathcal{U}}(D,D')=\phi$
  alors ECHEC ; FIN
  sinon SUCCES ; FIN
  fsi
sinon Soit $T:=\{t_k \mid 1\leq k\leq p\}$ les p terminaux de $A\nu(i)$.
fsi

(5) k:=0 ; j_k:=1 ; $\rho_k(j_k):=\varepsilon$

(6) k:=k+1

(7) $\forall 1\leq l\leq j_{k-1}$ recherche de toutes les substitutions $\sigma_k(j)$ telles que $\mu[D\rho_k(j), t_k\rho_k(j)]$ est un chemin valide dans $A\nu(i)$ avec $\rho_k(j)=\rho_{k-1}(l)\sigma_k(j)$; soit j_k le nombre de ces substitutions.

(8) *Si* $j_k=0$
alors i:=i+1 ;
  aller-à (3)
sinon *si* k=p
  alors $V:=\vec{\mathcal{U}}(D,D')\cup\{\rho_k(j) \mid 1\leq j\leq j_k\}$;
    $\vec{\mathcal{U}}(D,D'):=V$;
    i:=i+1 ; *aller-à* (3)
  sinon *aller-à* (6)
  fsi
fsi

II.2.4. Théorème de $\Longrightarrow$ - unification.

Théorème 2. Si deux D-termes D et D' sont $\Longrightarrow$ - unifiables, alors l'algorithme de $\Longrightarrow$ - unification se termine toujours, et quel que soit $\rho \in \vec{\mathcal{U}}(D,D')$ il existe D_i tel que $D \Longrightarrow D_i$ et ρ est l'UPG de D_i et D'.

démonstration. Remarquons tout d'abord que l'algorithme de $\Longrightarrow$ - unification de deux D-termes D et D' s'arrête toujours car l'arbre AEO est toujours fini (voir § II.5.4., chapitre cinquième), et que le nombre d'arbres de validation engendrés par AEO est également borné (voir § II.5.5., chapitre cinquième ; proposition 19). $\vec{U}(D,D')$ étant vrai, il existe au moins une paire $[D_i, \theta]$ telle que $D \Longrightarrow D_i$ et $D_i\theta=D'\theta$; soit $\rho\in\vec{\mathcal{U}}(D,D')$ telle que $D \Longrightarrow D_i$ et $D_i\rho=D'\rho$. Montrons que ρ est l'unificateur le plus général de D_i et D', c'est-à-dire qu'il existe la substitution λ telle que $\theta=\rho\lambda$. Pour ce faire, nous allons prouver par induction dans l'algorithme de $\Longrightarrow$ - unification que $\forall\ 1\leqslant k\leqslant p\ \exists\lambda_k$ telle que $\theta=\rho_k\lambda_k$.

Soit $\{d_1,..,d_k,..,d_p\}$ les terminaux de l'arbre de validation valide associé à la formule $D \Longrightarrow D_i$, et $\{t_1,..,t_k,..,t_p\}$ l'ensemble des D-termes de D' correspondants à $\{d_1,..,d_k,..,d_p\}$. Il en résulte que :

$$\{d_1,..,d_k,..,d_p\}\theta=\{t_1,..,t_k,..,t_p\}\theta$$

Pour k=0, on a $\rho_o=\varepsilon$ et $\theta=\rho_o\lambda_o$, d'où $\lambda_o=\theta$. Supposons que $\forall j\leqslant k\ \exists\lambda_j$ telle que $\theta=\rho_j\lambda_j$, et montrons par induction que $\theta=\rho_{k+1}\lambda_{k+1}$. Comme $\theta=\rho_k\lambda_k$, il en résulte que :

$$(1) \qquad \{d_1,..,d_k,d_{k+1},..,d_p\}\rho_k\lambda_k=\{t_1,..,t_k,t_{k+1},..,t_p\}\ \rho\lambda_k$$

Or à la kième itération de l'algorithme de $\Longrightarrow$ - unification (étape 7 de cet algorithme), on a :

$$\{d_1,..,d_k\}\rho_k=\{t_1,..,t_k\}\rho_k$$

A la k+1ième itération, l'algorithme calcule l'unificateur le plus général σ_{k+1} de $d_{k+1}\rho_k$ et $t_{k+1}\rho_k$, soit :

$$d_{k+1}\rho_k\sigma_{k+1}=t_{k+1}\rho_k\sigma_{k+1}$$

Or d'après (1), on a également :

$$d_{k+1}\rho_k\lambda_k=t_{k+1}\rho_k\lambda_k$$

Comme σ_{k+1} est l'UPG de $d_{k+1}\rho_k$ et $t_{k+1}\rho_k$, il en résulte qu'il existe une substitution λ_{k+1} telle que :

$$\lambda_k=\sigma_{k+1}\lambda_{k+1}$$

d'où :

$$\theta=\rho_k\lambda_k=\rho_k\sigma_{k+1}\lambda_{k+1}$$

Mais $\rho_k\sigma_{k+1}=\rho_{k+1}$, d'où :

$$\theta=\rho_{k+1}\lambda_{k+1} \qquad (2)$$

L'expression (2) est donc vraie $\forall 1\leqslant k\leqslant p$; en particulier pour k=p, on a :

$$\theta=\rho_p\lambda_{p+1}$$

soit en posant $\rho=\rho_p$ et $\lambda=\lambda_{p+1}$, il vient $\theta=\rho\lambda$. Il en résulte que ρ est l'UPG de D_i et D' (c.q.f.d.)

II.2.5. $\Longrightarrow$- Unification des structures.

définition 9. On dit que deux structures A_1 et A_2, prises dans cet ordre et respectivement de la forme $A(x_1,D_1)$ et $A(x_2,D_2)$ sont $\Longrightarrow$ - unifiables si et seulement si la paire $<x_1,x_2>$ est une composante de substitution et $\vdash\vec{U}(D_1,D_2)$.

Désignons par $\vec{\mathcal{U}}(D_1,D_2)$ l'ensemble des unificateurs de D_1 et D_2 ; l'ensemble U_s des unificateurs de A_1 et A_2 est alors défini comme suit :

$$U_s=\{\rho_i\,|\,\rho_i=<x_1,x_2>\cup\sigma_i,\sigma_i\in\vec{\mathcal{U}}(D_1,D_2)\}$$

Proposition 9. Si A_1 et A_2 sont deux structures et/ou prédicats utilitaires unifiables, alors A_1 et A_2 sont $\Longrightarrow$ - unifiables.

(démonstration immédiate).

Remarque 2.

Si A_1 et A_2 sont deux structures $\Longrightarrow$ - unifiables, alors en général les structures A_2 et A_1 ne sont pas $\Longrightarrow$ - unifiables.

II.3. Unification forcée.

II.3.1. Position du problème.

Supposons que dans ARCHES soit enregistrée la structure PERSONNE(JEAN,.(EG,POIDS,\$(80,.(ISA,UNITE,\$(KG,Λ))))), et que soit posée la question PERSONNE(x,.(GT,POIDS,\$(50,Λ))) ?. Le prédicat $\vec{U}$(.(EG,POIDS,\$(80,.(ISA,UNITE,\$(KG,Λ)))),.(GT,POIDS,\$(50,Λ))) est toujours faux car EG et GT sont deux constantes représentant des opérateurs différents. En conséquence les structures précédentes ne peuvent pas être déduites l'une de l'autre (voir § III.) et ARCHES ne peut fournir comme solution à la question posée la valeur x:=JEAN, alors qu'il est évident qu'elle constitue une réponse pertinente. Cette situation est dûe au fait que la définition de la $\Longrightarrow$- unification, fondée sur la relation de déduction $\Longrightarrow$ et sur l'unification de Robinson, est essentiellement syntactique et ne tient aucun compte des rapports sémantiques qui existent entre les éléments composant les D-termes.

Une question essentielle soulevée par la définition des D-termes atomiques représentant les termes descriptifs est donc celle de leur *interprétation* dont l'expression formelle permet de "forcer" l'unification des D-termes ayant la même signification, et ainsi de restituer au cours des opérations de recherche la totalité des solutions compatibles avec cette interprétation. Il en résulte que l'appariement des D-termes ne doit plus être fondé que sur l'identité des formes (comme dans le cas de la $\Longrightarrow$- unification), mais aussi sur les propriétés sémantiques véhiculées par ces formes. Ainsi tout couple de formes différentes peut être apparié à condition que soient validées certaines règles sémantiques données à priori. Cette appariement, médiatisé par la prise en compte de la sémantique, définit l'*unification forcée* des D-termes.

II.3.2. Modalités de formulation de l'unification forcée.

Si les fonctions de description ont pour résultats symboliques les valeurs précises des éléments de description attribués aux individus (par exemple .(ISA,COULEUR,\$(ROUGE,Λ)) ou .(EG,POIDS,\$(80,Λ))), alors l'interprétation est le résultat symbolique lui-même. Par contre ceci n'est plus le cas pour des résultats symboliques comme par exemple .(GT,POIDS,\$(50,Λ)) ou .(BETWEEN,LOCALISATION,#(\$(INS(LIT,L1,),Λ),\$(INS(FENETRE,F1),Λ))). Comment interpréter de tels résultats ? Pour les deux exemples précédents, les valeurs précises des éléments de description peuvent être déterminées en interprétant de manière existentielle ces deux résultats. Dans le premier cas, il existe au moins un t (valeur numérique) telle que l'interprétation de .(GT,POIDS,\$(50,Λ)) conduit à la valeur précise .(EG,POIDS,\$(t,Λ)), avec la condition $t>50$; une telle interprétation, intégrée dans ARCHES, permettrait ainsi de fournir dans l'exemple cité dans le paragraphe précédent la solution x:=JEAN. Dans le second cas, il existe au moins un t (un emplacement) tel que l'interprétation de ce résultat conduit à la valeur précise .(ISA,LOCALISATION,\$(t,Λ)), avec la condition $t \in]P_1,P_2[$, P_1 et P_2 étant respectivement les emplacements du lit et de la fenêtre.

Il est évident que l'interprétation des D-termes dépend de la sémantique des classes, des opérateurs et des relations qu'ils peuvent entretenir entre eux. Comme elle a pour *fonction essentielle de forcer l'unification, l'interprétation de tout D-terme est définie comme le résultat d'une procédure de transformation conditionnelle de ce D-terme en d'autres D-termes*. Les règles d'inférence pragmatiques, qui déterminent les lois générales des connaissances véhiculées par ARCHES (voir § IV.2.3., chapitre septième), permettent précisément d'exprimer formellement ce type de transformation conditionnelle. Comme elles sont représentées dans le calcul des prédicats par des clauses de Horn (voir § V.2., chapitre septième), nous définissons la procédure d'interprétation d'un D-terme comme suit :

On appelle *procédure d'interprétation d'un D-terme* un ensemble fini de clauses de Horn dont les têtes sont toujours représentées par des structures dans lesquelles le deuxième argument véhicule ce D-terme ; les queues des clauses déterminant ses interprétations et les conditions de leur validation.

Les procédures d'interprétation appartiennent donc à l'ensemble P qui est l'un des ensembles qui composent toute base de connaissances associée au système symbolique ARCHES (voir § V.2., chapitre septième ; définition 11).

Ainsi l'invocation automatique par le démonstrateur du système ARCHES (voir § V. ; et chapitre neuvième) des procédures d'interprétation permettra de forcer l'unification des D-termes qui n'ont pu être $\Longrightarrow$-unifiés.

Exemple 3.

Procédure d'interprétation du D-terme .(GT,T,\$(y,z)) pour le champ PERSONNE.

PERSONNE(x,.(GT,T,\$(y,z)))-PERSONNE(x,.(EG,T,\$(t,z)))-INF(y,t).
PERSONNE(x,.(GT,T,\$(y,z)))-PERSONNE(x,.(GT,T,\$(t,z)))-INF(y,t).
PERSONNE(x,.(GT,T,\$(y,z)))-PERSONNE(x,.(BTW,T,#(\$(t_1,z),\$(t_2,z))))
-INF(y,t_1).

Dans cette procédure INF(u,v) est un prédicat utilitaire d'inégalité qui exprime que u est inférieur à v.

III. PRINCIPE DE RESOLUTION POUR LE SYSTEME SYMBOLIQUE ARCHES.

III.1. Notion de clause réduite.

définition 10. On appelle *clause réduite* d'une clause C toute clause C_r obtenue en supprimant dans C toutes les structures $-A_i$ telles qu'il existe une structure $-A$ appartenant à C pour laquelle $\forall i \; A \vdash\!\!- A_i$.

Si C est de la forme $\{T,-A,-A_i,-Q_j \mid 1 \leq i \leq p, 1 \leq j \leq n\}$ avec $A=A_o(x,D)$ et $A_i=A_o(x,D_i)$ telles que $\forall 1 \leq i \leq p$ $D \Longrightarrow D_i$, alors la clause réduite C_r est de la forme $\{T,-A,Q_j \mid 1 \leq j \leq n\}$.

Il est a remarquer que si C_r est une clause réduite, toute instance $C_r\theta$ de C_r par θ n'est pas dans le cas général une clause réduite.

Proposition 10. Si C est une clause valide, alors toute clause réduite C_r de C est également valide.

(démonstration immédiate).

Cette proposition nous permet désormais de supposer que toutes les clauses manipulées par le langage ARCHES sont des clauses réduites.

III.2. Définition de la règle de résolution.

Ayant introduit les modalités d'unification des D-termes et la notion de clause réduite, nous pouvons désormais définir la règle de résolution pour le système symbolique ARCHES.

Soit C_1 et C_2 deux clauses quelconques (supposées réduites) n'ayant aucune variable en commun (sinon on procède au renommage des variables) :

$$\begin{cases} C_1 = \{T_1,-A_1,\ldots,-A_n\} \\ C_2 = \{T_2,-B,-B_1,\ldots,-B_m\} \end{cases}$$

Par ailleurs supposons que les structures T_1 et B, respectivement de la forme $T(x,D_1)$ et $T(y,D_2)$, soient $\Longrightarrow$- unifiables. Il en résulte que

$\vdash \vec{U}(D_1,D_2)$ et $\vec{\mathcal{U}}(D_1,D_2)=\{\rho_k \mid 1\leq k\leq p\}\neq \phi$; soit U_s l'ensemble des unificateurs de T_1 et B : $U_s=\{\sigma_k \mid \sigma_k=\langle x,y\rangle \cup \rho_k,\ 1\leq k\leq p\}$ (voir § II.2.5.). Nous définissons la résolvante de C_1 et C_2 comme suit :

définition 11. On appelle résolvante des deux clauses C_1 et C_2 toute clause D_k définie comme suit :

$$D_k=\{C_1\sigma_k-\{T_1\sigma_k\}\}\cup\{C_2\sigma_k-\{-B\sigma_k\}\}$$

Nous devons remarquer que $x\sigma_k=y\sigma_k$ et $D_1\sigma_k \Longrightarrow D_2\sigma_k$; d'après la règle d'inférence intra-champ , nous avons $T_1\sigma_k \vdash B\sigma_k$. Il en résulte que la résolvante D_k est une conséquence logique de cette règle d'inférence, moyennant la transitivité du symbole de dérivation $\vdash$ (voir proposition 11).

Etant donné deux clauses C_1 et C_2, si $T_1\in C_1$ n'est $\Longrightarrow$- unifiable avec aucune structure négative $\in C_2$ alors C_1 et C_2 n'ont pas de résolvante. Dans le cas contraire C_1 et C_2 ont en *général plusieurs résolvantes* construites à partir de T_1 et de B :

$$R(C_1,C_2)=\{D_k \mid 1\leq k\leq p\}$$

définition 12. On appelle *règle de résolution* l'opération qui associe à tout couple de clauses C_1 et C_2 l'ensemble des résolvantes construites à partir des structures T_1 et B telles que $T_1\in C_1$ et $-B\in C_2$.

La règle de résolution est une règle de déduction qui permet de déduire de deux clauses données à priori d'autres clauses construites selon des modalités bien définies. Elle est à l'origine de la définition de la R-déduction.

définition 13. Etant donné un ensemble de clauses E et C une clause quelconque, une *R-déduction de C à partir de E,* notée $E \vdash_R C$, est la liste finie des clauses $B_1,\ldots,B_n$ telles que :

(1) $B_n=C$

(2) $\forall\ 1\leq i\leq n \begin{cases} B_i\in E \\ \text{ou} \\ B_i\in R(B_j,B_k) \text{ pour } j,k<i \end{cases}$

Réfuter un ensemble de clauses E, c'est déduire de cet ensemble la clause vide : $E \vdash_R \Box$. Le *principe de réfutation,* fondé sur la règle de résolution dont on montrera la complétude (voir § V.3.), est à la base de la conception du démonstrateur déductif du système symbolique ARCHES (voir § V.1.).

Si $G\cup R\cup S\cup P$ est la base de connaissances associée au système symbolique ARCHES (voir § V.2., chapitre septième), et F un problème à résoudre tels que l'ensemble $G\cup R\cup S\cup P\cup\{-F\}$ est réfutable alors le problème F est résolu dans ARCHES :

Si $G\cup R\cup S\cup P\cup\{-F\} \vdash_R \Box$

Alors La structure F est démontrable dans ARCHES à partir de l'ensemble $\mathcal{F}_T$ des thèses (représenté par l'ensemble S) au moyen des règles d'inférence pragmatiques (représentées par l'ensemble P) et de la règle d'inférence intra-champ (utilisant en particulier l'ensemble R) (voir § IV.3., chapitre septième).

Remarque 3.

Dans ce cas l'ensemble G n'est pas utilisé car la démonstration du problème F ne met pas en oeuvre la règle d'inférence inter-champs.

III.3. Propriétés de la règle de résolution.

Proposition 11. Si C_1 et C_2 sont deux clauses vraies dans l'interprétation I, alors toute résolvante C de C_1 et C_2 est également vraie dans l'interprétation I.

démonstration. Soit C_1 et C_2 deux clauses de la forme $C_1=\{T_1\}\cup C_1^\circ$ et $C_2=\{-B\}\cup C_2^\circ$; et soit C une résolvante de C_1 et C_2 construite à partir de T_1 et de B :

$$C=\{C_1\sigma-\{T_1\sigma\}\}\cup\{C_2\sigma-\{-B\sigma\}\}=C_1^\circ\sigma\cup C_2^\circ\sigma$$

Supposons que C_1 et C_2 sont vraies dans l'interprétation I ; il en résulte que $C_1\sigma$ et $C_2\sigma$ sont également vraies dans I. Par ailleurs si $T_1\sigma$ est vraie dans I, alors $-B\sigma$ est fausse dans I ; et réciproquement si $T_1\sigma$ est fausse, alors $-B\sigma$ a une valeur de vérité indéterminée (conséquence de la définition 11). Supposons que $T_1\sigma$ soit fausse dans I ; alors $C_1\sigma$ qui est vraie dans I n'est pas une clause unaire, et $C_1^\circ\sigma$ est vraie dans I. On en déduit que $C=C_1^\circ\sigma\cup C_2^\circ\sigma$ est vraie dans I quel que soit la valeur de varité de $C_2^\circ\sigma$. De la même manière si on suppose que $-B\sigma$ est fausse dans I, on en déduit que $C_2^\circ\sigma$ est vraie dans I, et donc de même pour la résolvante C (c.q.f.d.).

Proposition 12. Si C'_1 et C'_2 sont respectivement des instances des clauses C_1 et C_2 et si C' est une résolvante de C'_1 et de C'_2, alors il y a une résolvante C de C_1 et C_2 telle que C' est une instance de C.

démonstration. Soit θ une substitution telle que $C'_1=C_1\theta$ et $C'_2=C_2\theta$ ($C_1\theta$ et $C_2\theta$ étant supposées réduites). Supposons que C'_1 et C'_2 soient respectivement de la forme $\{T'_1\}\cup C'^\circ_1$ et $\{-B'\}\cup C'^\circ_2$, T'_1 et B' étant les deux structures à partir desquelles est construite la résolvante C' ; il existe donc σ, unificateur le plus général de T'_1 et B', tel que :

$$C'=\{C'_1\sigma-\{T'_1\sigma\}\}\cup\{C'_2\sigma-\{-B'\sigma\}\}$$

La clause C_1 est toujours de la forme $\{T_1\}\cup C_1^\circ$, T_1 étant la tête (unique) de C_1. Il en résulte que $C_1\theta=\{T_1\theta\}\cup C_1^\circ\theta$, $C_1^\circ\theta$ étant supposée réduite ; d'où $T'_1=T_1\theta$. De même supposons que la clause C_2 soit de la forme $\{T_2\}\cup\{-B\}\bigcup_{i=1}^{p}\{-B_i\}\cup C_2^\circ$ telle que $\forall 1\leq i\leq p\ B\theta\vdash B_i\theta$ et telle que l'ensemble des clauses $\{B,B_i \mid 1\leq i\leq p\}$ corresponde à la clause B'. Il en résulte que $C_2\theta=\{T_2\theta\}\cup\{-B\theta\}\cup C_2^\circ\theta$, $C_2^\circ\theta$ étant supposée réduite et $B'=B\theta$. Comme T'_1 et B' sont $\Longrightarrow$- unifiables il en résulte que T_1 et B sont nécessairement de la forme : $T_1=T(x,D_1)$ et $B=T(y,D_2)$; d'où $T'_1=T(x',D_1\theta)$ et $B'=T(y',D_2\theta)$. T'_1 et B' étant $\Longrightarrow$- unifiables, il existe en particulier σ, unificateur le plus général de $D_1\theta$ et $D_2\theta$, tel que la formule

$D_1\theta\sigma \Longrightarrow D_2\theta\sigma$ est vérifiée. On en déduit que D_1 et D_2 sont $\Longrightarrow$-unifiables (proposition 7), et il existe au moins une substitution μ, unificateur le plus général de D_1 et D_2, telle que la formule $D_1\mu \Longrightarrow D_2\mu$ est vérifiée. Il existe donc une substituion λ telle que $\theta\sigma=\mu\lambda$ car μ est un unificateur plus général que $\theta\sigma$ (d'après le théorème 2). Dans ces conditions la clause C s'exprime comme suit :

$$\begin{aligned}C'&=\{C_1\theta\sigma-\{T_1\theta\sigma\}\}\cup\{C_2\theta\sigma-\{-B\theta\sigma\}\}\\&=\{C_1\mu\lambda-\{T_1\mu\lambda\}\}\cup\{C_2\mu\lambda-\{B\mu\lambda\}\}\end{aligned}$$

Or la clause :

$$C=\{C_1\mu-\{T_1\mu\}\}\cup\{C_2\mu-\{-B\mu\}\}$$

est une résolvante de C_1 et de C_2 car T_1 et B sont $\Longrightarrow$- unifiables par l'unificateur μ, d'où $C'=C\lambda$: la résolvante C' de C'_1 et C'_2 est donc bien l'instance d'une résolvante C de C_1 et C_2 (c.q.f.d.).

IV. MODALITES D'EXPLORATION DES R-GRAPHES.

IV.1. Propriétés du cheminement.

La règle d'inférence inter-champs permet d'affecter à tout individu des descriptions de plus en plus générales (voir § IV.2.2.a, chapitre septième). Son application itéré dépend des propriétés formelles de la relation SET qui définit les R-graphes du système ARCHES (voir § I.3. et III.3., chapitre troisième). Les mécanismes de transfert des descriptions d'un champ vers un autre, mis en oeuvre par cette règle d'inférence, sont donc réalisés à partir de l'exploration des R-graphes dont les modalités sont déterminées par les propositions suivantes.

Proposition 13. Si la relation $G \cup R \cup S \cup P \vdash_R A(x_o, D_o)$ est vraie et si $SET(A, A_1)$ appartient à G, alors l'énoncé $\vdash -A_1(I(A), D_o)$ est vrai.

