



HAL
open science

Lézard : acquisition des connaissances et gestion de l'incertitude dans un système expert de seconde génération

Jean Charlet

► **To cite this version:**

Jean Charlet. Lézard : acquisition des connaissances et gestion de l'incertitude dans un système expert de seconde génération. Intelligence artificielle [cs.AI]. Université Pierre & Marie Curie - Paris 6, 1989. Français. NNT: . tel-03560682

HAL Id: tel-03560682

<https://theses.hal.science/tel-03560682v1>

Submitted on 25 Feb 2022

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

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

THESE de DOCTORAT
de l'UNIVERSITÉ PARIS 6

Spécialité :

Informatique

présentée

par Monsieur Jean CHARLET

pour obtenir le grade de DOCTEUR DE L'UNIVERSITÉ PARIS 6

Sujet de la thèse :

LEZARD : Acquisition des Connaissances
et Gestion de l'Incertitude
dans un Système Expert de Seconde Génération

Soutenue le 20 juin 1989

devant le jury composé de :

Monsieur Norbert	COT	(Président)
Madame Marie-Odile	CORDIER	(Rapporteur)
Monsieur Michel	DEGLAS	(Rapporteur)
Monsieur Jean-François	BOISVIEUX	
Monsieur Olivier	GASCUEL	
Monsieur Vincent	MORICE	



Le Professeur Norbert Cot a bien voulu diriger cette thèse. Je tiens à lui exprimer ma gratitude pour la confiance qu'il m'a toujours témoignée.

Madame Marie-Odile Cordier, Professeur à Rennes, a accepté de juger mon travail. Ses remarques, nombreuses et précises, m'ont montré avec quelle attention elle l'a lue. Je suis flatté qu'elle ait bien voulu être rapporteur de cette thèse et je l'en remercie.

Monsieur Michel Deglas a accepté d'être rapporteur et de lire attentivement mon travail. Je l'en remercie. Ses remarques pertinentes et les discussions que nous avons eues ensemble m'ont permis d'éviter quelques bavures incertaines.

Olivier Gascuel a su guider mes premiers pas en Intelligence Artificielle et m'assurer d'un soutien scientifique de tous les instants pendant mes travaux sur le système Lézard. Je tiens à lui exprimer ma gratitude pour sa confiance et son amitié.

Vincent Morice a su guider mes errements littéraires et m'épauler pendant la rédaction de cette thèse. Les nombreuses discussions que j'ai eues avec lui m'ont permis de clarifier mon esprit et d'améliorer l'exposé de mon travail. Sa disponibilité de tous les instants m'a été d'un grand secours et je tiens vivement à l'en remercier.

Le Professeur Jean-François Boisvieux, m'a accueilli à l'INSERM il y a cinq ans et m'a offert d'excellentes conditions de travail tout au long de cette thèse. Je tiens à lui exprimer ma gratitude pour cela et pour son amitié.

Je remercie également Jack Bouaud pour ses, Ô combien grandes, compétences en mise en page et interprétations plus ou moins cristalliennes des commandes L^AT_EX, Bruno Bachimont pour la preuve ontologique de sa présence amicale, Pierre Zweigenbaum pour son amitié et sa disponibilité, Laurent Doré pour son amitié jamais donnée du bout des lèvres mais plutôt venue du off the lip, Hal1 Fauconnet pour accepter avec bonne grâce de faire de la géologie dans les topazes, cristaux et autres pierres précieuses et France Mentre pour son amitié et sa gentillesse.

Je tiens également à remercier les autres membres du laboratoire, Bernard, Alain, Marc, Véronique, Patrick, Brigitte et les autres...

Merci finalement à Véronique qui a su supporter mon caractère difficile durant la rédaction de cette thèse.

Paris, le 12 Juin 1989.



Contenu de la thèse

Préface	1
1 Motivations	1
2 Lézard	2
3 Présentation des chapitres de la thèse	2
1 Introduction	5
1 Les approches de l'incertitude	6
2 Objectifs du projet	10
3 Le système Lézard.	12
2 La représentation des connaissances	17
1 Problématique	17
2 Les connaissances médicales	19
3 Le langage de représentation des connaissances	21
4 Les exceptions	32
5 Interprétation du modèle physiopathologique	34
3 Le traducteur	37
1 Les connaissances profondes et les connaissances de surface	37
2 Acquisition des connaissances	40
3 Les mécanismes d'interprétation des connaissances	43
4 Traduction ; dérivation formelle des relations	52
5 Description d'un processus d'interprétation complet	59
6 Les méta-connaissances du traducteur	62
4 Le moteur d'inférences	65
1 La compilation des règles de production	65
2 Le fonctionnement du moteur	73
3 Connaissances de contrôle	85
4 Exécution, sortie des résultats, traces	86
5 Evaluation : application	91

1	Evaluation d'un système expert	92
2	Le système bayésien	93
3	La base des données de l'ARC & AURC	94
4	Les procédures d'évaluation	97
5	Le système Léopard et les douleurs aiguës de l'abdomen	106
6	Résultats	107
7	Conclusion et perspectives	114
6	Perspectives	115
1	Les améliorations envisagées	116
2	Perspectives	118
	Références Bibliographiques	123
1	Vocabulaire et définitions préliminaires	135
2	Le calcul des exclusions	138
3	Le calcul des négations	147
	Table des matières	159
	Liste des figures	163

Préface

Cette thèse décrit le système expert essentiel LÉZARD. Il a été conçu pour résoudre les problèmes de diagnostic dans un domaine incertain, c'est à dire un domaine qu'on ne peut décrire complètement, que ce soit par manque de connaissance de notre part ou en raison de la faiblesse du mode de description des connaissances.

1 Motivations

Les motivations qui ont présidé à l'écriture du système viennent de l'observation du comportement d'autres systèmes experts raisonnant dans un domaine incertain, en particulier SAM [Gascuel 85] et MYCIN [Shortliffe 76]. Dans ces systèmes, l'incertitude attachée au raisonnement est modélisée de façon *numérique* mais la signification des coefficients utilisés pour représenter l'incertitude des inférences n'est pas facile à appréhender.

D'autre part, ces systèmes ne peuvent représenter des connaissances complexes. Des systèmes comme ABEL [Patil et al. 82] ou SESAM [Ferrand 84] par exemple, permettent de décrire — en médecine — des modèles physiopathologiques complexes. Ils ont toutefois des difficultés à aborder le raisonnement incertain.

De plus les systèmes actuels n'ont pas la capacité, soit de faire des hypothèses en cas de données insuffisantes, soit de gérer des contradictions entre données initiales.

Les objectifs de notre travail découlent directement de ces constatations : il faut tout d'abord disposer d'un formalisme assez riche pour permettre une description aisée de mécanismes complexes. Pourtant, alors que la description d'un mécanisme via un modèle complexe peut être considérée, en première approximation, comme complète et donc certaine, ce modèle n'inclut pas toutes les connaissances nécessaires à des tâches diagnostiques : des incertitudes persistent à ce niveau. Des connaissances empiriques, dans lesquelles l'incertitude peut être donnée, doivent donc nécessairement être ajoutées au système. Un travail sur l'acquisition de ces connaissances est alors nécessaire. Enfin, le système doit être à même de tenir compte de l'incomplétude et des données manquantes : il doit permettre de modéliser un raisonnement hypothétique.

2 Lézard

A partir de ces objectifs, notre travail a été très pragmatique : une base de données de l'ARC & AURC¹ sur les douleurs aiguës de l'abdomen nous a fourni le domaine de travail du système. Un premier prototype du système LÉZARD a été développé [Gascuel et al. 85]. Le module d'acquisition des connaissances y était assez sommaire, alors que le moteur comportait déjà ses principales composantes.

L'étape suivante, qui constitue le principal travail de cette thèse, a été le développement d'un module d'acquisition des connaissances performant. Nous proposons une approche qui permet l'acquisition de nouvelles connaissances à partir du modèle : ce dernier a un contenu sémantique complexe qui autorise plusieurs interprétations logiques valides ; le système va aider l'expert à desambigüiser le modèle et lui permettre ainsi d'élaborer une base des connaissances opérationnelle.

Le système LÉZARD² a été développé au "Département Intelligence Artificielle et Médecine" de l'unité INSERM U194 et du service d'informatique médicale de l'Assistance Publique dans le cadre du PRC-GRECO/IA³.

Le système LÉZARD est écrit en COMMON-LISP et fonctionne actuellement sur VAX-VMS. LÉZARD été décrit, à différentes phases de son évolution dans [Gascuel et al. 85, Charlet & Gascuel 86, Charlet 87, charlet 89a]. Le code source occupe environ 5000 lignes.

3 Présentation des chapitres de la thèse

Cette thèse est organisée en 6 chapitres.

Dans le premier chapitre, une introduction au domaine, nous abordons les problèmes d'incertitude en général, ce que recouvre ce vocable et les approches qui sont habituellement proposées pour résoudre ce problème. Nous y discutons l'intérêt des approches *numériques* ou *non numériques* en fonction du type d'incertitude à prendre en compte. Nous faisons ressortir les problèmes qui vont se poser, que ce soit au niveau du modèle (acquisition des connaissances), ou au niveau des inférences (contrôle). Nous décrivons ensuite les choix effectués pour le système LÉZARD, en adéquation avec les objectifs.

Le second chapitre introduit les problèmes de représentation des connaissances et de la dualité représentation/interprétation des connaissances. Les connaissances nécessaires à un système expert en médecine y sont ensuite abordées. La représentation des connaissances du système LÉZARD est alors décrite : description du langage de représentation des connaissances choisi,

¹Association pour la Recherche Chirurgicale & Association Universitaire pour la Recherche Chirurgicale.

²LÉZARD n'est l'acronyme de rien du tout. Ceci vient de ... les arts...

³Programme de Recherche Coordonnée — Groupe de REcherche COordonnée en Intelligence Artificielle.

avec ses primitives et les concepts qu'elles permettent de représenter ; description des caractéristiques des objets du système, avec la séparation des objets en plusieurs classes (signe, syndrome, diagnostic), avec la définition des valeurs (courantes et par défaut) des objets, avec la fiabilité attachée à chaque objet ; signification des relations décrites avec les primitives et les objets. Finalement nous examinons comment se posent les problèmes d'interprétation des connaissances pour le système LÉZARD.

Le troisième chapitre aborde la description du TRADUCTEUR, réponse proposée aux problèmes de représentation des connaissances abordés dans le chapitre précédent. Le problème de l'acquisition des connaissances y est discuté. Nous examinons comment il est pris en compte par le TRADUCTEUR. Dans une première partie nous abordons les problèmes des *connaissances de surface* et des *connaissances profondes*. Dans une seconde partie nous décrivons le mécanisme d'acquisition des *connaissances de surface* à partir des *connaissances profondes*. Dans une troisième partie nous analysons tous les mécanismes mis en œuvre dans le processus d'acquisition, leur caractère logique (formalisable) et *heuristique* (informel). Nous décrivons alors leur mise en œuvre dans le TRADUCTEUR. Dans une cinquième partie nous décrivons un processus de traduction complet, en illustration des descriptions précédentes.

Les chapitres 2 et 3 et l'annexe 1 qui va avec, forment un ensemble où le problème de l'acquisition et de la représentation des connaissances sont traités en profondeur. Ils sont le noyau de cette thèse et en forment la première partie.

Le quatrième chapitre décrit le moteur d'inférences du système LÉZARD. Dans une première partie nous abordons le traitement que subissent les règles du système en vue d'une amélioration des inférences, dans le but d'une plus grande rapidité du système. Nous décrivons ensuite le fonctionnement du moteur : nous y abordons la *non monotonie* du moteur, comment y sont gérées les hypothèses, à quoi correspond le choix d'un raisonnement. Nous y décrivons comment des approches *symbolique* et *numérique* y coopèrent. En particulier nous y développons le problème des connaissances inférentielles — souvent des *heuristiques* — que doit posséder le moteur pour effectuer un raisonnement "incertain" de bonne qualité.

Le chapitre 4 décrit le moteur d'inférences du système LÉZARD. Il forme la seconde partie de la thèse

Le cinquième chapitre est la description de l'application du système LÉZARD au domaine des douleurs aiguës de l'abdomen. Une évaluation comparative critique y est menée avec un système "bayésien". La base des données de l'ARC & AURC, contenant les observations effectuées sur environ 7000 malades dans le domaine des douleurs aiguës de l'abdomen, est spécialement décrite. C'est elle qui a fourni les échantillons d'apprentissage pour développer le système LÉZARD et l'échantillon test pour le comparer au système bayésien. Les résultats de cette étude sont ensuite analysés.

Ce chapitre, exclusivement centré sur l'application, traite en profondeur le problème de la validation du système LÉZARD. Il peut être lu indépendamment des autres, si on ne s'intéresse qu'aux procédures d'évaluation des systèmes d'aide à la décision. Il forme la troisième et

dernière partie du développement de cette thèse.

Le sixième chapitre conclut cette thèse. Il dresse un bilan du travail effectué et des apports qu'il a permis, en particulier dans le domaine de l'acquisition des connaissances. Nous essayons ensuite, à travers les critiques que l'on peut formuler, de trouver de nouvelles ouvertures pour améliorer le système. Finalement, dans une partie prospective, nous abordons les problèmes généraux qui restent posés, en particulier au niveau du TRADUCTEUR, du point de vue de l'apprentissage et des perspectives qui s'ouvrent à lui dans ce domaine.

Deux annexes sont ajoutées à la fin de ce document.

La première annexe donne des précisions sur le "calcul des exclusions" qui est une méthode de génération de *connaissances de surface* abordée au chapitre quatre.

Cette annexe se rattache à la première partie, représentation et acquisition des connaissances. Elle peut être retenue pour une seconde lecture bien qu'elle éclaire grandement les problèmes de compatibilités et d'exclusion entre diagnostics.

En seconde annexe nous donnons la base des connaissances de haut niveau utilisée dans l'application de LÉZARD au domaine des douleurs aiguës de l'abdomen. Cette partie complète le chapitre cinq dédié à l'application du système.

Chapitre 1

Introduction

Introduction

Chaque chapitre débute par un paragraphe appelé "Introduction". Nous essaierons pour chacun d'entre eux d'y décrire leur but et leur rôle.

Dans ce premier chapitre, une introduction au domaine, nous abordons les problèmes d'incertitude en général, ce que recouvre ce vocable et les approches qui sont habituellement proposées pour résoudre ce problème.

Nous y discutons l'intérêt des approches numériques ou non numériques en fonction du type d'incertitude à prendre en compte. Nous faisons ressortir les problèmes qui vont se poser, que ce soit au niveau du modèle (acquisition des connaissances), ou au niveau des inférences (contrôle).

En fonction des approches décrites et des systèmes déjà réalisés, nous donnons les objectifs qui ont présidé à la réalisation du système LÉZARD.

Nous y décrivons ensuite les choix effectués pour le système, en adéquation avec les objectifs.

L'automatisation des processus d'aide à la décision est une branche d'activité importante de l'informatique. On y distingue essentiellement deux types de programmes : ceux fondés sur des approches numériques, généralement le calcul des probabilités, et ceux qui utilisent un grand nombre de connaissances sur le problème à résoudre : c'est le domaine de l'intelligence Artificielle (IA), et plus précisément des systèmes experts (SE) en diagnostic.

Les problèmes viennent principalement de ce que cette opération doit se faire dans un monde incertain : c'est particulièrement visible en médecine. Les programmes développés ne sont pas exempts de critiques. Celles principalement faites à l'encontre des systèmes numériques sont, d'une part la rigidité du modèle mathématique sous-jacent, et d'autre part l'absence de

capacité explicative sur les résultats obtenus (cf. 2 ch. 5). Cependant ces systèmes, en particulier les systèmes probabilistes "bayésiens" [Szolovitz & Pauker 78, Charniak & McDermott 86], ont le mérite d'être simples à mettre en œuvre et de fournir des résultats satisfaisants dans un grand nombre de situations.

Nous nous intéressons ici, uniquement à l'approche IA du problème qui vise à répondre aux critiques précédentes. La prise en compte de l'incertain dans la prise de décision a déjà été étudiée par de nombreux auteurs.

De nombreux modèles mathématiques de l'incertitude, fondés sur l'approche probabiliste, ont été proposés en IA : les principaux sont la théorie de l'Evidence de Shafer [Shafer 76, Bathnagar & Kanal 86] et la théorie des Possibilités de Zadeh [Zadeh 78, Dubois & Prade 85].

Cette incertitude peut aussi être prise en compte de manière non numérique dans des logiques particulières comme les logiques des défauts [Reiter 80, Kayser 88] ou plus généralement les logiques non monotones [McDermott & Doyle 80, McCarthy 80].

Devant le nombre de solutions proposées, il nous semble nécessaire de préciser les différentes significations du mot incertitude et comment les méthodes précédentes peuvent raisonnablement les prendre en compte. Nous introduirons ensuite les objectifs que nous avons cherché à atteindre. Puis nous donnerons les principales caractéristiques du système LÉZARD que nous avons développé. Enfin, nous terminerons l'introduction par une présentation succincte des différents chapitres de ce travail.

1 Les approches de l'incertitude

L'incertitude ne peut se concevoir que vis à vis d'un domaine précis — celui-ci pouvant s'étendre jusqu'au monde. Il y a incertitude quand, voulant décrire un domaine, on ne peut le faire complètement. Ainsi, un domaine incertain est un domaine qu'on ne peut décrire complètement, que ce soit par manque de connaissance de notre part ou en raison de la faiblesse du mode de description des connaissances.

La possibilité de modéliser l'incertitude dans un système informatique sous-entend donc l'existence d'un formalisme permettant de la représenter et de méthodes pour la manipuler. En pratique, la représentation de l'incertain doit être suffisamment simple et intuitive pour être comprise par un expert ; elle doit aussi être efficace, c'est à dire permettre les prises de décision les plus automatisées possibles en évitant un appel trop systématique à un expert [Bhatnagar & Kanal 86].

Mais cette incertitude est un concept assez vague qui se prête mal à une définition simple d'un formalisme de représentation. Nous allons donc essayer tout d'abord de recenser les différentes sources d'incertitude qui peuvent agir sur les connaissances manipulées par le système d'aide à la décision, avant d'aborder les formalismes permettant de la modéliser.

Nous avons choisi de retenir un classement des types d'incertitude proche de celui proposé par [Bonissone 87] :

- qualité de l'information

1. fiabilité de l'information saisie :

Les capteurs qui saisissent les données de la situation peuvent être bruités. Dans le domaine médical c'est typiquement : (1) le problème de la subjectivité des observations que le médecin effectue sur le patient ; (2) la fiabilité des résultats complémentaires. Il y a incertitude sur la véracité de l'information.

2. imprécision des observations :

Cette source d'incertitude correspond à l'imprécision qui caractérise l'information elle-même. Par exemple en médecine, une "température forte" est un terme imprécis.

- incomplétude des informations

3. incomplétude des données :

Les données d'un cas à traiter peuvent être incomplètes. Il peut être nécessaire de faire alors des hypothèses sur les données manquantes.

4. incomplétude du modèle :

On peut ici distinguer deux cas :

- La connaissance est encodée sous forme de règles de production comme dans MYCIN [Shortliffe 76]. Les connaissances que ces règles encodent sont souvent empiriques et parfois ésotériques mais leur simplicité d'expression les rend efficaces dans le processus inférentiel : ce sont des *connaissances de surface*. Le modèle existe même s'il n'apparaît pas explicitement. Dans ce cas, son incomplétude se traduit par l'absence d'un certain nombre de règles.
- La connaissance est fournie sous la forme d'un modèle plus structuré comme dans ABEL [Patil et. al 82] qui possède des primitives d'entrée des connaissances permettant de décrire des relations causales ou des hiérarchies : ces connaissances forment un *modèle profond* ; ce sont des *connaissances profondes*. L'incomplétude du premier cas se retrouve évidemment ici, mais elle se complique du problème de l'utilisation du modèle qui ne peut être pris en compte par un moteur d'inférences élémentaire. Des connaissances supplémentaires sont nécessaires pour décrire l'utilisation qui doit être faite du modèle : ce sont des *méta-connaissances* ; elles peuvent être elles aussi incomplètes.

- fiabilité des inférences

5. fiabilité inhérente aux inférences :

Il est parfois impossible de fournir une corrélation précise entre des événements et la conclusion que l'on peut en tirer. Cette corrélation peut être entre autres, une corrélation statistique comme dans la règle "si tabac alors cancer" (nous avons

ici une probabilité conditionnelle) ou une utilité comme dans la règle “si évolution dangereuse probable alors faire tel examen coûteux”. Ainsi la “force” de l’implication doit pouvoir être modulée. On est alors dans le cadre du raisonnement approximatif.

Nous allons maintenant examiner, en fonction de ces types d’incertitude, les approches les plus appropriées pour les modéliser. Ces approches peuvent être séparées en deux types, *numériques* et *non numériques* ou *symboliques*. Les approches *numériques* permettent d’associer un *degré de certitude* (que ce soit un nombre ou une fonction numérique, comme dans la *logique des possibilités*) à une information. Le raisonnement est conduit en combinant les *degrés de certitude* des différentes informations. Dans une approche *symbolique* “on cherche à raisonner *sur* l’incertitude plutôt que *dans* l’incertitude” [Sombé 89] dans la mesure où, plutôt que d’associer un *degré de certitude* à une information, on essaie de trouver quelles sont les justifications de cette information. On est alors conduit à faire des hypothèses sur des données ou des variables inconnues ou mal connues. Le raisonnement se poursuit alors sur des bases hypothétiques et peut être remis en cause [Cordier 86].

1.1 Qualité de l’information

La qualité d’une information peut être modélisée selon deux axes, son imprécision et son incertitude¹ [Dubois & Prade 85] :

Imprécision : les informations sont jugées ici sur leur contenu. Les qualificatifs renvoyant à l’imprécis sont entre autres, “vague”, “flou”, “ambigu”. Dire “il a entre 30 et 40 ans” ou “il a la trentaine”, sont des informations imprécises. La valeur de vérité attachée à une information est alors modulée.

Incertitude : les informations sont jugées quant à la croyance que l’on a sur leur véracité. Des qualificatifs comme “probable”, “possible”, . . . , peuvent représenter cette incertitude. Des informations peuvent, par exemple, être affectées d’un *degré de certitude* associé à la confiance que l’on a dans leur véracité. Dire “je pense qu’il a 30 ans” est une information incertaine.

Dire “je pense qu’il a entre 30 et 40 ans” est une information à la fois imprécise et incertaine.

L’imprécision et l’incertitude d’une information sont toutes deux modélisables dans le cadre des *logiques numériques* : la *logique probabiliste* ou la *théorie de l’Evidance* [Shafer 76] par exemple. En particulier, la *logique floue* de Zadeh [Zadeh 78] ou la *logique des possibilités* [Dubois & Prade] sont bâties en tenant compte de ces concepts². En revanche, dans le cadre

¹Il y a, semble-t-il des difficultés à définir clairement des concepts d’imprécision et d’incertitude en partant d’un concept d’information lui-même mal défini. Mais il n’est pas dans notre propos de discuter ce problème.

²Dans le cadre de ces logiques des problèmes d’acquisition des connaissances se posent en raison de la modélisation de l’imprécision et de l’incertitude via des fonctions numériques : trouver un jeu de fonctions cohérent et ayant une signification physique n’est pas facile.

d'une *logique qualitative*, l'imprécision ne peut plus être appréhendée de la même façon : dans un cadre formel de propositions et de formules bien formées ([Manna & Waldinger 85]) l'imprécision n'est plus un concept pertinent ; il doit être appréhendé via la sémantique propre des termes de la proposition : TEMPERATURE-ELEVEE est par exemple un terme imprécis.

1.2 Incomplétude des informations

L'incomplétude des informations est bien modélisée par les approches *non numériques*. En effet, l'incomplétude des informations nécessite souvent de faire des hypothèses et de les réfuter ensuite. La plupart des approches sont donc *non monotones*³.

L'incomplétude des données peut être modélisée dans le cadre d'une *logique multivalente*. Aux valeurs de vérité habituelles, *vrai* ou *faux*, sont adjointes d'autres valeurs de vérité comme *inconnu*. La *logique des défauts* de Reiter [Reiter 80] permet de prendre en compte l'incomplétude des données en faisant des hypothèses sur les valeurs correspondant aux données manquantes ; on est encore dans le cadre d'une *logique multivalente* dans la mesure où la valeur de vérité d'un objet peut être "*vrai, faux ou vrai jusqu'à la preuve du contraire...*"

L'incomplétude du modèle est plus difficile à prendre en compte. Si ce modèle n'est pas explicite, c'est difficile : compléter une base des connaissances sous forme de règles de production — des *connaissances de surface* — pose rapidement des problèmes de cohérence. Si le modèle est explicite — on a un modèle *profond* —, l'acquisition et la complétion des connaissances est facilitée : il est plus facile d'écrire une hiérarchie taxonomique que de détailler toutes les relations d'héritage qu'elle sous-tend. De plus, si certaines formes d'incomplétudes sont connues, on peut envisager de les modéliser via les primitives de description du modèle (une relation causale peut être déclarée non exhaustive si on ne modélise pas toutes les causes d'une pathologie). Malgré cela, l'acquisition des connaissances et surtout l'amélioration de celles-ci restent problématiques [Politakis & Weiss 84]. Quel que soit la profondeur du modèle, des *méta-connaissances* sont nécessaires pour diriger le processus d'acquisition, que ce soit au niveau de la cohérence ou de l'incomplétude.

Ce n'est qu'après avoir élaboré au mieux un processus d'acquisition des connaissances que l'on peut essayer de raisonner avec un modèle incomplet.

Les approches de type TMS [Doyle 79] comme l'ATMS de [de Kleer 86] ou le moteur d'inférences SHERLOCK de [Cordier 86], ont été élaborées pour permettre un raisonnement hypothétique en présence d'informations incomplètes. Mais la modélisation d'un raisonnement hypothétique n'est pas triviale. Si les approches de type TMS fournissent les outils nécessaires à garantir la cohérence des connaissances durant un raisonnement hypothétique, ils ne donnent pas les procédures de choix entre les raisonnements ou la manière de résoudre les conflits. Ces procédures sont le plus souvent *heuristiques* et elles doivent être incluses dans les connaissances

³Dans des approches comme l'ATMS de de Kleer [de Kleer 86], les raisonnements sont menés en parallèle ; la logique n'est plus non monotone mais des hypothèses sont toujours faites.

du système : de la même façon que le médecin connaît la façon d'utiliser les connaissances qu'il possède il faut fournir au moteur les *stratégies* nécessaires à l'exploitation de ces connaissances. Ces *stratégies* sont des *méta-connaissances* dans la mesure où elles guident l'utilisation des connaissances.

1.3 Fiabilité des inférences

Les connaissances inférentielles permettent de déduire de nouveaux faits à partir des données ou des faits précédemment déduits. La fiabilité de ces connaissances doit aussi être modélisée. Cette modélisation a été surtout étudiée pour des règles causales du type "si P alors Q", celles-ci pouvant s'appliquer sur des objets complexes comme des "frames" dans PIP [Pauker et al. 76].

Ce sont les approches *numériques* qui permettent une modélisation efficace de cette incertitude. C'est le cas de systèmes experts comme MYCIN [Shortliffe 76], CASNET [Kulikowski & Weiss 82] ou INTERNIST [Pople 82] qui utilisent des facteurs de certitude.

Les principaux reproches faits aux facteurs de certitudes comme à bien des théories *numériques* sont de deux types : il est difficile de développer un système de facteurs cohérent et la sémantique de ces facteurs et de leur combinaison est faible. Des travaux tendant à donner une signification à ces facteurs en les remplaçant par des modulations *non numériques* sont décrits dans [Cordier 86].

Finalement, les types d'incertitude et d'incomplétude pris en compte par les approches *numériques* et *non numériques* sont qualitativement différents. Les approches *numériques* prennent en compte l'incertitude via un processus de combinaison de *fonctions de certitude* et sont pour cela bien adaptées à la résolution des problèmes d'imprécision et d'incertitude sur les données ou les inférences. En revanche, les approches *non numériques* qui prennent en compte l'incertitude via des *logiques hypothétiques* sont mieux adaptées à la résolution des problèmes d'incomplétude des connaissances [Bhatnagar & Kanal]. Des connaissances *heuristiques* doivent être associées à ces dernières approches pour permettre de modéliser un raisonnement hypothétique dans tout ce qu'il a d'informel. En tout état de cause, un travail sur l'acquisition des connaissances est nécessaire.

2 Objectifs du projet

2.1 Caractéristiques du système

Le système expert LÉZARD a été écrit pour résoudre le problème du diagnostic dans un environnement incertain.

Nous avons rapidement vu que les systèmes experts doivent utiliser un mode de représenta-

tion des connaissances évolué, c'est à dire un modèle *profond*, pour limiter l'incomplétude des connaissances. Cela permet entre autres, de saisir différents niveaux pertinents d'abstraction dans le cadre d'un modèle⁴ complexe. Si ABEL [Patil et al. 82] ou SESAM [Ferrand 84] par exemple, permettent de décrire des modèles physiopathologiques complexes, ils ont toutefois des difficultés à aborder le raisonnement incertain.

Dans les systèmes où le raisonnement incertain est encodé de manière *numérique* (comme MYCIN ou SAM [Gascuel 85]) les coefficients apparaissant dans le raisonnement n'ont pas de signification physique. Il est difficile d'en donner des valeurs a priori. L'optimisation du système de coefficients est alors souvent nécessaire bien que sa signification du point de vue des connaissances médicales ne soit pas toujours très claire [Golmard & Rodary 85].

De plus les systèmes actuels n'ont pas la capacité, ou de faire des hypothèses en cas de données insuffisantes, ou de gérer les contradictions entre données initiales. Des programmes comme PIP [Pauker et al. 76], ABEL ou ANGINA [Long et al. 84] doivent par exemple faire appel à l'utilisateur pour au moins une de ces fonctions. Dans certains cas il est aussi simple de donner directement le diagnostic que de répondre à ce type d'interrogation.

De ceci découlent les principales contraintes que nous nous sommes imposées durant la réalisation de notre travail :

- Il faut utiliser un formalisme assez riche pour permettre de décrire aisément les mécanismes complexes.
- Le modèle peut porter sur des situations dont on sait que la description est incomplète : situations mal connues ou dont la description est trop compliquée pour la profondeur des connaissances utilisées. Dans ce cas le langage doit comporter les primitives nécessaires à l'expression de cette incomplétude.
- Le système doit être à même de faire des hypothèses sur des données manquantes.
- Des connaissances *heuristiques* doivent être intégrées au système pour permettre de modéliser ce raisonnement hypothétique dans tout ce qu'il a d'informel.
- Un travail sur la logique même de la décision ou, en d'autres termes, sur le raisonnement incertain est indispensable.
- Trouver un modèle acceptable pour le raisonnement incertain est un travail expérimental. C'est pourquoi il faut adopter une architecture générale du système qui permette de changer aisément de mode de raisonnement et de tester différentes hypothèses mais en conservant le formalisme externe. Ce qui nécessite deux niveaux de langage.
- Un travail sur l'acquisition des connaissances est en tout état de cause nécessaire.

⁴Nous employons ici "modèle" dans son sens courant et non dans son sens logique.

2.2 Choix du domaine

Le domaine médical est, de par sa complexité, un domaine de recherche intéressant : les contraintes qu'il impose, que ce soit du point de vue des connaissances ou du raisonnement rendent cruciaux les problèmes de représentation des connaissances comme de modélisation de l'incertitude.

Le système est appliqué aux douleurs aiguës de l'abdomen (cf. chapitre 5). Les raisons du choix de ce domaine sont principalement :

- un intérêt méthodologique : la caractéristique de ce domaine est qu'il impose un raisonnement incertain de "bonne qualité". L'élaboration des bases de ce raisonnement est alors le noyau du problème ;
- la présence d'une base de données sur le sujet (cf. chapitre 5).

3 Le système Léopard.

3.1 Choix des approches

En fonction des contraintes imposées par nos objectifs, nous avons essayé de privilégier certains axes du système LÉZARD :

- l'élaboration d'une procédure d'acquisition des *connaissances de surface* à partir de *connaissances profondes* — ce processus étant guidé par des *méta-connaissances* ;
- la mise en place d'un modèle de raisonnement incertain combinant à la fois une approche *symbolique* et une approche *numérique* ; des *stratégies heuristiques* étant utilisées pour contrôler le raisonnement.

Par la suite, nous emploierons l'adjectif *heuristique* pour qualifier toute connaissance empirique, c'est à dire à la fois les *connaissances de surface* et les *méta-connaissances* qui dirigent le processus d'acquisition des connaissances. Le terme de *stratégie* sera réservé pour les *méta-connaissances* utilisées au niveau du moteur d'inférences pour le choix des *meilleurs* raisonnements.

3.2 Structure du système Léopard

Le système LÉZARD est un système *essentiel* dans la mesure où il ne dépend pas d'une application. Toutefois, il a été élaboré pour effectuer de la discrimination diagnostique et ne

serait efficient que dans des tâches de même nature. L'architecture du système est modulaire (cf. figure 1.1).

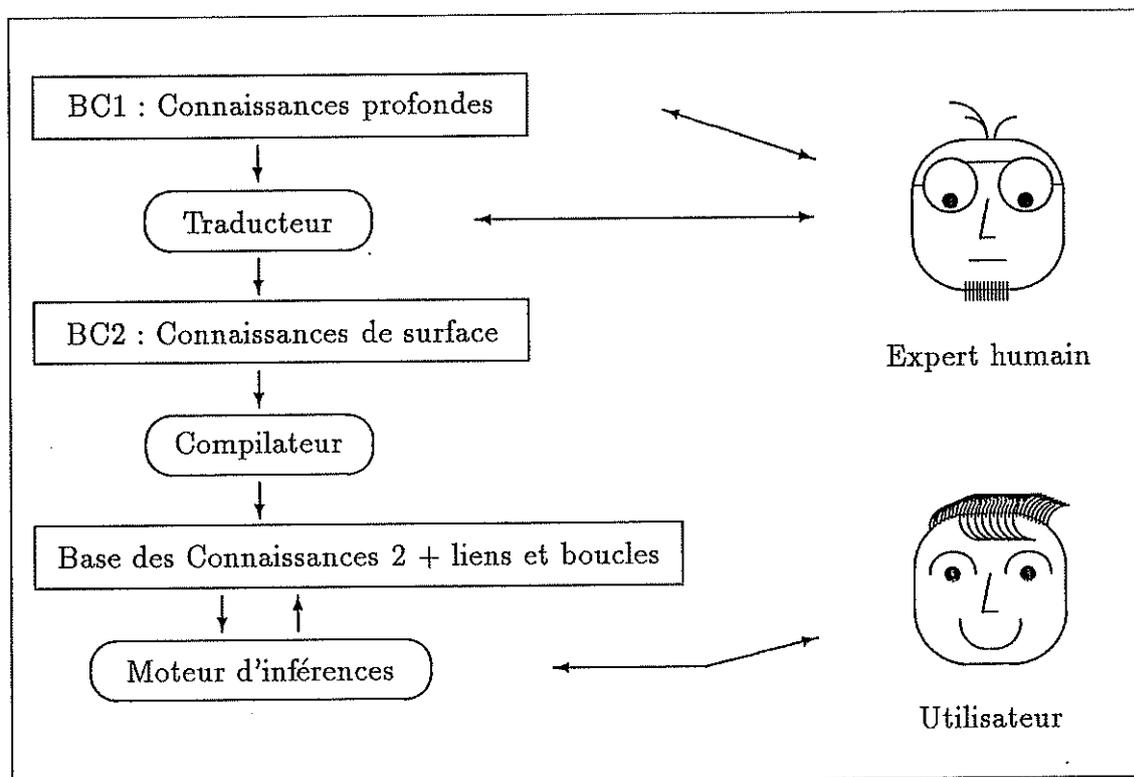


Figure 1.1: L'architecture du système LÉZARD.

La base des *connaissances profondes* est écrite dans un langage de haut niveau, aussi concis que possible. Deux types de connaissances sont représentées : les connaissances physiopathologiques⁵ et les connaissances sémiologiques. Les connaissances physiopathologiques, purement statiques, sont représentées dans un réseau sémantique⁶. Ce réseau est constitué d'objets (uniquement des diagnostics) reliés entre eux par deux types de relations : les relations taxonomiques et les relations causales. Les taxonomies sont de deux types : les taxonomies anatomiques et les taxonomies de spécialisation par processus pathologique. Les connaissances sémiologiques établissent les relations entre les signes et les diagnostics, entre les syndromes et les diagnostics, et plus généralement entre les faits et les diagnostics. Ces relations permettent d'exprimer le tableau caractéristique d'un diagnostic, de donner les signes permettant d'évoquer un diagnostic ou d'affecter un individu à une classe diagnostique.

La figure 1.2 montre un exemple dans le domaine des douleurs aiguës de l'abdomen. Ce réseau est exprimé, ici, dans le formalisme du système LÉZARD et comprend :

⁵Nous incluons dans ce terme les connaissances physiologiques et physiopathologiques.

⁶Le réseau est sémantique car les relations reliant les objets ont une signification (sémantique) qui est claire.

- Des taxonomies anatomiques :

APPENDICITE et OCCLUSION-ORGANIQUE is-a ATTEINTE-DIGESTIVE

- Une taxonomie de spécialisation des processus physiopathologiques :

OCCLUSION-FONCTIONNELLE et OCCLUSION-ORGANIQUE is-a OCCLUSION

- Des relations causales :

PERITONITE ou COLIQUE-NEPHRETIQUE peut-causer OCCLUSION-FONCTIONNELLE
 APPENDICITE ou INFARCTUS-MESENTERE ou OCCLUSION-ORGANIQUE peut-causer PERITONITE
 PERITONITE ou INFARCTUS-MESENTERE peut-causer OCCLUSION-ORGANIQUE

- Des connaissances sémiologiques :

DOULEUR-OMBILICALE évoque APPENDICITE ou ATTEINTE-VASCULAIRE
 DOULEUR-EPIGASTRIQUE évoque OCCLUSION-ORGANIQUE

Le TRADUCTEUR traduit les *connaissances profondes* dans le formalisme des règles de production : “(et $P_1 P_2 \dots P_n$) $\rightarrow C$ ”, où P_i et C sont des littéraux. Ce sont des *connaissances de surface* (cf. 1 ch. 3). Par exemple la taxonomie :

OCCLUSION-FONCTIONNELLE et OCCLUSION-ORGANIQUE constituent OCCLUSION

est traduite, entre autres règles, en :

(non OCCLUSION) \Rightarrow (non OCCLUSION-FONCTIONNELLE)
 (non OCCLUSION) \Rightarrow (non OCCLUSION-ORGANIQUE)
 OCCLUSION-FONCTIONNELLE \Rightarrow OCCLUSION

Au cours de cette phase les connaissances du système sont augmentées de manière interactive. Des règles sont générées par différents modules d'interprétation des connaissances et sont proposées à l'expert. Des *méta-connaissances* guident l'interrogatoire de ce dernier.

Les règles de la seconde base de connaissances subissent un prétraitement — une compilation [Konolige 79] — destiné à accélérer la phase d'interprétation [Gascuel et al. 85], et forment ainsi la base de connaissances associée au moteur d'inférences.