Démonstration. Si la relation $G \cup R \cup S \cup P \vdash_R A(x_o, D_o)$ est vraie, alors il existe nécessairement une thèse appartenant à $\mathcal{T}_T$ de la forme $A(x_o, D)$ telle que par application de la règle d'inférence intra-champ on puisse déduire dans ARCHES $A(x_o, D_o)$; *car la règle d'inférence inter-champs n'est pas utilisée au cours de ce processus de démonstration* (voir § III.2.). Ceci présuppose donc que la formule $D \Longrightarrow D_o$ est vérifiée. Il en résulte d'après la condition 1 de consistance des connaissances enregistrées (voir § III.3.2., chapitre septième) que la structure $A_1(I(A), D_o)$ n'est pas une thèse du système symbolique ARCHES. En d'autres termes l'énoncé $\vdash A_1(I(A), D_o)$ n'est pas valide, on en déduit donc que l'énoncé $\vdash -A_1(I(A), D_o)$ est vrai (c.q.f.d.).

Cette proposition indique que si dans le champ A la description D_o décrit localement un quelconque individu x_o alors D_o ne doit pas être une description globale du concept A. Il fixe ainsi les modalités d'enregistrement de l'information dans ARCHES : il impose de caractériser tout individu x_o, appartenant à un champ donné, par la description D_o qui l'identifie en tant que tel (i.e. description spécifique). Il en résulte que les descriptions doivent être réparties en partant des

plus spécifiques (attribuées aux individus appartenant aux champs associés aux terminaux des R-graphes) pour aboutir aux plus générales (attribuées aux individus appartenant aux champs associés aux racines des R-graphes) de telle manière qu'il n'y ait *aucune redondance d'information* dans cette répartition ; comme le montre la proposition suivante :

Proposition 14. Etant donné un chemin quelconque $\mu[A_1, A_2]$ dans un R-graphe donné, si $G \cup R \cup S \cup P \vdash_R A_1(x_o, D_o)$ alors quel que soit les sommets A et A' tels que $A' \neq A_1$, $SET(A, A') \in G$ et $A, A' \in \mu[A_1, A_2]$, l'énoncé $\vdash -A'(I(A), D_o)$ est toujours vrai.

démonstration. Soit $B_1, B_2 \in \mu[A_1, A_2]$ tels que $SET(A_1, B_1)$, $SET(B_1, B_2) \in G$; Comme $S \vdash_R A_1(x_o, D_o)$, d'après la proposition 13 on en déduit l'énoncé $\vdash -B_1(I(A_1), D_o)$ (1). Montrons que $\vdash -B_2(I(B_1), D_o)$; supposons pour cela que $\vdash B_2(I(B_1), D_o)$. La règle d'inférence inter-champs permet alors d'affirmer que $\forall x \vdash B_1(x, D_o)$, donc en particulier pour $x = I(A_1)$ l'énoncé $\vdash B_1(I(A_1), D_o)$, ce qui est contradictoire avec l'énoncé (1). On en déduit que $\vdash -B_2(I(B_1), D_o)$. En utilisant de manière itéative la règle d'inférence inter-champs on généralise très facilement ce résultat pour montrer que quel que soit A et $A' \in \mu[A_1, A_2]$ tels que $SET(A, A') \in G$ l'énoncé $\vdash -A'(I(A), D_o)$ est toujours vrai (c.q.f.d.).

Comme entre tout noeud d'un R-graphe donné et sa racine R il existe au moins un chemin (voir § III.3., chapitre troisième), la proposition 14 reste donc pertinente si A_2 coïncide avec R (figure 8.2.).

Corollaire. Si $\mu_1[A, R]$ et $\mu_2[A, R]$ sont deux chemins distincts vérifiant les trois conditions suivantes - /1/ $\mu_1[A, R]$ et $\mu_2[A, R]$ ont en commun un noeud B différent de A et de R ; /2/ il existe un sommet $A_1 \in \mu_1[A, B]$ tel que l'énoncé $S \vdash_R A_1(x_1, D)$ est vrai ; /3/ l'énoncé $\vdash -C_1(I(C), D)$ est vrai pour tout couple de sommets C et $C_1 \in \mu_2[A, B]$ tel que $SET(C, C_1) \in G$ - alors l'énoncé $\vdash -C'_1(I(C'), D)$ est vrai pour tout couple de sommets C' et $C'_1 \in \mu_2[A, R]$ tel que $SET(C', C'_1) \in G$.

La démonstration est immédiate à partir de la proposition 14 (figure 8.3.).

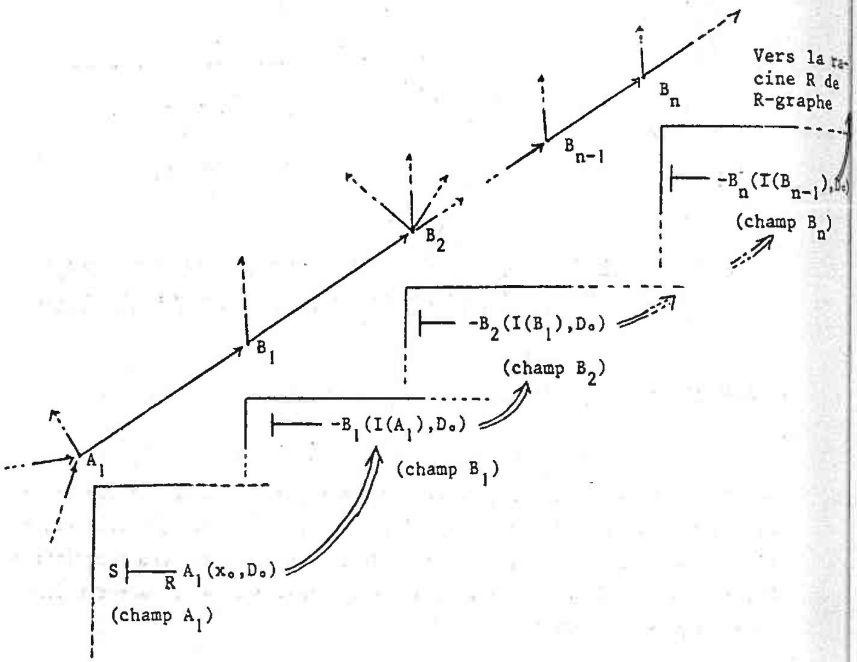


figure 8.2. - Modalités de répartition des descriptions dans le système ARCHES.

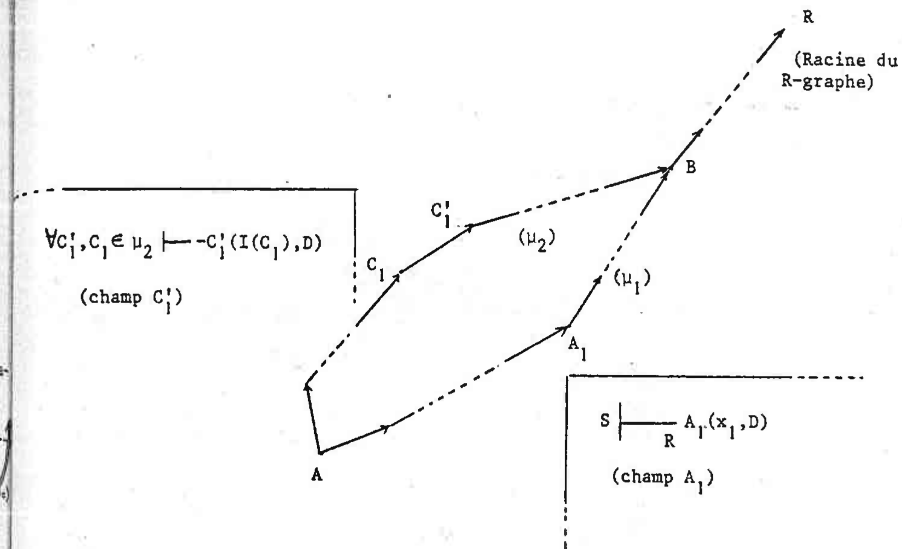


figure 8.3. - Compatibilité des répartitions des descriptions entre champs associés à deux chemins ayant un sommet commun.

IV.2. Algorithme de cheminement.

Les R-graphes sont explorés toutes les fois que le système ARCHES tente de montrer au moyen de la règle d'inférence inter-champs que tout individu appartenant à un champ donné A est caractérisé par une description D qui ne l'identifie pas en tant que tel, c'est à dire une description plus générale (figure 8.4.).

Soit R la racine du R-graphe contenant le sommet A. L'algorithme d'exploration de ce R-graphe à partir de A consiste à rechercher sur tous les chemins compris entre A et R l'ensemble des champs A_i tels que les énoncés du type $G \cup R \cup S \cup P \vdash_R A_i(x,D)$ sont vérifiés, les modalités de parcours de tous ces chemins étant déterminées par la proposition 14 et son corollaire : tout chemin $\mu[A,R]$ est examiné tant que $\forall$ concept $A_i \in \mu[A,R]$ l'énoncé $G \cup R \cup S \cup P \vdash_R A_i(x,D)$ n'a pu être établi ; cet examen étant borné par la racine R (pour une implémentation en PROLOG de cet algorithme voir {114}).

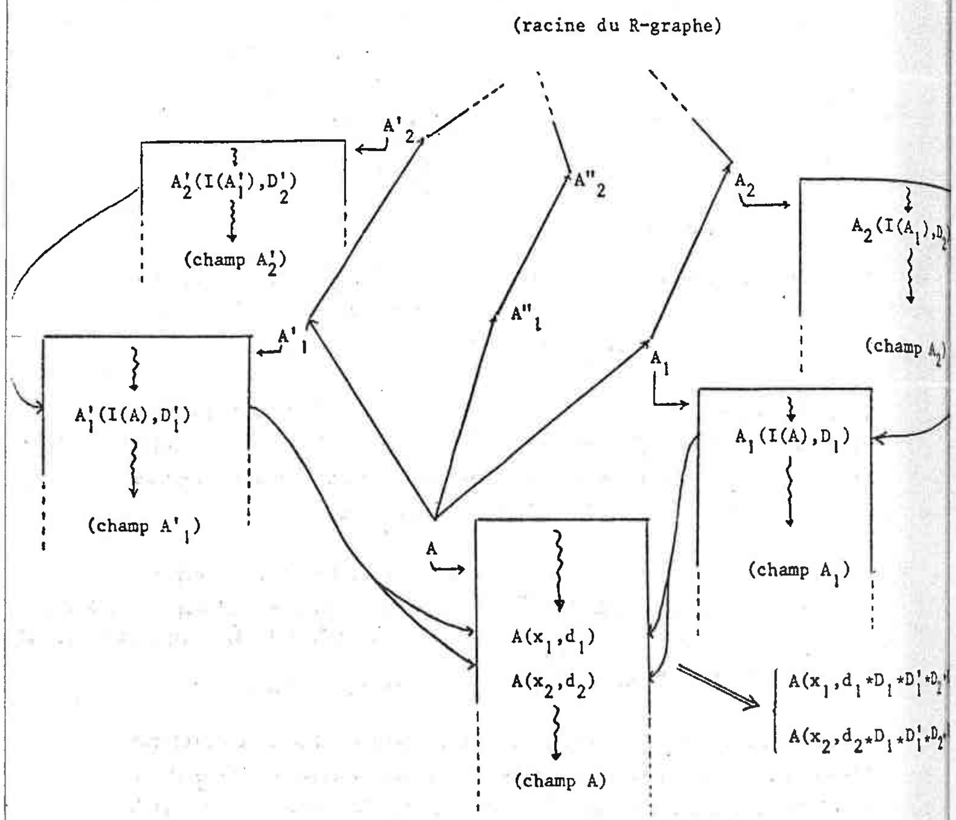


figure 8.4. - *Modalités de transfert des descriptions à l'intérieur d'un R-graphe.*

V. LE DEMONSTRATEUR DEDUCTIF DU SYSTEME SYMBOLIQUE ARCHES.

V.1. Organisation générale du démonstrateur.

Désignons par $\mathcal{B} = G \cup R \cup S \cup P$ la base de connaissances associée au système symbolique ARCHES, et par F un problème à résoudre à partir de $\mathcal{B}$, F étant une structure de la forme T(x, D).

Résoudre le problème F dans ARCHES, c'est déduire, dans le calcul des prédicat, de la base $\mathcal{B}$ la clause {F} en utilisant la règle de résolution (donc la règle d'inférence intra-champ) et les modalités d'exploration des R-graphes (donc la règle d'inference inter-champs).

Ces deux règles de déduction sont à la base de la construction algorithmique du démonstrateur déductif $\mathcal{D}_D(\mathcal{B}, F)$ qui se termine à l'étape SUCCES si F est résolu, sinon il s'arrête à l'étape ECHEC (voir § IV.3., chapitre septième).

$$\mathcal{D}_D(\mathcal{B}, F) = \text{SUCCES} \Leftrightarrow F \text{ est démontrable dans ARCHES.}$$

Pour démontrer F à partir de $\mathcal{B}$ le démonstrateur tente de prouver que l'ensemble de clauses $\mathcal{B} \cup \{-F\}$ est insatisfiable (i.e. $\mathcal{B} \cup \{-F\} \vdash_R \square$) ; si c'est le cas alors F est un théorème et le démonstrateur s'arrête à l'étape SUCCES. Dans le cas contraire le démonstrateur explore le R-graphe dont l'un des sommets est le concept T repérant le champ qui contient le problème F à résoudre. Les modalités d'exploration de ce R-graphe à partir du sommet T lui permettent de déterminer les p structures telles que $\forall 1 \leq i \leq p \ F_i \vdash F$ par application de la règle d'inférence inter-champs. Si il existe au moins une structure F_i telle que l'ensemble $\mathcal{B} \cup \{-F_i\}$ est insatisfiable (voir § III.2..), alors le théorème F est démontré ; dans le cas contraire il y a échec.

définition de l'algorithme $\mathcal{D}_D(\mathcal{B}, F)$

(1) *si* $\mathcal{B}\cup\{-F\} \vdash_R \square$
alors l'ensemble $\mathcal{C}$ des substitutions qui rendent insatisfiables $\mathcal{B}\cup\{-F\}$ est la solution au problème posé ;
SUCCES ; FIN
fsi
Soit $\{F_i | F_i \in \text{champ } T_i,\ 1 \leq i \leq p\}$ les p structures déterminées à partir des modalités d'exploration des R-graphes et telles que quel que soit i $F_i \vdash F$ par application de la règle d'inférence inter-champs.
$\mathcal{C} := \phi$
$i := 0$
(2) $i := i+1$
si $i > p$ *alors* *aller-à* (3) *fsi*
si $\mathcal{B}\cup\{-F_i\} \vdash_R \square$
alors $\mathcal{C} := \mathcal{C} \cup \mathcal{C}_i$; ($\mathcal{C}_i$ étant l'ensemble des substitutions qui rendent insatisfiables l'énoncé $\mathcal{B}\cup\{-F_i\}$)
aller-à (2)
sinon *aller-à* (2)
fsi
(3) *si* $\mathcal{C} = \phi$
alors ECHEC ; FIN
(4) *sinon* SUCCES ; FIN
fsi

Théorème 3. Le problème F de la forme T(x,D) est résolu dans le système ARCHES, représenté par la base de connaissances $\mathcal{B}$, si et seulement si $\mathcal{D}_D(\mathcal{B}, F) = \text{SUCCES}$.

démonstration. Si le problème F est résolu dans ARCHES, alors deux cas mutuellement exclusifs sont possibles (proposition 13) :

Premier cas. D est une description spécifique auquel cas T(x,D) est nécessairement démontrable à partir du champ repéré par le concept T, car il n'existe aucun autre champ à partir duquel la structure F_i peut

être démontrée telle que $F_i \vdash F$ par application de la règle d'inférence inter-champs (propositions 13 et 14). Il en résulte qu'il existe une R-déduction de T(x,D) à partir de $\mathcal{B}$: $\mathcal{B} \vdash_R \{T(x,D)\}$ (1). Affirmer l'énoncé (1) équivaut à rendre insatisfiable l'ensemble de clauses $\mathcal{B}\cup\{-T(x,D)\}$ (voir § III.3., proposition 12) ; on peut donc déduire de cet ensemble la clause vide : $\mathcal{B}\cup\{-T(x,D)\} \vdash_R \square$ (complétude de la règle de résolution, voir démonstration § V.3.) . Dans ces conditions l'algorithme $\mathcal{D}_D(\mathcal{B}, F)$ se termine à l'étape (1) qui est précisément une de ses sorties SUCCES.

Deuxième cas. D est une description générale auquel cas T(x,D) ne peut pas être démontrée directement à partir du champ repéré par le concept T. Soit R la racine du R-graphe qui contient le sommet T, et désignons par $\mathcal{U} := \{\mu_j | 1 \leq j \leq m\}$ l'ensemble de tous les chemins qui existent entre les sommets T et R. Puisque F est résolu et que $\mathcal{B}\cup\{-F\} \not\vdash_R \square$, alors il existe nécessairement au moins un champ T_i tel que $\mathcal{B}\cup\{-T_i(I(T_{i-1}), D)\} \vdash_R \square$ (2), avec $SET(T_{i-1}, T_i) \in G(2')$; en effet de (2) et (2') on déduit par application de la règle d'inférence inter-champs l'énoncé $\vdash T_{i-1}(I(T_{i-2}), D)$, et par application itérée de cette règle on doit obtenir l'énoncé $\vdash T(x,D)$ (car T(x,D) est supposé démontrable). Il en résulte qu'il existe au moins un chemin $\mu \in \mathcal{U}$ de la forme $(T, T_2, \ldots, T_i, \ldots, T_{n-1}, R)$ tel que l'énoncé (2) est vérifié ; d'après la proposition 14 le champ correspondant repéré par le concept T_i est unique ce qui fixe de manière précise les modalités d'exploration des R-graphes. En explorant le R-graphe de racine R selon ces modalités (voir § IV.), on en déduit que l'algorithme doit trouver nécessairement au moins une structure F_i de la forme $T_i(I(T_{i-1}), D)$ telle que $F_i \vdash F$ par application itérée de la règle d'inférence inter-champs; dans ces conditions l'algorithme $\mathcal{D}_D(\mathcal{B}, F)$ se termine à l'étape (4) qui est précisément une de ses sorties SUCCES. Réciproquement si $\mathcal{D}_D(\mathcal{B}, F) = \text{SUCCES}$, alors on montre très facilement en s'appuyant sur la complétude de la règle de résolution (voir § V.3.) et sur les modalités d'exploration des R-graphes que la problème F est résolu (c.q.f.d.).

V.2. Arbres sémantiques du système ARCHES.

V.2.1. Définitions des souches et des arbres sémantiques.

Soit $\mathcal{B}$ la base de connaissances associée à ARCHES, et $H(\mathcal{B})$ l'univers de Herbrand correspondant. Désignons par $A(\mathcal{B})$ l'ensemble des atomes (formules propositionnelles) construits à partir des structures et des prédicats utilitaires qui appartiennent à $\mathcal{B}$ et dont les arguments sont déterminés par énumération des éléments de $H(\mathcal{B})$, {30}. Soit enfin Q_o tout ensemble d'atomes appartenant à $A(\mathcal{B})$ défini comme suit :

$$Q_o = \{B_{ij} \mid \forall ij\ B_o \vdash B_{ij} \quad \text{avec } B_o \notin Q_o \text{ et } B_o, B_{ij} \in A(\mathcal{B})\}$$

Si B_o est un atome construit sur un prédicat utilitaire alors $Q_o = \phi$. Par contre si B_o est un atome de la forme $B(b,D)$ construit à partir de la structure $B(x,y)$, alors tous les B_{ij} qui valident l'énoncé $B_o \vdash B_{ij}$ sont de la forme $B(b,D_{ij})$ telle que la formule $D \Longrightarrow D_{ij}$ est toujours vérifiée.

Dans le cas général l'ensemble d'atomes Q_o est également organisé par la règle d'inférence inter-champs. Soit $\mathcal{A}(Q_o)$ la structure déductive représentée sous forme d'arborescence de $\{B_o\} \cup Q_o$ (figure 8.5.) : $B_{11}, B_{12}, \ldots, B_{1i}, \ldots$ sont immédiatement déductibles de B_o et quel que soit i les sommets de la sous-arborescence de racine B_{1i} définissent l'ensemble Q_{1i} ; $B_{21}, \ldots, B_{2p1}$ sont immédiatement déductibles de B_{11}, et ainsi de suite pour tous les B_{1i} ; le deuxième niveau de l'arborescence $\mathcal{A}(Q_o)$ est construit de telle manière que pour tout j si $\exists k<j$ tel que $B_{2k} = B_{2j}$, alors on supprime sur ce niveau l'atome B_{2j} ; en d'autres termes quel que soit k et j donnés on a toujours $B_{2k} \neq B_{2j}$. Enfin si B_{ij} désigne le jième atome situé au ième niveau de $\mathcal{A}(Q_o)$ alors B_{ij} engendre une sous-arborescence dont les sommets définissent Q_{ij}, et quel que soit k et j donnés on a toujours $B_{ik} \neq B_{ij}$.

A partir de $A(\mathcal{B})$ et de tous les ensembles Q_o^k construits à partir des B_o^k tels que $A(\mathcal{B}) = \bigcup_k [\{B_o^k\} \cup Q_o^k]$, nous introduisons la notion d'arbres sémantiques dont la définition et les propriétés permettent de prouver la complétude de la règle de résolution (voir § III.2., et § V.3.).

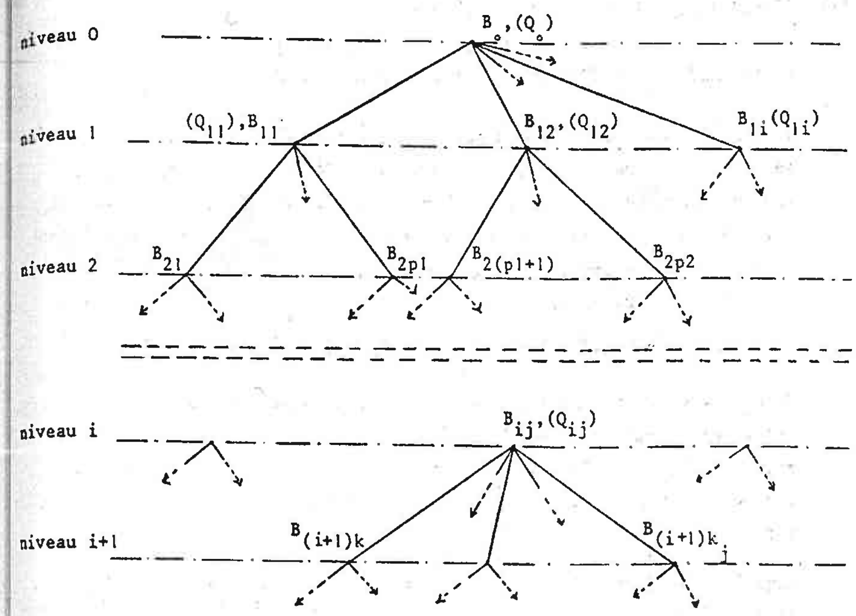


figure 8.5. - Structure déductive de $\{B_o\} \cup Q_o$.

définition 14. On appelle *souche sémantique* associée au doublet $<B_o^k, Q_o^k>$ un arbre binaire dont la construction obéit aux règles ci-après :

(1) Si $Q_o^k = \phi$, alors au doublet $<B_o^k, \phi>$ correspond un arbre binaire réduit à un seul niveau dont les arcs gauche et droit sont étiquetés respectivement par B_o^k et $-B_o^k$.