Le moteur d'inférences, d'ordre 0, est un moteur de type “chaînage avant” fonctionnant dans le cadre d'une *logique non monotone*. L'incertitude du raisonnement y est appréhendée via une *logique symbolique*. Le raisonnement supportant une conclusion C est l'arbre *ET* des règles déclenchées qui amènent à C . Il est associé à cette conclusion C et en est la justification. Il

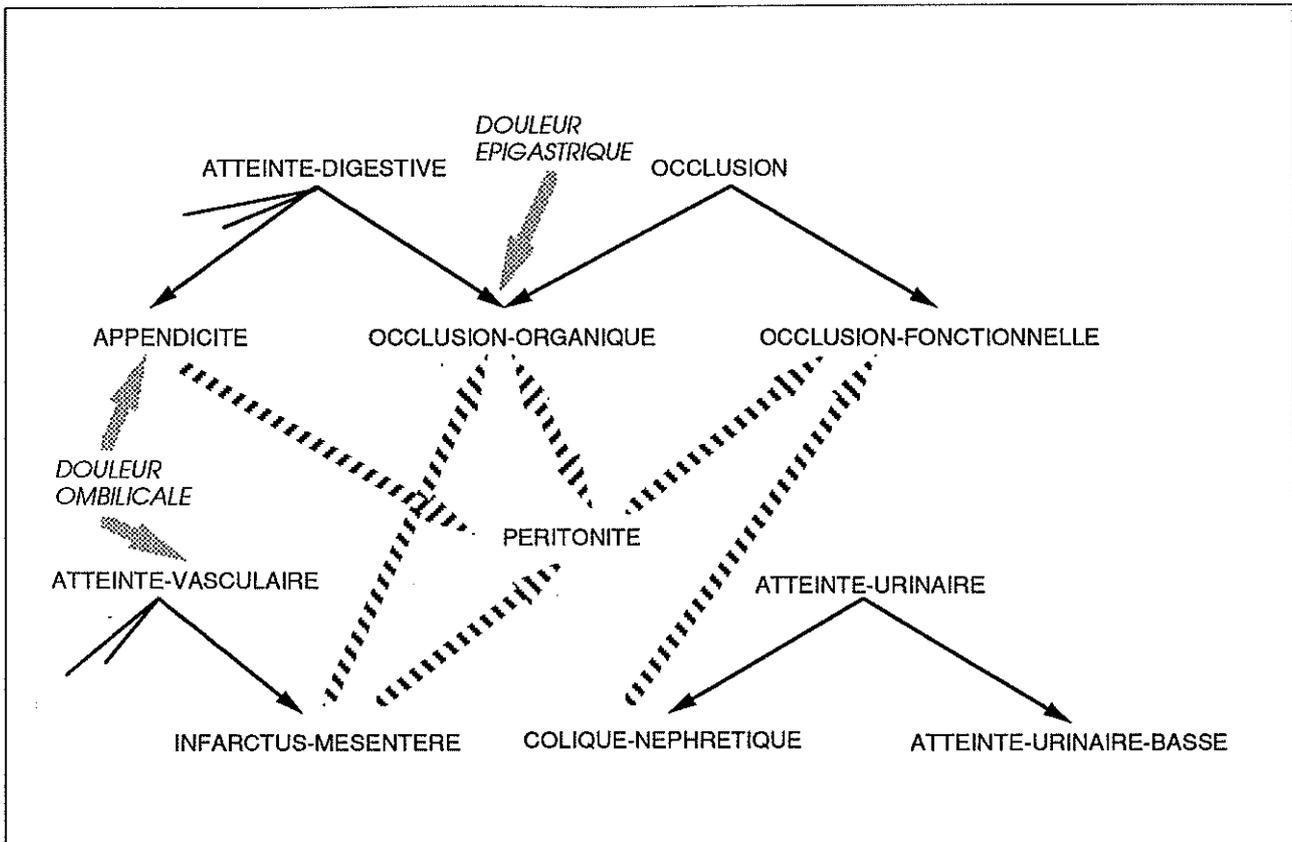


Figure 1.2: Représentation du réseau sémantique. Les flèches épaisses interrompues représentent les liens causaux, les grisées les relations d'évocation, les noires les liens taxonomiques.

est nécessaire de faire des hypothèses quand apparaît un conflit entre deux raisonnements. Les hypothèses que l'on fait portent sur le choix des raisonnements : choisir entre deux hypothèses revient à résoudre un conflit entre deux raisonnements. Des *stratégies heuristiques* sont mises en œuvre pour choisir le meilleur raisonnement : des procédures de discrimination *qualitatives* (*non numériques*) s'appuyant sur les justifications sont utilisées en premier lieu, puis si le choix n'a pu se faire, une procédure *quantitative* (*numérique*) est activée.

Chapitre 2

La représentation des connaissances

Introduction

Dans ce chapitre nous nous intéressons à la problématique de la représentation des connaissances, de l'acquisition des connaissances et à la dualité inévitable qui apparaît entre représentation et interprétation des connaissances.

Cela nous oblige d'abord à recenser quel type de connaissance l'on doit fournir à un système expert en médecine. C'est ce que nous ferons dans une seconde partie.

Dans une troisième partie nous décrivons complètement le langage de représentation des connaissances du système LÉZARD.

Finalement nous discutons de la manière dont la représentation des connaissances choisie résout la problématique de l'acquisition des connaissances. Nous montrons comment les solutions retenues imposent la présence et les caractéristiques d'un TRADUCTEUR qui fera l'objet du prochain chapitre.

1 Problématique

Acquisition des connaissances

Quel que soit le mode de représentation des connaissances choisi, la phase d'acquisition et d'amélioration de celles-ci est problématique : Il est ainsi très difficile de se rendre compte de l'influence sur les résultats fournis par le moteur d'un ajout dans la base des connaissances. Cela est particulièrement vrai dans les systèmes à règles de production dans lesquels toute la connaissance est au même niveau de généralité, et la cohérence de la base difficile à tester¹.

Dans des systèmes dont la représentation des connaissances est de plus haut niveau (à base de réseau sémantique par exemple) l'influence d'une modification est plus facile à appréhender et la cohérence est plus aisée à maintenir dans le processus d'amélioration.

Ainsi, plus le mode de représentation des connaissances est clair, plus les conséquences d'un changement dans les connaissances sont visibles, plus l'amélioration des connaissances sera facile.

Dualité Représentation/Interprétation des connaissances

Bien qu'un système expert soit censé être déclaratif dans l'expression de ses connaissances, il ne peut l'être totalement. Le langage de représentation des connaissances qui permet au système de mémoriser les connaissances d'un domaine est à la source d'un problème inévitable : ce langage ne peut être découplé de l'usage qui va être fait des informations qu'il contient. Il peut être déclaratif dans la mesure où les connaissances représentées sont modifiables à l'envi, mais sa forme et sa signification (sémantique) dépendent de l'usage que l'on veut en faire. Le problème de la représentation des connaissances est inséparable du problème dual, l'interprétation de ces connaissances.

De ce point de vue les rapports entre le langage de représentation des connaissances et son interpréteur — souvent le moteur d'inférences — sont primordiaux : Ainsi, les capacités d'expression des langages sont telles — et ce d'autant plus que le langage est élaboré — que les connaissances représentées ont un contenu sémantique complexe. Il existe alors plusieurs interprétations possibles d'un langage et si l'expert ne sait pas quelle est l'interprétation choisie par le moteur, il ne la maîtrise pas et des résultats inattendus peuvent être obtenus. Il faut donc informer l'expert de l'utilisation et de l'interprétation qui sera faite de ses connaissances.

Raisonnements fondamentaux et structure des connaissances

Un système expert de diagnostic possède des principes et des modes de raisonnement fondamentaux directement liés à la structure de la base des connaissances. Ces modes de raisonnement sont ceux que l'on répertorie habituellement comme étant nécessaires à la simulation du processus diagnostic. Rappelons-en quelques uns dans le domaine médical en notant que contrôler le nombre d'hypothèses est le premier principe à appliquer dans l'élaboration d'un programme d'aide au diagnostic en intelligence artificielle [Szolovits et al. 88] :

— évocation d'hypothèses par les signes d'un patient ;

¹Si les règles sont pondérées (cf. 1.1.3) cela est encore plus difficile. Par exemple, de l'avis même de ses concepteurs, la représentation des connaissances du système essentiel EMYCIN est telle que son utilisation et son amélioration est difficilement accessible à une personne ne connaissant pas le cœur du système et ses processus de raisonnement.

- réduction du nombre d'hypothèses par l'utilisation de signes nécessaires à la présence de certains diagnostics ;
- structuration des diagnostics en hiérarchies pour raisonner à divers niveaux et en particulier pour essayer de conclure sur les diagnostics les plus généraux avant d'affiner la recherche ; cela permet de réduire ainsi le nombre de diagnostics retenus à tout moment du raisonnement ;
- regroupements de signes dans le cadre d'un syndrome pour réduire la complexité des raisonnements.

Dans le cadre du moteur d'inférences, les connaissances liées à ces modes de raisonnement sont souvent codées de manière procédurale. Il est évidemment plus élégant — et formellement plus juste du point de vue de la structure du système expert — d'encoder les connaissances d'un domaine de manière déclarative. Mais des raisons d'efficacité et de représentation obligent souvent à en programmer certaines de manière procédurale. La difficulté est alors de les expliciter pour pouvoir les faire intervenir à bon escient dans le processus de contrôle de l'inférence. L'expert peut ainsi les connaître si ce n'est influencer dessus.

Nous allons donc essayer de recenser les connaissances que doit "comprendre" un système expert en médecine et indiquer comment elles doivent être encodées pour répondre à notre problématique.

2 Les connaissances médicales

Les réflexions faites par P. Szolovits [Szolovits et al. 88] sur les connaissances nécessaires au diagnostic médical ont permis de dégager des principes généraux de structuration des connaissances qui répondent à la problématique développée au début de ce chapitre.

Les connaissances médicales nécessaires à un système expert sont de trois types : des connaissances dites statiques, des connaissances dites pragmatiques, toutes deux contenues dans la base, et les connaissances de contrôle qui ont trait à l'utilisation des connaissances précédentes, à la manière de raisonner du système.

Dans le cas d'un cadre diagnostic plus général que la médecine, les mêmes composantes existent. Nous les décrirons dans le contexte médical en notant les caractéristiques hors contexte quand l'analogie avec un domaine diagnostic plus général est intéressante.

Les connaissances statiques

Le domaine d'application du système expert, son univers de travail, peut être décrit par : d'une part un ensemble de connaissances bien définies, acceptées par tous les experts et d'autre

part des connaissances moins absolues, traduisant plutôt le savoir faire d'un expert donné. Nous appelons connaissances statiques le premier ensemble de connaissances. Ces connaissances statiques doivent permettre de décrire les maladies dont un patient peut souffrir (les pannes dont peut être atteint le sujet de l'étude). Elles doivent permettre aussi de décrire les signes observables sur un patient. Pour décrire ces connaissances nous avons besoin d'un formalisme qui permette de représenter les états du patient, normaux ou anormaux, en termes de primitives connues par le système.

Le système doit donc savoir prendre en compte la connaissance médicale suivant plusieurs composantes structurelles fondamentales qui permettront la mise en place du raisonnement médical :

- une composante anatomique qui comprend des hiérarchies sur les organes, des positions et des inclusions entre organes ;
- une composante physiologique qui doit permettre de décrire les états normaux de fonctionnement du système humain dans le domaine considéré ;
- une composante physiopathologique qui permet de décrire les états anormaux du patient à travers (i) des taxonomies de processus pathologiques, et (ii) des relations causales qui décrivent comment le changement d'un état influe sur les autres états.
- ...

Les connaissances pragmatiques

Les connaissances pragmatiques forment le second volet de la description de l'univers de travail. Moins certaines que les précédentes, ce sont principalement elles que l'expert modifiera pendant la phase de mise au point de la base. Typiquement elles sont représentées à l'aide de primitives décrivant l'étiologie² des maladies et doivent permettre, comme précédemment, d'exprimer les raisonnements nécessaires au fonctionnement du système :

- primitive permettant l'évocation d'hypothèses par les signes d'un patient ;
- primitive exprimant qu'un signe est absolument nécessaire pour poser un certain diagnostic ;
- primitive permettant le regroupement de signes dans le cadre d'un syndrome ;
- ...

²L'étiologie est la science des causes des maladies.

Les connaissances de contrôle

Les connaissances de contrôle ont trait aux *stratégies*³ de raisonnement retenues pour le système. Elles permettent de guider l'utilisation des connaissances précédentes (statiques ou pragmatiques). Elles sont parfois données au système de manière totalement déclarative et ce sont des méta-règles. Mais le plus souvent elles ne sont pas explicites, car implémentées de manière procédurale (c'est par exemple souvent le cas quand la structure même des connaissances sur lesquelles elles agissent entre en jeu dans la formalisation des connaissances). Citons-en quelques unes [Charlet 88] :

- la focalisation sur les niveaux d'abstraction les plus élevés au sein du modèle physiopathologique ;
- la recherche d'un processus physiopathologique unique ;

Ces connaissances sont parfois l'expression médicale de connaissances plus générales : le moteur d'inférences inclut des *stratégies* de raisonnement d'ordre général indépendantes du domaine médical (comme le raisonnement à plusieurs niveaux) mais pouvant être, par exemple, utiles ou nécessaires à l'accomplissement d'une tâche diagnostique. La séparation entre connaissances indépendantes du domaine et dépendantes du domaine n'est pas toujours facile à faire et dépend plus des buts du système que des qualités intrinsèques des connaissances ; mais dans tous les cas cela n'a aucune incidence sur leur représentation.

Nous ne décrivons pas le mode de représentation de ces *stratégies* pour la raison qu'il dépend de chacune. Nous verrons leur mise en œuvre au niveau du traducteur — à ce niveau elles servent à contrôler l'acquisition des connaissances et sont appelées *méta-connaissances* —, et au niveau du moteur d'inférences.

3 Le langage de représentation des connaissances

3.1 Les restrictions du langage

Sur les considérations abordées dans la section 2 nous avons développé une représentation des connaissances dans un langage de "haut niveau" incluant la plupart des connaissances et propriétés décrites.

Certains concepts ne sont pas représentés dans LÉZARD et ce pour deux raisons :

1. Dans une première version du système il a paru utile de ne pas surcharger l'expert qui écrit la base des connaissances de concepts dont il n'a pas toujours l'usage. En particulier

³Une *stratégie* est un ensemble d'actions coordonnées en vue d'un but. En Intelligence Artificielle cette coordination est souvent informelle et donc heuristique.

certains concepts sont sémantiquement très proches les uns des autres et leur traitements sont apparus a posteriori identiques. Nous n'avons alors introduit qu'une primitive pour leur représentation.

2. L'expert peut agir sur l'interprétation que fait LÉZARD de certaines primitives. Cela nous a paru souvent plus cohérent que de lui permettre d'exprimer deux concepts très proches et de vérifier ensuite s'il a utilisé le bon. Par exemple la primitive *peut-causer* recouvre toutes les significations liées aux relations de causalité, quel que soit le niveau d'abstraction des objets qu'elles relient (et donc la profondeur du modèle causal qu'elles décrivent ; c'est par exemple le cas dans le système ABEL).

Nous allons maintenant décrire les objets et primitives du langage de représentation des connaissances retenu. Le vocabulaire utilisé pour les primitives du langage est orienté vers la discrimination diagnostique dans le domaine médical. Mais la signification (l'interprétation) des mots de ce vocabulaire dépasse largement ce cadre. Ainsi ce vocabulaire est valide dans le cadre de la discrimination diagnostique en général.

Pour ce qui est des objets nous utiliserons des mots extraits du domaine d'application "douleur aiguës de l'abdomen". Ces mots sont choisis par l'utilisateur et seraient évidemment différents dans un autre domaine d'application.

3.2 La description d'un objet

La logique des propositions

En préliminaire à la description des connaissances du système il faut noter que le moteur d'inférences travaille dans le cadre d'une logique des propositions étendue : une logique bivalente permettant de manipuler les valeurs de vérité *vrai* et *faux*, mais enrichie de prédicats élargissant les moyens d'expression.

Nous pouvons remarquer que la logique des propositions n'est pas pénalisante du point de vue de la capacité de représentation du système : en médecine l'outil usuel de description des patients et des maladies est la logique des propositions. Cela est autorisé par l'étendue du vocabulaire médical où tout est nommé et où tous les objets possibles sont répertoriés. Même les évolutions des maladies, si elles sont notables, ont un nom. Le fait de rester dans une logique des propositions simple facilite la formalisation du système et nous allons voir maintenant comment sont utilisées les valeurs de vérité et comment elles sont enrichies.

3.2.1 Définition et valeurs de vérité d'un objet

L'objet est un mot du domaine choisi par l'utilisateur. Par exemple APPENDICITE, PNEUMOPERITOINE. Les objets ne sont pas valués ; ce sont donc des objets binaires ne pouvant prendre

que les valeurs *présent* ou *absent*. Le système LÉZARD raisonne dans une logique des propositions bivalente. Les valeurs de vérité possibles des objets sont au nombre de deux : *vrai*, *faux*.

Il existe des relations évidentes entre les 2 valeurs possibles d'un objet, et les valeurs de vérité *vrai* ou *faux* qu'on peut donner aux propositions "OBJET *présent*" et "OBJET *absent*". Ainsi :

"APPENDICITE *présent*" est *vraie* \equiv "APPENDICITE *absent*" est *fausse*
 "APPENDICITE *présent*" est *fausse* \equiv "APPENDICITE *absent*" est *vraie*

Pour simplifier à la fois l'écriture des propositions et leur manipulation par le moteur, nous avons choisi d'utiliser uniquement la valeur *présent* d'un objet. Toute proposition contenant la valeur *absent* pouvant, à l'aide des règles précédentes, se ramener à une proposition sur la valeur *présent*. De plus, la proposition correspondant à la présence d'un objet sera alors résumée par l'expression de l'objet seul.

APPENDICITE est *vrai* correspond alors
 à la valeur de vérité de "APPENDICITE *présent*" est *vraie*

C'est cette notation qui sera retenue dans la suite de l'exposé.

Valeur de vérité inconnue

En plus des valeurs de vérité *vrai* et *faux* on associe une "pseudo" valeur de vérité aux objets : "inconnu". Celle-ci représente le fait que l'on ne sait pas si l'objet est *faux* ou *vrai*. Cette information ne peut être représentée par une valeur de vérité dans notre logique des propositions. En effet, dans ce cadre la valeur de vérité *inconnu* ne permet pas une formalisation cohérente du modèle et, entre autres, de la conjonction de faits⁴. Cette "pseudo" valeur de vérité va donc être utilisée dans le système comme *vrai* ou *faux* mais il ne sera pas fait d'opération logique avec ; en revanche, *inconnu* sera utilisée dans le cadre des prédicats portant sur les valeurs de vérité comme nous allons le voir dans le paragraphe suivant⁵.

Prédicats portant sur les valeurs de vérité des objets

Les prédicats portant sur les valeurs de vérité des objets font partie des outils d'expression des connaissances du langage de représentation des connaissances : c'est le cas du prédicat *oui* sous entendu dans les formules du système et du prédicat *non*.

⁴Dans une logique des propositions simple, si l'on a P *inconnu* et Q *inconnu* on ne peut construire la table de vérité de "P \vee Q" ou de "P \wedge Q".

⁵Dans la suite de l'exposé, *inconnu* sera considérée et notée comme une valeur de vérité. Il sera fait état de sa nature particulière si nécessaire.

1. (*oui* OBJET) est *vraie* si OBJET est *vrai*, la formule est *fausse* si OBJET est *faux* et *inconnue* si OBJET est *inconnue*. Comme vu précédemment, le prédicat est omis dans l'écriture des relations.
2. (*non* OBJET) est *vraie* si OBJET est *faux*, la formule est *fausse* si OBJET est *vrai* et *inconnue* si OBJET est *inconnu*.

Deux autres prédicats permettent d'étendre les moyens d'expression du langage : *faux-ou-inconnu* et *vrai-ou-inconnu*. Ces deux prédicats sont à rapprocher du THNOT de PLANNER [Hewitt 72] ou de la modalité M de MacDermott [MacDermott & Doyle 79]⁶.

3. (*faux-ou-inconnu* OBJET) est *vraie* si OBJET est *faux* ou *inconnu* ; elle est *fausse* si OBJET est *vrai*.
4. (*vrai-ou-inconnu* OBJET) est *vraie* si OBJET est *vrai* ou *inconnu* ; elle est *fausse* si OBJET est *faux*.

Ces deux prédicats sont associés aux valeurs de vérité des *signes*, *syndromes* et *diagnostics*. Ils traduisent l'état des objets mais il n'est fait aucune hypothèse sur la possibilité qu'un objet *inconnu* puisse devenir *vrai* ou *faux* via une quelconque démonstration.

Ils peuvent apparaître directement dans une relation si l'expert l'estime nécessaire mais sont, en général, issus du processus de généralisation (cf. 3.3).

Ces quatre prédicats s'appliquant sur les objets du domaine permettent alors de construire les formules du système et, à l'aide des primitives, de construire toutes les relations comprises par le système.

Items valués et logique des propositions

Vu que le système LÉZARD raisonne dans une logique des propositions, il pourrait y avoir des problèmes pour représenter les items valués comme l'âge ou le poids ou ... En fait, nous avons adopté la solution courante qui consiste à définir une partition des valeurs que peut prendre l'item valué ; chaque partie est alors un intervalle auquel on fait correspondre une proposition, un *signe* ici. Ainsi pour l'âge, les objets binaires suivants ont été définis⁷ :

- (1) AGE < 15
- (2) AGE ≥ 15 - < 30
- (3) AGE ≥ 30

⁶Ces modalités sont la base d'une logique non monotone : les logiques monotones ne peuvent prendre en compte le phénomène des nouvelles informations ; de telles modalités le permettent et autorisent ainsi la construction de logique non monotone.

L'expert ayant éprouvé la nécessité d'un quatrième intervalle qui en recouvre plusieurs autres, il a fallu alors définir l'objet $AGE < 30$, les conditions de véracité de la proposition associée et donc poser la règle : $(AGE < 15 \vee AGE \geq 15 - < 30) \Rightarrow AGE < 30$.

En revanche, il n'a pas été utile de poser des relations d'exclusion comme "AGE < 15 et AGE \geq 15 - < 30 s'excluent". En effet, la base des données sur laquelle travaille le système ne donne qu'une instanciation de chaque item valué. Ainsi, des tests de cohérence sur les intervalles se rapportant à un item se sont avérés inutiles.