(2) Si $Q_o^k \neq \phi$, l'arc gauche issu de la racine de la souche sémantique associée à $<B_o^k, Q_o^k>$ est étiqueté par la formule $B_o^k \bigwedge_{ij} B_{ij}^k$ telle que $\forall_{ij} B_{ij}^k \in Q_o^k$; l'arc droit étant étiqueté par $-B_o^k$.

(3) L'extrémité de l'arc étiqueté par $-B_o^k$ engendre la souche sémantique associée au doublet $<B_{11}^k, Q_{11}^k>$.

(4) Tous les terminaux de la souche sémantique associée à $<B_{11}^k, Q_{11}^k>$ engendrent la souche sémantique associée à $<B_{12}^k, Q_{12}^k>$, et ainsi de suite pour tous les atomes immédiatement déductibles de B_o^k et appartenant à Q_o^k.

Les deux lemmes ci-après nous permettent de démontrer une proposition relative aux formules qui étiquettent les deux arcs issus de tout sommet N d'une souche sémantique ; et qui est nécessaire pour prouver la complétude de la règle de résolution.

Lemme 1.
$$\frac{\vdash D \Longrightarrow D'}{\vdash [A(x,D) \wedge A(x,D')] \vee -A(x,D)}$$

démonstration. Posons $B=A(x,D)$ et $B'=A(x,D')$ et supposons que la formule $D \Longrightarrow D'$ soit toujours vérifiée :

$\vdash B \vee -B$

$\vdash B$
$\overline{\vdash B'}$ (règle d'inférence intra-champ)
$\vdash B \wedge B'$
$\vdash (B \wedge B') \vee -B$

$\vdash -B$
$\overline{\vdash -B \vee (B \wedge B')}$

$\vdash (B \wedge B') \vee -B$ (c.q.f.d.)

On généralise très facilement cette proposition à l'ensemble fini des D_i tels que quel que soit i la formule $D \Longrightarrow D_i$ est toujours vérifiée ; si on pose $B=A(x,D)$ et $B_i=A(x,D_i)$, il vient :

Lemme 2.
$$\frac{\vdash D \Longrightarrow D_i \quad \forall i}{\vdash (B \wedge B_i) \vee -B}$$

Proposition 15. Si F_{1N} et F_{2N} sont les formules qui étiquettent les deux arcs issus de tout sommet N d'une souche sémantique, alors la formule suivante est valide : $\vdash F_{1N} \vee F_{2N}$.

(démonstration immédiate à partir des lemmes 1 et 2).

Exemple 4.

Les figures 8.6. et 8.7. montrent respectivement la structure déductive de $\{B_o\} \cup Q_o$, et la souche sémantique associée au doublet $<B_o, Q_o>$, dans lesquelles B_o et Q_o sont définis comme suit :

$B_o = B(b, D_1)$
$Q_o = \{B(b,D_1), B(b,D_3), B(b,D_4)\}$
$D_1 \Longrightarrow D_2 \Longrightarrow D_3$
$D_1 \Longrightarrow D_4$

définition 15. On appelle *arbre sémantique de* ARCHÈS un arbre binaire construit de haut en bas en greffant, les unes sur les autres, les souches sémantiques selon les règles ci-après :

(1) La racine de l'arbre sémantique engendre la souche sémantique associée à un doublet quelconque $<B_o^k, Q_o^k>$.

(2) Les terminaux de la souche sémantique associée à $<B_o^k, Q_o^k>$ engendrent la souche sémantique associée à $<B_o^j, Q_o^j>$, et ainsi de suite par *énumération sans répétition* de tous les doublets $<B_o^i, Q_o^i>$.

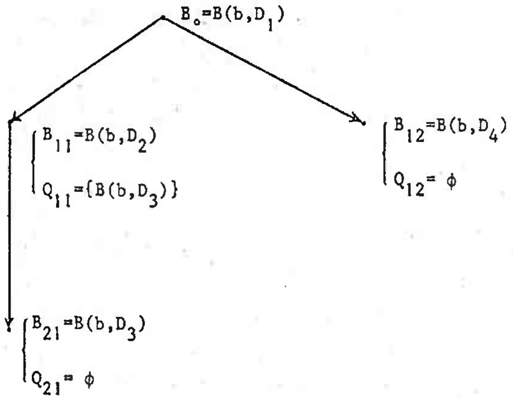


figure 8.6. - *Structure déductive de* $\{B(b,D_i)\ |1\leqslant i\leqslant 4\}$.

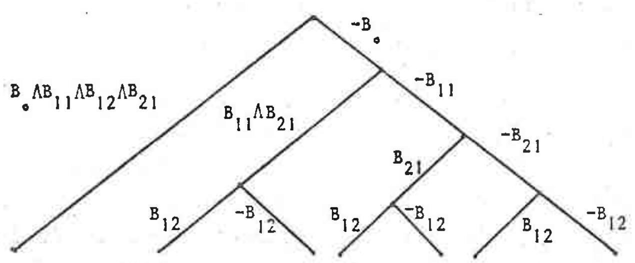


figure 8.7. - *Souche sémantique associée à* $\langle B(b,D_1),\{B(b,D_i)\,|\,2\leqslant i\leqslant 4\}\rangle$.

Désignons par $\mathcal{H}(N)$ l'union de toutes les formules étiquetant les arcs formant le chemin compris entre la racine de l'arbre sémantique et le sommet N. $\mathcal{H}(N)$ ne contient aucune paire complémentaire car l'arbre sémantique est construit par énumération sans répétition des souches sémantiques (règle (2) ci-dessus). Il en résulte que $\mathcal{H}(N)$ *est une interprétation partielle* de $\mathcal{B}$ dans l'univers de Herbrand.

Remarque 4.

Si quelque soit k $Q_o^k = \phi$, alors la définition des arbres sémantiques du système ARCHES est la même que celle, classique, mise en oeuvre pour la démonstration de théorèmes {30} ; cette situation est tout à fait légitime car dans ce cas l'ensemble $A(\mathcal{B})$ ne possède aucune structure algébrique. Les arbres sémantiques de ARCHES se présentent donc comme une généralisation des arbres sémantiques classiques ; généralisation qui prend en compte l'organisation déductive de $A(\mathcal{B})$ déterminée par la règle d'inférence intra-champ, elle-même définie à partir de la relation de déduction $\Longrightarrow$.

Exemple 5.

Considérons l'ensemble S de clauses ci-après :

(C1) $P(a,D_1)-R(a,D_2')$

(C2) $T(a,D_3)-P(a,D_1')$

(C3) $R(a,D_2)$

(C4) $-T(a,D_3')$

dans lequel d'une part les structures P, R et T sont des formules sans variable, et d'autre part les D-termes D_1, D_1', D_2, D_2', D_3, D_3' vérifient les formules $D_1 \Longrightarrow D_1'$, $D_2 \Longrightarrow D_2'$ et enfin $D_3 \Longrightarrow D_3'$. L'ensemble des atomes A(S) construit sur S est donc de la forme :

$$A(S) = \{P(a,D_1), P(a,D_1'), R(a,D_2), R(a,D'_2), T(a,D_3), T(a,D_3') \ldots\}$$

avec

$$\begin{cases} B^1_\circ = P(a,D_1) & ; \quad B^1_{11} = P(a,D_1') & ; \quad Q^1_\circ = \{B^1_{11}\} \\ B^2_\circ = R(a,D_2) & ; \quad B^2_{11} = R(a,D_2') & ; \quad Q^2_\circ = \{B^2_{11}\} \\ B^3_\circ = T(a,D_3) & ; \quad B^3_{11} = T(a,D_3') & ; \quad Q^3_\circ = \{B^3_{11}\} \end{cases}$$

Dans ces conditions la figure 8.8. montre le début de l'arbre sémantique de ARCHES pour l'ensemble S de clauses ayant les propriétés définies ci-dessus.

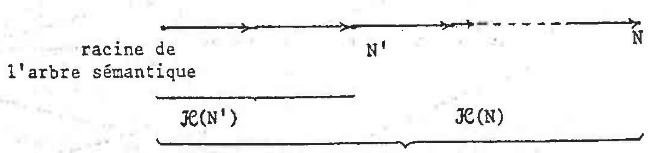
V.2.2. Propriétés définitionnelles.

Dans ce paragraphe nous donnons quatre propriétés définitoires des arbres sémantiques du système ARCHES ; propriétés qui sont nécessaires pour prouver la complétude de la règle de résolution mise en oeuvre au cours des raisonnements déductifs dans ARCHES (voir § V,3.).

définition 16. On dit qu'un arbre sémantique pour un ensemble S de clauses est *complet* si et seulement si pour tout terminal N, l'ensemble $\mathcal{K}(N)$ contient quel que soit i A_i ou $-A_i$ tel que $A_i \in A(S)$.

définition 17. On dit que le sommet N est un *noeud d'échec* si $\mathcal{K}(N)$ réfute une quelconque instance d'une clause $C \in S$, mais $\mathcal{K}(N')$ ne réfute aucune instance d'aucune clause de S pour n'importe quel ancêtre N' de N.

Supposons que C soit de la forme $\{T, -A_i \mid 1 \leqslant i \leqslant p\}$; N est un noeud d'échec si $\exists \sigma$ telle que $-\sigma T \in \mathcal{K}(N)$ et $\forall i\ \sigma A_i \in \mathcal{K}(N)$ et si $\forall$ N' ancêtre de N il n'existe aucune substitution ρ telle que $-\rho T \in \mathcal{K}(N')$ et $\forall i\ \rho A_i \in \mathcal{K}(N')$.



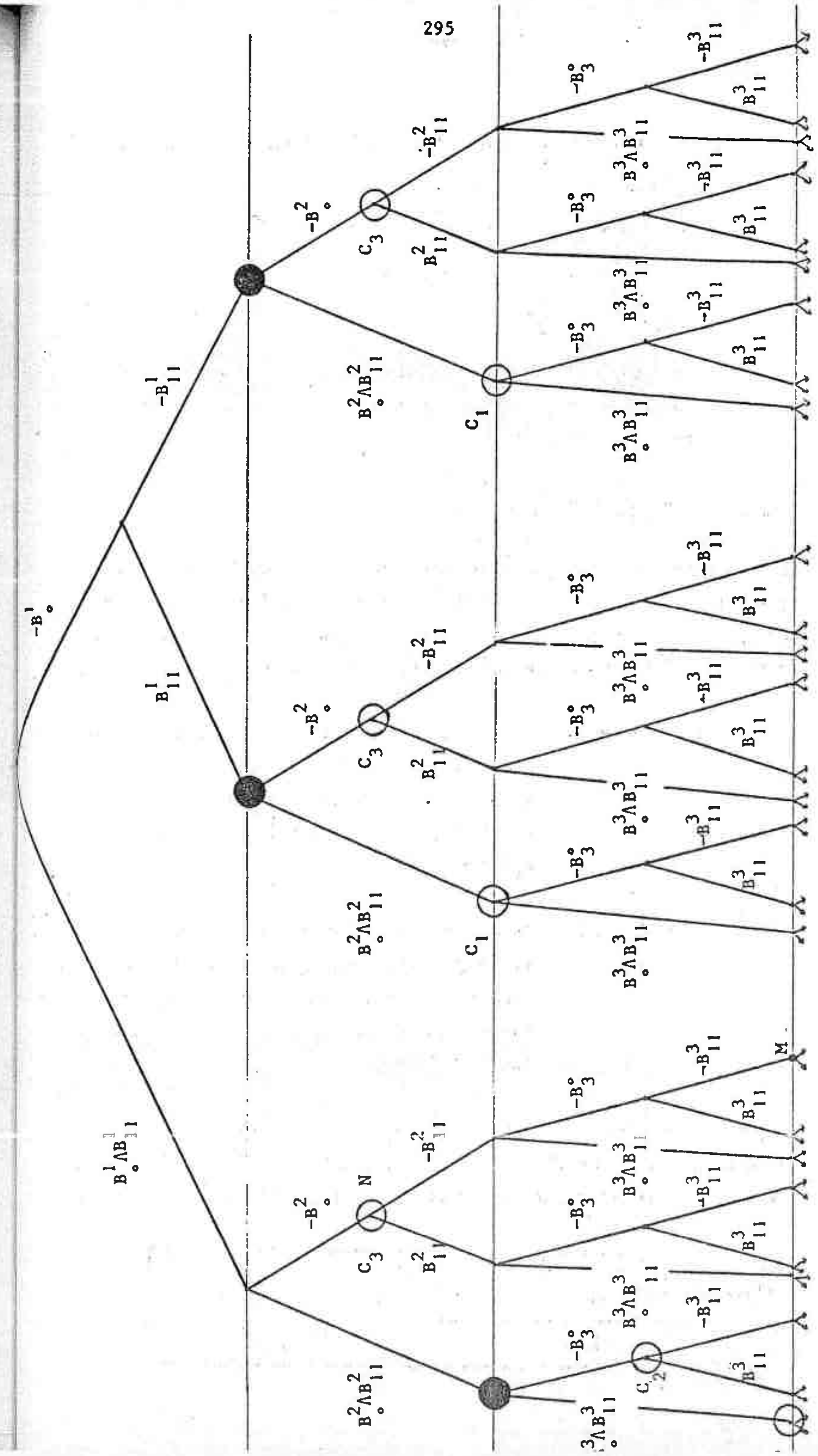


figure 8.8. - Arbre sémantique de ARCHES pour l'ensemble S de clauses.

Remarque 5.

Si $\mathcal{H}(N)$ est un noeud d'échec, il existe en général plusieurs substitutions σ_i qui réfutent une même clause $C \in S$ (voir § II.2., et § III.2.).

Exemple 6.

Reprenons l'ensemble S de clauses dans l'exemple 5 ; et considérons le chemin OM dans l'arbre sémantique associé à S (figure 8.8.). Le sommet $N \in OM$ réfute la clause $C_3=\{R(a,D_2)\}$ car $-B^2_{\circ}$ ou $-R(a,D_2) \in \mathcal{H}(N)$. La figure 8.8. montre tous les noeuds d'échec associés à l'ensemble S (sommets entourés d'un cercle indicé par la clause qui est réfutée) ; dans cet exemple on a toujours $\sigma=\varepsilon$ car les structures sont des formules sans variable.

définition 18. On dit qu'un arbre sémantique de ARCHES est *fermé* si et seulement si tous ses chemins se terminent par un noeud d'échec.

Exemple 7.

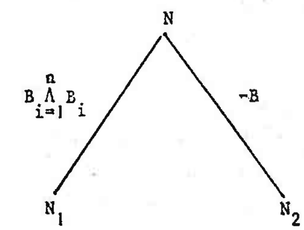
L'arbre sémantique de la figure 8.8. est un arbre sémantique fermé.

définition 19. On dit qu'un noeud N est un *noeud d'inférence* si et seulement si les deux successeurs immédiats de N sont des noeuds d'échec.

Dans la figure 8.8. les noeuds d'inférence sont entourés d'un cercle noirci.

Proposition 16. Si N est un noeud d'inférence alors il existe un noeud N' ancêtre de N (ou éventuellement confondu avec N) tel que N' est un noeud d'échec.

démonstration. Si N est un noeud d'inférence alors ses deux successeurs immédiats N_1 et N_2 sont des noeuds d'échec (définition 19), les arcs $\widehat{NN_1}$ et $\widehat{NN_2}$ étant étiquetés respectivement par $B \bigwedge_{i=1}^{n} B_i$ (ou B si $Q=\phi$) et par $-B$:



N_1 étant un noeud d'échec, il existe $C_k \in S$ et au moins une substitution σ_k (mais en général plusieurs, voir Remarque 5) telle que $C_k\sigma_k$ est réfutée en N_1. Comme N n'est pas un noeud d'échec, il en résulte que $-B$ et/ou $-B_i \in C_k\sigma_k$ (définition 17) ; mais $C_k\sigma_k$ étant supposée une clause réduite (définition 10), il existe une seule formule $B_{\circ}$ identique à B ou (exclusivement) à l'une des B_i telle que $-B_{\circ} \in C_k\sigma_k$. De même on montre qu'il existe $C_j \in S$ et au moins une substitution σ_j telle que $C_j\sigma_j$ est réfutée en N_2 et $B \in C_j\sigma_j$. $C_k\sigma_k$ et $C_j\sigma_j$ sont donc de la forme :

$$C_k\sigma_k = \{T, -B_{\circ}, -B_l \mid 1 \leqslant l \leqslant p\}$$
$$C_j\sigma_j = \{B, -B'_l \mid 1 \leqslant l' \leqslant p'\}$$

avec la condition $B \vdash\!\!- B_{\circ}$; ce qui indique que B et $B_{\circ}$ sont $\Longrightarrow$- unifiables (voir § II.2.5.), l'unificateur le plus général ρ pour les substitutions σ_j et σ_k étant défini par l'équation $(\sigma_j \cup \sigma_k) = \rho.\eta$ (en général il y a plusieurs unificateurs les plus généraux). Il en résulte que si C' est la résolvante de $C_j\sigma_j$ et de $C_k\sigma_k$, alors $\forall Q \in C'$ on a :

$$-Q \in \mathcal{H}(N_1) \cup \mathcal{H}(N_2) - \{B, -B, B_i \mid 1 \leqslant i \leqslant n\}$$

Il existe donc un noeud N' identique à N ou ancêtre de N tel que $\forall Q \in C'$ on a $-Q \in \mathcal{H}(N')$. Par ailleurs d'après la proposition 12 la clause C' est une instance de la clause C qui est une des résolvantes des clauses C_j et C_k : $C \in R(C_j, C_k)$. On en déduit que $\mathcal{H}(N')$ réfute une instance de la clause C ; en conséquence N' est un noeud d'échec pour l'ensemble $S \cup \{C\}$ (c.q.f.d.).

V.3. Complétude de la règle de résolution.

Pour déduire le problème F de la base $\mathcal{B}$, l'algorithme $\mathcal{D}_D(\mathcal{B},F)$ tente de montrer que l'ensemble $\mathcal{B}\cup\{-F\}$ est insatisfiable en le réfutant, c'est-à-dire en prouvant l'énoncé $\mathcal{B}\cup\{-F\} \vdash_R \square$ (voir § III.2. et § V.1.) Nous justifions dans ce paragraphe cette démarche en démontrant d'une part une proposition qui indique que $\mathcal{B} \vdash_R \{F\}$ si et seulement si $\mathcal{B}\cup\{-F\}$ est insatisfiable, et d'autre part deux théorèmes qui prouvent la complétude de la règle de résolution.

Proposition 17. L'énoncé $\mathcal{B} \vdash_R \{F\}$ est valide si et seulement si l'ensemble $\mathcal{B}\cup\{-F\}$ est insatisfiable.

démonstration. Supposons que $\mathcal{B} \vdash_R \{F\}$ soit valide. Deux cas sont alors possibles : d'une part si $\mathcal{B}$ est vrai dans l'interprétation I alors $\{F\}$ est également vrai dans I (proposition 11) ; il en résulte que $\{-F\}$ est faux dans I, et donc aussi $\mathcal{B}\cup\{-F\}$. D'autre part si $\mathcal{B}$ est faux dans I, alors quel que soit la valeur de vérité de $\{F\}$ l'ensemble $\mathcal{B}\cup\{-F\}$ est faux dans I. On en déduit que pour toute interprétation I l'ensemble $\mathcal{B}\cup\{-F\}$ est faux ; donc $\mathcal{B}\cup\{-F\}$ est insatisfiable. Réciproquement supposons que $\mathcal{B}\cup\{-F\}$ soit insatisfiable. Comme $\mathcal{B}$ est satisfiable il existe au moins une interprétation I telle que $\mathcal{B}$ est vrai ; donc $\{-F\}$ est faux, soit $\{F\}$ vrai dans I. Il en résulte que toutes les fois que $\mathcal{B}$ est vrai dans une interprétation alors $\{F\}$ est également vrai dans cette interprétation. On en déduit que $\mathcal{B} \vdash_R \{F\}$ est valide (c.q.f.d.).

Théorème 4. Un ensemble de clauses E est insatisfiable si et seulement si E possède un arbre sémantique fermé fini.

La démonstration de ce théorème est tout à fait comparable à celle du théorème de HERBRAND , {30}, car les arbres sémantiques du système ARCHES sont également binaires (voir § V.2.), et possèdent la propriété définie par la proposition 15.

Théorème 5. Un ensemble de clauses E est insatisfiable si et seulement si il existe une R-déduction de la clause vide ($\square$) à partir de E.

démonstration. Supposons que E soit insatisfiable ; d'après le théorème précédent il possède un arbre sémantique fermé fini $\mathcal{A}$. L'arbre $\mathcal{A}$ possède au moins un noeud d'inférence car sinon il aurait une branche infinie ; soit $N\in\mathcal{A}$ ce noeud d'inférence. D'après la proposition 16 il existe donc dans $\mathcal{A}$ un noeud N' ancêtre de N (ou éventuellement confondu avec N) tel que N' est un noeud d'échec pour la résolvante C_1 de deux clauses C' et $C''\in E$. Désignons par $\mathcal{A}_1$ l'arbre sémantique obtenu en supprimant dans $\mathcal{A}$ le sous-arbre binaire engendré par le sommet N' ; $\mathcal{A}_1$ représente l'arbre sémantique fermé de l'ensemble $E\cup\{C_1\}$, et il est strictement plus petit que $\mathcal{A}$ (car il contient moins de sommets). En itérant ce processus un nombre fini de fois on aboutit nécessairement à la racine de $\mathcal{A}$ qui correspond à la clause vide (car aucune clause de E ne peut être réfutée par cette racine). On obtient donc une liste finie de clauses $\{C_1,\ldots,C_i,\ldots,C_n=\square\}$ telle que pour tout i $C_i\in E$ ou $C_i=R(C_j,C_k)$ pour j,k<i ; il en résulte que $E \vdash_R \square$ (voir § III .2.). Réciproquement supposons que l'énoncé $E \vdash_R \square$ soit valide, et soit $\{C_1,\ldots,C_i,\ldots C_n=\square\}$ la liste des résolvantes dans cette R-déduction. Supposons que E soit satisfiable, et soit I l'interprétation correspondante. D'après la proposition 11 si C' et C" sont vraies dans I alors toute résolvante $\in R(C',C'')$ est également vraie dans I. Il en résulte que $\forall 1\leqslant i\leqslant n$ C_i est vraie dans I. Cependant ceci est impossible car la clause vide ($C_n=\square$) est toujours fausse quelle que soit l'interprétation de I ; il en résulte que l'ensemble de clauses E est insatisfiable (c.q.f.d.).

CHAPITRE NEUVIEME.

RAISONNEMENT ANALOGIQUE
DANS ARCHES

-=-=-=-=-=-=-=-=-=-

I. INTRODUCTION.

De tout temps l'analogie a été *considérée comme un facteur essentiel d'invention*, comme le souligne par exemple DOROLLE dans {51} : "La faculté inventive par excellence, c'est la faculté d'identification, la faculté de percevoir des ressemblances et des différences. Et le jeu de cette faculté suppose une singulière aptitude à penser par analogies". Mais elle a été *regardée avec méfiance comme instrument de preuve* ; le raisonnement par analogie ne semble guère avoir trouvé de crédit auprès des logiciens. La raison essentielle, selon nous, qui rend complexe la description logique de ce mode de raisonnement, est que le mot même d'analogie a un sens très imprécis. Si Aristote lui donne le sens d'égalité de rapports - "A est à B comme C est à D" -, la signification du mot analogie s'est affaiblie pour désigner une certaine ressemblance entre des choses ou des qualités qui par ailleurs diffèrent. Dès que l'égalité de rapports cesse d'être une égalité mathématique (la forme rigoureuse est en effet représentée par la proportion géométrique), la notion commence à se dégrader. Et le *principe d'identité*, essentiel dans toute construction logique, *est mis en défaut*. Il en résulte que le raisonnement analogique *ne permet pas de conclure avec certitude* comme c'est le cas dans le raisonnement déductif. De plus, le degré de vraisemblance des résultats obtenus est très difficile à évaluer. Aussi le raisonnement par analogie a été relativement peu étudié pour lui même, à l' exception de quelques rares tentatives principalement au niveau épistémologique (se reporter par exemple à {6}, {51} et {108}).

Cependant, le raisonnement analogique est très souvent utilisé dans des domaines de connaissances empiriques non théorisés, comme par exemple le champ des sciences de l'homme (voir à ce sujet {48}, {72} et {82}). Aussi la valeur argumentative de ce mode de raisonnement a été étudiée de manière plus systématique par des logiciens des langues naturelles comme GRIZE ou PERELMAN. Le premier a essayé de dégager quelques éléments pour une "logique naturelle", et d'étudier ainsi les mécanismes par lesquels les activités logico-discursives produisent des analogies ({70}, {71}, {100}). Le second a mis d'avantage l'accent sur l'élaboration de schémas d'inférence fondés sur la ressemblance de rapports. Et il a évalué au niveau du discours scientifique le rôle que peut jouer l'argumentation dans la validation des résultats obtenus quand ces discours portent sur des *univers incomplètement décrits* ({105},{106},{107}). C'est là une deuxième faiblesse du raisonnement par analogie qui tend à produire de nouvelles connaissances à partir d'univers dont la description n'est ni exhaustive, ni universelle (inférences qui peuvent conclure du particulier au particulier, de général au général ou encore du particulier au général et inversement).

Ces différents travaux ont atteint un niveau de formalisation tel qu'ils sont loin de permettre une mécanisation effective de ce type de raisonnement. Pourtant les recherches en intelligence artificielle, dans le but de comprendre et d'automatiser le fonctionnement de certains modes de "raisonnement naturel" (voir § IV., chapitre deuxième),se sont intéressées et s'intéressent de plus en plus - à côté du raisonnement déductif - au *raisonnement approché*. Dans ce cadre le raisonnement par analogie joue un rôle important, bien que la plupart des travaux relevant de ce secteur d'activités ne se réclament pas explicitement de l'analogie, comme dans {64} par exemple. C'est KLING {87} qui définit pour la première fois, à notre connaissance, un paradigme pour le raisonnement analogique. Il a implémenté un système particulier (ZORBA-1) qui permet de démontrer un nouveau théorème T' grâce à une analogie avec la démonstration connue d'un théorème T déjà prouvé, ceci afin de limiter le nombre de clauses à générer. A partir de là, d'autres travaux très récents - notamment en représentation des connaissances - ont mis en oeuvre explicitement ce mode de raisonnement, fondé essentiellement sur la ressemblance de structures que le concept de "Matching", bien connu en intelligence artificielle, permet d'évaluer de manières diverses ; citons par exemple {32}, {65} et {88}.

Compte tenu des hypothèses qui fondent le cadre de notre recherche (en particulier mise en oeuvre de raisonnements hypothético-déductifs sur des univers incomplètement décrits), nous avons étendu l'activité inférentielle du système symbolique ARCHES en définissant - à côté du

raisonnement déductif (voir chapitre huitième) - un type particulier de raisonnement par analogie. Le modèle analogique que nous proposons (voir § II.) et la règle d'inférence analogique correspondante (voir § III.) permettent de construire un démonstrateur analogique (voir § IV.) qui produit des preuves d'énoncés qui *rendent satisfiable et consistant le système ARCHES.*

II. PARADIGME ANALOGIQUE DANS LE SYSTEME SYMBOLIQUE ARCHES.

II.1. Hypothèses sur les fondements du raisonnement analogique.

Nous supposons que les données véhiculées par ARCHES peuvent être *incomplètes* sous le double point de vue de leur caractérisation et de leur organisation. Les travaux relatifs à des domaines de connaissances empiriques non théorisés relèvent le plus souvent d'une telle situation (voir § II., chapitre premier). Or les raisonnements déductifs - et donc valides - ne sont pas totalement opératifs dans des univers où les connaissances sont incomplètement définies. Là est la raison essentielle de l'insertion dans ARCHES d'un *mode de raisonnement fondé sur l'analogie* (voir § I.) : Pour pallier à ces insuffisances, les spécialistes mettent en oeuvre dans leurs secteurs d'activités des raisonnements argumentatifs dont l'objet est de pouvoir inférer de nouvelles connaissances en confrontant d'autres connaissances qui se "ressemblent" ; l'argumentation développée pour justifier la nature et la pertinence de ces ressemblances prenant appui sur leur savoir et leur savoir-faire (voir § II.1., chapitre premier ; pour une utilisation de l'analogie dans les sciences de l'homme se reporter à {48} et {72}, par exemple).

On appelle *sous-structure* d'une structure $S(x, \mathcal{D}(x))$ toute formule de la forme $S(x, d_x)$ telle que $\mathcal{D}(x) \Longrightarrow d_x$.

Nous rappelons que dans ARCHES tout individu est décrit en extension par une seule structure (voir § III.3., chapitre septième). Il en résulte que $S(x, d_x)$ n'est pas explicitement stockée dans ARCHES, et que la relation $S(x,\mathcal{D}(x)) \longmapsto S(x, d_x)$ est valide (voir § IV.2.2.a, chapitre septième). Par ailleurs il est évident que toute structure est sous-structure d'elle-même, du fait de la réflexité de la relation $\Longrightarrow$ (voir § IV. , chapitre quatrième). Dans la suite, nous désignons par P et T deux structures quelconques (donc enregistrées dans ARCHES), par A et B deux sous-structures de P, et enfin par C et D deux sous-structures de T.

Compte-tenu des objectifs qui ont présidé à la conception de ARCHES (en particulier, voir chapitre deuxième), le paradigme analogique formalisé dans ce système est fondé sur la *similitude de sous-structures* dont la formulation la plus générale est :

"A est à B ce que C est à D"

Il exprime une *ressemblance de rapports* dont le type parfait est la proportion géométrique (voir § I. ; le symbole == exprimant la ressemblance entre les rapports $\frac{A}{B}$ et $\frac{C}{D}$) :

$$\frac{A}{B} == \frac{C}{D}$$

Ce paradigme correspond à nos objectifs, car il permet effectivement de déduire au vue de "certaines ressemblances" une connaissance nouvelle (D) à partir de connaissances connues (A, B, et C) mais éventuellement incomplètes (en effet dans le cas général A et C sont différentes respectivement des structures P et T, qui elles-mêmes sont incomplètement décrites). L'*inférence analogique* correspondant à ce paradigme s'exprime comme suit (figure 9.1.) :

	A de P *ressemble* à C de T
OR	B *dérive* de P
DONC	D *dérive* de T

Un cas particulier important est celui qui correspond à un *transfert* de description partielle de la structure P vers la structure T ; donc les sous-structures B et D véhiculent des descriptions qui ont en commun des éléments de description.

Cette inférence analogique fournit des énoncés de conséquence de la forme : $T \vdash D$ (voir § V., chapitre septième), dont la *validité logique n'est pas postulée* car elle opère sur des univers incomplètement décrits (voir § I.). Il est évident que le non-contrôle de cette validité peut conduire à produire de nouvelles connaissances qui rendent le système ARCHES contradictoire comme le montre l'exemple suivant.

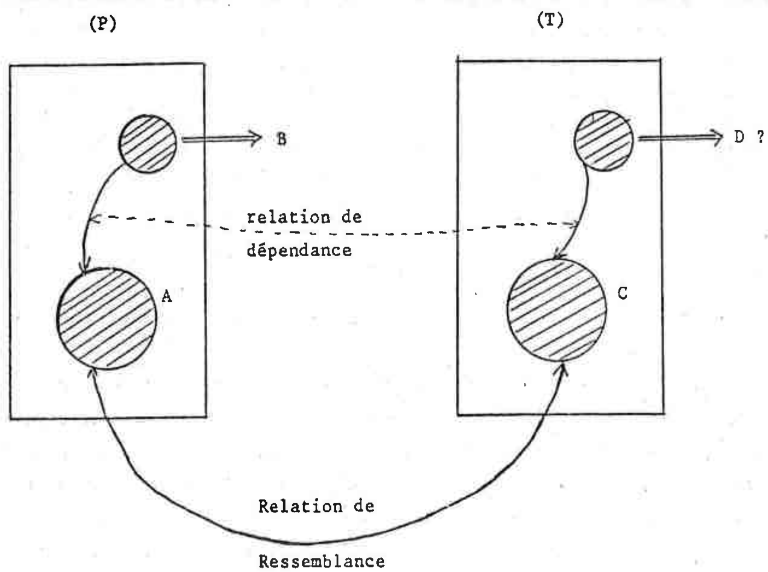


figure 9.1. - *Représentation du paradigme analogique du système ARCHES.*

Exemple 1.

Soit P et T les deux structures ci-après :

$$\begin{cases} \text{ASTRE(TERRE,}\mathcal{D}\text{(TERRE))} & \text{(P)} \\ \text{ASTRE(LUNE,}\mathcal{D}\text{(LUNE))} & \text{(T)} \end{cases}$$

dans lesquelles $\mathcal{D}$(TERRE) et $\mathcal{D}$(LUNE) sont respectivement les descriptions de la TERRE et de la LUNE à l'intérieur du champ ASTRE.

Posons :

A = ASTRE(TERRE,ISA(FORME,RONDE)) avec $\mathcal{D}$ (TERRE) $\Longrightarrow$ ISA(FORME,RONDE)

B = ASTRE(TERRE,ISA(PEUPLEMENT,HOMME)) avec $\mathcal{D}$ (TERRE) $\Longrightarrow$ ISA(PEUPLEMENT, HOMME)

C = ASTRE(LUNE,ISA(FORME,RONDE)) avec $\mathcal{D}$ (LUNE) $\Longrightarrow$ ISA(FORME,RONDE)

Nous souhaitons démontrer l'énoncé "D=ASTRE(LUNE,ISA(PEUPLEMENT,HOMME))?" , sachant qu'il n'a pu être prouvé au moyen du raisonnement déductif (voir chapitre huitième). La règle d'inférence analogique définie ci-dessus se particularise, à l'aide des quatre sous-structures A, B, C et D, de la manière suivante :

	La TERRE et la LUNE ont la même forme
OR	La TERRE est habitée par des hommes
DONC	La LUNE est habitée par des hommes

Bien que la ressemblance soit totale entre les descriptions véhiculées par les sous-structures A et C (i.e. la *même* forme), cette inférence est manifestement non valide. Cette non-validité est dûe d'abord au fait que l'inférence est fondée sur la ressemblance de deux descriptions très incomplètes de la TERRE et de la LUNE : en effet, une seule propriété a été prise en compte pour appliquer la règle d'inférence. Mais elle est dûe également au fait que les propriétés "la TERRE est ronde" et "la TERRE est habitée par des hommes" n'entretiennent aucune relation sémantique (évidente et/ou connue), ou d'une manière plus générale aucune relation de dépendance. Enfin, nous avons fait implicitement l'hypothèse que le transfert d'une propriété de la structure P vers la structure T est valide.

Cet exemple (et l'analyse de sa non-validité) est très représentatif des problèmes que pose la mise en oeuvre des raisonnements analogiques. Il nous permet d'énoncer les hypothèses qui fondent le paradigme analogique du système ARCHES.

Hypothèse 1. *Le raisonnement analogique est mis en oeuvre toutes les fois qu'un énoncé D, sous-structure d'une structure T, n'a pu être prouvé à partir du raisonnement déductif.*

Cette hypothèse montre clairement que l'une de nos préoccupations essentielles est d'étendre l'activité inférentielle du système ARCHES. Si le raisonnement déductif est intéressant pour une large classe de problèmes (voir chapitre huitième), il est bien évident que ce type de raisonnement est insuffisant pour dériver de nouvelles connaissances à partir de domaines incomplètement décrits. C'est l'objet du raisonnement analogique. La composition de ces deux modes de raisonnement - le premier valide, et le second dépendant des applications, donc des interprétations - permet ainsi de construire par emboîtement deux catégories de connaissances nouvelles : les connaissances valides et les connaissances satisfiables pour une interprétation donnée (voir § III.3.).

Hypothèse 2. *Si B et D, respectivement sous-structures des structures P et T, ont été prouvées dans ARCHES et se ressemblent alors il existe au moins deux énoncés A et C, respectivement sous-structures de P et T et distincts de B et D, qui se ressemblent.*

La ressemblance est une notion vague est très générale dont la mesure dépend non seulement de l'organisation des informations à comparer, mais aussi des problèmes à résoudre (pour des mesures particulières de la ressemblance voir, par exemple, dans {15} les chapitres II et III). Cette notion est définie dans ARCHES à partir de la relation de déduction $\Longrightarrow$ et de ses propriétés (voir § III.1.). La pertinence de cette ressemblance contribue à accroitre dans une inférence analogique le *degré de vraisemblance* de la conclusion D quand on connaît les trois autres termes A, B et C. Ce degré de vraisemblance dépend de deux facteurs : d'une part, il est d'autant plus grand que la ressemblance entre les sous-structures A et C est forte, c'est à dire que cette ressemblance tend vers l'*identité* ; d'autre part, il est d'autant plus grand que la quantité d'information véhiculée par les sous-structures A et C est importante, c'est à dire que cette quantité tend vers l'*exhaustivité* des informations véhiculées par P et T. Ainsi la limite du raisonnement analogique dans ARCHES est déterminée par deux valeurs particulières : l'identité représentant la ressemblance maximum, et l'exhaustivité exprimant la prise en compte d'un univers complètement décrit ; c'est donc le raisonnement déductif (voir § I.).

Hypothèse 3. *Si A' (respectivement C'), sous-structure de P (respectivement T), ne ressemble à aucune sous-structure de T (respectivement P) alors A' (respectivement C') n'a aucune influence dans le déclenchement d'une inférence analogique.*

Cette hypothèse a pour objet de ne prendre en compte dans le raisonnement analogique mis en oeuvre dans ARCHES que des *informations qui tendent à favoriser l'inférence analogique* (qu'on pourrait qualifier de critères positifs, voir hypothèse 2), au détriment de *celles qui tendent à s'opposer au déclenchement de l'inférence analogique* (qu'on pourrait qualifier de critères négatifs). A priori il n'y a aucune raison de privilégier les critères positifs par rapport aux critères négatifs. Ces derniers peuvent même jouer un rôle essentiel dans le contrôle de la validité de l'inférence analogique. C'est en particulier le cas dans l'exemple 1 ; en effet la prise en compte du critère négatif "Présence/Absence d'une atmosphère autour d'une planète" aurait interdit le déclenchement de l'inférence analogique. Cependant ces choix trouvent leur justification dans la définition de l'hypothèse 4.

Hypothèse 4. *Une inférence analogique ne peut être déclenchée que s'il existe une relation de dépendance effective, notée DEPENDANCE, entre d'une part les sous-structures A et B, et d'autre part entre les sous-structures C et D.*

Cette relation de dépendance indique que la réalisation de B (resp. D) est conditionnée par l'existence de A (resp. C) (figure 9.1.). Le degré de vraisemblance de la conclusion D est d'autant plus élevé que la relation de dépendance est plus forte. Les *relations de dépendance* peuvent être fixées par des spécialistes qui utilisent le savoir et le savoir-faire des domaines dans lesquels ils opèrent pour les justifier. Elles *définissent des hypothèses* contribuant à résoudre les problèmes posés, leur validation constituant une étape essentielle et indispensable de toute démarche expérimentale cohérente (voir § II.3., chapitre premier). Cependant elles peuvent également être déterminées automatiquement par le système ARCHES (voir § III.3.). Dans l'exemple 1 il n'y a aucune dépendance entre les relations d'état (FORME,ISA) et (PEUPLEMENT,ISA). Par contre si on avait indiqué au système que la réalisation de la relation (PEUPLEMENT,ISA) était conditionnée par l'existence d'une atmosphère entourant les planètes (relation d'état (ELEMENT,ISA) par exemple) alors l'inférence analogique n'aurait pu être déclenchée. Remarquons que cette relation de dépendance permet de véhiculer des critères négatifs (voir hypothèse 3). Nous les exprimerons dans ces termes toutes les fois qu'ils auront une influence essentielle dans les mécanismes analogiques. Il est évident que cette manière de prendre en compte les critères négatifs est beaucoup plus efficace : sélection argumentée des critères négatifs influents, gain de place-mémoire, etc.

Hypothèse 5. *Les sous-structures B et D véhiculent des descriptions qui ont en commun des éléments de description.*

En d'autres termes l'activité inférentielle du raisonnement analogique défini dans ARCHES a pour objet essentiel de *transférer une description partielle* (celle véhiculée par B) de la structure P vers la structure T.

Ces cinq hypothèses, qui fixent le cadre méthodologique d'un mode particulier de raisonnement analogique, sont à la source de la construction du *modèle analogique* mis en oeuvre dans ARCHES (voir § II.2.) et de la définition de la *règle d'inférence analogique* correspondante (voir § III.2.).

II.2. Le modèle analogique dans le système ARCHES.

II.2.1. Analyse de quelques exemples.

Exemple 2.

Considérons le champ VILLE décrivant la géographie des villes européennes (géographie humaine, physique, économique, climatique,...). Et soit P et T les deux structures associées aux villes MARSEILLE et ROME :

$$\begin{cases} \text{VILLE(MARSEILLE,}\mathcal{D}\text{(MARSEILLE))} & \text{(P)} \\ \text{VILLE(ROME,}\mathcal{D}\text{(ROME))} & \text{(T)} \end{cases}$$

Par ailleurs désignons par A, B et C les sous-structures ci-après :

$$\begin{cases} \text{A = VILLE(MARSEILLE,ISA(SITUATION,CÔTIERE)*ISA(LATITUDE,MOYENNE))} \\ \text{B = VILLE(MARSEILLE,ISA(CLIMAT,TEMPERE))} \\ \text{C = VILLE(ROME,ISA(SITUATION,CÔTIERE)*ISA(LATITUDE,MOYENNE))} \end{cases}$$

Nous souhaitons démontrer l'énoncé "D=VILLE(ROME,ISA(CLIMAT,x))?", sachant qu'il n'a pu être prouvé à l'aide du raisonnement déductif (hypothèse 1). Pour ce faire, *nous supposons que le climat des villes européennes ne dépend que de leur situation géographique et de leur latitude* (hypothèse 4)(voir § III.3.1.; figure 9.3.) :

$$\begin{cases} \text{DEPENDANCE(ISA(CLIMAT,x),ISA(SITUATION,y))} \\ \text{DEPENDANCE(ISA(CLIMAT,x),ISA(LATITUDE ,z))} \end{cases}$$

Désignons par U_E et R_E les ensembles dont les éléments sont respectivement les traits et les relations d'état représentées par les doublets (classe, opérateur) qui composent toute description E. Soit $R_A \cup U_A$, $R_B \cup U_B$, $R_C \cup U_C$ et $R_D \cup U_D$ les ensembles associés respectivement aux sous-structures A, B, C et D ; et soit f la fonction qui transforme $R_A \cup U_A$ en $R_C \cup U_C$ et $R_B \cup U_B$ en $R_D \cup U_D$:

$$\{\text{(SITUATION,ISA),(LATITUDE,ISA)}\} \cup \{\text{CÔTIERE,MOYENNE}\} \xrightarrow{f} \{\text{(SITUATION,ISA),}$$
$$\text{(LATITUDE,ISA)}\} \cup \{\text{CÔTIERE,MOYENNE}\}$$

$$\{\text{(CLIMAT,ISA)}\} \cup \{\text{TEMPERE}\} \xrightarrow{f} \{\text{(CLIMAT,ISA)}\} \cup \{x\}$$

Cette transformation suppose nécessairement que $R_A \cap R_C \neq \phi$ et que $R_B \cap R_D \neq \phi$; il en résulte que f ne peut exister que si certaines relations d'état sont communes respectivement aux sous-structures A et C d'une part et aux sous-structures B et D d'autre part. Par ailleurs elle tient compte de la sémantique des traits : deux traits ne peuvent être associés dans cette transformation que s'ils appartiennent à la même classe, et les éléments qui sont identiques dans les ensembles cible et source se correspondent de manière biunivoque.

Pour démontrer l'énoncé D, il faut être en mesure de déterminer la valeur que prend la fonction f au point TEMPERE : x = f(TEMPERE). Nous remarquons à cet effet que f représente la *fonction d'identité* pour tous les points autresque le point TEMPERE. Cette fonction d'identité indique tout simplement que la ressemblance entre les descriptions véhiculées par A et C est maximum (hypothèse 2). Nous sommes donc dans un cas extrêmement favorable : d'une part les critères de dépendance sont vérifiés ; d'autre part les sous-structures A et C véhiculent des descriptions identiques. *Le raisonnement analogique consiste alors à inférer que f représente également la fonction d'identité au point TEMPERE* (ce qui est conforme avec l'hypothèse 5) ; il en résulte que :

$$x = f(\text{TEMPERE}) = \text{TEMPERE}$$

L'énoncé D est donc démontré.

Remarque.

Il est possible de trouver des contre-exemples de même nature où l'inférence analogique conduit à une contradiction. Par exemple NAPLES et NEW-YORK ont la même situation géographique et des latitudes comparables ; pourtant elles ont des climats très différents. Mais dans ce cas NEW-YORK n'appartient pas à l'ensemble des villes européennes pour lesquelles il est supposé que leur climat ne dépend que de leur situation géographique et de leur latitude. Si on étend l'aire géographique d'investigation, il faut bien évidemment compléter et/ou modifier les critères de dépendance. Pour cet exemple, il faudrait, en particulier, préciser que le climat dépend aussi de la LONGITUDE et de la nature des COURANTS MARINS. Cette remarque montre bien le caractère hypothétique des critères de dépendance que seuls les spécialistes sont à même d'évaluer et de justifier par rapport à leur savoir. Vu sous cet angle, le raisonnement analogique apparait comme un raisonnement hypothético-déductif.

Exemple 3.