Cette représentation des items valués n'est pas pénalisante dans la mesure où, dans la plupart des cas, le médecin est intéressé par des "fourchettes" (des intervalles) sur les items et non des valeurs précises. Il se sert même de termes *flous* qui caractérisent les intervalles choisis : le médecin se sert, par exemple, de l'information "température élevée" plutôt que de la valeur exacte "température=39.5" ou de l'information "adolescent" plutôt que de la valeur exacte "age=17". Il se pose alors le problème de la correspondance entre ces termes *flous* et les intervalles portant sur les items ; la solution n'est pas triviale dans la mesure où ces termes ne correspondent pas à des intervalles absolus mais peuvent changer selon le patient étudié. Nous ne discuterons pas plus avant dans cette direction puisque dans notre application nous n'avons que des intervalles à prendre en compte.

Valeur de vérité par défaut

Les objets du domaine ont une valeur de vérité par *défaut* qui leur est affectée avant tout raisonnement. Cette valeur de vérité dépend de la *classe* de l'objet :

- Les *signes* se voient affecter initialement *par défaut* une des 3 valeurs de vérité au moment de l'entrée des connaissances. Ce qui permet, par exemple, de considérer qu'un *signe* est a priori *faux* si aucune information n'est fournie à ce sujet dans les données du malade. En revanche les *signes* présents dans les données le sont avec leurs valeurs de vérité qui sont prépondérantes sur la valeur *par défaut*.
- Les *syndromes* et *diagnostics* sont initialement *inconnus* et ne peuvent être affectés de valeurs de vérité différentes qu'au cours du raisonnement.

Considérer qu'un objet est *faux* si aucune information n'est fournie à son sujet correspond à la convention CWA (Closed-World-Assumption) du monde clos, c'est à dire "tout ce qui n'est pas affirmé vrai est considéré comme faux" [Cordier 86]. Cela permet une économie de représentation mais, avec nos conventions, on ne peut profiter de cette économie puisque seule une partie des objets est dans ce cas.

⁷Cette définition de la partition aurait pu ne dépendre que de l'expert. En fait, elle est directement liée aux signes répertoriés dans la base des données sur laquelle a été testée l'application choisie (cf. chapitre 5).

3.2.2 Classe d'un objet

Les objets appartiennent à l'une des trois *classes* suivantes : *signe*, *syndrome*, *diagnostic*. Chaque *classe* recouvre un certain nombre de concepts :

1. Le *signe* ou observation est un objet dont on obtient la valeur directement. Aucune inférence n'est faite dessus. Ce peut être :
 - des signes cliniques (SOUFFLE-ABDOMINAL, DEFENSE, ...)
 - des antécédents (ANTECEDENT-DOULEUR-SEMBLABLE, ...)
 - des résultats d'examen (LEUCOCYTES>15000, TEMPERATURE>38.5, ...)
2. Le *syndrome* est un regroupement de signes dont l'origine peut varier :
 - des regroupements de signes décrits par l'expert. Par exemple CONTRACTURE, RESP-ABD-NON-VISIBLE, TR-DOULEUR-DOUGLAS, sont regroupés sous le terme SIGNES--PERITONITE ; ce dernier est alors un *syndrome*.
 - des regroupements de signes directement générés par le système à partir de relations mettant en jeu plusieurs signes. Dans ce dernier cas le *syndrome* permet de réduire le nombre de prémisses à instancier dans les règles générées à partir des relations précédentes (cf. 5).
3. Le *diagnostic* est l'objet recherché par LÉZARD. Les *diagnostics* sont finals s'ils correspondent aux maladies répertoriées dans le système comme étant les pathologies recherchées (APPENDICITE, PANCREATITE, ...). Autrement ce sont des *diagnostics* de niveau d'abstraction élevé (ATTEINTE-GYNECO, ATTEINTE-VASCULAIRE, ...) utilisés pour discriminer les *diagnostics* finaux.

Un ordre de précedence est défini entre les *classes* des objets. Il permet de les ranger selon leur place dans le raisonnement : un *signe* est un objet initial du raisonnement, un *syndrome* est un objet intermédiaire du raisonnement et un *diagnostic* est un objet terminal du raisonnement. On peut alors écrire : *signe* < *syndrome* < *diagnostic*.

Cet ordonnancement n'est pas indispensable au raisonnement (à ceci près qu'il faut savoir reconnaître les *signes*, objets sur lesquels il n'est pas fait de déduction). On pourrait assigner à tous les objets une *classe* unique, le programme ferait les mêmes inférences [Gascuel 85]. Mais nous verrons plus loin que cet ordre est utile dans le cadre de l'acquisition des connaissances.

3.2.3 Fiabilité d'un objet

A chaque objet est associée une des 3 *fiabilités* suivantes : *peu-fiable*, *fiable*, *incontestable*. Cette *fiabilité* représente le degré de confiance qu'a l'expert dans la présence de l'objet donc

dans la proposition "(OBJET présent) est vrai"⁸. Par exemple AGE<15 est *incontestable* alors que DEBUT-BRUTAL est *peu-fiable*.

Cette *fiabilité* est utilisée de la même manière que les coefficients de certitude de MYCIN (cf. 1.3 ch. 1), à la différence qu'ici nous avons une pondération discrète. Un algorithme de propagations des coefficients de certitude, adapté à leur caractère discret, permet d'associer un tel coefficient à tout objet durant le raisonnement (cf. 2.4.2.c ch. 4).

Selon leur *classe*, les objets ne sont pas traités de la même manière vis à vis de ces coefficients : Les *signes* ont initialement une des 3 catégories. Celle-ci est fixée dans la description du domaine. Cela n'empêche évidemment pas la remise en cause d'un signe pendant le raisonnement, surtout si sa *fiabilité* est faible. A l'inverse, Les *syndromes* et les *diagnostics* n'ont pas de *fiabilité* au départ puisque, à l'inverse des *signes*, ils ne font pas partie des données du problème. Celle-ci leur est attribuée durant le raisonnement.

Un ordre de précedence est défini entre les *fiabilités* des objets : *peu-fiable* < *fiable* < *incontestable*. C'est cet ordre qui permet d'affirmer qu'un objet a une *meilleure fiabilité* via un raisonnement plutôt que via un autre.

3.3 Les relations

Les relations entre les objets sont exprimées à l'aide de primitives propres au langage de représentation des connaissances. Leur signification est plus ou moins proche de celle qu'elles pourraient avoir dans le langage courant mais le vocabulaire choisi peut être modifié aisément. On peut classer les relations en plusieurs catégories :

1. Les relations du modèle du domaine. Elles décrivent les relations existant entre les *diagnostics* : des relations causales, des relations hiérarchiques (taxonomies anatomiques ou physiopathologique). Ces axiomes forment, en médecine, le modèle physiopathologique du domaine.
2. Les relations sémiologiques ayant trait aux *signes*. Elles permettent de décrire les relations existant entre *signes*, entre *signes* et *syndromes*, entre *signes* et *diagnostics*, et entre *syndromes* et *diagnostics*. Ce sont pour la plupart des relations pragmatiques (par la suite, quand ce ne sera pas le cas, il en sera fait état).

Nous n'avons pas, dans l'application qui sert de base à la description du langage, modélisé le "bon fonctionnement" des organes, ce qui correspondrait au modèle physiologique du domaine ; ce n'était pas nécessaire. Mais rien ne s'opposerait à cette modélisation si cela s'avérait utile

⁸Cette confiance est aussi attribuée à la confiance qu'a l'expert dans l'absence de l'objet donc dans la proposition "(OBJET présent) est faux" (ceci parce que l'expert n'a pas exprimé le désir d'avoir des degrés de confiance différents dans l'absence ou la présence d'un objet).

dans un autre domaine (le système ABEL par exemple, dont le domaine d'application est le diagnostic et la thérapie des troubles électrolytiques et acido-basiques, inclut des connaissances physiologiques sur les équilibres acido-basiques). Il suffirait de créer les primitives adéquates et de déclarer leur signification (cf. chapitre 3).

3.3.1 La signification des relations

La syntaxe

Les relations reliant les objets sont des relations $(n + 1)$ -aires, reliant n fils F_i à 1 père P (cf. figure 2.1).

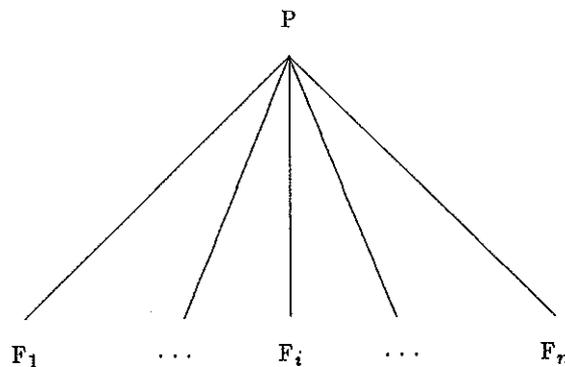


Figure 2.1: relations reliant un ensemble d'objets fils F_i à un objet père P

Les relations du modèle

locaconstituent

Cette primitive permet de construire une relation qui décrit un élément de taxonomie anatomique dans la physiologie ou la pathophysiologie du domaine. Les fils sont supposés exhaustifs.

ATTEINTE-DIGESTIVE et ATTEINTE-URINAIRE et ATTEINTE-VASCULAIRE et ATTEINTE-GYNECO et AFFECTION-POUMON *locaconstituent* DOULEUR-SPECIFIQUE ;

Dans le domaine de la maintenance informatique on pourrait avoir :

ECRAN-DE-VISUALISATION et CLAVIER et SOURIS et CÂBLES *locaconstituent* POSTE-DE-TRAVAIL-INFORMATIQUE ;

Ces deux exemples sont différents dans la mesure où le premier permet de donner les éléments d'un ensemble et le second les composantes d'un ensemble. La primitive prend en compte ces deux types de relations qui seront traitées de la même manière.

locaconstituent-n

Signification semblable à la primitive précédente sauf que l'élément de taxonomie est non exhaustif.

processconstituent

Cette primitive permet de construire une proposition qui décrit un élément de taxonomie dans un processus physiologique ou pathophysiologique du domaine. Les fils sont supposés exhaustifs.

POUSSEE-ULCERE et PERFO-ULCERE *processconstituent* ULCERE ;

Dans le domaine des pannes de voiture on pourrait avoir :

PNEU-DEGONFLE et PNEU-CREVE *processconstituent* PNEU-A-PLAT ;

processconstituent-n

Signification semblable à la primitive précédente sauf que l'élément de taxonomie est non exhaustif.

PERITONITE et OCCLUSION *processconstituent-n* DOULEUR-SPECIFIQUE ;

peut-causer

Permet de relier des diagnostics qui ont une relation de cause à effet possible. Chaque fils présent dans la relation *peut-causer* le père. La relation est censée fournir toutes les causes possibles du père.

OCC-ORGANIQUE-SE ou OCC-ORGANIQUE-PR ou PERFO-ULCERE ou APPENDICITE ou CHOLECYSTITE ou SIGMOIDITE ou GEU ou FAUSSE-COUCHE ou SALPINGITE ou INFARCTUS-MESENTERE *peut-causer* PERITONITE-SE ;

peut-causer-n

Signification semblable à la primitive précédente sauf que la proposition est non exhaustive ; on ne connaît donc pas toutes les causes possibles du père.

ANEVRYSME-AORTE ou COLIQUE-NEPHRETIQUE ou APPENDICITE ou PANCREATITE ou SIGMOIDITE ou PERITONITE-SE ou PERITONITE-PR *peut-causer-n* OCC-FONCTIONNELLE ;

impliquent

Permet de représenter une implication directe : la présence simultanée des fils implique celle du père.

Dans le domaine des pannes de voiture on pourrait avoir :

PANNE-LANCEUR *implique* PANNE-DEMARREUR ;

Les relations sémiologiques (relations où interviennent des signes)

impliquent

Permet de représenter une implication directe : la présence simultanée des fils implique

celle du père. Cette implication est contrainte par le fait qu'elle doit suivre l'ordre normal du raisonnement : des *signes* vers les *syndromes*, puis vers les *diagnostics*. La traduction de cette contrainte sur les littéraux (les objets ou négations d'objets) contenus dans la règle est alors : La *classe* de la conclusion (le père) doit être \succeq (du point de vue de l'ordre défini précédemment : *signe* \prec *syndrome* \prec *diagnostic*) à la *classe* de la prémisse (un fils) dont la *classe* est la plus élevée.

$$\forall i, \text{classe}(A) \succeq \text{Max}(\text{classe}(F_i))$$

Cette primitive permet entre autres de représenter l'implication d'un *diagnostic* par un *signe* pathognomonique⁹ comme le montre le premier exemple :

LAVEMENT-BARYTE-TUMEUR implique CANCER-INTESTIN ;
 DEFENSE et DOULEUR-PROV-LOC-FIG impliquent SIGMOIDITE ;

constant-dans

Relie un *signe* ou un *syndrome* (le "père" de la relation) aux *diagnostics* (les "fils"). La présence du *signe* est une condition sine qua non à la présence d'un des *diagnostics*. Son absence les élimine.

DOULEUR-MOBILISATION-UTERUS constant-dans FAUSSE-COUCHE ;

signale-tous

Définit un *syndrome* par un ensemble de *signes*. La présence simultanée de tous les *signes* est nécessaire à l'affirmation du *syndrome*. Dans la structure retenue on peut définir un *syndrome* à partir d'autres *syndromes*. On a donc :

$$\text{classe}(P) = \text{syndrome}$$

$$\forall i, \text{classe}(F_i) \preceq \text{classe}(P)$$

SYNDROME-AUX-1 signale-tous ARRET-MATIERE et ARRET-GAZ ;

signale

Signification identique à la relation précédente si ce n'est qu'un seul *signe* suffit à affirmer le *syndrome*.

UIV-ANORMALE signale CALCUL-UIV ou UIV-DILATATION ou UIV-REIN-MUET ;

evoque

Relie un *signe* ou un *syndrome* avec tous les *diagnostics* qu'il évoque. La liste des *diagnostics* évoqués est censée être exhaustive. En fait elle ne l'est que pour des *diagnostics* raisonnablement évoqués par le *signe*. Par exemple, si S évoque A ou B fortement et C faiblement, l'expert aura tendance à omettre C. Cette déviation est prise en compte au moment de l'interprétation de la relation.

D-PROV-LOC-HYPOC-D évoque ABCES-FOIE ou CHOLECYSTITE ou CANCER-INT ou OCC-ORGANIQUE-PR ou ATT-URINAIRE ;

⁹Un *signe* pathognomonique est un *signe* qu'on ne rencontre que dans une maladie donnée et dont la présence suffit à établir le diagnostic.

La primitive *evoque* permet aussi de relier directement un ensemble de p signes ou *syndromes* avec les n diagnostics qu'ils évoquent. C'est une facilité d'écriture qui permet à l'expert de déclarer une relation $(n + p)$ -aire. La traduction en p relations *evoque* $(n + 1)$ -aire est effectuée automatiquement par le système. Dans l'exemple suivant on passe d'une relation $(5 + 2)$ -aire à 2 relations $(5 + 1)$ -aires

TV-MASSE ou TR-MASSE *evoque* SIGMOIDITE ou SALPINGITE ou KYSTE-OVAIRE ou GEU ou OCCLUSION ;

est transformée en :

TV-MASSE *evoque* SIGMOIDITE ou SALPINGITE ou KYSTE-OVAIRE ou GEU ou OCCLUSION

;

TR-MASSE *evoque* SIGMOIDITE ou SALPINGITE ou KYSTE-OVAIRE ou GEU ou OCCLUSION

;

evoquent

Relie p signes avec les n diagnostics qu'ils évoquent. La présence simultanée de tous les signes est nécessaire à l'évocation des diagnostics. Implicitement, cela définit un *syndrome* par un ensemble de signes. Le retour aux relations standard *evoque* et *signale* par le biais de la génération automatique d'un *syndrome* permet de traduire une relation $(n + p)$ -aire en deux relations ; une $(n + 1)$ -aire et une $(p + 1)$ -aire. Dans l'exemple suivant, on passe d'une relation $(1 + 2)$ -aire à 1 relation $(1 + 2)$ -aire et 1 relation $(1 + 1)$ -aire ($n = 1$).

ARRET-MATIERE et ARRET-GAZ *evoquent* OCCLUSION ;

est transformée en :

SYNDROME-AUX-1 *signale-tous* ARRET-MATIERE et ARRET-GAZ ;

SYNDROME-AUX-1 *evoque* OCCLUSION ;

3.3.2 Fiabilité d'une relation versus fiabilité d'une règle

Comme les objets, les relations ont une *fiabilité* prise dans l'ensemble {*incontestable, fiable, peu-fiable*}. Comme nous l'avons vu en 3.2 ch. 1, les relations sont destinées à être traduites en règles de production.

La *fiabilité* de la relation est donc destinée à être associée aux règles qui en découlent. En raison de cette association les relations ont une *fiabilité* obtenue de différentes manières :

1. Si c'est une relation ne contenant que des *diagnostics*

- (a) si c'est une taxonomie (*locaconstituent(-n)*, *processconstituent(-n)*) elle est *incontestable* puisque décrivant la hiérarchie des *diagnostics* retenue.
- (b) sinon, sa *fiabilité* est demandée à l'expert. Dans notre cas il ne reste plus que les relations de causalité (*peut-causer(-n)*) ou des implications (*impliquent*) ; elles peuvent effectivement se traduire avec plus ou moins de force.

2. Si c'est une relation reliant des *signes* ou des *syndromes* à un *syndrome* elle sera *incontestable* (*signale(-tous)*) ; sa traduction correspondant à une règle n'étant qu'une étape "passerelle", parfois artificielle dans le raisonnement, elle ne doit pas pouvoir modifier la *fiabilité* du raisonnement et doit donc être pour cela *incontestable*.
3. Si c'est une relation contenant des *syndromes* ou des *signes* en prémisses et un *diagnostic* en conclusion, la *fiabilité* sera demandée à l'expert au vu de la règle (*evoque, évoquent, constant-dans, impliquent, implique*). Mais ces relations contenant des *signes* ou des *syndromes*, la *fiabilité* sera contrainte par celle des objets la constituant : si ce sont des *signes* ce seront eux qui contraindront la *fiabilité* de la relation, si ce sont un ou des *syndromes* ce seront les *signes* associés à ou aux *syndromes* qui contraindront la *fiabilité* de la relation selon l'inégalité :

$$fiabilité(règle) \leq \text{Min} [fiabilité(signes)].$$

L'expert qui va donner la *fiabilité* de la relation va en fait donner la *fiabilité* de la relation combinée à celle des *signes*, donc en fait, la *fiabilité* de la conclusion (un *diagnostic*) de la règle qui en découle. En effet, l'expert a tendance à tenir compte de ces *signes*. On se trouve alors dans une situation où la *fiabilité* de la conclusion sera fonction de celle de la règle et de celles des prémisses par une fonction du type

$$fiabilité(conclusion) = \mathcal{F} [fiabilité(règle), fiabilité(prémisses)] \text{ avec} \\ fiabilité(conclusion) \leq \text{Min} [fiabilité(règle), fiabilité(prémisses)]$$

Ainsi dans le cadre de l'algorithme de propagation de certitude mis en place (cf. 2.4.2.c) la *fiabilité* de la conclusion de ces règles sera déjà connue.

Comme pour les objets, un ordre de précedence peut être défini entre les *fiabilités* des relations. Il est donné par : *peu-fiable* \prec *fiable* \prec *incontestable*.

4 Les exceptions

4.1 Introduction

Les règles de la base des connaissances du moteur sont "plus ou moins vraies" : nous avons utilisé, pour caractériser chacune des règles, une *fiabilité* prise dans l'ensemble {*incontestable, fiable, peu-fiable*}. Ces *fiabilités* sont une manière d'exprimer des degrés de certitude de façon non numérique. Dans le cadre d'un raisonnement probabiliste on peut considérer qu'une règle est généralement vraie en fixant un seuil s et en posant que la règle " $A \Rightarrow B$ " est généralement vraie si $P(B/A) > s$.

Cette définition est à la fois simple et naturelle. Elle est utilisée par Halpern et McAllester dans un article [Halpern & McAllester 84] où ils étudient les liens entre logique et probabilités. Le type de question qui se pose à eux dans leur raisonnement probabiliste ou à nous est : si la

règle R_1 : "A \Rightarrow B" est généralement vraie et si la règle R_2 : "B \Rightarrow C" est généralement vraie peut on dire que la déduction de A à C est généralement vraie ? Bien sûr la réponse est non. Il suffit de remplacer A par ORNITHORYNQUE, B par MAMMIFERE et C par VIVIPARE pour en être convaincu. Pourtant un raisonnement probabiliste qui permettra de vérifier si $P(C/A) > S$ ou un raisonnement à base de degré de certitude (*numérique* ou *non numérique*) ne sont pas capables de fournir une bonne réponse : ils ne savent pas gérer les exceptions.