Considérons le champ AMPHORE décrivant un ensemble d'amphores sous des points de vue différents (morphologique, historique, fonctionnel,...) ; et soit P et T deux structures particulières représentant les amphores N1 et N2 (figure 9.2.) :

$$\begin{cases} \text{AMPHORE}(\text{N1}, \mathcal{D}(\text{N1})) & \text{(P)} \\ \text{AMPHORE}(\text{N2}, \mathcal{D}(\text{N2})) & \text{(T)} \end{cases}$$

Par ailleurs désignons par A, B et C les trois sous-structures ci-après :

A = AMPHORE(N1,ISA(PARTIE,\$(INS(TIMBRE,T1),ON(LOC,LEVRE)))*ISA(PARTIE,
\$(INS(TIMBRE,T1),ISA(FORME,\$(CIRCONFERENCE,
EG(DIAMETRE,30)))))*ISA(PARTIE,\$(INS(TIMBRE,
T1),ISA(TEXTE,"ASCL"))))

B = AMPHORE(N1,IN(LOC,INS(FOUR.F1)))

C = AMPHORE(N2, ISA(PARTIE,\$(INS(TIMBRE,T2),ON(LOC,LEVRE)))*ISA(PARTIE,
\$(INS(TIMBRE,T2),ISA(FORME,\$(CIRCONFERENCE,
EG(DIAMETRE,30)))))

A indique que l'amphore N1 possède un timbre T1 qui est localisé sur la lèvre, qui a la forme d'une circonférence de diamètre égal à 30, et enfin

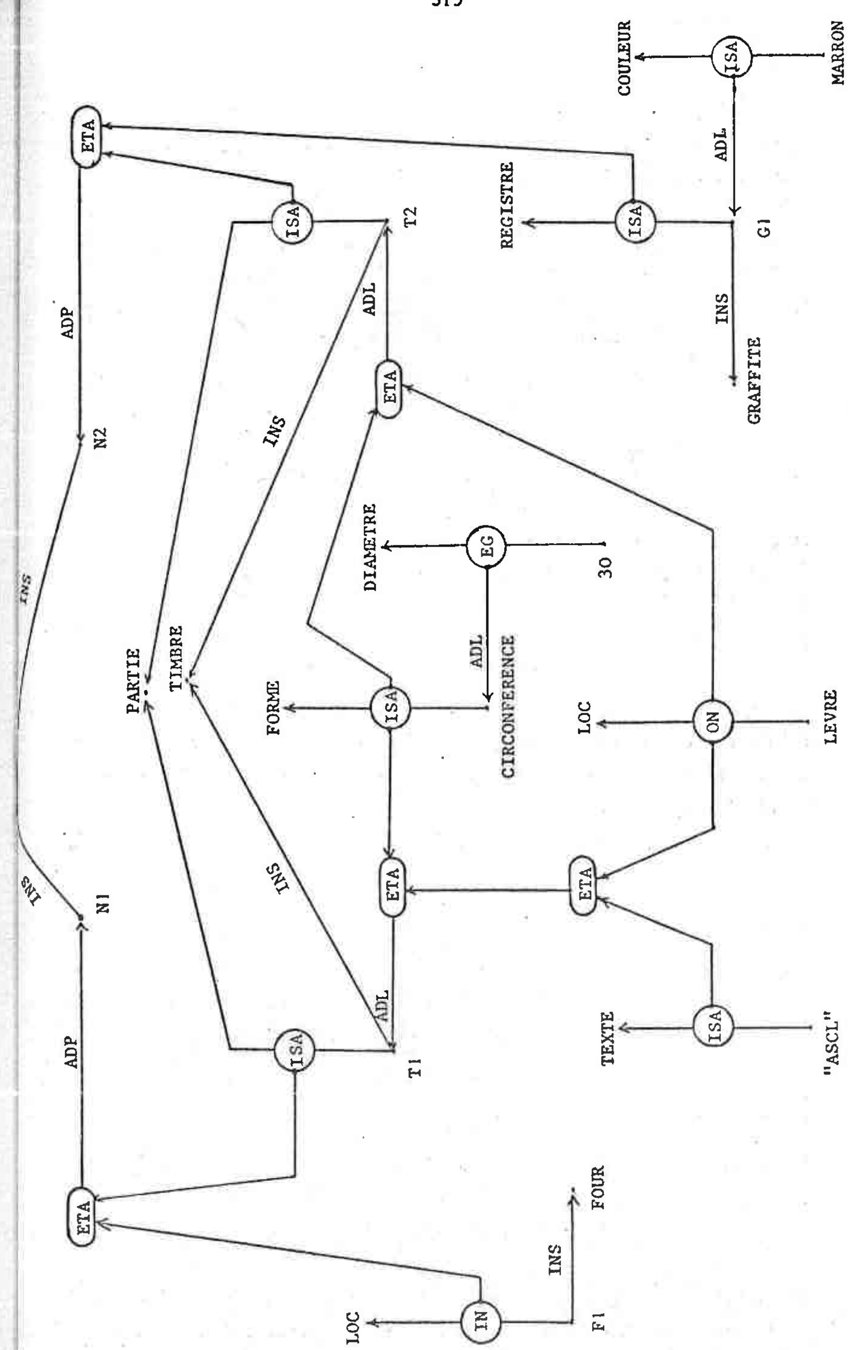


figure 9.2. - Extrait du champ AMPHORE représentant des éléments de description des amphores N1 et N2.

qui porte l'inscription "ASCL". B indique que l'amphore N1 est localisée dans le four F1. Enfin C indique que N2 possède un timbre T2 qui est localisé sur la lèvre, et qui a la forme d'une circonférence de diamètre égal à 30 (l'inscription de ce timbre est effacée).

Nous souhaitons démontrer l'énoncé "D=AMPHORE(N2,IN(LOC,INS(FOUR,x))) ?". Pour ce faire nous supposons que la localisation des amphores étudiées ne dépend que de la description de leur timbre défini comme une partie d'amphore (hypothèse 4) (voir § III.3.1. ; figure 9.3.) :

DEPENDANCE(IN(LOC,x),ISA(PARTIE,$(INS(TIMBRE,y),z)))

Comme dans l'exemple précédent, soit f la fonction qui transforme d'une part l'ensemble $R_A \cup U_A$ en l'ensemble $R_C \cup U_C$, et d'autre part l'ensemble $R_B \cup U_B$ en l'ensemble $R_D \cup U_D$. Cette transformation est possible car $R_A \cap R_C \neq \phi$ et $R_B \cap R_D \neq \phi$.

{(PARTIE,ISA),(LOC,ON),(FORME,ISA),(DIAMETRE,EG),(TEXTE,ISA)} $\xrightarrow{f}$ {(PARTIE,ISA),(LOC,ON),(FORME,ISA),(DIAMETRE,EG)}

{T1,LEVRE,CIRCONFERENCE,30,"ASCL"} $\xrightarrow{f}$ {T2,LEVRE,CIRCONFERENCE,30}

{(LOC,IN)}∪{F1} $\xrightarrow{f}$ {(LOC,IN)}∪{x}

Là aussi pour prouver D, il faut déterminer la valeur que prend la fonction f au point F1 : x=f(F1). Mais dans ce cas elle est plus complexe que dans l'exemple 2. Nous sommes en présence de trois situations différentes : /1/ f représente localement la fonction d'identité comme f(LEVRE)=LEVRE ou f(30)=30 par exemple ; /2/ f représente localement une fonction quelconque définie en extension comme c'est le cas au point T1 : f(T1)=T2 ; /3/ enfin f a des points singuliers, c'est à dire que certaines valeurs n'ont pas de transformées (ou inversement) comme le doublet (TEXTE,ISA) ou le trait "ASCL". Dans ces conditions quelle est la valeur de x dans l'énoncé x=f(F1) ? En d'autres termes comment définir la fonction f au point F1 ? Pour ce faire, *nous supposons que les structures A et C se ressemblent si et seulement si elles possèdent en commun des éléments de description, c'est à dire si f représente localement la fonction d'identité* (hypothèse 2). Comme cette hypothèse est vérifiée (les timbres T1 et T2 ont la même description à l'exception de

l'inscription qui est présente dans T1 et effacée dans T2), le raisonnement analogique consiste à inférer que f représente la fonction d'identité au point F1 (hypothèse 5), il en résulte que :

x=f(F1)=F1

L'énoncé D est donc démontré.

Exemple 4.

Considérons dans le champ ANIMAL les deux structures suivantes :

ANIMAL(HOMME, $\mathcal{D}$(HOMME))	(P)
ANIMAL(POISSON, $\mathcal{D}$(POISSON))	(T)

De plus désignons par A, B et C les trois sous-structures ci-après :

A = ANIMAL(HOMME,ISA(ORGANE,POUMON))
B = ANIMAL(HOMME,ISA(ORGANE,$(POUMON,ISA(FONCTION,RESPIRATOIRE))))
C = ANIMAL(POISSON,ISA(ORGANE,BRANCHIE))

(Remarquons que dans ce cas B $\longmapsto$ A , par application de la règle de décomposition)

Nous souhaitons prouver "D=ANIMAL(POISSON,ISA(ORGANE,$(BRANCHIE,ISA (FONCTION,RESPIRATOIRE))))?".
Pour ce faire nous supposons que la fonction dépend de l'organe :

DEPENDANCE(ISA(FONCTION,x),ISA(ORGANE,y))

Désignons par f la fonction qui transforme $R_A \cup U_A$ en $R_C \cup U_C$ d'une part, et d'autre part $R_B \cup U_B$ en $R_D \cup U_D$ (transformation possible car $R_A \cap R_C \neq \phi$ et $R_B \cap R_D \neq \phi$) :

{(ORGANE,ISA)}∪{POUMON} $\xrightarrow{f}$ {(ORGANE,ISA)}∪{BRANCHIE}

{(ORGANE,ISA),(FONCTION,ISA)}∪{POUMON,RESPIRATOIRE} $\xrightarrow{f}$ {(ORGANE,ISA),(FONCTION,ISA)}∪{BRANCHIE,RESPIRATOIRE}

Dans ce cas, on a BRANCHIE=f(POUMON) : les structures A et C ne se ressemblent donc pas car f ne représente jamais localement la fonction d'identité. On ne peut donc pas conclure que RESPIRATOIRE=f(RESPIRATOIRE) (hypothèse 2 et 5). Il en résulte que le raisonnement analogique ne permet pas de démontrer l'énoncé D (il s'agit ici d'une analogie de fonction, voir à ce sujet {51}).

Ces trois exemples montrent que le raisonnement analogique peut être modélisé en termes d'applications particulières qui transforment les éléments de A et de B respectivement en ceux de C et de D. L'évaluation de ces applications (resp. l'impossibilité de leur évaluation) permet alors de dériver (resp. de ne pas démontrer) l'énoncé D à partir de la donnée des trois termes A, B et C comme dans les exemples 2 et 3 (resp. l'exemple 4). L'objet du paragraphe suivant est de présenter précisément la description formelle de ce modèle.

II.2.2. Définition du modèle analogique.

Le modèle analogique de ARCHES est défini comme une application particulière entre deux structures P et T, dont l'évaluation fondée sur des rapports de ressemblance entre sous-structures permet de dériver dans T une connaissance nouvelle par transfert d'une description partielle de P vers T.

Plus précisément le modèle analogique est composé par les éléments suivants :

α) Une structure T à partir de laquelle doit être dérivée la sous-structure D (qui n'a pu être prouvée à l'aide du raisonnement déductif). Soit R_D et U_D respectivement l'ensemble des doublets (classe, opérateur) (i.e. les relations d'état) et l'ensemble des traits qui composent la sous-structure D.

β) Une relation de dépendance entre $R'_D \subset R_D$ et un ensemble R_e de relations d'état donné à priori.

γ) Une sous-structure B de P qui donne naissance aux ensembles R_B et U_B, et qui ressemble à D. On dit que B et D se ressemblent si et seulement si $R_B=R_D$; il en résulte que R_B dépend également de R_e.

δ) Deux sous-structures A et C, respectivement de P et T, qui donnent naissance aux ensembles R_A, R_C, U_A et U_C tels que $R_e \subset R_A$ et $R_e \subset R_C$ (on a donc $R_A \cap R_C \neq \phi$).

ε) Une application f qui permet de transformer d'une par $R_A \cup U_A$ en $R_C \cup U_C$ et d'autre par $R_B \cup U_B$ en $R_D \cup U_D$:

$$\begin{tikzcd} R_A \cup U_A \arrow[r] & R_C \cup U_C \\ R_B \cup U_B \arrow[r] \arrow[u, "\text{DEPENDANCE}"] & R_D \cup U_D \arrow[u, "\text{DEPENDANCE}"'] \end{tikzcd}$$

Cette fonction f n'a d'existence que si les deux conditions suivantes sont vérifiées : /1/ $R_B=R_D$; /2/ $R_A \cap R_C \neq \phi$ (conditions imposées par les points (γ) et (δ) du modèle).

ζ) Une méthode d'évaluation de f qui est définie comme suit : /1/ les éléments qui sont identiques dans les deux groupes d'ensembles se correspondent de manière biunivoque, on a donc en particulier $f(R_B)=R_D=R_B$ et $f(R_e)=R_e \subset R_A \cap R_C$; /2/ l'application respecte la sémantique des traits : deux traits se correspondent dans l'application f si et seulement si ils appartiennent à la même classe ; /3/ l'application f n'est pas définie pour les autres éléments.

η) Une règle d'évaluation analogique de la preuve de D qui s'exprime comme suit : Si f représente localement la fonction d'identité dans la transformation $U_A \longrightarrow U_C$, c'est à dire qu'il existe $U'_A \subset U_A$ tel que $f(U'_A)=U'_A \subset U_C$ et si U'_D désigne l'ensemble de toutes les variables, inclus dans U_D et éventuellement vide, tel que $U'_D = f(U'_B)$ avec $U'_B \subset U_B$, alors f représente la fonction d'identité pour les points de U'_B, d'où $U'_D = U'_B$. Si après évaluation on a $U_B \cap U_D \neq \phi$, alors *on infère analogiquement D*. En effet dans ce cas, B et D véhiculent des descriptions qui ont en commun des éléments de description (précisément ceux définis par $U_B \cap U_D$), ce qui est tout à fait conforme avec l'hypothèse 5 sous-jacente à la définition du modèle analogique (voir § II.1.).

Nous devons remarquer que c'est la fonction d'identité qui permet d'évaluer la ressemblance entre les sous-structures A et C. Si f ne représente jamais la fonction d'identité, alors A et C sont différents et on ne peut pas prouver par analogie la sous-structure D.

III. MODALITES DE MISE EN OEUVRE DU RAISONNEMENT ANALOGIQUE DANS LE SYSTEME SYMBOLIQUE ARCHES.

III.1. Définitions préliminaires.

Nous nous proposons de définir à partir du modèle analogique (voir § II.2.2.) les modalités de mise en oeuvre du raisonnement analogique dans le système ARCHES. Ces modalités utilisent quelques notions particulières dont les définitions et les propriétés sont présentées dans ce paragraphe.

définition 1. On appelle *schéma de description* toute description dans laquelle tous les traits sont représentés par des variables.

Les schémas de description sont désignés par des expressions de la forme E(x,y,z,...) dans laquelle l'ordre des variables x, y, z,... correspond à celui dans lequel elles se rencontrent pour la première fois quand l'expression E est explorée de la gauche vers la droite.

Exemple 5.

E(x,y,z)=ISA(PARTIE,$(INS(TIMBRE,x),ON(LOC,y)))*
ISA(PARTIE,$(INS(TIMBRE,x),ISA(FORME,z)))

définition 2. Etant donné un schéma de description $E(v_1,..,v_i,.,v_n)$ tel que $\forall ij\ v_i \neq x_j$ (en renommant éventuellement les variables si cela est nécessaire), on appelle *x-standardisation* de $E(v_1,..,v_i,.,v_n)$ l'expression $E\xi$ dans laquelle ξ est la *substitution de standardisation*.

$$\xi=\{<v_i,x_i>|1\leq i\leq n\}$$

La x-standardisation a pour objet de renommer dans tout schéma de description E ses variables par les variables standard $x_1,\dots,x_i,\dots,x_n$, x_i devant remplacer la Iième variable qui se rencontre pour la première fois dans E.

définition 3. Deux schémas de description x-standardisés E_1 et E_2 sont égaux si et seulement si les chaînes de caractères qui composent E_1 et E_2 sont identiques :

$$E_1=E_2$$

(cette notation se justifie car on a bien évidemment $E_1 \Longrightarrow E_2$ et $E_2 \Longrightarrow E_1$, voir § IV.1.2., chapitre quatrième).

Propriété 1. Si deux schémas de description E_1 et E_2 sont tels que $E_1\xi=E_2\xi$, alors E_1 et E_2 sont $\Longrightarrow$-unifiables (évident).

La définition 3 et la propriété 1 permettront désormais de supposer que tous les schémas de description sont x-standardisés. A cet effet, nous définissons le prédicat SQELET(D,E) qui est vrai si et seulement si E est le schéma de description x-standardisé de la description D.

Propriété 2. Si $\vdash$ SQELET(D,E) alors D et E sont $\Longrightarrow$-unifiables (évident).

Propriété 3. Si les deux relations $D \Longrightarrow D_1$ et SQELET(D_1,E) sont simultanément vérifiées, alors D et E sont $\Longrightarrow$ -unifiables.

(immédiat à partir de la propriété 2 et de la transitivité de la relation $\Longrightarrow$.)

La propriété 4 ci-après est une conséquence immédiate de la propriété 3 :

Propriété 4. Etant donné la structure A(x,D), si les deux relations $A(x,D) \longmapsto A(x,D')$ et $\vdash$SQELET(D',E) sont simultanément vérifiées, alors la relation $A(x,D) \longmapsto A(x,E)$ est également vérifiée.

définition 4. On dit qu'un *schéma de description est lié* si et seulement si au moins une de ses variables est liée.

En d'autres termes un schéma de description est lié si et seulement si il existe au moins un de ses traits qui a pour valeur une constante ; en particulier une description est un schéma de description lié. Toutes les définitions et propriétés caractérisant les schémas de description sont également valides pour les schémas de description liés. Nous supposerons donc que tous les schémas de description liés sont x-standardisés. A cet effet nous définissons le prédicat SQTLIE(E) qui est vrai si et seulement si E est un schéma de description lié.

Remarquons en particulier que si deux schémas de description liés E_1 et E_2 sont égaux alors ils possèdent en commun des éléments de description. En effet, si on désigne par R_1, R_2, U_1 et U_2 les ensembles de relations d'état (doublet (classe, opérateur)) et de traits associés respectivement à E_1 et E_2, on a :

$$R_1 = R_2 \quad \text{et} \quad U_1 \cap U_2 \neq \phi$$

Il en résulte que les *deux schémas de description liés E_1 et E_2 se ressemblent au sens du modèle analogique* (voir § II.2.2., point (n)). Il est donc possible d'évaluer la ressemblance de deux sous-structures quelconques à l'aide de la relation de déduction $\Longrightarrow$ (voir § IV., chapitre quatrième).

définition 5. Deux sous-structures $P(x, \mathcal{A}_x)$ et $T(y, \mathcal{C}_y)$ se ressemblent dans le système ARCHES si et seulement si les trois relations ci-après sont simultanément vérifiées :

$$\left|\begin{array}{ll} \mathcal{A}_x \Longrightarrow E & (1) \\ \mathcal{C}_y \Longrightarrow E & (2) \\ \mathrm{SQTLIE(E)} & (3) \end{array}\right.$$

En d'autres termes deux sous-structures se ressemblent si et seulement si on peut *dériver* de leurs descriptions le même schéma de description lié. En effet on peut déduire de (1) et de (2) par application dans ARCHES de la règle d'inférence intra-champ les deux relations d'inférence

$P(x, \mathcal{A}_x) \longmapsto P(x,E)$ et $T(y, \mathcal{C}_y) \longmapsto T(y,E)$ (voir § IV.2.2.a, chapitre septième). Ces deux sous-structures possèdent donc en commun des éléments de description ce qui est tout à fait conforme à la notion de ressemblance qui a été introduite dans le modèle analogique (voir § II.2.2.): *Cette ressemblance est donc fondée essentiellement sur l'opération de* $\Longrightarrow$ *- unification* (voir § II.2., chapitre huitième).

Par ailleurs si nous supposons que les deux descriptions $\mathcal{A}_x$ et $\mathcal{C}_y$ ont le même schéma de description, alors leur ressemblance peut s'exprimer à l'aide d'un prédicat particulier. Cette hypothèse consiste à affirmer que les trois relations suivantes sont simultanément vérifiées : $\vdash \mathrm{SQELET}(\mathcal{A}_x, E_1)$; $\vdash \mathrm{SQELET}(\mathcal{C}_y, E_2)$; et enfin $E_1 = E_2$. Dans ces conditions $\mathcal{A}_x$ et $\mathcal{C}_y$ se ressemblent si et seulement si *elles possèdent en commun* un schéma de description lié E. Soit RECSQT(D_1, D_2, E) le prédicat qui est vrai si et seulement si E est le schéma de description lié commun aux descriptions D_1 et D_2. Il en résulte que $\mathcal{A}_x$ et $\mathcal{C}_y$ se ressemblent si et seulement si $\vdash \mathrm{RECSQT}(\mathcal{A}_x, \mathcal{C}_y, E)$.

III.2. Règle d'inférence analogique.

II.2.1. Définition.

Nous souhaitons démontrer dans la structure $T(y, \mathcal{D}(y))$ la sous-structure $T(y, \mathcal{E}_y)$, qui n'a pu être prouvée à l'aide du raisonnement déductif (chapitre huitième). On se donne pour ce faire une structure $P(x, \mathcal{D}(x))$ à partir de laquelle $P(x, \mathcal{E}_x)$ a pu être dérivée telle que la relation $\vdash \mathrm{RECSQT}(\mathcal{E}_x, \mathcal{E}_y, E)$ est vérifiée, i.e. les descriptions $\mathcal{E}_x$ et $\mathcal{E}_y$ ont des éléments de description en commun (voir § II.2.2. ; point (β)). Le théorème $T(y, \mathcal{E}_y)$ est alors démontré analogiquement si les prémisses de la règle de dérivation définie ci-après (définition 6) sont toutes vérifiées. Cette règle sera appelée *règle d'inférence analogique du système ARCHES* (voir § IV.2.2.b, chapitre septième).

définition 6. Si les conditions de ressemblance, au sens du modèle analogique, sont vérifiées entre deux structures $P(x, \mathcal{D}(x))$ et $T(y, \mathcal{D}(y))$ telles que les relations $\mathcal{D}(x) \Longrightarrow \mathcal{E}_x$ et $\mathrm{RECSQT}(\mathcal{E}_x, \mathcal{E}_y, E)$ sont vérifiées, alors la structure $T(y, \mathcal{E}_y)$ est déductible de la séquence de structures $P(x, \mathcal{D}(x))$ et $T(y, \mathcal{D}(y))$:

$\mathcal{D}(x) \Longrightarrow \mathcal{C}_x$; (1)

RECSQT($\mathcal{C}_x$, $\mathcal{C}_y$,E) ; (2)

$\mathcal{D}(x) \Longrightarrow \mathcal{A}_x$; (3)

$\mathcal{D}(y) \Longrightarrow \mathcal{B}_y$; (4)

$\mathcal{A}_x \Longrightarrow D$; (5)

$\mathcal{B}_y \Longrightarrow D$; (6)

SQTLIE(D) ; (7)

$\mathcal{D}(y) \not\Longrightarrow \neg\mathcal{C}_y$; (8)

$P(x,\mathcal{D}(x)),\ T(y,\mathcal{D}(y)) \vdash\!\!\longrightarrow T(y,\mathcal{C}_y)$

Cette règle d'inférence composée de huit prémisses évalue en particulier la ressemblance des deux descriptions $\mathcal{A}_x$ et $\mathcal{B}_y$ (prémisses 5, 6 et 7 ; voir définition 5). Par contre elle suppose que $\mathcal{C}_y$ d'une part et d'autre part $\mathcal{A}_x$ et $\mathcal{B}_y$. entretiennent certaines relations de dépendance dont l'existence conditionne son application en déterminant les descriptions $\mathcal{A}_x$ et $\mathcal{B}_y$ (voir § II.2.2., point (B) et § III.3.).

Cette règle d'inférence analogique peut être représentée dans le calcul des prédicats par la règle d'inférence ci-après dans laquelle nous avons supposé que la structure "témoin" $P(x,\mathcal{D}(x))$ est une thèse (voir § V.1., chapitre septième) :

$P(x,\mathcal{D}(x)) \vdash P(x,\mathcal{C}_x)$; (1')

$\vdash$ RECSQT($\mathcal{C}_x$, $\mathcal{C}_y$, E) ; (2')

$P(x,\mathcal{D}(x)) \vdash P(x,\mathcal{A}_x)$; (3')

$T(y,\mathcal{D}(y)) \vdash T(y,\mathcal{B}_y)$; (4')

$P(x,\mathcal{A}_x) \vdash P(x,D)$; (5')

$T(y,\mathcal{B}_y) \vdash T(y,D)$; (6')

$\vdash$ SQTLIE(D) ; (7')

$T(y,\mathcal{D}(y)) \not\vdash T(y,\neg\mathcal{C}_y)$; (8')

$T(y,\mathcal{D}(y)) \vdash T(y,\mathcal{C}_y)$

Exemple 6.