Les auteurs montrent toutefois qu'en rajoutant dans les prémisses de chaque règle toutes ses exceptions, soit pour notre exemple : "(MAMMIFERE \wedge (*non* ORNITHORYNQUE)) \Rightarrow VIVIPARE", D'une certaine façon il s'agit d'un constat d'échec car il est communément admis en Intelligence Artificielle (cf. par exemple [McCarthy 80]) que les règles ne peuvent en général comporter toutes leurs exceptions. Cependant si le système ne sait pas qu'un ORNITHORYNQUE est ovipare il a peu de chance de l'inventer tout seul.

Notre opinion est que les exceptions doivent bien figurer dans la base de connaissances mais qu'elles ne doivent pas apparaître dans les prémisses de chaque règle. Dans notre exemple on doit avoir : la règle R_2 reste inchangée et la règle R_3 : "ORNITHORYNQUE \Rightarrow (*non* VIVIPARE)" est présente dans la base de connaissances. Le nouveau problème qui apparaît est que les règles R_1 , R_2 et R_3 sont contradictoires. Elles le sont si l'on reste dans le schéma classique de la déduction. Si le système sait que R_3 est une exception de la déduction de A à C par R_1 et R_2 , elles ne le sont plus.

La représentation des exceptions et des raisonnements d'exceptions a fait l'objet de nombreux travaux qui sont décrits par exemple dans [Reiter 80, Touretzki 84, Moinard 87]. Nous proposons deux façons de donner des exceptions. Chacune d'elles entraîne une modification de la déduction. Elles font l'objet des deux paragraphes suivants [Gascuel et al. 85].

4.2 Donner une exception avec une règle shunt

Cette première façon de donner les exceptions est une simple extension de l'exemple poursuivi au cours du paragraphe précédent. Pour donner une exception à une suite de règles "A \Rightarrow $S_1 \Rightarrow S_2 \dots \Rightarrow C$ " on utilise une règle shunt "A \Rightarrow (*non* C)".

Dans la déduction A peut être une situation connue initialement ou une situation déduite. Avant de conclure à C sur la base de A le module déductif — le moteur — regarde s'il n'est pas possible de passer directement de A à (*non* C) par une seule règle. Une méthode similaire a déjà été utilisée par Touretzki [Touretzki 84] pour résoudre les conflits entre héritages contradictoires de propriétés dans un réseau sémantique.

4.3 Donner une exception à l'aide d'une règle contradictoire

L'idée est de nier une règle générale " $S \Rightarrow C$ " en reprenant ses prémisses et en ajoutant une nouvelle prémisse " $(S \wedge S') \Rightarrow (\text{non } C)$ " qui spécifie les cas où la règle est fautive. Les effets de l'exception peuvent être directs comme ci-dessus ou indirects. Dans ce cas la suite de règles " $S \Rightarrow S_1 \dots \Rightarrow C$ " est contredite par la suite de règles " $(S \wedge S') \Rightarrow S'_1 \dots \Rightarrow (\text{non } C)$ ".

Dans la déduction, S peut être une situation connue initialement ou une situation déduite, par contre S' doit être une situation connue initialement (être certaine) pour que l'on tienne compte de l'exception. Avant de conclure à C sur la base de S il faut regarder s'il n'existe pas une règle utilisant S et la donnée initiale S' qui donnerait directement ou indirectement la conclusion ($\text{non } C$).

4.4 Conséquences

Nous avons décrit ici une façon de tenir compte des exceptions dans un raisonnement de sens commun. Cette description montre, si cela était encore nécessaire, la dualité qui existe entre représentation et utilisation des connaissances. Nous ne décrirons pas plus avant les problèmes de spécificité des raisonnements d'exception. Nous le ferons dans le cadre du moteur d'inférences, dans l'étude des critères de choix des raisonnements.

Pour terminer, notons qu'il faut informer l'expert de ces façon de donner les exceptions.

5 Interprétation du modèle physiopathologique

Les relations construites à l'aide des primitives du langage, des objets du domaine et des prédicats s'appliquant sur ces derniers permettent ainsi de décrire les connaissances médicales du domaine.

Comme dans les systèmes experts récents, ce mode de représentation des connaissances permet de saisir différents niveaux pertinents d'abstraction [Szolovits et al. 88], et des modèles physiopathologiques complexes (CASNET [Kulikowski & Weiss 82], ABEL [Patil et al. 82], SESAM [Ferrand 84], ANGINA [Long et al. 84]). Selon la terminologie habituelle [Chandrasekaran & Mittal 83], ces connaissances sont *profondes* ("deep knowledge") et ces systèmes sont dits de *seconde génération* [Steels 88]. La description de la base des connaissances par un modèle *profond* facilite l'acquisition des connaissances — il est plus facile d'écrire une hiérarchie taxonomique que de détailler toutes les relations d'héritage qu'elle sous-tend. Mais les problèmes abordés au début de ce chapitre sont toujours présents sous une forme liée à la représentation des connaissances choisie :

L'acquisition des connaissances et surtout l'amélioration de celles-ci restent problé-

matiques [Politakis & Weiss 84]. De plus leur utilisation n'est pas triviale. De la même façon que le médecin connaît la façon d'utiliser les connaissances qu'il possède il faut fournir au moteur les *méta-connaissances* nécessaires à l'interprétation et à l'exploitation de ces connaissances.

La création de *connaissances de surface* nécessaires à la réalisation d'une tâche précise est une des phases de cette interprétation. Cette tâche peut être diagnostique ou thérapeutique, entre autres [Sembugamoorthy & Chandrasekaran 86]. Prenons un exemple : soient trois diagnostics A B et C tels que A peut causer B et B peut causer C — des *connaissances profondes*. Un programme d'aide au diagnostic aura tendance à proposer la séquence : A a causé B qui a causé C. Or il existe des cas où A et C sont en fait incompatibles. Si la connaissance "A \Rightarrow non C" et "C \Rightarrow non A" — des *connaissances de surface* — n'est pas explicite, le programme produira des erreurs. Ainsi, des connaissances empiriques ou *de surface*, de gestion plus efficace, doivent être intégrées au système et cette intégration devra nécessairement être guidée par des *méta-connaissances*.

La structure de la base des connaissances et les primitives choisies impliquent déjà des choix de raisonnements. Il faut en tenir compte quand on réalise le moteur d'inférences.

Dans le prochain chapitre nous allons voir comment le problème de l'interprétation des connaissances de haut niveau et leur complétion sont abordés dans le système LÉZARD et quelles solutions y sont proposées.



Chapitre 3

Le traducteur

Introduction

Nous avons vu dans le chapitre précédent les problèmes d'acquisition et d'interprétation des connaissances qui apparaissent inévitablement dans un système expert. Le TRADUCTEUR est le module du système LÉZARD développé pour apporter une solution à ces problèmes.

Nous allons d'abord définir les concepts de connaissances de surface et de connaissances profondes et nous verrons quel est leur utilité.

Nous discuterons ensuite le problème de l'acquisition et de l'interprétation des connaissances dans le cadre du système LÉZARD et comment il est pris en compte dans le TRADUCTEUR. Puis nous décrirons les mécanismes effectivement mis en œuvre pour l'acquisition et pour la traduction des connaissances. Finalement nous décrirons complètement le processus d'acquisition des connaissances du système.

1 Les connaissances profondes et les connaissances de surface

Nous avons introduit précédemment la distinction entre *connaissances profondes* et *connaissances de surface*. Nous allons voir ce que recouvrent ces qualificatifs et quelle est l'utilité de cette partition.

1.1 Définitions

Un expert utilise différents types de connaissances pour résoudre un problème. En général il utilise des connaissances que l'on dit *de surface* ou *compilées* ou *empiriques* ou *expérimentales*. Ces connaissances sont des raccourcis ou des résumés de connaissances fondées sur une compréhension plus profonde du phénomène. Ces dernières connaissances sont dites *profondes* ou *fonctionnelles* ou *physiques* [Fink 87].

1.1.1 Les connaissances de surface

Une connaissance du type " $A \Rightarrow (\text{non } C)$ " est une règle de production. C'est typiquement une *connaissance de surface* : ce caractère de la règle de production n'est pas lié à sa syntaxe mais au peu de sémantique qu'elle transporte. Ses principales caractéristiques sont les suivantes [Fink 87, Klein & Finin 87, Koton 85, Steels & Van de Velde 85] :

- elle décrit une situation où une conclusion est directement tirée des faits caractérisant la situation ;
- la connaissance qu'elle encode est souvent empirique ; dans ce cas comme dans le précédent elle est un vecteur efficace de l'inférence — du point de vue de la rapidité du raisonnement.

Les défauts inhérents aux propriétés des *connaissances de surface* sont :

- elles peuvent difficilement être données directement par les experts, car trop granulaires et trop nombreuses. De plus, leurs interactions sont difficiles à appréhender ; quand un modèle sous-jacent existe, il n'est pas explicité ;
- elles sont pour cela difficiles à maintenir dans le processus d'amélioration de la base ;
- de par leur existence même, elles sont capables de fournir des justifications¹ au raisonnement construit. Mais des explications² du raisonnement ne peuvent être données par la trace des règles utilisées. Cette trace montre parfois l'utilisation d'arguments abscons et le raisonnement ainsi décrit est souvent un raccourci pas toujours compréhensible par le non-expert ;

¹Nous employons ici "justification" dans le sens qu'on lui donne habituellement dans le langage des TMS : la "justification" d'une proposition est une somme d'informations sur le raisonnement mené jusque là et permettant d'appuyer la croyance que l'on a en cette proposition sur la croyance que l'on a dans les autres propositions impliquées dans le raisonnement. Cette justification d'une proposition ne préjuge pas de sa signification.

²L'"explication" d'une proposition rend compte de la compréhension que l'on a des causes qui ont entraîné notre conviction.

- elles ne sont efficaces — et même plus, utilisables — que dans les conditions pour lesquelles elles ont été créées : en dehors des conditions qui satisfont toutes leurs prémisses, les règles de production, par exemple, ne s'appliquent pas (nous verrons que le problème ne se pose pas de la même manière pour les *connaissances profondes*). De plus des préconditions sous-jacentes sont souvent requises pour leur application. Il s'agit le plus souvent de préconditions implicites comme "toutes choses étant égales par ailleurs" ou "dans le contexte du domaine d'application".

Ces règles de production sont caractéristiques des systèmes experts de *première génération* comme MYCIN [Shortliffe 76] ou PROSPECTOR [Konolige 79].

1.1.2 Les connaissances profondes

Les limitations des systèmes de *première génération* — dans la représentation des connaissances en particulier — ont amené leur évolution vers les systèmes de *seconde génération*. Ces derniers permettent eux aussi l'écriture de règles empiriques, mais ils modélisent en plus des *connaissances profondes*. Ces *connaissances profondes* sont typiquement des modèles causaux — ces modèles décrivent, éventuellement à plusieurs niveaux d'abstraction, la fonction de chaque élément représenté et s'attachent à représenter les relations de causalité entre leurs différentes propriétés [Steels & Van de Velde 85, Fink 87]. En médecine, les modèles causaux sont des modèles physiologiques et physiopathologiques, les éléments en étant des entités physiologiques et des diagnostics (dans notre cas les objets du domaine).

Les principales caractéristiques des *connaissances profondes* sont alors les suivantes [Koton 85, Klein & Finin 87] :

- elles explicitent le modèle représenté ;
- en plus des justifications qu'elles peuvent produire elles permettent de générer des explications causales satisfaisantes ;
- elles sont plus facile à maintenir que les *connaissances de surface* car plus générales ;
- elles peuvent aborder des problèmes à la limite du modèle qu'elles décrivent ; ceci parce que quand on décrit un modèle on aborde souvent des considérations qui débordent du strict cadre de l'application : par exemple, dans le domaine des douleurs aiguës de l'abdomen, on va décrire un modèle physiopathologique dont la frontière ne sera pas obligatoirement précise, englobant des pathologies associées comme des mécanismes physiopathologiques hors du domaine des douleurs aiguës mais dont la localisation sera l'abdomen. De ce point de vue les *connaissances profondes* sont moins restrictives que les *connaissances de surface* et donnent encore des informations dans des domaines où les *connaissances de surface* n'auraient pas été écrites.

- mais il est généralement admis qu'elles sont d'utilisation plus complexe et plus lente que les *connaissances de surface*. En particulier, elles ont plus d'interprétations possibles que les *connaissances de surface* et la structure de contrôle du système est plus complexe.

1.1.3 Relativité de ces deux concepts

Les *connaissances profondes* ne sont pourtant pas si différentes des *connaissances de surface* qu'il semblerait à première vue. Il n'y a pas de frontière absolue séparant les *connaissances profondes* des *connaissances de surface*. Les *connaissances profondes* sont aussi profondes que le désire celui qui écrit la base des connaissances. En médecine par exemple, on peut s'arrêter à une description causale qui ne s'intéresse qu'aux relations entre des diagnostics généraux ; mais on peut aussi s'intéresser à un modèle fonctionnel beaucoup plus précis. Ainsi, ABEL [Patil 82] s'intéresse aux troubles acido-basiques d'un patient et contient pour cela des connaissances métaboliques très précises.

La frontière *connaissances profondes*–*connaissances de surface* est redessinée pour chaque nouveau domaine d'application. Elle dépend des modèles que possède l'expert sur le domaine et de la précision de diagnostic demandée.

2 Acquisition des connaissances

Le problème de l'interprétation des connaissances n'est pourtant pas résolu pour les *connaissances profondes*. Cette interprétation passe par l'acquisition de *connaissances de surface* (cf. 5 ch. 2) ; Ceci étant, les défauts et avantages des différents types de connaissances expliquent pourquoi les systèmes experts de *seconde génération* modélisent à la fois des *connaissances de surface* et des *connaissances profondes*. Les *connaissances profondes* y sont alors principalement utilisées :

- pour pallier les défaillances des *connaissances de surface* quand celles-ci ne permettent pas d'aboutir à un diagnostic ; les *connaissances profondes* permettent alors de poursuivre le raisonnement et d'acquérir les *connaissances de surface* faisant défaut [Steels & Van de Velde 85] ;
- pour générer de bonnes explications.
- pour contrôler l'acquisition des *connaissances de surface*. C'est par exemple le cas dans le système ADELE [Reynaud 89] et dans LÉZARD.

2.1 Acquisition des connaissances dans le système Lézard

L'idée qui a présidé à l'élaboration du processus d'acquisition est que l'on a intérêt à ce que ce processus, et en particulier l'acquisition des *connaissances de surface*, ait lieu avant toute inférence et soit le plus complet et le plus efficace possible. Le processus d'acquisition des connaissances dans LÉZARD peut alors être décrit étape par étape :

Acquisition des connaissances profondes et de surface

Un langage de représentation des connaissances de haut niveau (cf. 3 ch. 2) permet de décrire le modèle *profond* et de représenter les *connaissances de surface* (la primitive *impliquent*, par exemple, décrit typiquement une *connaissance de surface*).

Acquisition de nouvelles connaissances de surface à partir des connaissances profondes

Le système va interpréter³ les *connaissances profondes*, en extraire des connaissances *heuristiques*, de nouvelles *connaissances de surface*. Ce processus d'extraction est guidé par des *méta-connaissances* acquises séparément et a lieu en interaction avec l'expert : Les connaissances acquises et donc générées à partir de la base des connaissances initiale doivent être cohérentes avec le modèle. Mais des problèmes de validité se posent cependant en raison de l'incertitude attachée aux connaissances de départ. La solution proposée est donc de laisser l'expert indiquer, lorsque cela semble nécessaire, si une connaissance générée est ou non valide. LÉZARD permet à l'expert de donner un avis plus fin que seulement accepter ou rejeter une connaissance. Puisque les difficultés viennent des incertitudes, il est possible d'accepter une relation tout en la considérant peu fiable. Une *fiabilité* sera donc attachée à chaque nouvelle relation (cf. 3.3.2 ch. 2) ; elle pourra être directement imposée par le système ou soumise à l'appréciation de l'expert. Notons qu'une relation générée à partir d'une relation initiale possèdera une *fiabilité* inférieure ou égale à la *fiabilité* de la relation initiale.

Durant cette seconde étape, les *connaissances de surface* générées le sont sous la forme de relations écrites dans la syntaxe du langage de haut niveau puis toutes les connaissances — *profondes* ou *de surface* — écrites par l'expert ou générées par le système sont mises sous la forme comprise par le moteur d'inférences : Pour le système LÉZARD la représentation retenue est la règle de production à conjonction de prémisses et avec une seule conclusion, donc une règle du type : " $(A \wedge B) \Rightarrow S$ ".

La coopération (introduite dans la seconde étape) entre le système et l'expert n'est pas obligatoire, dans la mesure où l'expert n'est pas indispensable. Mais il doit alors être remplacé par un autre processus qui contrôle les connaissances acquises : en effet le système ne peut interpréter les *connaissances profondes* tout seul : il doit, ou recevoir des conseils de l'expert comme ici, ou tester les connaissances qu'il aura générées sur des exemples.

³Cette interprétation ne signifie pas que le moteur n'en fasse plus. Le moteur doit toujours faire une interprétation sinon il ne peut faire d'inférences. Mais tout ce qui sera fait ne sera plus à faire et simplifiera d'autant la tâche du moteur.

Le tableau 3.1 résume le processus d'acquisition des connaissances dans le système LÉZARD.

0	Acquisition des <i>méta-connaissances</i>
1	Acquisition des <i>connaissances profondes</i> et de <i>surface</i> via le langage de haut niveau
2	Interprétation de ces connaissances
(a)	Interprétation et complétion des <i>connaissances profondes</i> en termes de <i>connaissances de surface</i>
(b)	Acquisition de nouvelles <i>connaissances de surface</i> nécessaires à une tâche diagnostique
(c)	Mise sous forme de règles de production de l'ensemble des <i>connaissances de surface</i>

Figure 3.1: Acquisition des connaissances dans le système LÉZARD.

2.2 Avantages et inconvénients de l'approche développée

L'intérêt de cette approche est multiple :

- Un moteur fait obligatoirement une interprétation des connaissances qu'il utilise. Si ces dernières sont des *connaissances profondes*, donc de "haut niveau", elles ont un contenu sémantique complexe qui autorise plusieurs interprétations logiques valides. La traduction des *connaissances profondes* en *connaissances de surface* oblige à ne retenir que les interprétations sémantiquement valides. Ces *connaissances de surface* sont alors traitées sans ambiguïté par le moteur tout en permettant un contrôle plus précis ;
- l'information fournie par les *connaissances profondes* est complétée de manière très précise au niveau des *connaissances de surface* très granulaires ; les *méta-connaissances* mises en œuvre pour cette complétion permettent de résoudre de façon satisfaisante les problèmes d'incertitude (cf. 3 ch. 3) ;
- la génération des *connaissances de surface* au niveau d'une phase préliminaire aux inférences permet de découpler les processus ; on peut plus facilement modifier les fonctionnalités du TRADUCTEUR sans interférer sur le moteur d'inférences et en particulier on peut ainsi configurer les connaissances du système en fonction des tâches qui lui sont dévolues ;

- l'acquisition des connaissances est rapide et de qualité ; cette affirmation ne pouvant être justifiée que par l'expert et au niveau des résultats du système ;
- l'unité de représentation sous forme de règles de production et leur compilation (cf. 1 ch. 4) permet une plus grande rapidité d'exécution du moteur ;
- les avantages des deux types de représentation sont conservés. Seule la génération d'explication demande une petite gymnastique dans la mesure où il est nécessaire de conserver la représentation sous forme de *connaissances profondes* pour que les *connaissances de surface* puissent y faire appel en cas de demande d'explications.

Mais il ne faut pas dissimuler les choix et les contraintes que cela impose dans la description des connaissances, en particulier parce que les *connaissances profondes* sont incomplètes et interprétables de plusieurs façons. Ainsi l'interprétation des *connaissances profondes* doit être connue de l'expert : il doit savoir quelles règles vont être créées à partir des *connaissances profondes* fournies pour connaître la signification précise de ce qu'il exprime.

3 Les mécanismes d'interprétation des connaissances

Nous allons décrire maintenant les mécanismes mis en œuvre pour, à partir des *connaissances profondes*, générer de nouvelles *connaissances de surface* qui déterminent l'interprétation des premières. Nous verrons d'abord le mécanisme général : LÉZARD propose des *connaissances de surface* et l'expert les accepte ou les refuse. Puis nous détaillerons les connaissances qu'il peut proposer : règles d'exclusion, de généralisation, d'abduction...

3.1 Les connaissances implicites ou explicites

3.1.1 Problématique dans le cadre de la dérivation formelle

Nous allons voir sur un exemple de traduction que dans un domaine dans lequel l'imprécision de l'information est de mise il est souvent abusif d'appliquer des mécanismes qui sont licites dans une logique classique.