Claude BERNARD observe que la couleur du sang dans la veine rénal est rutilante. Il suppose que ce fait est en relation avec la secrétion de l'urine. Par analogie, la même chose doit se produire pour toute secrétion. Et il découvre qu'il en est ainsi en effet pour toutes les glandes, spécialement la sous-maxillaire (exemple extrait de {51}, chapitre II).

Ce mode de raisonnement analogique est représenté par la règle d'inférence analogique suivante :

ANIMAL(HOMME,$\mathcal{D}$(HOMME)) $\vdash$ ANIMAL(HOMME,ISA(ORGANE,\$(REIN,ISA(PARTIE, \$(VEINE,ISA(COULEUR,\$(ROUGE,ISA(NUANCE, BRILLANT)))))))

ANIMAL(HOMME, $\mathcal{D}$(HOMME)) $\vdash$ ANIMAL(HOMME,ISA(ORGANE,\$(REIN,ISA(FONCTION, SECRETION)))) [A]

ANIMAL(HOMME, $\mathcal{D}$(HOMME)) $\vdash$ ANIMAL(HOMME,ISA(ORGANE,\$(SOUS-MAXILLAIRE, ISA(FONCTION,SECRETION)))) [C]

A $\vdash$ ANIMAL(HOMME,ISA(ORGANE,\$(x1,ISA(FONCTION,SECRETION))))

C $\vdash$ ANIMAL(HOMME,ISA(ORGANE,\$(x1,ISA(FONCTION,SECRETION))))

$\vdash$ SQTLIE(ISA(ORGANE,\$(x1,ISA(FONCTION,SECRETION))))

ANIMAL(HOMME,$\mathcal{D}$(HOMME)) $\vdash$ ANIMAL(HOMME,ISA(ORGANE,\$(SOUS-MAXILLAIRE, ISA(PARTIE,\$(VEINE,ISA(COULEUR,\$(ROUGE, ISA(NUANCE,BRILLANT))))))))

<u>Remarque</u>.

Nous avons supposé que le raisonnement déductif ne permet pas de démontrer la négation de l'énoncé à prouver (absence de la premisse (8)). Nous avons également admis que la couleur de l'organe caractérisée par sa nuance dépend de sa fonction (voir § III.3.), le couple de classes (PARTIE, COULEUR) ayant la propriété d'héritage :

DEPENDANCE(ISA (ORGANE,$(x,ISA(COULEUR,$(y,ISA(NUANCE,z))))),
ISA'(ORGANE,$(x,ISA(FONCTION,t))))

Enfin, nous devons remarquer que dans cet exemple les structures P et T sont confondues.

III.2.2. Propriétés.

Propriété 5. Le règle d'inférence analogique ne rend pas le système symbolique ARCHES contradictoire.

démonstration. La conclusion $T(y,\mathcal{C}_y)$ de la règle d'inférence analogique ne peut être admise que si la premisse (8) est vraie, c'est à dire que si l'énoncé $\vdash T(y,\neg\mathcal{C}_y)$ ne peut être démontré au moyen du raisonnement déductif (voir chapitre huitième). Il en résulte que l'énoncé $T(y,\neg\mathcal{C}_y * \mathcal{C}_y)$ est toujours faux ; on ne peut donc pas déduire de la structure $T(y,\mathcal{D}(y))$ l'énoncé $T(y,\Lambda)$ (voir § III.3.1, chapitre septième). Le système symbolique ARCHES est donc consistant (c.q.f.d.).

Propriété 6. Dans le cas général la règle d'inférence analogique n'est pas valide, c'est à dire qu'elle n'est vraie que pour certaines interprétations.

Nous entendons par là le fait que la *formule logique du calcul des prédicats associée à ce schéma de dérivation n'est pas une tautologie* dans la partie du système ARCHES où est mis en oeuvre le raisonnement déductif. Autrement dit le raisonnement analogique est un mode de raisonnement qui ne tend qu'à *rendre satisfiable le système ARCHES* (voir § III.3.)
Cependant la règle d'inférence analogique conduit à une tautologie dans le cas particulier suivant :

$$\mathcal{A}_x = \mathcal{B}_y \quad \text{et} \quad \mathcal{C}_x = \mathcal{D}$$

Dans ce cas, il s'agit d'un raisonnement déductif non explicité (du fait de la transitivité du symbole $\longmapsto$, voir § IV., chapitre septième).

III.3. Le graphe de dépendance.

III.3.1. Définition.

Ne pouvant valider à l'aide du raisonnement déductif la sous-structure $T(y,\mathcal{C}_y)$, nous *construisons un système dans lequel on cherche à la rendre satisfiable par rapport à une interprétation déterminée.* Cette interprétation est fondée sur les relations de dépendance qui existent entre les descriptions $\mathcal{A}_x$, $\mathcal{B}_y$ et $\mathcal{C}_y$ (voir § II.2.2., point (B) du modèle analogique). La relation de dépendance est exprimée à l'aide du prédicat DEPENDANCE à deux places d'arguments ; le premier argument indique le terme descriptif dont l'existence est subordonnée à la réalisation du deuxième argument qui représente également un terme descriptif. Le même terme descriptif peut dépendre naturellement de plusieurs autres termes descriptifs ; un tel cas donne naissance à plusieurs relations de dépendance reliées par la conjonction ET.

Ces *relations de dépendance sont des hypothèses* qui sont fixées à priori par les utilisateurs du système ARCHES. Cependant pour toute application donnée, ARCHES peut déterminer automatiquement toutes les relations de dépendance dont les premiers arguments sont fixés par les utilisateurs. Pour chaque relation de dépendance à construire, dont le premier argument a pour valeur B, par exemple, le système ARCHES recherche dans la base de connaissances la description D (différente de Λ et de B) commune à toutes les stuctures $P(x_i,\mathcal{D}(x_i))$ pour lesquelles ont été prouvés à l'aide du raisonnement déductif les énoncé $P(x_i,B)$. Il infère alors que B dépend de D : $\vdash$ DEPENDANCE(B,D). Dans cette construction ARCHES fait une *hypothèse fondamentale* sur la nature des liens de dépendance qui existent entre les éléments de description : *Si une même sous-structure a été prouvée à l'aide du raisonnement déductif à partir de n structures différentes (n>1), alors les éléments de description de cette sous-structure dépendent de la description commune aux n structures.* Il est évident que ce mode d'évaluation des relations de dépendance n'a pas le degré de pertinence que peut avoir celui qui est mis en oeuvre par les utilisateurs eux-mêmes. Ces derniers ont en effet la possibilité d'argumenter et de justifier leurs choix grâce à leurs connaissances (le plus souvent extra-système) des domaines sur lesquels ils opèrent (il s'agit ici du savoir et du savoir-faire qui composent les domaines empiriques ; voir § II., chapitre premier). Mais ce mode d'évaluation est malgré tout intéressant car il peut jouer un *rôle heuristique* important dans la découverte par les utilisateurs de nouvelles relations de dépendance.

D'une manière générale, on se donne pour toute application plusieurs relations de dépendance du type DEPENDANCE(B,D) ; d'où la définition suivante :

> *définition 7.* On appelle *graphe de dépendance* tout ensemble particulier de relations de dépendance : deux sommets A_i et A_{ij} sont reliés si et seulement si $\vdash$ DEPENDANCE(A_i,A_{ij}).

La donnée d'un graphe de dépendance détermine une interprétation particulière qui fixe la nature des rapports de subordination entre les termes descriptifs. Changer les éléments de ce graphe revient à changer d'interprétation.

> *Propriété 7.* Le graphe de dépendance est un graphe ET.

En effet nous avons observé que tout terme descriptif peut dépendre de plusieurs autres termes descriptifs qui donnent naissance à une conjonction de relations de dépendance.

Exemple 7.

La figure 9.3. montre deux extraits de graphes de dépendance issus des exemples 2 et 3 (voir § II.2.1.). Dans le graphe correspondant à l'exemple 2, les sommets ne sont étiquetés que par les classes car tous les termes descriptifs sont de la forme ISA(classe, variable).

III.3.2. Axiomes.

La relation DEPENDANCE possède les deux propriétés axiomatiques suivantes :

> *Axiome 1.* DEPENDANCE est antiréflexive :
>
> $\vdash \forall$X-DEPENDANCE(X,X)

> *Axiome 2.* DEPENDANCE est antisymétrique :
>
> $\vdash \forall$XY(DEPENDANCE(X,Y) $\supset$ -DEPENDANCE(Y,X))

Ces deux axiomes sont nécessaires pour s'assurer de la cohérence intrinsèque du démonstrateur analogique (voir § IV.2.2.).

Par ailleurs ces deux axiomes font que les graphes de dépendance ne sont ni symétriques, ni réflexifs, et n'ont pas de circuit.

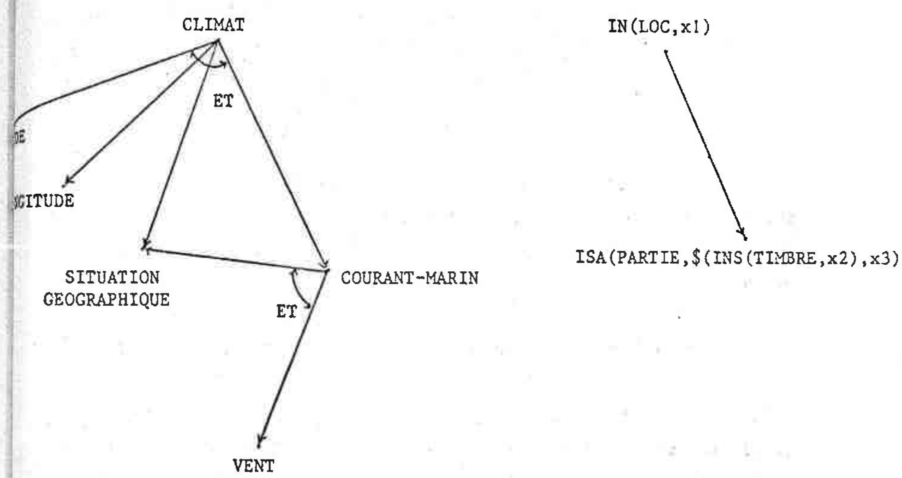


figure 9.3. - Extraits de graphes de dépendance.

III.3.3. Modalités d'utilisation de la règle d'inférence analogique.

L'application de la règle d'inférence analogique dépend du graphe de dépendance, c'est à dire d'une interprétation $\mathcal{M}$ fixée à priori par les utilisateurs du système ARCHES. En effet la conclusion de cette règle ne peut être admise que si les éléments qui composent $\mathcal{C}_y$ dépendent de ceux formant $\mathcal{A}_x$ et $\mathcal{L}_y$. En d'autres termes les schémas des termes descriptifs composant ces éléments doivent être des sommets de ce graphe reliés dans le sens de la dépendance (i.e. de $\mathcal{C}_y$ vers $\mathcal{A}_x$ et $\mathcal{L}_y$).

Ce mode d'utilisation de la règle d'inférence analogique - *filtrée par les graphes de dépendance* - contribue à produire, à partir d'un ensemble de structures logiquement vraies (i.e. valides), des ensembles de structures dont la vérité dépend des interprétations $\mathcal{M}_1, .., \mathcal{M}_j, ..., \mathcal{M}_n$ définies par ces graphes de dépendance. Notre dispositif permet ainsi de construire, pour chaque interprétation $\mathcal{M}_j$, *l'ensemble maximal consistant formé de deux parties* : l'une *valide* correspondant aux thèses du système ARCHES et aux énoncés prouvés à l'aide du raisonnement déductif ; l'autre *satisfiable* produite par l'application de la règle d'inférence analogique (figure 9.4.).

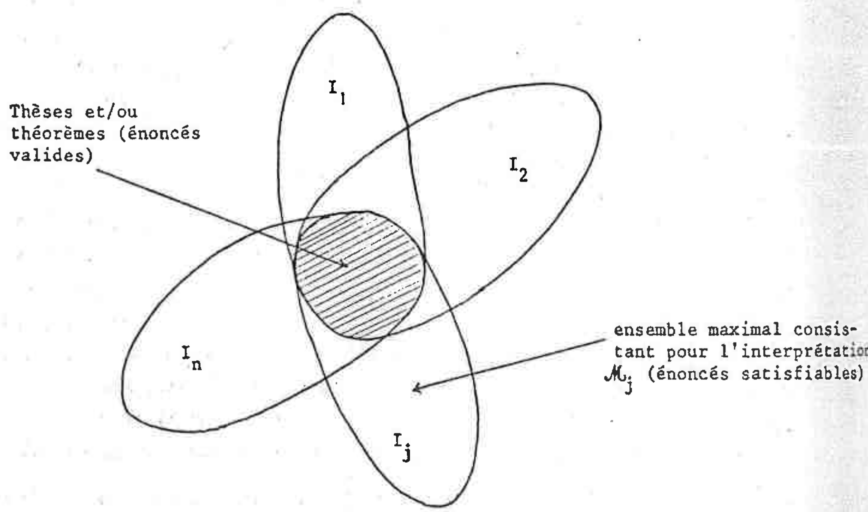


figure 9.4. - Construction des différents types de solutions prouvées par ARCHES.

IV. LE DEMONSTRATEUR ANALOGIQUE DU SYSTEME SYMBOLIQUE ARCHES.

IV.1. Organisation générale du démonstrateur.

Soit $\mathcal{B}$ la base de connaissances associée au système symbolique ARCHES (voir § V.2., Chapitre septième), et G_D un graphe de dépendance particulier .Désignons par $T(y, \mathcal{C}_y)$ la sous-structure à prouver dans laquelle $\mathcal{C}_y$ *représente un terme descriptif*. Et nous supposons que $T(y, \mathcal{C}_y)$ n'a pu être démontrée à l'aide du raisonnement déductif ; nous avons donc (voir § V., chapitre huitième) :

$$\mathcal{D}_D(\mathcal{B}, T(y, \mathcal{C}_y)) = \text{ECHEC}$$

Soit par ailleurs $P(x, \mathcal{D}(x))$ une thèse dont on suppose qu'elle ressemble à la structure $T(y, \mathcal{D}(y))$; c'est une donnée de raisonnement analogique. Dans ces conditions, ARCHES peut mettre en oeuvre le raisonnement analogique en essayant d'appliquer la règle d'inférence analogique filtrée par le graphe de dépendance G_D. Ce schéma de dérivation et ses conditions d'utilisation sont à la base de la construction algorithmique du démonstrateur analogique $\mathcal{D}_A(\mathcal{B}, G_D, P, T(y, \mathcal{C}_y))$ qui se termine à l'étape SUCCES si $\mathcal{C}_y$ est prouvée par analogie; sinon il s'arrête à l'étape ECHEC (figure 9.5.):

$\mathcal{D}_A(\mathcal{B}, G_D, P, T(y, \mathcal{C}_y)) = \text{SUCCES} \Leftrightarrow T(y, \mathcal{C}_y)$ est démontrable dans ARCHES par analogie.

Si le démonstrateur déductif prouve que l'énoncé $T(y, \neg\mathcal{C}_y)$ est un théorème, alors $T(y, \mathcal{C}_y)$ ne peut être démontré. Dans le cas contraire, ARCHES recherche dans le graphe de dépendance G_D le sommet A_i tel que $\mathcal{C}_y$ et A_i sont $\Rightarrow$-unifiables, i.e. $\vdash \vec{U}(\mathcal{C}_y, A_i)$ (voir § II.2., chapitre huitième). Si le sommet A_i n'existe pas, ou si A_i n'est relié à aucun autre sommet de G_D, alors il y a échec. Dans le cas contraire désignons par $F = \{A_{ij} \mid 1 \leqslant j \leqslant p\}$ l'ensemble des p sommets qui sont reliés à A_i. Le schéma de description (éventuellement lié) $A = A_{i1} * .. * A_{ij} * .. * A_{ip}$ construit à partir de F permet de déterminer les deux descriptions $\mathcal{A}_x$ et $\mathcal{A}_y$. Si ces deux descriptions se ressemblent (au sens de la définition 5), alors ARCHES détermine la description $\mathcal{C}_x$ telle qu'il existe un schéma de description lié E qui rende vrai le prédicat $\text{RECSQT}(\mathcal{C}_x, \mathcal{C}_y, E)$

(i.e. $\mathcal{E}_x$ et $\mathcal{E}_y$ ont des éléments de description en commun, voir § II, 2.2. et § III.1.) ; et tente de prouver la sous-structure $P(x, \mathcal{E}_x)$. Si nous avons $\mathcal{D}_D(\mathcal{B}, P(x, \mathcal{E}_x))$=SUCCES alors l'énoncé $T(y, \mathcal{E}_y)$ est prouvé ; dans le cas contraire il y a échec.

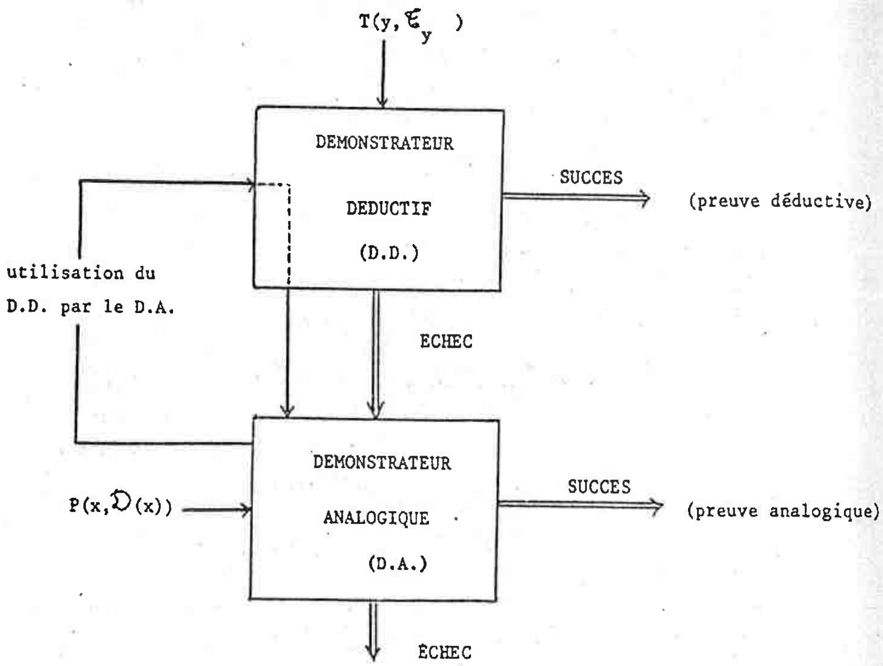


figure 9.5. - *Schéma d'organisation des démonstrateurs déductif et analogique.*

IV.2. Construction algorithmique du démonstrateur.

IV.2.1. Définition de l'algorithme $\mathcal{D}_A(\mathcal{B}, G_D, P, T(y, \mathcal{E}_y))$.

L'algorithme $\mathcal{D}_A$ a pour objet de décrire et de représenter la règle d'inférence analogique (voir § III.2.1., définition 6). Pour ce faire, il utilise en particulier l'algorithme $\mathcal{D}_D(\mathcal{B}, F)$ définissant le demonstrateur déductif (voir § V., chapitre huitième).

Définition de l'algorithme $\mathcal{D}_A(\mathcal{B}, G_D, P, T(y, \mathcal{E}_y))$

1 *si* $\mathcal{D}_D(\mathcal{B}, T(y, \neg\mathcal{E}_y))$=SUCCES
alors ECHEC ; FIN
fsi
si $\exists A_i \in G_D$ tel que $\vdash \vec{U}(\mathcal{E}_y, A_i)$ $[$soit $\vec{\mathcal{U}}(\mathcal{E}_y, A_i) = \{\theta_k \mid 1 \leq k \leq n\} \neq \phi]$
alors $F := \{A_{ij} \mid \text{DEPENDANCE}(A_i, A_{ij}), 1 \leq j \leq p\}$
sinon ECHEC ; FIN
fsi
si $F = \phi$
alors ECHEC ; FIN
2 *sinon* $A := A_{i1} * A_{i2} * \ldots * A_{ij} * \ldots * A_{ip}$
fsi
3 *si* $\mathcal{D}_D(\mathcal{B}, P(x,A))$=ECHEC
alors ECHEC ; FIN
sinon $\mathcal{E}_1 := \{\sigma_i^1 \mid 1 \leq i \leq n_1\}$; $\Leftrightarrow$ (ensemble des substitutions fournies par $\mathcal{D}_D$ et qui rendent insatisfiable l'ensemble $\mathcal{B} \cup \{-P(x,A)\}$)
$\mathcal{A}_x := \overset{n_1}{\underset{i=1}{*}} A\sigma_i^1$
fsi
4 *si* $\forall k \in [1,n]$ $\mathcal{D}_D(\mathcal{B}, T(y, A\theta_k))$= ECHEC
alors ECHEC ; FIN
sinon $\mathcal{E}_2 := \bigcup_{k=1}^{n} \mathcal{E}_{2k}$ avec $\mathcal{E}_{2k} := \{\sigma_{kj}^2 \mid 1 \leq j \leq m(k)\}$;
$\mathcal{E}_y := \overset{n}{\underset{k=1}{*}} [\overset{m(k)}{\underset{j=1}{*}} A\theta_k \sigma_{kj}^2]$
fsi

5 Soit $\mathcal{E}_1' \subset \mathcal{E}_1$ tel que $\forall \sigma_q^1 \in \mathcal{E}_1'$ $\exists 1 \leq k \leq n$ et $1 \leq j \leq m(k)$ qui rendent vrai le prédicat RECSQT$(A\sigma_q^1, A\theta_k\sigma_{kj}^2, E)$, et posons $\rho_1 = \sigma_q^1 \theta_k$; désignons par $\mathcal{E}$ l'ensemble des substitutions ρ_1.

Suite de l'algorithme page suivante.

si $\mathcal{E} = \phi$
alors ECHEC ; FIN
sinon *aller-a* 5
fsi
6 $\forall \rho_1 \in \mathcal{E}$ on détermine la description $\mathcal{C}_x$ de la forme $A_i\rho_1 = A_i\sigma^1_q\theta_k$
si $\forall \rho_1 \in \mathcal{E}$ $\mathcal{D}'_D(\mathcal{B}, P(x, A_i\rho_1))$=ECHEC
alors ECHEC ; FIN
7 *sinon* $S := \{\sigma_i \mid 1 \leq i \leq m\}$; (solutions fournies par $\mathcal{D}_D$, et repré-
SUCCES ; FIN sentant l'ensemble des substitutions
qui rendent insatisfiable $\mathcal{B} \cup \{-P(x, A_i\rho_1)\}$
fsi

.2.2. <u>Théorème relatif à la mise en oeuvre du démonstrateur</u>.

Théorème. Si la sous-structure $T(y, \mathcal{C}_y)$ n'a pu être prouvée à l'aide du raisonnement déductif à partir de la base $\mathcal{B}$ associée au système ARCHES, et si G_D est un graphe de dépendance particulier, alors $T(y, \mathcal{C}_y)$ est démontrée à partir du raisonnement analogique utilisant une structure P de comparaison si et seulement si
$$\mathcal{D}_A(\mathcal{B}, G_D, P, T(y, \mathcal{C}_y)) = \text{SUCCES}.$$

démonstration. Si $T(y, \mathcal{C}_y)$ est résolue de manière analogique, alors il existe nécessairement une structure $P(x, \mathcal{D}(x))$ à partir de laquelle a été démontrée au moins une sous-structure du type $P(x, \mathcal{C}_x)$; on a donc : $\mathcal{D}_D(\mathcal{B}, P(x, \mathcal{C}_x))$=SUCCES. Il en résulte que l'algorithme $\mathcal{D}_A$ se termine à l'étape 7 qui est précisément sa sortie SUCCES. Réciproquement si $\mathcal{D}_A(\mathcal{B}, G_D, P, T(y, \mathcal{C}_y))$=SUCCES , alors l'algorithme $\mathcal{D}_A$ réalise les opérations suivantes : en 1 il vérifie que $T(y, \neg \mathcal{C}_y)$ n'est pas un théorème, ce qui correspond à la validation de la prémisse (8) de la règle d'inférence analogique (voir § III.2.1.; définition 6). En 2 il détermine le schéma de description A (éventuellement lié) qui conditionne la réalisation de $\mathcal{C}_y$; Il suppose que si $\mathcal{C}_y \Longrightarrow A_i$ est vérifiée (i.e. $\vdash \vec{U}(\mathcal{C}_y, A_i)$) et si A_i dépend de A (i.e. de A_{i1} *et* de A_{i2} *et...et* de A_{ip}), alors $\mathcal{C}_y$ dépend de A. Remarquons que ce schéma est calculé à partir du graphe de dépendance dont l'exploration est toujours finie (axiomes 1 et 2). En 3 il détermine $\mathcal{A}_x$ à partir de A en construisant l'ensemble $\mathcal{E}_1$ des substitutions qui rendent insatisfiable $\mathcal{B} \cup \{-P(x,A)\}$. $\forall \sigma^1_i \in \mathcal{E}_1$ on a $P(x, \mathcal{D}(x)) \longmapsto P(x, A\sigma^1_i)$, il en résulte que $P(x, \mathcal{D}(x)) \longmapsto P(x, \mathcal{A}_x)$ (1) avec $\mathcal{A}_x = A\sigma^1_1 * A\sigma^1_2 * \ldots * A\sigma^1_p$. En 4 il détermine $\mathcal{L}_y$ à partir de $A\theta_k$ car $\mathcal{C}_y$ dépend de $\mathcal{L}_y$ (voir § II.1.). Pour ce faire il procède comme en 3 en construisant l'ensemble $\mathcal{E}_2$ des substitutions qui rendent insatisfiable $\mathcal{B} \cup \{-P(x, A\theta_k)\}$; il en résulte que $T(y, \mathcal{D}(y)) \longmapsto T(y, \mathcal{L}_y)$ (2) avec $\mathcal{L}_y = A\theta_1\sigma^2_{11} * A\theta_1\sigma^2_{12} * \ldots * A\theta_p\sigma^2_{pm(p)}$. Les énoncés (1) et (2) correspondent à la validation des prémisses (3) et (4) de la règle d'inférence analogique. A l'étape 5 l'algorithme $\mathcal{D}_A$ tente d'établir la ressemblance des descriptions $\mathcal{A}_x$ et $\mathcal{L}_y$ à l'aide du prédicat RECSQT car $\mathcal{A}_x$ et $\mathcal{L}_y$ ont le même schéma de description A (voir § III.1. ; définition 5) ; ces deux descriptions se ressemblent donc selon les modalités définies dans les prémisses

(5), (6) et (7) de la règle d'inférence analogique. Enfin en (6) $\mathcal{D}_A$ construit $\mathcal{C}_x$ à partir de $A\sigma_q^1\theta_k$ car $\mathcal{C}_x$ dépend de $\mathcal{A}_x$ [substitution σ_q^1] et possède en commun avec $\mathcal{C}_y$ un schéma de description lié [substitution θ_k] ; ce qui valide bien la prémisse (2) de la règle d'inférence analogique. Puis $\mathcal{D}_A$ prouve l'énoncé $P(x, \mathcal{D}(x)) \longmapsto P(x, \mathcal{C}_x)$, ce qui est conforme à la prémisse (1) de la règle d'inférence analogique. Il en résulte que les huit prémisses de la règle d'inférence analogique sont validées au cours du déroulement de l'algorithme $\mathcal{D}_A$ qui se termine à l'étape (7) qui est précisément sa sortie SUCCES. La conclusion peut être postulée, soit $T(y, \mathcal{D}(y)) \longmapsto T(y, \mathcal{C}_y)$. La sous-structure $T(y, \mathcal{C}_y)$ est donc résolue à l'aide de la règle d'inférence analogique (c.q.f.d.).

IV.3. Exemple de fonctionnement du démonstrateur analogique.

A titre d'illustration nous examinons dans ce paragraphe le déroulement des opérations mises en oeuvre par le démonstrateur analogique pour prouver la structure ci-après (voir exemple 3) :

AMPHORE(N2,IN(LOC,INS(FOUR,x)))?

Considérons la base $\mathcal{B}$ formée de plusieurs champs, dont le champ AMPHORE qui contient en particulier les deux thèses ci-après :

AMPHORE(N1,$\mathcal{D}$(N1)) (P)
AMPHORE(N2,$\mathcal{D}$(N2)) (T)

Ces deux thèses P et T sont telles que les relations suivantes sont vérifiées :

$\mathcal{D}$(N1) $\Longrightarrow$ ISA(PARTIE,\$(INS(TIMBRE,T1),ON(LOC,LEVRE)))*ISA(PARTIE,\$(INS(TIMBRE,T1),ISA(FORME,\$(CIRCONFERENCE,EG(DIAMETRE,30)))))*ISA(PARTIE,\$(INS(TIMBRE,T1),ISA(TEXTE,"ASCL")))

$\mathcal{D}$(N1) $\Longrightarrow$ IN(LOC,INS(FOUR,F1))

$\mathcal{D}$(N2) $\Longrightarrow$ ISA(PARTIE,\$(INS(TIMBRE,T2),ON(LOC,LEVRE)))*ISA(PARTIE,\$(INS(TIMBRE,T2),ISA(FORME,\$(CIRCONFERENCE,EG(DIAMETRE,30)))))

Par ailleurs nous supposons que le graphe de dépendance G_D associé à la base $\mathcal{B}$ contient la relation ci-après :

DEPENDANCE(IN(LOC,x),ISA(PARTIE,\$(INS(TIMBRE,y),z))) (4)

Nous souhaitons démontrer par analogie cette structure sachant qu'elle n'a pu être prouvée à l'aide du démonstrateur déductif ; on a donc :

$\mathcal{D}_D$($\mathcal{B}$,AMPHORE(N2,IN(LOC,INS(FOUR,x))))=ECHEC (5)
$\mathcal{D}_D$($\mathcal{B}$, AMPHORE(N2,¬IN(LOC,INS(FOUR,x))))=ECHEC (6)

Pour ce faire on se donne pour structure de comparaison la thèse P=AMPHORE(N1,$\mathcal{D}$(N1)).

Appliquons sur ces données le démonstrateur analogique $\mathcal{D}_A$ (voir § IV.2.1.). L'instruction (1) de l'algorithme $\mathcal{D}_A$ est vérifiée du fait de la condition (6). Recherchons dans le graphe G_D le point A_i tel que la relation $\vec{U}$(IN(LOC,INS(FOUR,x)),A_i) soit valide. D'après la condition (4) ce point existe, et fournit après renommage des variables une seule substitution θ_1:θ_1=<x,INS(FOUR,x1)>. Il en résulte que A=ISA(PARTIE,\$(INS(TIMBRE,y),z)) (étape (2) de l'algorithme $\mathcal{D}_A$).

A ce stade de la démonstration, nous devons construire les descriptions $\mathcal{A}_{N1}$ et $\mathcal{C}_{N2}$:

A l'étape (3), on a $\mathcal{D}_D$($\mathcal{B}$,AMPHORE(N1,A))=SUCCES du fait de la condition (1) ; ce qui fournit les solutions suivantes :

$\mathcal{C}_1$={{<y,T1>,<z,ON(LOC,LEVRE)>},
{<y,T1>,<z,ISA(FORME,\$(CIRCONFERENCE,EG(DIAMETRE,30)))>},
{<y,T1>,<z,ISA(TEXTE,"ASCL")>}}

d'où

$\mathcal{A}_{N1}$ =ISA(PARTIE,\$(INS(TIMBRE,T1),ON(LOC,LEVRE)))*ISA(PARTIE,\$(INS(TIMBRE,T1),ISA(FORME,\$(CIRCONFERENCE,EG(DIAMETRE,30)))))*ISA(PARTIE,\$(INS(TIMBRE,T1),ISA(TEXTE,"ASCL")))

De même à l'étape (4) , on a $\mathcal{D}_D(\mathcal{B}$,AMPHORE(N1,$A\theta_1$))=SUCCES car $A\theta_1$=A et que la condition (3) est vérifiée ; ce qui fournit les solutions suivantes :

$\mathcal{C}_2$={{<y,T2>,<z,ON(LOC,LEVRE)},

{<y,T2>,<z,ISA(FORME,$(CIRCONFERENCE,

EG(DIAMETRE,30)))>}}

d'où

$\mathcal{C}_{N2}$ =ISA(PARTIE,$(INS(TIMBRE,T2),ON(LOC,LEVRE)))*

ISA(PARTIE,$(INS(TIMBRE,T2),ISA(FORME,$(CIRCONFERENCE,

EG(DIAMETRE,30)))))

Dans ces conditions $\mathcal{A}_{N1}$ et $\mathcal{C}_{N2}$ se ressemblent analogiquement, et on a $\mathcal{C}=\{\varepsilon\}$ (substitution vide). Il en résulte que $\mathcal{C}_{N1}=A_i\varepsilon=A_i$; par suite on a $\mathcal{D}_D(\mathcal{B}$,AMPHORE(N1,IN(LOC,INS(FOUR,x1))))=SUCCES du fait de la condition (2) ; ce qui fournit la solution :

$\mathcal{S}$={<x1,F1>}

et la structure AMPHORE(N2,IN(LOC,INS(FOUR,x))) est prouvée analogiquement pour x=F1 (après renommage des variables).

CONCLUSION

-=-=-=-=-=-

Les objectifs, que nous nous sommes fixés dans le cadre de cette recherche, étaient de *concevoir* et de *formaliser* un système symbolique particulier de représentation et de traitement de connaissances- le système ARCHES - dont les *bases épistémologiques* soient clairement explicitées. Aussi, notre travail se situait d'emblée sur l'axe des recherches conduites en intelligence artificielle sur la problématique de la représentation informatique des connaissances et des raisonnements ; ces recherches se développant précisément dans un contexte méthodologique où sont examinés plus à fond les principes et les hypothèses qui fondent l'organisation et l'activité inférentielle des bases de connaissances. Cependant, comme nous l'avons souligné dans le chapitre premier, ce travail puise également sa source dans l'examen systématique d'un ensemble d'expériences relatives à la résolution de problèmes dans des domaines empiriques et réels, comme les sciences de l'homme par exemple. Cette démarche a pour but de produire des formalismes plus généraux que ceux rencontrés à propos de telle ou telle expérience particulière, l'élaboration d'un quelconque modèle abstrait et/ou formel de représentation de connaissances n'étant possible que s'il résulte de la volonté d'analyser et d'expliquer un très grand nombre d'observations (rôle inductif de la méthode expérimentale).

En d'autres termes, ARCHES est conçu comme un *méta-système* qui peut être utilisé pour générer des *systèmes d'information intelligents* dans différents domaines du discours scientifique. La classe des faits réels qui est adéquate à son architecture est circonscrite par un groupe précis d'hypothèses fondées sur la linguistique, la logique et la nature des univers de connaissances étudiés. Nous visons à représenter des phénomènes factuels exprimant des états ou des changements d'état. La définition opératoire que nous en donnons s'appuie sur des éléments théoriques issus de la linguistique, ce qui permet de déterminer les "phrases atomiques" du langage naturel (plus petites phrases porteuses de sens par rapport à ARCHES) qui véhiculent les états ainsi que les connecteurs qui les articulent sur le plan logique. Phrases atomiques et connecteurs sont à l'origine de la définition des structures de données qui organisent le système ARCHES dont la formalisation doit rester dans les limites de la logique du premier ordre. Par ailleurs les modes de raisonnement que nous souhaitons mettre en oeuvre dans ARCHES sont comparables à ceux utilisés dans les sciences expérimentales, c'est-à-dire proches de la réalité observée : on part de suppositions ou d'hypothèses pour en tirer ensuite les conséquences que l'on rend indépendantes des hypothèses de départ par des règles appropriées. Cette position de principe permet de définir les modes de raisonnement élémentaires autorisés dans ARCHES à partir de techniques formelles comparables aux méthodes de la déduction naturelle. Ainsi l'ensemble de ces hypothèses, qui détermine le cadre de conception du système ARCHES, peut être utilisé également comme une *aide* à la représentation et à la constitution des bases de connaissances qui lui sont associées.

Partant de ces hyptohèses, nous avons conçu un système particulier qui *distingue très nettement deux niveaux de représentation* : la représentation des connaissances et la représentation des raisonnements. Cette séparation, qui est conforme à une certaine intuition et conduit à la conception de deux sous-systèmes interdépendants, est totalement justifiée eu égard à l'un au moins des objectifs que nous nous sommes fixés au début de ce travail, à savoir la formalisation de ARCHES considéré comme un système formel spécialisé. A cet effet, rappelons brièvement les caractéristiques principales de ces deux niveaux de représentation qui marquent, selon nous, l'intérêt et l'originalité de notre étude.

L'architecture du système ARCHES est fondée sur un découpage en champs et en domaines des faits réels enregistrés. Elle a pour conséquence d'introduire une grande *modularité* dans l'analyse des connaissances investiguées et une grande *souplesse* dans les modalités de leur représentation, de leur mise à jour et de leur évolution. En particulier, le découpage en champs permet d'exprimer les différents points de vue descriptifs sous lesquels les individus peuvent être analysés.

Les confusions communément rencontrées dans la conception des réseaux sémantiques à propos de la signification des notions de noeuds et de liens qui y sont attachés ont été totalement *supprimées* en définissant de manière très précise les quatre liens SET, INS, ADP et ADL (ou $). Ainsi ces liens permettent de clarifier les rapports qui existent entre caractérisés et

caractérisations : SET relie les concepts entre eux, INS relie les individus aux concepts, le lien ADP définit les rapports entre individus et descriptions en donnant naissance à la notion de structure qui permet de représenter les faits étudiés, et enfin le lien ADL exprime les rapports entre éléments de description se caractérisant les uns les autres.

La puissance de représentation des faits est considérablement accrue pour quatre catégories de raisons : /1/ les couples <classes, opérateur> mettent en relation aussi bien les individus entre-eux que les individus et leurs valeurs d'attributs. Ceci a pour conséquence de créer une organisation des champs fondée sur la nature relationnelle des individus qui les composent ; cette organisation dynamique complétant l'organisation statique des champs en domaines qui est définie par la relation SET. /2/ Les termes descriptifs permettent non seulement de caractériser les individus, mais aussi de véhiculer - grâce à leur structure récursive et à la définition du lien ADL - des descriptions locales exprimant des propriétés de propriétés ou d'une manière plus générale des relations d'état de relations d'état. /3/ La richesse des connecteurs est telle qu'elle contribue à définir des descriptions véhiculant des faits réels complexes et évolutifs. En particulier, signalons la représentation explicite de la négation et la formalisation de son interprétation fondée sur le principe du "système de description clos", les modalités d'évolution des descriptions exprimées par deux connecteurs temporels (le futur immédiat et le futur médiat), et enfin la définition formelle du ET de succession représentant une des valeurs de la conjonction PUIS du langage naturel. /4/ Enfin les propriétés sémantiques des relations d'état (i.e. les couples <classe, opérateurs>) et les rapports logiques qu'elles entretiennent sont exprimés et représentés par le système de règles de substitution-réduction. De même les lois générales, définissant l'organisation intentionnelle des univers de connaissances étudiés, son véhiculées par les règles d'inférence pragmatiques. Peuvent être exprimées également par de telles règles les propriétés de cohérence et de consistante des bases de connaissances correspondantes. Ainsi, *ARCHES combine à la fois les avantages des réseaux sémantiques et des systèmes de production*. En effet, d'une part son langage objet permet de représenter des connaissances complexes et structurées à travers les notions de concepts, d'individus, de R-graphes, de termes descriptifs, de connecteurs, etc. ; d'autre part les règles d'inférence pragmatiques jouent le même rôle que les règles qui composent les systèmes de production, mais ces dernières sont des règles de réécriture alors que les règles d'inférence pragmatiques définissent des opérations de déduction élémentaires.

L'activité inférentielle du système symbolique ARCHES est fondée sur la structure algébrique de l'ensemble des descriptions. Cette structure, qui détermine des modalités spécifiques de dérivation des descriptions, est définie par une relation de déduction vérifiant un ensemble de conditions qui permet de fixer d'une part les critères formels d'utilisation du système de règles de substitution-réduction, et d'autre part les propriétés logiques des connecteurs. Ces règles permettent de transformer les termes descriptifs selon des modalités déterminées par une relation de réécriture définie sur l'ensemble des termes descriptifs, et pour laquelle nous avons construit un algorithme original de décidabilité. Elles peuvent produire des termes descriptifs qui caractérisent *incomplètement* des individus (règle de décomposition), qui représentent des descriptions locales *héritées* par des individus (règles d'héritage et de transitivité), ou enfin qui sont formés de classes dont la portée sémantique a été *étendue* à d'autres valeurs (règle d'extension). Par ailleurs, les propriétés logiques des connecteurs permettent de définir et de justifier formellement les règles d'insertion et d'élimination de ces mêmes connecteurs. En particulier nous avons prouvé que l'ensemble des descriptions muni de la négation et de son interprétation est *consistant*. Enfin nous avons élaborer une procédure de décision pour la relation de déduction ; sa construction utilise la méthodologie de résolution des problèmes par décomposition et construction de graphes ET/OU correspondants.

En outre nous avons donné une *interprétation complète et précise* du système symbolique ARCHES. La définition de cette interprétation présente au moins deux avantages : /1/ elle détermine la sémantique de tous les éléments formels qui composent ARCHES ; et en particulier celle - non classique - des connecteurs qui marquent l'évolution des descriptions. Ceci constitue une aide précieuse et un garde-fou essentiel pour la mise en oeuvre cohérente de ce système. /2/ Elle est indispensable au niveau formel car elle a permis d'établir la consistance de ARCHES ; et en particulier elle a permis de prouver la validité des conditions que vérifie la relation de déduction définie sur l'ensemble des descriptions.

Pour augmenter la capacité déductive de ARCHES, nous l'avons muni de deux modes de raisonnement étroitement solidaires l'un de l'autre : le raisonnement déductif et le raisonnement analogique. Le raisonnement déductif est déterminé par deux règles d'inférence structurales - la règle inter-champs et la règle intra-champ - dont les définitions ne dépendent que de l'organisation et des propriétés formelles du système ARCHES. Pour prouver les théorèmes à l'aide de ce mode de raisonnement, nous avons défini en particulier une *règle de résolution* et des *modalités spécifiques d'unification* des descriptions. Et nous avons démontré sa *complétude* à partir de la définition d'*arbres sémantiques* pour le système ARCHES. Quant au raisonnement analogique, il est déterminé à partir d'un modèle analogique défini comme une *application* particulière qui permet d'exprimer une certaine ressemblance de rapports entre deux couples quelconques de descriptions. Cette ressemblance est évaluée par une *fonction de filtrage* définie à partir des modalités d'unification des descriptions et de prédicats particuliers qui précisent la nature de ce filtrage. Ceci nous a permis de définir une règle d'inférence analogique permettant de transférer une sous-description d'une structure vers une autre à partir de prémisses qui ont pour objet essentiel d'une part d'évaluer précisément la ressemblance entre ces deux structures, et d'autre part de ne pas rendre inconsistant le système ARCHES. L'application de cette règle est *guidée* par le graphe de dépendance qui fixe pour chaque interprétation les rapports d'influence existant entre certains couples de relations d'état, et dont le rôle est de contribuer à rechercher l'existence de telles relations entre les éléments des deux couples de descriptions.

Enfin, ARCHES se présente comme un système d'expression et de résolution de problèmes. Chaque problème est décomposé en une conjonction de sous-problèmes dont la résolution est réalisée par les démonstrateurs déductif et analogique que nous avons construits à partir des règles d'inférence, et validés formellement.

Ce bilan montre que les objectifs que nous nous étions fixés sont atteints. En effet notre système a été construit en s'appuyant sur la logique, ce qui a permis d'établir ses principales propriétés formelles (axiomatisation des éléments de représentation, règles d'inférence basiques, cohérence, complétude, interprétation, etc.). Cependant, ARCHES en tant que projet de recherche nous incite à effectuer d'autres travaux dans le but non seulement de pallier à ses insuffisances et limites, mais aussi de le mettre à l'épreuve. Ces développements concernent principalement trois aspects : l'*implémentation*, les *applications* et enfin l'*évolution et les tentatives de perfectionnement* du système symbolique ARCHES.

La représentation et la manipulation par des clauses de Horn des différentes composantes de ARCHES ont permis de réaliser les premiers essais d'implémentation de certains aspects de ce système en utilisant le langage PROLOG et la méthodologie de programmation en logique du premier ordre qui lui est sous-jacente. Nous continuerons à développer ce travail en sachant bien que nous devrons imaginer des méthodes d'implémentation originales et efficaces, d'autant que notre système sera implanté sur micro-ordinateur : écriture optimale des procédures PROLOG, élimination locale de non déterminisme quand cela est possible, utilisation de la modularité des connaissances pour faciliter l'accès aux données en mode direct, contrôle de la stratégie de recherche de type "retour-arrière" (backtracking) pour la rendre moins aveugle (inhibition ou validation de règles, activation de primitives de contrôle pour supprimer des choix qui conduisent sûrement à des échecs, etc.), définition et mise en oeuvre d'heuristiques particulières.

Comme nous l'avons déjà signalé, la conception de ARCHES procède en particulier de l'examen de plusieurs études de cas. En retour, nous avons pu testé sa puissance de représentation et de traitement en l'appliquant à certaines de ces études. Cependant, nous voudrions élargir le champ de nos essais en utilisant ARCHES dans des domaines et des expériences qui ne sont pas à l'origine de sa conception. A cet effet, nous envisageons la possibilité de le mettre en oeuvre dans certains secteurs d'activité où nous avons de bonnes raisons de croire qu'il est potentiellement adéquat pour résoudre certaines classes de problèmes (comme les sciences juridiques ou la linguistique par exemple). Le déroulement de ces applications, si elles se développent conformément à nos voeux, nous amènera sans doute à réexaminer certains problèmes d'implémentation et certains aspects méthodologiques concernant les perfectionnements du système ARCHES.

Enfin, en plus des tentatives de perfectionnements liées essentiellement à l'utilisation de ARCHES, nous voudrions étendre sa capacité déductive en mettant l'accent sur des recherches plus fondamentales relatives à la formalisation des raisonnements. Dans ce cadre, nous voudrions approfondir, en particulier, l'étude du raisonnement par analogie : examen plus diversifié de la définition de la notion de ressemblance, évaluation de cette ressemblance par des fonctions de filtrage plus sophistiquées, création dynamique d'analogies en fonction des connaissances nouvellement acquises, création de structures de comparaison par apprentissage,..., et d'une manière plus générale, développement d'axes de recherche théorique et méthodologique contribuant à automatiser ce mode de raisonnement.

BIBLIOGRAPHIE

-=-=-=-=-=-=-=-=-=-=-=-

{1} ANSCOMBE G.E. - Before and after. Somerville College, Oxford, 1976 (rapport interne).

{2} ANTOINE G. - La coordination en français. Thèse d'état, d'Arthrey, Paris, 1962.

{3} BACRI N. - Fonctionnement de la négation. Mouton, Paris - La Haye, 1976.