Ceci étant, comment utiliser la connaissance $C : (S \equiv (A \vee B))$ dont une écriture dans un formalisme proche du système LÉZARD pourrait être (A et B CONSTITUENT S) ?

Une méthode, si l'on reste en logique classique, est la dérivation formelle. La connaissance S peut, par dérivation formelle, produire le groupe de propositions qui suit :

P1 : $A \Rightarrow S$	P5 : $((non\ A) \wedge (non\ B)) \Rightarrow (non\ S)$
P2 : $B \Rightarrow S$	P6 : $(S \wedge (non\ A)) \Rightarrow B$
P3 : $(non\ S) \Rightarrow (non\ A)$	P7 : $(S \wedge (non\ B)) \Rightarrow A$
P4 : $(non\ S) \Rightarrow (non\ B)$	

Ces propositions sont des règles de production au sens classique du terme. Elles peuvent être directement utilisées par un moteur d'inférences simple. Pourtant en présence d'incertitude la dérivation formelle n'est pas toujours licite et la proposition P6 peut être fausse.

Plaçons nous par exemple dans un cadre probabiliste où l'on considère qu'une proposition est vraie si elle a une probabilité supérieure à 0.8 et donnons un sens à S, A et B :

- S : jouer trois fois de suite à pile ou face et observer les résultats
- A : sur ces trois fois on observe au moins une fois pile et une fois face
- B : on observe trois fois face.

La proposition C est alors "Si l'on joue trois fois de suite à pile ou face on observe au moins une fois face". Elle peut être considérée comme vraie (un calcul probabiliste donne 7/8). Par contre la proposition P6 : "Si l'on joue trois fois de suite à pile ou face et si l'on observe trois résultats identiques alors on observe trois fois face" est fausse (un calcul probabiliste donne 1/2 et il est évident que l'on peut observer trois fois pile avec la même probabilité). La dérivation qui permet de passer de C à P6 n'est donc pas valide.

Dans cette exemple on croit la proposition C : $(S \equiv (A \vee B))$ et donc $(S \Rightarrow (A \vee B))$. Implicitement, on croit alors P6 $((S \Rightarrow (A \vee B)) \equiv (\neg S \vee (A \vee B)) \equiv (\neg S \vee A \vee B) \equiv (\neg(S \wedge \neg A) \vee B) \equiv (S \wedge (non\ A)) \Rightarrow B)$, et c'est là que l'on fait erreur car $(S \Rightarrow (A \vee B))$ est une connaissance incertaine et l'équivalence logique entre C et P6 est fausse dans un cadre incertain.

3.1.2 Croyance implicite/Croyance explicite

Parmi toutes les dérivations formelles possibles, il faut donc sélectionner celles qui sont valides. Ce sera à l'expert d'accepter ou de refuser les connaissances dérivées. Cependant il n'est pas raisonnable de lui demander son avis sur toutes les connaissances obtenues, leur nombre devenant rapidement trop important. C'est donc le système qui choisit les connaissances à soumettre à l'appréciation de l'expert. Ce choix est fait à partir de considérations logiques, mais également *heuristiques*. Ces *heuristiques* peuvent conduire à un ensemble de questions sous-optimal, en ce sens que certaines d'entre elles ne sont pas absolument nécessaires pour éviter des dérivations abusives. Mais il est préférable d'avoir un ensemble un peu trop grand de questions plutôt que laisser faire des dérivations non valides.

En cherchant un cadre logique à ce processus on constate qu'on se rapproche ici des concepts de *croyance implicite* et *croyance explicite* développés dans la logique L de Levesque [Levesque 84] (voir aussi [Charlet & Gascuel 86]) : La *croyance implicite* est la croyance possible. Si

l'on croit P et $(P \Rightarrow Q)$ alors croire Q est une *croissance implicite*. La *croissance implicite* obéit aux règles classiques de dérivation. La *croissance explicite* est la croissance directe. Elle obéit à des règles de dérivation beaucoup plus limitées. Par exemple on pourra croire explicitement $(A \vee B)$ sans croire explicitement A ou croire explicitement B : cela signifie que dans certains contextes l'on ne peut pas appréhender A ou B isolément et que seul $(A \vee B)$ est valide (ce qui est illustré dans l'exemple précédent). En revanche si l'on croit explicitement $(A \wedge B)$ on croira explicitement A et on croira explicitement B .

Dans le système LÉZARD nous mettons en œuvre une réflexion similaire à celle de Levesque où l'expert complète les connaissances originellement disponibles par des connaissances implicites sur lesquelles il est interrogé. Ainsi, interroger l'expert sur la validité des propositions P6 et P7 de l'exemple précédent revient à lui demander si $[\text{croire } (A \vee B)] \Rightarrow [\text{croire } A] \vee [\text{croire } B]$. Mais à la différence de la logique L qui donne un cadre aux inférences lors de l'exécution du moteur nous situons notre réflexion au niveau de l'interprétation du modèle. Le système sera ainsi déchargé d'autant au moment de l'inférence proprement dite. La mise en œuvre de cette réflexion dans le TRADUCTEUR n'est pas indépendante des autres mécanismes. A l'inverse, la logique L donne un cadre aux mécanismes du TRADUCTEUR — tous ceux dans lesquels des questions doivent être posées pour générer de la connaissance — et n'est pas une fin en soi.

La figure 3.2 décrit de manière synthétique les questions à envisager en fonction des connaissances abordées.

3.2 La compatibilité/incompatibilité entre diagnostics

Dans les systèmes experts de discrimination, et pour des raisons d'efficacité on a souvent été obligé de rajouter des informations sur les diagnostics mutuellement exclusifs. Cela est mieux que de supposer tous les diagnostics exclusifs a priori⁴.

Ainsi dans PIP, chaque maladie est un "frame" qui a une relation permettant de répertorier les diagnostics différentiels, c'est à dire des diagnostics dont la tableau clinique est proche du premier mais qui sont exclusifs avec lui. Les exclusions ne sont donc répertoriées, dans PIP, qu'au niveau d'un diagnostic sur lequel on s'est focalisé. Dans ABEL, des "relations associatives" relient un objet du domaine à ceux dont sa présence influence l'absence ou la présence. De la même manière que PIP ces "relations associatives" décrivent des exclusions par rapport à chaque objet du domaine. Dans ces deux exemples la déclaration des exclusions est délocalisée, reportée sur chaque objet. Dans SAM des relations d'exclusions peuvent être nommément entrées dans les connaissances. Mais la déclaration des exclusions dans un modèle physiopathologique un peu complexe pose rapidement des problèmes de cohérence. Il est difficile de vérifier a priori si une exclusion mutuelle déclarée n'entre pas en conflit avec les interprétations du modèle que les données d'un patient permettent de construire⁵.

⁴Les systèmes bayesiens que nous décrivons au chapitre 5 supposent l'exhaustivité et l'exclusivité des diagnostics. Ce sont des contraintes de ce type qui les ont fait critiquer d'un point de vue conceptuel.

⁵Le mot interprétation est pris ici dans son sens logique. Les PSM de ABEL sont typiques de ces interprétations.

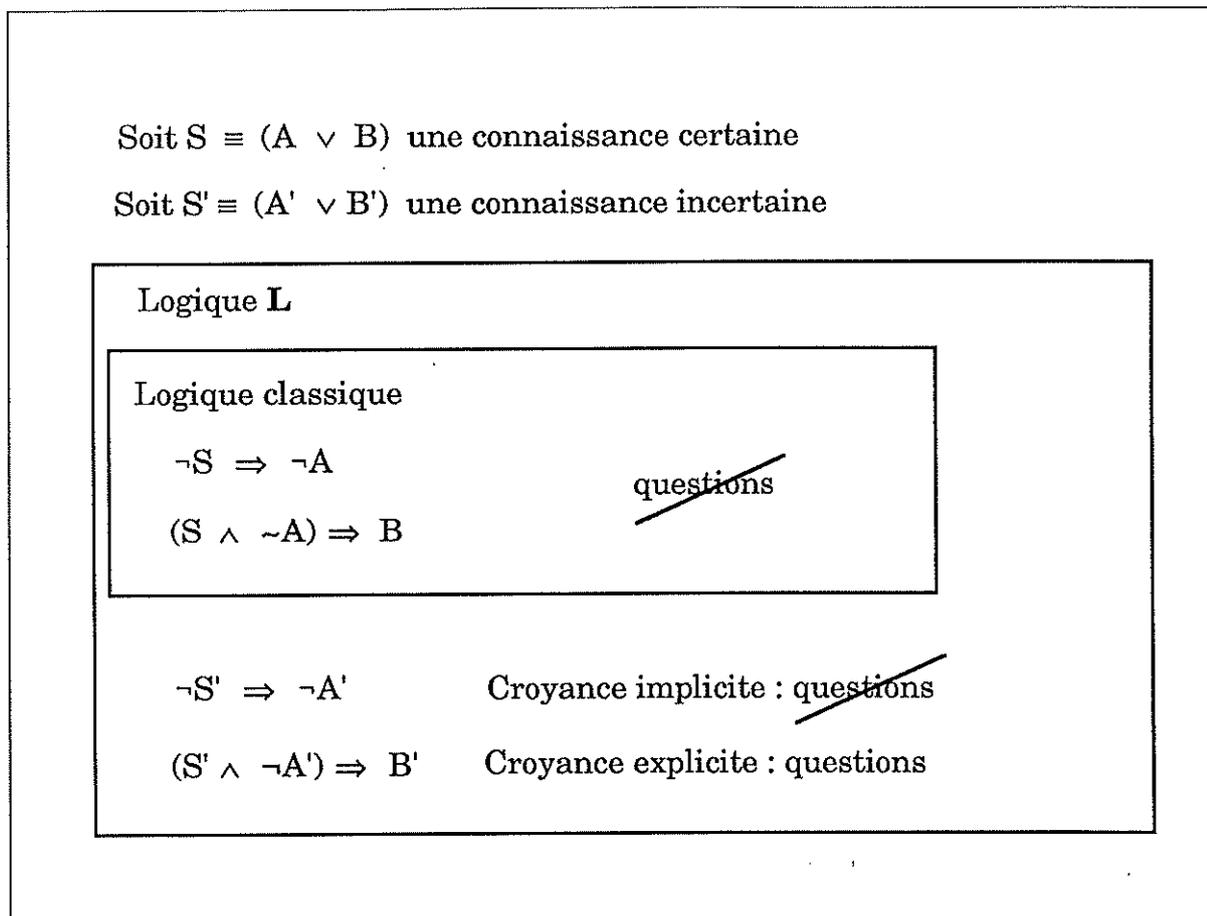


Figure 3.2: Prise en compte des connaissances incertaines dans le cadre de la logique L.

On peut aussi se poser la question de la validité de ces exclusions. Celle-ci est à rechercher dans "l'hypothèse du processus unique".

3.2.1 L'hypothèse du processus unique

Comme ANGINA [Long et al. 82] ou ABEL [Patil et al. 82], le système LÉZARD fait, de façon explicite, ce qu'on peut appeler "l'hypothèse du processus unique". Cette hypothèse est qu'un malade donné peut présenter plusieurs maladies mais que ces pathologies doivent être reliées entre elles par des relations de cause à effet et donc être l'expression d'un processus pathologique unique. Par exemple un malade peut souffrir d'un cancer intestinal qui a provoqué une occlusion organique, laquelle a provoqué une péritonite.

En raffinant la description, il faut noter que le système précise légèrement l'hypothèse du processus unique en tenant compte de la chronicité éventuelle de certaines pathologies. L'hypo-

thèse devient : "un processus unique et éventuellement les manifestations d'une ou plusieurs pathologies chroniques" [Szolovits et al. 88]. Par exemple un malade peut souffrir d'un ulcère chronique superposé à une péritonite indépendante de cet ulcère.

Cette hypothèse est médicalement valide : il s'agit d'une hypothèse que fait un médecin en début de consultation en présence d'un patient. La seule exception à cette hypothèse est la présence d'un patient dans un état général déficient — un vieillard par exemple — qui souffrirait de plusieurs pathologies indépendantes.

3.2.2 Compatibilité et exclusions mutuelles

Sur la base de l'"hypothèse du processus unique" le système interprète le modèle et propose un certain nombre d'*exclusions mutuelles* ou au contraire de *compatibilités logiques* entre diagnostics. Pour ce faire nous avons mis en place un algorithme qui examine le réseau sémantique, et génère plusieurs types de relations toutes cohérentes avec le réseau. Le principe de base de cette génération est : "deux diagnostics s'excluent mutuellement si ils n'ont pas de lien de cause à effet" (les détails de l'algorithme sont donnés en annexe ; cf. annexe I). Les relations générées sont les suivantes :

1. Des relations de *compatibilité logique*. Par exemple :

"APPENDICITE et OCCLUSION sont-compatibles"

qui signifie que rien n'interdit que les diagnostics APPENDICITE et OCCLUSION soient simultanément déclarés *vrais*. Le premier intérêt de ce concept est de permettre de contrôler le modèle déclaré, le second de générer des relations d'*exclusion* en interaction avec l'expert (ce concept et son utilisation sont précisés dans l'annexe I).

2. Des relations d'*exclusions mutuelles*. Par exemple :

"ATTEINTE-DIGESTIVE et ATTEINTE-URINAIRE *sexcluent*" qui signifie

"ATTEINTE-DIGESTIVE \Rightarrow (*non* ATTEINTE-URINAIRE)" et

"ATTEINTE-URINAIRE \Rightarrow (*non* ATTEINTE-DIGESTIVE)".

3. Des relations d'*exclusions mutuelles conditionnelles*. Par exemple :

"ATTEINTE-VASCULAIRE et OCCLUSION-ORGANIQUE *sexcluent-en-l-absence-de* INFARCTUS-MESENTERE" qui signifie

"(ATTEINTE-VASCULAIRE \wedge (*non* INFARCTUS-MESENTERE))

\Rightarrow (*non* OCCLUSION-ORGANIQUE)" et

"(OCCLUSION-ORGANIQUE \wedge (*non* INFARCTUS-MESENTERE))

\Rightarrow (*non* ATTEINTE-VASCULAIRE)".

Ces relations, cohérentes avec le modèle physiopathologique, vont être proposées à l'expert. Elles ne peuvent être déclarées sans autre forme de procès car elles ont un rôle à jouer dans la validation du modèle (cf. annexe I pour plus de précisions). La figure 3.3 donne un extrait du dialogue proposé à l'expert après la génération des relations de *compatibilité logique* sur la base du réseau décrit à la figure 1.2.

Il faut noter que l'ordre de génération des trois types de relations n'est pas anodin. En effet, on montre que les réponses données par l'expert aux relations de *compatibilité logique* générées influent sur la génération des *incompatibilités logiques* (cf. annexe I et plus particulièrement 2.2 page 140 et 2.5 page 146).

Etes-vous d'accord avec le fait que
 APPENDICITE et OCCLUSION sont compatibles (o/n) ? o
 Etes-vous d'accord avec le fait que
 ATTEINTE-VASCULAIRE et OCCLUSION-FONCTIONNELLE sont compatibles (o/n) ? o
 Etes-vous d'accord avec le fait que
 OCCLUSION-ORGANIQUE et OCCLUSION-FONCTIONNELLE sont compatibles (o/n) ? n
Attention. Nier cette compatibilité peut remettre en cause le modèle décrit ou générer une relation d'exclusion entre les 2 diagnostics.
 (1) Vous niez le modèle, (2) vous confirmez l'incompatibilité,
 (3) vous acceptez la compatibilité (1/2/3) ? 2
 OCCLUSION-ORGANIQUE et OCCLUSION-FONCTIONNELLE *sexcluent*.

Voulez-vous une règle d'exclusion du type :
 ATTEINTE-DIGESTIVE et ATTEINTE-URINAIRE *sexcluent* (o/n) ? o
 Voulez-vous une règle d'exclusion du type :
 ATTEINTE-VASCULAIRE et ATTEINTE-URINAIRE *sexcluent* (o/n) ? o
 Voulez-vous une règle d'exclusion du type :
 OCCLUSION et ATTEINTE-URINAIRE-BASSE *sexcluent* (o/n) ? o
 Voulez-vous une règle d'exclusion du type :
 COLIQUE-NEPHRETIQUE et ATTEINTE-URINAIRE-BASSE *sexcluent* (o/n) ? o
 Voulez-vous une règle d'exclusion du type :
 ATTEINTE-VASCULAIRE et APPENDICITE *sexcluent* (o/n) ? n
 Voulez-vous une règle d'exclusion du type :
 OCCLUSION-ORGANIQUE et ATTEINTE-URINAIRE *sexcluent* (o/n) ? o

Figure 3.3: Dialogue entre le système Lézard et l'expert au sujet des relations d'*exclusions mutuelles* et de *compatibilité logique* générées.

3.2.3 Exclusions et négations

A partir du concept d'*exclusion* entre diagnostics et de sa mise en œuvre on peut générer le concept de *négation* des diagnostics par les signes : si un signe évoque un certain nombre de diagnostics et que la liste de ceux-ci est exhaustive, ce même signe nie alors un certain nombre de diagnostics — ceux qui ne sont en aucune manière reliés aux diagnostics évoqués (cf. l'annexe I pour les détails de l'algorithme).

L'intérêt de ce concept (traduit par la primitive *nie*) est qu'il permet une meilleure focalisation du moteur d'inférences sur les bons diagnostics. En effet, la seule présence d'un signe qui évoque plusieurs diagnostics permet déjà d'en éliminer un certain nombre sans avoir besoin de faire le choix entre les diagnostics évoqués. Si nous avons la relation (S évoque A ou B) et les deux *exclusions* (A et C *sexcluent*), (B et C *sexcluent*), alors on peut croire en (A ou B) sans croire isolément chacun d'eux et générer la relation (S nie C). Mais la relation évoque étant entachée d'incertitude l'expert doit être interrogé pour valider la relation générée. L'intérêt de cette relation apparaît si on envisage son absence : dans ce cas, si on ne peut se décider sur A ou B on est obligé de conserver C.

Comme pour les *exclusions*, les *négations* peuvent être *conditionnelles*.

3.2.4 Utilité et discussion

Le "calcul des exclusions" permet de s'affranchir de l'écriture des exclusions et des tests de cohérence qu'il faudrait faire dessus. Les exclusions calculées sont complètes dans la mesure où une autre relation d'exclusion serait redondante⁶ avec celles calculées et obligatoirement cohérentes avec le modèle physiopathologique écrit par l'expert. Si des incohérences existent dans le modèle elles peuvent apparaître dans les exclusions calculées. En revanche rien n'assure que ces exclusions sont utiles. C'est la raison pour laquelle il faut interroger l'expert dans certains cas :

1. Les *compatibilités logiques* permettent à la fois un contrôle du modèle physiopathologique et la génération de nouvelles connaissances. Elles permettent entre autres de générer des exclusions sémantiquement exactes mais que la structure du modèle ne permet pas de générer sans poser de questions à l'expert. Par ailleurs, en cas d'incohérences il peut être nécessaire de modifier le modèle physiopathologique (voir en annexe I, 2.5 page 146 sur les interactions avec l'expert).
2. Les *exclusions mutuelles* n'ont pas besoin d'être sujettes à questions : les essais que nous avons menés ont montré que ces exclusions sont peu nombreuses⁶, exactes⁷ et presque toujours considérées comme utiles par l'expert. Des questions pourraient cependant être

⁶Les *exclusions* sont peu nombreuses et non-redondantes en raison de l'algorithme utilisé pour les calculer : il cherche à créer les *exclusions* au plus haut niveau possible dans les hiérarchies.

utiles pour tester la validité du modèle, mais ce contrôle est en fait réalisé au niveau des *compatibilités logiques*.

3. Les *exclusions mutuelles conditionnelles* sont par contre sujettes à questions. Les connaissances qu'elles génèrent, bien que n'étant pas fausses apparaissent souvent inutiles ou incompréhensibles à l'expert. Des problèmes d'utilité se posent ; elles sont alors susceptibles d'être généralisées (cf. 3.3 ch. 3).
4. Les *négations* simples tirées des relations *evoque* ne sont pas toujours exactes car les relations *evoque* ne sont pas toujours exhaustives (cf. 3.3.1 ch. 2).
5. Les *négations conditionnelles*, quant à elles, ne sont pas toujours utiles (cf. I). Il faut donc interroger l'expert, qu'il s'agisse de *négations* simples ou *conditionnelles*.

3.3 La généralisation des règles

3.3.1 Utilité des connaissances

Une des principales contraintes imposées au TRADUCTEUR est que les règles produites soient effectivement utilisées dans des inférences faites sur des données réelles. Dans un premier prototype du TRADUCTEUR il est apparu que de nombreuses règles inutiles étaient créées. Il est difficile de dire qu'une règle est inutile. On peut simplement constater que, durant une évaluation, certaines règles ne sont utilisées sur aucun des cas testés. Cependant il n'est pas prouvé qu'elles ne serviront jamais. Tout ce que l'on peut dire est que si une règle a trop de prémisses, il est peu probable qu'elle soit utilisée. De plus elle ne sera pas acceptée par les utilisateurs qui la jugeront trop compliquée. Cette remarque rejoint une constatation de Michalski [Michalski 83] qui considère, qu'en apprentissage, il ne faut pas créer de règles ayant trop de prémisses.

3.3.2 Généralisation

Les règles inutiles sont, entre autres, les règles qui ont un grand nombre de prémisses. Une manière de les rendre utiles est de les généraliser, c'est à dire d'étendre leur domaine d'application. Les deux prédicats *faux-ou-inconnu* et *vrai-ou-inconnu* (cf. 3.2.1 ch. 2) permettent cette généralisation. Celle-ci est proposée à l'expert sur un certain nombre de relations (cf. 5 ch. 3). Soit par exemple la règle :

$$\begin{aligned} (\text{ATTEINTE-DIGESTIVE} \wedge (\text{non OCCLUSION}) \wedge (\text{non APPENDICITE})) \\ \Rightarrow \text{ATTEINTE-FOIE} \quad (1) \end{aligned}$$

et les propriétés des prédicats *non*, *faux-ou-inconnu* et *vrai-ou-inconnu* que nous rappelons ici :

⁷Sauf si un diagnostic est *chronique*, comme nous l'avons vu en 3.2.1 ch. 3. Dans ce cas l'expert est interrogé.

(*non* X_1) : est *vrai* si la valeur de vérité de X_1 est *faux*
 (*faux-ou-inconnu* X_1) : est *vrai* si la valeur de vérité de X_1 est *faux* ou *inconnu*
 (*vrai-ou-inconnu* X_1) : est *vrai* si la valeur de vérité de X_1 est *vrai* ou *inconnu*.

La règle 1 peut alors se généraliser par exemple en :

$$\begin{aligned}
 & (\text{ATTEINTE-DIGESTIVE} \wedge (\text{non OCCLUSION}) \wedge (\text{faux-ou-inconnu APPENDICITE})) \\
 & \Rightarrow \text{ATTEINTE-FOIE} \quad (2)
 \end{aligned}$$

On crée de cette manière une règle (2) qui s'applique plus souvent que la règle 1. La règle 1 se déclenche seulement pour ATTEINTE-DIGESTIVE *vrai* et OCCLUSION *faux* et APPENDICITE *faux*. La règle 2 se déclenche pour ATTEINTE-DIGESTIVE *vrai* et OCCLUSION *faux* et APPENDICITE *faux* ou *inconnu*.

Ces deux règles ne sont ni incompatibles ni redondantes. La règle 2 s'applique sur un domaine plus grand et elle est moins contraignante que la règle 1 ; sa *fiabilité* est donc inférieure ou égale à celle de la règle 1. Cette *fiabilité* est donnée par l'expert. Dans le cas où les *fiabilités* des règles 1 et 2 sont identiques, la règle 2 s'appliquant dans un plus grand nombre de cas remplace la règle 1. Dans le cas contraire les deux règles sont conservées.

Il faut noter tout de même que la généralisation n'est pas la panacée et que les règles qui ont trop de prémisses ne doivent pas être générées. En effet, la généralisation ne réduit pas le nombre de prémisses même si elle augmente le champ d'application de la règle ; les problèmes de compréhension ne sont pas levés. On pourrait généraliser en supprimant des prémisses comme cela se fait souvent en apprentissage mais dans notre cas les règles à généraliser sont tirées d'un modèle a priori exact et supprimer des prémisses crée des règles inacceptables par l'expert.

3.4 L'interprétation des liens causaux

L'interprétation des liens causaux est une opération faite systématiquement qui va permettre de générer de nouvelles règles. Le but en est de créer de nouvelles hypothèses. En effet, il y a dualité entre un objet hypothèse et un raisonnement hypothèse : choisir une règle plutôt qu'une autre c'est choisir un raisonnement plutôt qu'un autre et c'est donc choisir les objets sur lesquels s'appuie un raisonnement plutôt que d'autres sur lesquels s'appuie un autre raisonnement.

3.4.1 Transformation de la relation causale en implication

Une interprétation possible des liens causaux est la création de règles d'implication renforçant ces liens. Son principe est : "si je sais que P peut-causer Q alors je peux supposer que P cause toujours Q". Ainsi, à partir de "P peut-causer Q" on peut supposer "P \Rightarrow Q".

Par exemple, si le TRADUCTEUR sait que COLIQUE-NEPHRETIQUE peut causer PERITONITE il proposera à l'expert la règle : "COLIQUE-NEPHRETIQUE \Rightarrow PERITONITE". De façon implicite il demande à l'expert si l'on peut supposer que la COLIQUE-NEPHRETIQUE a toujours pour conséquence PERITONITE.

3.4.2 Abduction

L'abduction est l'un des mécanismes de raisonnement le plus employé en médecine [Fieschi 81]. Il est employé dans le TRADUCTEUR et permet de proposer une autre interprétation logique des relations de causalité. Son principe est : "Si je sais que Q est vrai et que P peut-causer Q alors je peux supposer que P est vrai". L'abduction permet, à partir de "P peut-causer Q", de supposer que "Q \Rightarrow P".

Par exemple, si le TRADUCTEUR sait que APPENDICITE peut causer PERITONITE il proposera à l'expert la règle : "PERITONITE \Rightarrow APPENDICITE". De façon implicite il demande à l'expert si l'on peut supposer que la PERITONITE a presque toujours une cause et que cette cause est précisément l'APPENDICITE.

Ce même principe peut s'appliquer aux relations *constant-dans*. Par exemple, si le TRADUCTEUR sait que INFLAMMATION se manifeste toujours par une TEMPERATURE-ELEVEE il proposera à l'expert la règle : "TEMPERATURE-ELEVEE \Rightarrow INFLAMMATION". De façon implicite il demande à l'expert si l'on peut supposer que la TEMPERATURE-ELEVEE est toujours la manifestation d'une INFLAMMATION.

4 Traduction ; dérivation formelle des relations

Les relations écrites par l'expert — des *connaissances profondes et de surface* — comme les relations générées par le TRADUCTEUR — des *connaissances de surface* — subissent une dérivation formelle qui permet de générer des règles de production.

Pour LÉZARD, une règle de production est une conjonction de prémisses qui implique une conclusion unique (cf. 2 ch. 3). Une prémisses ou la conclusion est un littéral (objet ou négation d'objet). Le moteur utilisera les règles en chaînage avant uniquement, des prémisses vers la conclusion. De plus, la *classe* de la conclusion est toujours inférieure ou égale à la *classe* des prémisses ; et le raisonnement se fait vers les *classes* croissantes. Aucune règle contraire à cette contrainte ne sera générée. Ce qui est une des raisons pour lesquelles des dérivations possibles ne sont pas faites. Ainsi, si S est un *signe* et A un *diagnostic*, $(S \Rightarrow A) \equiv (\neg A \Rightarrow \neg S)$ mais seule la première règle sera générée.

Dans notre système nous utilisons des relations dont la signification doit être précisée (en raison des problèmes d'incertitude notamment). A partir de là on pourra parler de dérivation formelle dans le système LÉZARD et générer les règles de production du système. Il nous faut

donc décrire l'interprétation minimale imposée de chacune des relations utilisées dans le système pour que l'application de la dérivation formelle ait un sens.

Les relations à traduire sont le plus souvent des relations $(n + 1)$ -aires, reliant n fils F_i à 1 père P (cf. figure 3.4).

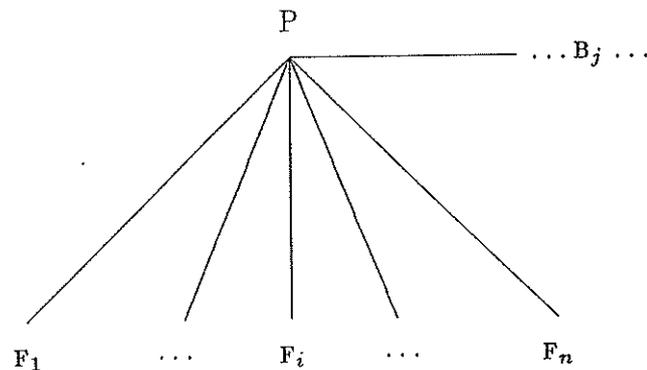


Figure 3.4: relations reliant un ensemble d'objets fils F_i à un objet père P et optionnellement un ensemble d'objets B_j

Les dérivations que nous utiliserons sur ces relations sont :

1) affirmation par les fils :

- $(F_1 \wedge F_2 \wedge \dots \wedge F_n) \Rightarrow P$

2) affirmation par un fils :

- $(F_i) \Rightarrow P \forall i \in [1, n]$

3) élimination par un fils :

- $(\text{non } F_i) \Rightarrow (\text{non } P) \forall i \in [1, n]$

4) élimination par les fils :

- $((\text{non } F_1) \wedge (\text{non } F_2) \wedge \dots \wedge (\text{non } F_n)) \Rightarrow (\text{non } P)$

5) affirmation par défaut :

- $(P \wedge (\text{non } F_1) \wedge \dots \wedge (\text{non } F_{i-1}) \wedge (\text{non } F_{i+1}) \wedge \dots \wedge (\text{non } F_n)) \Rightarrow F_i$
 $\forall i \in [1, n]$

6) élimination par le père :

- $(\text{non } P) \Rightarrow (\text{non } F_i) \forall i \in [1, n]$

7) négation par le père :

- $P \Rightarrow (\text{non } F_i) \forall i \in [1, n]$

8) négation conditionnelle par le père : (P nie F_i si un des B_j est *faux*) :

- $(P \wedge (\text{non } B_j)) \Rightarrow (\text{non } F_i) \forall j \in [1, p], \forall i \in [1, n]$

Deux autres formes de dérivation sont utilisées pour les relations d'exclusion et de négation générées par le "calcul des exclusions" (cf. annexe I). Elles traduisent des relations reliant 1 ou 2 objets, respectivement A_1 ou A_2 , et optionnellement un ensemble d'objets B_j (cf. figure 3.5) :

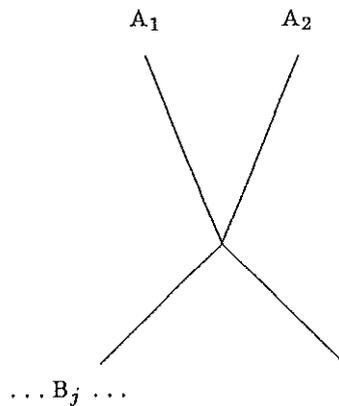


Figure 3.5: relations reliant deux objets A_1 et A_2 et optionnellement un ensemble d'objets B_j

9) exclusion mutuelle (A_1 et A_2 *sexcluent*) :

- $A_1 \Rightarrow (\text{non } A_2)$
- $A_2 \Rightarrow (\text{non } A_1)$

10) exclusion mutuelle conditionnelle (A_1 et A_2 *sexcluent* si un des B_j est *faux*) :

- $(A_1 \wedge (\text{non } B_j)) \Rightarrow (\text{non } A_2) \forall j \in [1, p]$
- $(A_2 \wedge (\text{non } B_j)) \Rightarrow (\text{non } A_1) \forall j \in [1, p]$

4.1 Dérivation de chaque primitive

Nous allons maintenant donner, en les illustrant par des exemples, quelles sont les dérivations retenues pour chaque primitive. Selon les primitives certaines traductions sont sujettes à validation et d'autres pas. Nous verrons pourquoi durant la description de chacune d'entre elles. Décider des traductions possibles, de celles qui seront automatiques ou qui génèreront des questions représente une connaissance permettant d'agir sur les connaissances : c'est une *méta-connaissance*.

loca(process)constituent

POUSSEE-ULCERE et PERFO-ULCERE *processconstituent* ULCERE

- affirmation par un fils (2)
 $POUSSEE-ULCERE \Rightarrow ULCERE$
 $PERFO-ULCERE \Rightarrow ULCERE$
- élimination par les fils (4)
 $((non\ POUSSEE-ULCERE) \wedge (non\ PERFO-ULCERE)) \Rightarrow (non\ ULCERE)$
- affirmation par défaut (5)
 $(ULCERE \wedge (non\ PERFO-ULCERE)) \Rightarrow POUSSEE-ULCERE$
 $(ULCERE \wedge (non\ POUSSEE-ULCERE)) \Rightarrow PERFO-ULCERE$
- élimination par le père (6)
 $(non\ ULCERE) \Rightarrow (non\ PERFO-ULCERE)$
 $(non\ ULCERE) \Rightarrow (non\ POUSSEE-ULCERE)$

Toutes ces dérivations sont automatiques et n'ont pas besoin d'être validées par l'expert. En effet la taxonomie est une connaissance de base du modèle et est donc considérée comme certaine. A l'instar de toutes les primitives décrivant le modèle (où seuls les *diagnostics* sont concernés) les relations sont *incontestables*, les dérivations sont certaines et les règles produites aussi *incontestables*. Cela est vrai pour les primitives *locaconstituent*, *processconstituent* et *peut-causer*.

Pour les primitives non-exhaustives correspondantes (*locaconstituent-n* et *processconstituent-n*) il en est de même si ce n'est que l'affirmation par défaut (5) ne peut être acceptée.

loca(process)constituent-n

PERITONITE et OCCLUSION *processconstituent-n* DOULEUR-SPECIFIQUE

- affirmation par un fils (2)
 $PERITONITE \Rightarrow DOULEUR-SPECIFIQUE$
 $OCCLUSION \Rightarrow DOULEUR-SPECIFIQUE$
- élimination par le père (6)
 $(non\ DOULEUR-SPECIFIQUE) \Rightarrow (non\ OCCLUSION)$
 $(non\ DOULEUR-SPECIFIQUE) \Rightarrow (non\ PERITONITE)$

- pas d'élimination par les fils (4) et pas d'affirmation par défaut (5)

peut-causer

OCC-ORGANIQUE-SE ou OCC-ORGANIQUE-PR ou PERFO-ULCERE ou APPENDICITE ou CHOLECYSTITE ou SIGMOIDITE ou GEU ou FAUSSE-COUCHE ou SALPINGITE ou INFARCTUS-MESENTERE *peut-causer* PERITONITE-SE

- élimination par les fils (4)

$$\begin{aligned} & ((\text{non OCC-ORGANIQUE-SE}) \wedge (\text{non OCC-ORGANIQUE-PR}) \wedge (\text{non PERFO-ULCERE}) \\ & \wedge (\text{non APPENDICITE}) \wedge (\text{non CHOLECYSTITE}) \wedge (\text{non SIGMOIDITE}) \wedge (\text{non GEU}) \\ & \wedge (\text{non FAUSSE-COUCHE}) \wedge (\text{non SALPINGITE}) \wedge (\text{non INFARCTUS-MESENTERE})) \Rightarrow \\ & (\text{non PERITONITE-SE}) \end{aligned}$$
- affirmation par défaut (5)

$$\begin{aligned} & (\text{PERITONITE-SE} \wedge (\text{non OCC-ORGANIQUE-PR}) \wedge (\text{non PERFO-ULCERE}) \wedge (\text{non AP-} \\ & \text{PENDICITE}) \wedge (\text{non CHOLECYSTITE}) \wedge (\text{non SIGMOIDITE}) \wedge (\text{non GEU}) \wedge (\text{non} \\ & \text{FAUSSE-COUCHE}) \wedge (\text{non SALPINGITE}) \wedge (\text{non INFARCTUS-MESENTERE})) \Rightarrow \text{OCC-} \\ & \text{ORGANIQUE-SE}) \dots \end{aligned}$$

peut-causer-n

Il faut rappeler que, comme les primitives précédentes, la primitive *peut-causer-n* a un rôle structurel dans le modèle physiopathologique et est prise en compte dans le "calcul des exclusions".

- Pas de proposition de dérivation.

Sauf pour *evoque*, les dérivations retenues pour les primitives suivantes sont automatiques. Leur *fiabilité* suit les contraintes décrites en 3.3.2 ch. 2.

impliquent

SEXE-MASCULIN *impliquent* non ATTEINTE-GYNECO

- affirmation par les fils (1)

$$\text{SEXE-MASCULIN} \Rightarrow (\text{non ATTEINTE-GYNECO})$$

constant-dans

SIGNES-URINAIRES *constant-dans* ATTEINTE-URINAIRE

- élimination par le pere (6)

$$(\text{non SIGNES-URINAIRES}) \Rightarrow (\text{non ATTEINTE-URINAIRE})$$

signale

CHOL-IV-ANORMALE *signale* CHOL-IV-LITHIASE ou CHOL-IV-VESICULE-NON-VISIBLE

- affirmation par un fils (2)
 $\text{CHOL-IV-LITHIASE} \Rightarrow \text{CHOL-IV-ANORMALE}$
 $\text{CHOL-IV-VESICULE-NON-VISIBLE} \Rightarrow \text{CHOL-IV-ANORMALE}$
- élimination par les fils (4)
 $((\text{non CHOL-IV-LITHIASE}) \wedge (\text{non CHOL-IV-VESICULE-NON-VISIBLE})) \Rightarrow (\text{non CHOL-IV-ANORMALE})$

signale comme *signale-tous* sont des relations *incontestables* (cf. 3.3.2 ch. 2).

signale-all

SYNDROME-AUX-1 *signale-tous* *ARRET-MATIERE* et *ARRET-GAZ* ;

- affirmation par les fils (1)
 $(\text{ARRET-MATIERE} \wedge \text{ARRET-GAZ}) \Rightarrow \text{SYNDROME-AUX-1}$
- élimination par un fils (3)
 $(\text{non ARRET-MATIERE}) \Rightarrow (\text{non SYNDROME-AUX-1})$
 $(\text{non ARRET-GAZ}) \Rightarrow (\text{non SYNDROME-AUX-1})$

evoque

D-PROV-LOC-HYPOC-D *evoque* *ABCES-FOIE* ou *CHOLECYSTITES* *CANCER-INT* ou *OCC-ORGANIQUE-PR* ou *ATT-URINAIRE* ;

- affirmation par défaut (5) soumise à l'appréciation de l'expert

l'affirmation par défaut (5) ne peut être automatiquement acceptée en raison de la non exhaustivité de la plupart des relations *evoque* décrites par l'expert ; elle est donc proposée à l'expert. En cas d'acceptation la *fiabilité* des règles d'affirmation par défaut est demandée en respectant la contrainte suivante : la *fiabilité* de la règle par défaut est \preceq à la *fiabilité* de la règle d'origine.

si n *signes* ou *syndromes* précèdent la primitive *evoque*, la relation se traduit par la décomposition en n relations *evoque*. La mise en œuvre de ce processus sera abordée dans le cadre de la description du processus d'interprétation complet (cf. 5 ch. 3).

evoquent

Se traduit par la création automatique d'un *syndrome* intermédiaire et la décomposition en relations *evoque* et *signale-tous* (le processus est décrit comme précédemment en 5 ch. 3).

sexcluent (simple)

ATTEINTE-DIGESTIVE et *ATTEINTE-URINAIRE* *sexcluent*

- exclusion mutuelle (9)

ATTEINTE-DIGESTIVE \Rightarrow (*non* ATTEINTE-URINAIRE)

ATTEINTE-URINAIRE \Rightarrow (*non* ATTEINTE-DIGESTIVE)

A moins qu'un des *diagnostics* ne corresponde à une maladie *chronique* ces dérivations sont automatiques et de catégorie *incontestables*. Si ce n'est pas le cas ces dérivations doivent être validées par l'expert. En cas d'acceptation, la *fiabilité* est demandée (cf. 3.2.4 ch. 3).

sexcluent (conditionnelle)

ATTEINTE-VASCULAIRE et OCCLUSION-FONCTIONNELLE *sexcluent-en-l-absence-de* PERITONITE ou INFARCTUS-MESENTERE.