{4} BELY N., BORILLO A., VIRBEL J., SIOT-DECAUVILLE N. - Procédures d'analyse sémantique appliquées à la documentation scientifique. Gauthier-Villars Editeurs, Paris, 1970.

{5} BERGE C. - Théorie des graphes et ses applications. Dunod, Paris, 1963.

{6} BLANCHE R. - Le Raisonnement. P.U.F., Paris, 1973.

{7} BLEDSOE W.W. - Non-resolution theorem proving. *Artificial intelligence*, Vol. 9, n° 1, 1-35, august 1977.

{8} BOBROW D.G., RAPHAEL B. - New programming languages for artificial intelligence research. *ACM computing surveys*, Vol. 6, n° 3, 153-174, september 1974.

{9} BOBROW D.G. - Dimensions of representation. *Representation and understanding, Studies in cognitive science*. BOBROW D.G., COLLINS A. (ed.), Academic Press, Inc., N.Y., 1975.

{10} BOBROW D.G., COLLINS A. (ed.) - Representation and understanding, Studies in Cognitive Science. Academic Press, Inc., New York, 1975.

{11} BOBROW D.G., WINOGRAD T. - An overview of KRL, a knowledge representation language. *Cognitive science*, Vol. 1, n° 1, 3-46, january 1977.

{12} BORILLO A., BORILLO M., VIRBEL J. - Propriétés remarquables d'un système de représentation et de traitement pour l'analyse du discours relatif à un domaine scientifique déterminé. *Analyse et validation dans l'étude des données textuelles*. Editions du CNRS, Paris-Marseille, 1977.

{13} BORILLO A. - Représentation des connaissances : de l'analyse syntaxique de la phrase à son interprétation. *La compréhension : acquisition, représentation et utilisation des connaissances*, Journées de travail IRIA/SESORI, Arc et Senans, mai 1978.

{14} BORILLO M., FERNANDEZ DE LA VEGA W., GUENOCHE A., JANON J., VIRBEL J. - Une expérience de recherche historique à partir de l'analyse d'un corpus d'inscriptions funéraires latines. *Antiquités africaines*, t. 9, 127-144, 1975.

{15} BORILLO M. - Raisonner, calculer. *Raisonnement et méthodes mathématiques en archéologie*, Editions du CNRS, Paris-Marseille, 1977.

{16} BORILLO M., FARIÑAS DEL CERRO L., VIRBEL J. - Validation problems in pattern recognition, study of a particular case. *Congrès IFIP-77*, Toronto, Canada, 1977.

{17} BORILLO M., VIRBEL J. - Statut scientifique de l'archéologie et formalisation de l'analyse des textes ; exemple d'un métalangage d'analyse du corpus des inscriptions latines. *Archéologie et calcul.* Union Générale d'Editions, série 7, Paris, 1978.

{18} BORILLO M., BOURRELLY L., FARIÑAS DEL CERRO L., VIRBEL J. - Le projet AVEROES : Une expérience de formalisation du raisonnement dans les sciences sociales. *Colloque international sur "la représentation des connaissances et raisonnement dans les sciences de l'homme", St Maximim, IRIA, 458-475, 1979.*

{19} BOURCIER D. - Information et signification en droit. Expérience d'une explicitation automatique de concepts. *Langage*, N° 53, 9-32, Mars 1979.

{20} BOURRELLY L., CHOURAQUI E. - Le système documentaire SATIN 1 : description générale et manuel d'utilisation (tome 1, 1974) ; Génération et aide à la mise au point (tome 2, 1978). Editions du CNRS, Paris-Marseille, 1974-1978.

{21} BOURRELLY L., CHOURAQUI E. - La description des données dans le système documentaire SATIN 1. *Automatique et systèmes informatiques*, Paris, février 1977.

{22} BOURRELLY L., FARIÑAS DEL CERRO L. - Simulation of a specific empirical reasoning. *Congrès EURO-IFIP-79*, Londres, 25-28 septembre 1979.

{23} BRACHMAN R.J. - What's in a concept : structural foundations for semantic networks. *International Journal of Man-machine studies*, Vol. 9, n° 2, 127-152, march 1977.

{24} BRACHMAN R.J. - On the epistemological status of semantic networks. Report n° 3807, Bolt Beranek and Newman Inc., april 1978.

{25} CARBONELL J.R. - AI in CAI : An artificial intelligence approach to computer-aided instruction. *IEE Transactions on man-machine systems*, MMS-11, n° 4, 190-202, 1970.

{26} CARNAP R. - Signification et synonymie dans les langues naturelles. *Langages ; logique et linguistique*, 1° année, 2, 108-123, juin 1966.

{27} CERCONE N. - Semantic networks, frames, and conventional data bases methodology. Computing Science Department, Simon Fraser University, Burnaby, British Columbia, Canada, 1979.

{28} CHAFE W.L. - Directionality and paraphrase. *Language* , Vol. 47, n° 1, 1-26, march 1971.

{29} CHANG C.L., SLAGLE J.R. - An admissible and optimal algorithm for searching and/or graphs. *Artificial Intelligence*, Vol. 2, n° 2, 117-128, 1971.

{30} CHANG C.L., LEE R.C.T. - Symbolic logic and mechanical theorem proving. Academic Press, N.Y., 1973.

{31} CHARNIAK E., WILKS Y. - Computational semantics. North-Holland, N.Y., 1976.

{32} CHEN D.T-W., FINDLER N. - Toward analogical reasoning in problem solving by computers. *Journal of cybernetics*, Vol. 9, 369-397, 1979.

{33} CHOMSKY N. - Aspects of the theory of syntax. The M.I.T. Press, Cambridge, Massachussetts, 1965.

{34} CHOURAQUI E., BAUDRY M.F., et al. - L'informatique et l'inventaire général. Ministère de l'Education Nationale/CNRS, Paris, mai 1972.

{35} CHOURAQUI E., VIRBEL J. - Les systèmes de représentation et de traitement des textes du point de vue des méthodes des sciences humaines et des méthodes informatiques. *Colloque international DGRST-CNRS*, Marseille, 11/13 décembre 1975.

{36} CHOURAQUI E., CORBIER P., JANON M., VIRBEL J. - Le SYCIL, un système documentaire pour l'exploitation d'un corpus d'inscriptions latines. *Antiquités africaines*, t. 9, 63-96, 1976.

{37} CHOURAQUI E., VIRBEL J. - Construction d'un dispositif expérimental pour la représentation et le traitement des données textuelles illustré sur un exemple en histoire. 6ième *Congrès international de linguistique computationnelle, COLING*, Ottawa, Canada, 28 Juin/ 2 Juillet 1976.

{38} CHOURAQUI E. - The index and filing system used by the "Inventaire Général des Monuments et des Richesses Artistiques de la France", *Computers and the Humanities*, Vol. 7, n° 5, 273-285, mai 1973.

{39} CHOURAQUI E. - Sur la représentation de la description des concepts. *R.A.I.R.O., Informatique/Computer Science*, Vol. 13, n° 3, 265-308, 1979.

{40} CHOURAQUI E. - Construction of data structures for representing real-world knowledge. *Congrès EURO-IFIP-79*, London, 25-28 septembre 1979.

{41} CHOURAQUI E. - Sur un réseau sémantique actif. 2ième *Congrès "Reconnaissance des formes et intelligence artificielle"*, Toulouse, 12-14 septembre 1979. (Egalement : *Colloque international sur "la Représentation des connaissances et Raisonnement dans les sciences de l'homme"*, St Maximin, IRIA, 215-221, septembre 1979).

{42} CHOURAQUI E. - Sur certains aspects du raisonnement analogique dans le réseau ARCHES. *Journées sur l'analogie*, Paris, 21-22 Février 1980.

{43} CLARK K.L. - Negation as failure. *Logic and data bases.* GALLAIRE H., MINKER J. (ed.), Plenum Press, New York, 1978.

{44} COLLINS A.M., QUILLIAN M.R. - Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior*, Vol. 8, 240-247, 1969.

{45} COULON D., AKNIN Y., LANCEL J.M., MONFILS M. - A language to describe knowledge by a network of procedures. *Séminaire international sur les systèmes intelligents de question-réponse et grandes banques de données*, IRIA, 164-171, Bonas, France, 1977.

{46} COYAUD M. - Linguistique et documentation. Collection "Langue et Langage", Librairie Larousse, Paris, 1972.

{47} DAVIS R., KING J. - An overview of production systems. *Machine Intelligence 8*, ELCOCK E.W., MICHIE D. (ed.), John Wiley and Sons Inc., N.Y., 300-332, 1977.

{48} DE COSTER M. - L'analogie en sciences humaines. P.U.F., Paris, 1978.

{49} DELIYANNI A., KOWALSKI R.A. - Logic and semantic networks. *Communications of the A.C.M.*, Vol. 21, n° 3, 184-192, march 1979.

{50} DOPP J. - Logiques construites par une méthode de déduction naturelle. Gauthier-Villars, Paris, 1962.

{51} DOROLLE M. - Le Raisonnement par analogie. P.U.F., Paris, 1949.

{52} DRESHER B.E., HORNSTEIN N. - On some supposed contributions of artificial intelligence to the scientific study of language. *Cognition*, 4, 321-398, 1976.

{53} DUCROT O. - Deux mais. *Syntaxe et sémantique du français, Cahier de linguistique*, Les presses de l'université du Québec, n° 8, 109-120, 1978.

{54} FAHLMAN S.E. - Thesis progress report : a system for representing and using real-world knowledge. Memo n° 331, Artificial Intelligence Laboratory, M.I.T., may 1975.

{55} FARIÑAS DEL CERRO L. - La sémantique des logiques non classiques et la "Représentation des connaissances". *R.A.I.R.O. Informatique/ Computer Science*, Vol. 13, n° 4, 351-366, 1979.

{56} FIKES R., HENDRIX G. - A network-bases knowledge representation and its natural deduction system. *5th International Joint Conference on artificial intelligence 1977, Proceedings of the Conference*, Vol. 1, 235-246, 1977.

{57} FILLMORE C.J. - The case for case. *Universals in linguistic theory*, BACH E., HARMS R.T. (ed.), Rinehart & Winston,Inc., New York, 1968.

{58} FINDLER N. (ed.) - Associative networks, Representation and use of knowledge by computers. Academic Press, N.Y., 1979.

{59} FREGE G. - Ecrits logiques et philosophiques. Editions du Seuil, Paris, 1971.

{60} GALLAIRE H., MINKER J. - Logic and data bases. Plenum Press, N.Y., 1978.

{61} GALLAIRE H., MINKER J., NICOLAS J.M. - An overview and introduction to logic and data bases. *Logic and data bases*. GALLAIRE H., MINKER J. (ed.), Plenum Press, New York, 1978.

{62} GALMICHE M. - Sémantique générative. Collection "Langue et Langage", Librairie Larousse, Paris, 1975.

{63} GARDIN J.C. - Une archéologie théorique. Collection l'Esprit Critique, Hachette, Paris, 1979.

{64} GELERNTER H. - Realization of a geometry - theorem proving machine. *Computer and thought*, McGraw-Hill book company, Inc., N.Y., 134-152, 1963.

{65} GICK N.L., HOLYOAK K.J. - Analogical problem solving. *Cognitive psychology*, Vol. 12,N°3, 306-355, July 1980.

{66} GODEL R. - Verbes d'état et verbes d'évènement. *Cahiers Ferdinand de Saussure*, Société Genevoise de Linguistique, 9, 33-50, Genève, 1950.

{67} GOLDMAN N.M. - Sentense paraphrasing from a conceptual base. *Communications A.C.M.*, Vol. 18, n° 2, 96-106, 1975.

{68} GRISWOLD R.E., POAGE J.F., POLONSKY I.P. - The SNOBOL 4 programming language, Prentice-Hall, Inc., Englewood Cliffs, New-Jersey, 1971.

{69} GRISWOLD R.E. - String and list processing in SNOBOL 4, Techniques and Applications. Prentice-Hall, Inc., Englewood Cliffs, New-Jersey, 1975.

{70} GRIZE J.B. (ed.) - Discours et analogie (LAD II) n° 30. Université de Neuchâtel, Centre de Recherches Sémiologiques, Neuchâtel, Suisse, mai 1977.

{71} GRIZE J.B. - Le discours analogique. *Colloque international sur "la Représentation des connaissances et raisonnement dans les sciences de l'homme"*, St Maximin, IRIA, 428-435, 1979.

{72} Groupe de Travail sur l'analogie. *Recueil de textes présentés aux journées sur l'analogie*, M.S.H., n° 1, mai 1980.

{73} GUENOCHE A. - Présentation d'un système de traitements documentaires et statistiques adapté aux calculs en sciences humaines. *Congrès AFCET "Panorama de la nouveauté informatique en France"*. Tome 1, 181-189, Paris/Gif-sur-Yvette, novembre 1976.

{74} GUENOCHE A., VIRBEL J. - Structure linguistique d'un sous langage naturel pour la consultation de banques de données archéologiques. *Communication au Congrès ICCH/3*, Waterloo, août 1977.

{75} HAYES P.J. - Some problems and non-problems in representation theory. *Proceedings of the AISB summer conference*, Brighton University of Sussex, 63-79, 1974.

{76} HENDRIX G.G. - Expanding the utility of semantic networks through partionning. *4th International joint conference on artificial intelligence 1975*, Proceedings of the conference M.I.T., 1975.

{77} HERBRAND M.J. - Recherches sur la théorie de la démonstration. Thèse d'état, Faculté des sciences de Paris, 1930.

{78} HESNARD A., VIRBEL J. - Analyse sémantique d'un champ lexicologique se rapportant à des objets en vue d'une traduction automatique dans un langage formalisé. *Communication au 5ième Congrès international de linguistique computationnelle, COLING*, Pise, Italie, août 1973.

{79} HUET G. - Confluent reductions : abstract properties and applications to term rewriting systems. *18th IEEE Symposium on Foundations of computer science*, 30-45, 1977.

{80} HUET G. - Equations and rewrite rules, a survey. *Conference on formal language theory*, Santa Barbara, december 1979.

{81} JACKENDOFF R.S. - An Interpretative theory of negation. *Foundations of Language*, Vol. 5, n° 2, 218-241, may 1969.

{82} KALINOWSKI G. - Le Raisonnement en sciences juridiques. *Colloque international sur "la Représentation des connaissances et Raisonnement dans les sciences de l'homme"*, St Maximin, IRIA, 476-485, 1979.

{83} KATZ J.J., FODOR J.A. - The structure of a semantic theory. *Language*, 39, 170-210.

{84} KAYSER D., BONNET A., JAKOB F. - Natural language comprehension based on approximate reasoning. *Séminaire international sur les systèmes intelligents de question-réponse et grandes banques de données*, I.R.I.A., 172-181, Bonas, France, 1977.

{85} KLEENE S.C. - Introduction to meta-mathematics. North-Holland Publishing Company, Amsterdam, 1971.

{86} KLINA E.S. - Negation in english. *The structure of language*. FODOR J.A., KATZ J.J. (ed.), Prentice Hall, Inc., N.J., 246-323, 1964.

{87} KLING R.E. - A paradigm for reasoning by analogy. *Artificial intelligence*, Vol. 2, 147-178, 1971.

{88} KLIX F., Van Der MEER E. - Analogical reasoning - An approach to mechanisms underlying human intelligence performances. *Human and artificial intelligence*, KLIX F. (ed.), North-Holland Publishing Company, Amsterdam, 193-241, 1979.

{89} KNEEBONE G.T. - Mathematical logic and the foundations of mathematics, An introductory survey. D. Van Nostrand Company Limited, London, 1963.

{90} KNUTH D.E. - The art of computer programming, Fundamental algorithms. Volume 1, Addison-Wesley publishing company, Menlo Park, Californie, 1969.

{91} LAKOFF G. - Stative adjectives and verbs in english. Report NSF-17, Harvard Computational Laboratory, 1966.

{92} LAURIERE J.L. - Raisonnement par analogie en intelligence artificielle : les systèmes de production. *Groupe de travail sur l'analogie, Recueil de textes*, 22-33, Paris, mai 1980.

{93} LESCA H. - L'étude du langage des organisations. *Séminaire sur l'informatique d'organisation, les systèmes d'information et de décision*, IRIA, Aix-en-Provence, 10-11 avril 1975.

{94} MACCARTHY M. - Meaning and intentional objects. *Semiotica*, Vol. 23, 1/2, 165-190, 1978.

{95} McCARTHY J. - LISP 1.5, "Programmer's manuel. M.I.T., Cambridge, Massachussetts, 1964.

{96} McCAWLEY J.D. - The role of semantics in a grammar. *Universals in Linguistic theory*, BACH E., HARMS R.T. (ed.), Rinehart & Winston, Inc., New York, 1968.

{97} McDERMOTT D.V. - Very large planner-type data bases. A.I. Memo n° 339, Artificial Intelligence Laboratory, M.I.T., september 1975.

{98} McSKIMIN J.R., MINKER J. - A Prédicate calculus based semantic network for deductive searching. *Associative network, representation and use of knowledge by computer*, FINDLER N. (ed.), Academic Press, N.Y., 205-238, 1979.

{99} MANNA Z. - Mathematical theory of computation. McGraw-Hill Kogakusha, Ltd, Tokyo, 1974.

{100} MIEVILLE D. - Exemples, analogie - un essai de schématisation. *Langue et discours II, Colloque de Basançon-Neuchâtel*, Neuchâtel, Suisse, n° 34, 53-73, 2-4 octobre 1978.

{101} MINSKY M. - A Framework for representing knowledge. Memo n° 306, Artificial Intelligence Laboratory, M.I.T., june 1974.

{102} NEWELL A., SIMON H.A. - Computer science as empirical inquiry : symbols and search. *Communications of the A.C.M.*, Vol. 19, n° 3, march 1976.

{103} NILSSON N.J. - Problem solving methods in artificial intelligence. McGraw-Hill Bask Company, New York, 1971.

{104} PAIR C., GAUDEL M.C. - Les structures des données et leur représentation en mémoire. I.R.I.A. éditeurs, Rocquencourt, 1977.

{105} PERELMAN Ch., OLBRECHTS-TYTECA L. - Traité de l'argumentation - La nouvelle rhétorique. Edition de l'Institut de Sociologie, Université libre de Bruxelles, Bruxelles, 1970.

{106} PERELMAN Ch. - Le champ de l'argumentation. Presses Universitaires de Bruxelles, Bruxelles, 1976.

{107} PERELMAN Ch. - L'empire rhétorique - Rhétorique et argumentation. VRIN, Paris, 1977.

{108} POLYA G. - Mathematics and plausible reasoning. Princeton University Press, Princeton, 1962.

{109} PORTE J. - Recherches sur la théorie générale des systèmes formels. Gauthier-Villars, Paris, 1965.

{110} QUILLIAN M.R. - "Semantic memory". *Semantic information processing,* MINSKY(ed.), M.I.T. Press, 1968.

{111} QUILLIAN M.R. - The Teachable language comprehender : a simulation program and theory of language. *Communications of the ACM.* Vol. 12, n° 8, 459-476, 1969.

{112} REICHENBACH H. - Elements of symbolic logic. The MacMillan Company, New York, 1947.

{113} RIBBENS D. - Programmation non numérique LISP 1.5. Dunod, Paris, 1969.

{114} RICARD M. - Les graphes de résolution dans le réseau ARCHES, algorithmes de contrôle et de démonstration. D.E.A., Groupe d'Intelligence Artificielle, Faculté des Sciences de Luminy, Université d'Aix-Marseille II, 1980.

{115} ROBINSON J.A. - A Machine-oriented logic based on the resolution principle. *Journal of the A.C.M.,* Vol. 12, n° 1, 23-41, january 1965.

{116} ROUSSEL Ph. - PROLOG - Manuel de référence et d'utilisation. Groupe d'Intelligence Artificielle, Université d'Aix-Marseille II, Marseille-Luminy, septembre 1975.

{117} RUMELHART D.E., NORMAN D.A. - Explorations in cognition. W.H. Freeman and Company, San Francisco, 1975.

{118} RUMELHART D.E., LEVIN J.A. - A Language comprehension system. *Explorations in cognition.* NORMAN D.A., RUMELHART D.E. (ed.), W.H. Freeman and Company, San Francisco, 1975.

{119} SABAH G. - Contribution à la compréhension effective d'un récit. Thèse de doctorat d'état, Université Pierre et Marie Curie, Paris VI, 1978.

{120} SAUVAN J. - Connaissance et informatique. *Dialectica,* Vol. 26, n° 1, 3-10, 1972.

{121} SCHANK R.C., COLBY K.M. - Conceptual information processing. North-Holland Publishing Company, Amsterdam, 1975.

{122} SCHUBERT L.K. - Extending the expressive power of semantic networks. *Artificial Intelligence,* Vol. 7, n° 2, 163-198, 1976.

{123} SCHWIND C. - A completeness proof for a logic of action. LISH/172/Avril 1980 (rapport interne).

{123a} SCOTT D. - Rules and derived rules. *Logical theory and semantic analysis,* STENLUND S. (ed.), D. Reidel publishing Company, Boston, 146-161, 1975.

{124} SHORTLIFFE E. - Computer-based medical consultations : MYCIN. American Elsevier, N.Y., 1976.

{125} SIMMONS R.F. - Semantic Networks - their computation and use for understanding english sentences. *Computer models of thought and languages,* SCHANK R.C. and COLBY K.M. (ed.), Freeman, San Francisco, California, 63-113, 1973.

{126} SMITH L.C. - Artificial intelligence in information retrieval systems. *Information processing & management,* vol.12, N°3, 189-222, 1976.

{127} SUSSMAN G.J., McDERMOTT D.V. - Why conniving is better than planning. A.I. Memo n° 255 A, Artificial Intelligence Laboratory, M.I.T., april 1972.

{128} TESNIERE L. - Eléments de syntaxe structurale. Librairie C. Klincksieck, Paris, 1959.

{129} VAN EMDEN M.H. - Programming with resolution logic. *Machine intelligence 8.* ELCOCK E.W., MICHIE D. (ed.), Ellis Horwood Ltd, N.Y., 1977.

{130} WANG H.W. - Logic of many sorted theory. *The Journal of Symbolic Logic,* Vol. 17, n° 2, 105-116, 1952.

{131} WANG H. - Toward mechanical mathematics. *I.B.M. Journal Res. & Dev.,* n° 4, 2-22, january 1960.

{132} WEIZENBAUM J. - "ELIZA" : a computer program for the study of natural language communication between man and machine. *Communications A.C.M.,* Vol. 9, n° 1, 36-45, 1966.

{133} WINOGRAD T. - Understanding natural language. Academic Press, Inc., New York, 1972.

{134} WOODS W.A. - What's in a link, Foundations for semantic networks. *Representation and understanding, studies in cognitive science.* BOBROW D.G., COLLINS A. (ed.), Academic Press Inc., New York, 1975.

TABLE DES MATIERES

-=-=-=-=-=-=-=-=-=-=-

CHAPITRE PREMIER - *PRESENTATION DU CADRE DE NOS RECHERCHES.*

CHAPITRE DEUXIEME - *HYPOTHESES SOUS-JACENTES A LA CONCEPTION DU SYSTEME SYMBOLIQUE ARCHES.*

CHAPITRE TROISIEME - *L'ARCHITECTURE DU SYSTEME SYMBOLIQUE ARCHES.*

CHAPITRE QUATRIEME - *LE SYSTEME FORMEL S_Δ DE CARACTERISATION DES DESCRIPTIONS.*

CHAPITRE CINQUIEME - *LES PROPRIETES DU SYSTEME FORMEL* S_Δ.

CHAPITRE SIXIEME - *REPRESENTATION DE L'EVOLUTION DES DESCRIPTIONS : EXTENSION DU SYSTEME FORMEL* S_Δ.