- exclusion mutuelle conditionnelle (8)

(ATTEINTE-VASCULAIRE \wedge (*non* PERITONITE)) \Rightarrow (*non* OCCLUSION-FONCTIONNELLE)

(ATTEINTE-VASCULAIRE \wedge (*non* INFARCTUS-MESENTERE)) \Rightarrow (*non* OCCLUSION-FONCTIONNELLE)

(OCCLUSION-FONCTIONNELLE \wedge (*non* PERITONITE)) \Rightarrow (*non* ATTEINTE-VASCULAIRE)

(OCCLUSION-FONCTIONNELLE \wedge (*non* INFARCTUS-MESENTERE)) \Rightarrow (*non* ATTEINTE-VASCULAIRE)

Ces dérivations doivent être validées par l'expert. En cas d'acceptation la *fiabilité* est demandée (cf. 3.2.4 ch. 3).

nie (simple)

DOULEUR-LOCALISATION-EPIGASTRIQUE *nie* ATTEINTE-URINAIRE

- négation par le père (7)

DOULEUR-LOCALISATION-EPIGASTRIQUE \Rightarrow (*non* ATTEINTE-URINAIRE)

Ces dérivations doivent être validées par l'expert. En cas d'acceptation la *fiabilité* est demandée ; comme ces règles mettent en jeu des *signes* ou des *syndromes*, elles ont donc une *fiabilité* qui en dépend suivant les contraintes habituelles (cf. 3.3.2 ch. 2).

nie (conditionnelle)

DOULEUR-LOCALISATION-OMBILICALE *nie* OCCLUSION-FONCTIONNELLE *en-l-absence-de* PERITONITE

- négation conditionnelle par le père (10)

(DOULEUR-LOCALISATION-OMBILICALE \wedge (*non* PERITONITE)) \Rightarrow (*non* OCCLUSION-FONCTIONNELLE)

Comme pour la *négation* simple ces dérivations doivent être validées par l'expert. En cas d'acceptation la *fiabilité* est demandée en respectant les mêmes contraintes.

4.2 Création de nouvelles primitives

Actuellement les primitives semblent suffisantes pour représenter les connaissances nécessaires. La structure modulaire du système autorise cependant la création de nouvelles primitives avec un minimum de programmation : il suffit de rajouter une primitive à celles déjà existantes en spécifiant dans le TRADUCTEUR les dérivations retenues et celles qui seront sujettes à questions.

5 Description d'un processus d'interprétation complet

Conditions d'utilisation

Les mécanismes de génération de connaissances et de dérivation décrits précédemment doivent être mis en place en suivant une procédure précise. Un ordre partiel contraint leur utilisation. Par exemple les connaissances du modèle vont être dérivées et traduites, et il en sera de même des connaissances sémiologiques. Si on modifie le modèle on est obligé de refaire la dérivation des connaissances se rapportant à cette modification y compris, et surtout, parmi les connaissances sémiologiques. On ne décrira pas ici toutes les contraintes qui se posent en cas de modification partielle des connaissances ; nous décrirons simplement l'application des différents mécanismes dans le cas d'un processus de traduction complet à partir des relations décrites par l'expert.

Déroulement du processus

1. Lecture du fichier où sont enregistrées les relations décrivant le modèle et les relations sémiologiques. Dans cette phase :

- (a) Chaque relation contenant la primitive *evoquent* est transformée en deux relations contenant respectivement les primitives *evoque* et *signale-tous* et il y a création d'un nouveau *syndrome* :

S_1 et S_2 *evoquent* A

est transformée en

SYNDROME_AUX_J *evoque* A

SYNDROME_AUX_J *signale-tous* S_1 et S_2

- (b) Chaque relation contenant la primitive *evoque* précédée de n signes est transformée en n relations à 1 signe :

$S_1 S_2$ *evoque* A

est transformée en

S_2 évoque A

S_1 évoque A

2. Tous les objets sont répertoriés et leurs *classe*, *fiabilité* et valeur de vérité *par défaut* sont demandées à l'expert en fonction des contraintes suivantes (et si elles ne sont pas suffisantes) :

- (a) Contraintes sur les *classes* en fonction des relations :

Dans chaque relation *locaconstituent(-n)*, *processconstituent(-n)*, *sexcluent* et *peut-causer(-n)* les objets sont des *diagnostics*.

Dans chaque relation "F signale S_1 ou ... ou S_n ", "F signale-tous S_1 et ... et S_n " F est un *syndrome* et les S_i des *signes*.

Dans chaque relation " B_1 et ... et B_n impliquent A" la *classe* de A est \supseteq à la plus grande des *classes* des B_i .

Dans chaque relation "S constant-dans B_1 et ... et B_n " chaque B_i est un *diagnostic* et S est un *signe* ou un *syndrome*.

Dans chaque relation "S évoque B_1 ou ... ou B_n " chaque B_i est un *diagnostic* et S est un *signe* ou un *syndrome*.

- (b) Dans le tableau (cf. 3.6) sont représentées les *fiabilités* et valeurs de vérité *par défaut* que peut prendre un objet en fonction de sa *classe*.

<i>classe</i>	<i>fiabilité</i>	valeur <i>par défaut</i>
<i>signe</i>	<i>incontestable</i> ou <i>fiable</i> ou <i>peu-fiable</i>	<i>vrai</i> ou <i>faux</i> ou <i>inconnu</i>
<i>syndrome</i>	indéterminée	sans objet
<i>diagnostic</i>	indéterminée	sans objet

Figure 3.6:

De plus les *diagnostics* pouvant représenter des maladies *chroniques* ou *non-chroniques*, l'expert est questionné à ce sujet.

3. La *fiabilité* des relations est demandée à l'expert en respectant des contraintes suivantes :

Une relation ne contenant que des *diagnostics* est *incontestable* (le modèle étant supposé tel).

Si il y a un *syndrome* dans une relation celle-ci ne peut avoir une *fiabilité* \succ à celle des *signes* associés au *syndrome*.

Si il y a un *signe* dans une relation celle-ci ne peut avoir une *fiabilité* \succ à celle du *signe*.

La seule exception est pour les relations *signale* et *signale-tous* qui sont *incontestables*.

4. Le "calcul des exclusions" est alors lancé. La validation et la complétion de ces relations est alors activée en même temps.

Pour les *exclusions* :

- (a) Les relations de *compatibilité logique* sont générées en premier et proposées à l'expert pour contrôle. Un refus de sa part peut remettre en cause le modèle ou permettre la génération d'une nouvelle relation d'*exclusion mutuelle*.
- (b) les relation d'*exclusions mutuelles* ne sont alors générées qu'ici pour tenir compte des relations d'*exclusion* générées à partir des *compatibilités logiques*.
- (c) Les relations de *exclusions mutuelles conditionnelles* ne sont générées qu'ici pour tenir compte des relations d'*exclusion* générées à partir des *compatibilités logiques*. Celles qui sont acceptées contiennent le prédicat *non* dans leurs prémisses ; on peut tenter de les généraliser en introduisant le prédicat *faux-ou-inconnu*. Ces généralisations sont proposées à l'expert et, en cas d'acceptation, la *fiabilité* est demandée en respectant la contrainte suivante : la *fiabilité* de la généralisation est \preceq à la *fiabilité* de la règle d'origine (*incontestable* en l'occurrence). En cas d'égalité, seule la règle généralisée est conservée.

Pour les *négations* :

- (c) Si la *négation* est simple, la relation est proposée à l'expert. Il peut en accepter une partie (*négation* d'une partie seulement des *diagnostics* considérés). En cas d'acceptation de la *négation* complète, sa *fiabilité* est demandée en respectant la contrainte suivante : la *fiabilité* de la *négation* est \preceq à la *fiabilité* du *signe* évocateur ou des *signes* associés au *syndrome* évocateur. En cas d'acceptation de seulement une partie de la *négation*, le système crée une *négation* par *diagnostic* et sa *fiabilité* est demandée en respectant la même contrainte que précédemment.
- (d) si la *négation* est *conditionnelle* le processus est analogue à celui utilisé pour les *négations* simples et ces relations contenant le prédicat *non* dans leur prémisses, elles pourront être généralisées comme dans le cas des *exclusions mutuelles conditionnelles*.

Puis les relations d'*exclusion* et de *négation* générées sont traduites suivant le processus décrit en 4.1.

5. Dérivation des relations de haut niveau suivant le processus décrit en 4.1 ch. 3. Durant cette dérivation de nouvelles règles sont proposées :
- (a) proposition de généralisation des affirmations par défauts générées sur les relations *locaconstituent*, *processconstituent*, *peut-causer* (génération automatique) et *locaconstituent-n*, *processconstituent-n*, *peut-causer-n*, *evoque* (générées après validation de l'expert).
 - (b) proposition de relations *impliquent* tirées des relations *peut-causer* et *peut-causer-n*. En cas d'acceptation, la *fiabilité* est demandée en respectant la contrainte suivante : la *fiabilité* de la règle validée est \leq à la *fiabilité* de la règle d'origine (*incontestable* en l'occurrence).
 - (c) proposition d'abductions tirées des relations *constant-dans*, *peut-causer* et *peut-causer-n*. En cas d'acceptation, la *fiabilité* est demandée en respectant la contrainte suivante : la *fiabilité* de la règle validée est \leq à la *fiabilité* de la règle d'origine (*incontestable* pour les relations *peut-causer(-n)*).
6. Fin. Toutes les connaissances sont ainsi interprétées. La base des connaissances du système est alors exclusivement formée de règles de production.

6 Les méta-connaissances du traducteur

Dans les mécanismes décrits précédemment nous avons plusieurs fois utilisé des *méta-connaissances*. Nous allons essayer de préciser leurs conditions d'utilisation.

Dans le cadre du TRADUCTEUR les *méta-connaissances* représentent les connaissances qui permettent de diriger la génération des règles en essayant de se limiter aux plus pertinentes ; corrélativement elles permettent de diminuer le nombre de questions posées à l'expert.

Les *méta-connaissances* abordées au long du chapitre 2 et de celui-ci font partie des connaissances élaborées par l'expert et nous-mêmes pour permettre au traducteur d'interpréter la base des connaissances écrite dans un langage de haut niveau. Elles sont plus ou moins déclaratives (cf. 2 ch. 2), plus ou moins informelles (la logique L est, elle, totalement formalisée), plus ou moins indépendantes du domaine : certaines sont des *heuristiques* générales — comme la généralisation —, d'autres sont plus particulières à la démarche diagnostique — comme l'hypothèse du processus unique —, d'autres encore sont typiquement médicales — comme l'abduction (même si le mécanisme mis en œuvre dans l'abduction peut être utilisé hors du domaine médical).

Il ne faut pas omettre dans cette énumération le fait que les principes fondamentaux de raisonnement retenus pour le système ne sont pas directement explicités au niveau du TRADUCTEUR mais sont des *heuristiques* qui influent directement la structure des connaissances (cf. 1 ch. 2).

Nous venons de voir que la frontière "dépendantes du domaine-indépendantes du domaine" n'est pas très précise et dépend du but défini pour le système plutôt que des propriétés inhérentes aux *méta-connaissances* [Kulikowski 87]. Mais ce n'est pas un problème. En revanche, là où il y a problème, c'est dans la difficulté qu'il y a à identifier, acquérir et utiliser ces *méta-connaissances* : notre démarche a été très pragmatique et le résultat n'est pas parfait mais il a l'avantage d'être bien adapté à chaque problème qui s'est posé à nous.



Chapitre 4

Le moteur d'inférences

Introduction

Ce chapitre est consacré au moteur d'inférences du système LÉZARD. Il sera découpé en quatre parties.

Dans la première nous décrirons la compilation que subissent les règles de production fournies par le traducteur et les avantages que l'on peut en retirer.

Dans la seconde partie nous nous attacherons à décrire les connaissances de contrôle du moteur d'inférences, les stratégies mises en place pour un meilleur raisonnement.

Dans la troisième partie nous décrirons le fonctionnement du moteur d'inférences, avec ses principes et ses outils.

Nous terminerons par la description des grandeurs caractéristiques de l'exécution du moteur d'inférences, des sorties qu'il propose et des explications qu'il peut donner.

1 La compilation des règles de production

L'idée de base est d'ordonner les règles avant les inférences afin d'éviter au moteur la recherche des règles déclençables à chaque pas de l'exécution.

Dans LÉZARD nous étendons une idée développée par K. Konolige [Konolige 79] dans PROSPECTOR et reprise par O. Gascuel dans SAM [Gascuel 81] pour la compilation des règles en logique 0. Dans le cadre de la logique des prédicats la compilation des règles vise le même but mais pose des problèmes plus ardues abordés par de nombreux auteurs [Ghallab 88, Bouaud 89]. Nous ne les aborderons pas ici.

1.1 Ordonnancer les règles de production

Rappelons que les règles sont de la forme “(conjonction de prémisses) \Rightarrow conclusion” ; une prémisses ou la conclusion étant un littéral (objet ou la négation d'objet).

Pour qu'une règle soit déclenchable il faut que toutes ses prémisses soient vérifiées. C'est donc dans le cas où toutes les inférences permettant d'obtenir ces prémisses ont déjà été faites que la règle a le plus de chances d'être déclenchable et déclenchée à juste titre (puisqu'on ne changera pas d'avis sur une prémisses).

Une relation “naturelle” de précédence entre règles de production est donc une relation \mathcal{P} , telle que :

Définition 1 $R_1 \mathcal{P} R_2$ si la conclusion de R_1 est une prémisses de R_2 . Autrement dit, une règle de conclusion A précède une règle dont A est une prémisses.

Précisons ici que nous ne tenons pas compte d'une éventuelle négation (prédicat *non*) ou d'un prédicat *faux-ou-inconnu*, qui précéderait A dans l'une ou les deux règles. L'idée est qu'il faut obtenir le maximum d'informations sur A avant de l'utiliser. Cette relation est *transitive* et peut être étendue :

Définition 2 Un chemin de règles est une suite de règles (R_1, R_2, \dots, R_n) telle que pour tout $i \in \{1, 2, \dots, n-1\}$, la conclusion de toute règle R_i du chemin soit une prémisses de la règle R_{i+1} . R_1 est l'origine du chemin, R_n est l'extrémité.

Définition 3 Une règle R_1 précède une règle R_n ($R_1 \mathcal{P} R_n$) s'il existe un chemin de règles (R_1, R_2, \dots, R_n) tel que R_1 en soit l'origine et R_n l'extrémité.

L'utilisation de cette relation de précédence pour ranger les règles a priori n'est possible que dans la mesure où :

- D'une part, on sait reconnaître que la conclusion d'une règle est la prémisses d'une autre règle. Cela n'offre aucune difficulté dans LÉZARD dont les règles sont dans le formalisme de la logique des propositions. Ce ne serait pas le cas pour un système autorisant des variables.
- D'autre part, une même règle ne doit pas pouvoir être rangée à la fois avant et après une autre règle : autrement dit la relation doit interdire les boucles de règles.

Cette dernière contrainte ne peut malheureusement être imposée à la base de règles de LÉZARD : la hiérarchie des diagnostics, par exemple, conduit toujours le TRADUCTEUR à générer des règles bouclantes.

Ainsi pour toutes les règles ne faisant pas partie d'une boucle, la relation \mathcal{P} est une *relation d'ordre strict*. Notons que cet *ordre* peut être *partiel* : si deux règles ne sont pas ordonnées l'une par rapport à l'autre, cela signifie que la conclusion de l'une quelconque de ces règles n'a aucune influence sur les prémisses de l'autre ; l'ordre d'exécution de ces deux règles sera donc sans influence sur les résultats. Nous allons maintenant montrer comment ce problème est abordé dans le système LÉZARD après avoir vu les solutions retenus dans d'autres systèmes à règles de production.

Ordonnancement dans Sam et Prospector

Dans SAM et PROSPECTOR des solutions sont proposées pour ordonner les règles :

- L'algorithme développé par K. Konolige pour ranger les règles de PROSPECTOR ne résout pas le problème : les boucles y sont purement et simplement interdites. Nous avons déjà indiqué que ce choix ne pouvait être fait pour LÉZARD.
- Dans le système SAM, système de diagnostic médical, la connaissance est représentée d'une part par des règles de production, et d'autre part par des "tautologies" ou axiomes du modèle physiopathologique. Contrairement à LÉZARD, les tautologies ne sont pas traduites en règles de production et leur interprétation est faite par le moteur, au moment de l'exécution. Les boucles sont interdites dans les règles de production ; seules sont autorisées celles contenues implicitement dans les tautologies. L'ordonnancement dans SAM consiste donc à regrouper en paquets les tautologies concernant les mêmes objets ; à ranger, selon \mathcal{P} , toutes les règles de production ; et à insérer les paquets de tautologies entre les règles de manière à ce que toutes les règles dont les conclusions sont des objets d'un paquet se trouvent avant lui et que toutes les règles dont les prémisses sont des objets d'un paquet se trouvent après lui.

L'ordonnancement dans Lézard

Pour LÉZARD, toute la connaissance est traduite sous forme de règles de production et il n'est pas aussi simple que dans SAM de séparer a priori les boucles du reste des règles. Nous décrivons l'algorithme utilisé en 1.2.

Nous allons définir une boucle en "compréhension" puis nous étendrons cette définition en abordant la boucle à travers ses composantes.

Définition préliminaire 4 Une boucle est un chemin de règles (R_1, R_2, \dots, R_n) tel que $R_1 \equiv R_n$.

Cette définition préliminaire est insuffisante dans la mesure où une boucle peut être formée de

plusieurs chemins mais elle permet de donner une idée de la structure d'une boucle. On peut définir une boucle en "extension" :

Définition 5 Une règle R_i fait partie d'une boucle s'il existe au moins un chemin $(R_1, \dots, R_i, \dots, R_n)$ tel que $R_1 \equiv R_n$. Autrement dit, toutes les règles R_i, R_j d'une boucle sont telles que $R_i \mathcal{P} R_j$ et $R_j \mathcal{P} R_i$.

Il est possible d'étendre la relation \mathcal{P} pour ordonner une boucle B_k et une règle R_i n'en faisant pas partie :

Définition 6 Une règle R_i externe à une boucle B_k , précède cette boucle ($R_i \mathcal{P} B_k$) si la conclusion de la règle fait partie des prémisses d'une des règles (équivalentes de ce point de vue) de la boucle.

Définition 7 Une boucle B_k précède une règle R_i externe à cette boucle ($B_k \mathcal{P} R_i$) si la conclusion d'une des règles (équivalentes de ce point de vue) de la boucle fait partie des prémisses de la règle.

Plus généralement, par transitivité :

Définition 8 Une règle R_1 précède une boucle s'il existe un chemin de règles (R_1, R_2, \dots, R_n) tel que R_1 en soit l'origine et R_n l'extrémité, que R_1 ne fasse partie de la boucle, et que la conclusion de la règle R_n fasse partie des prémisses d'une des règles de la boucle.

Définition 9 Une boucle précède une règle R_n s'il existe un chemin de règles (R_1, R_2, \dots, R_n) tel que R_1 en soit l'origine et R_n l'extrémité, que R_1 ne fasse partie de la boucle, et que la conclusion d'une des règle de la boucle fasse partie des prémisses de la règle R_1 .

Il est clair que ces définitions permettent aussi de classer des boucles les unes par rapport aux autres. On obtient de la même manière qu'entre les règles non bouclantes, un *ordre strict partiel* entre les règles non-bouclantes et les boucles.

La relation de précédence \mathcal{P} peut être utilisée pour ranger les règles d'une même boucle à condition d'arrêter le rangement quand la boucle se referme. Cet ordonnancement n'a pas beaucoup de sens (cf. 2.5 ch. 4 page 82 et chapitre 6 où sont abordés d'éventuels développements) car il n'est pas unique et la première règle de la boucle est aléatoire mais il a l'avantage de proposer un certain ordre. Notons que dans une boucle cette relation, qui est *transitive*, devient en plus *réflexive* et *symétrique* ; ce n'est donc plus une *relation d'ordre*, mais une *relation d'équivalence* et toutes les règles d'une même boucle font partie de la même *classe d'équivalence*.

1.2 La recherche des boucles dans Lézard

La détection des boucles revient à déterminer tous les chemins possibles entre les règles de la base des connaissances en utilisant la seule propriété de \mathcal{P} qui est conservée dans les boucles et en dehors, la *transitivité*. L'idée est de calculer la fermeture transitive du graphe orienté à n sommets défini sur les objets du domaine : les objets du domaine sont les n sommets du graphe et les règles reliant k prémisses à une conclusion définissent chacune k arcs orientés (prémisse $_j$, conclusion avec $j \in \{1, 2, \dots, k\}$), entre chaque prémisse et la conclusion.

L'algorithme de Roy-Warshall-Nolin [Roy 70] permet de calculer la fermeture transitive d'un graphe. Il se déroule en plusieurs étapes :

1. Fabriquer la matrice $M_0(i, j)$ de dimension $n \times n$ des sommets i et j , telle que $M_0(i, j) = 1$ si l'objet i est prémisse d'au moins une règle dont la conclusion est l'objet j (i est relié à j par au moins un arc) ; $M_0(i, j) = 0$ sinon.

Posons quelques définitions :

Définition 1 Un arc orienté (X, Y) est une relation (binaire ici) entre les sommets X et Y . X est l'origine de l'arc, Y en est l'extrémité.

Définition 2 Un chemin est une suite de sommets (X_1, X_2, \dots, X_p) telle que pour tout $i \in \{1, 2, \dots, p-1\}$, il existe un arc (X_i, X_{i+1}) . X_1 est l'origine du chemin, X_p est l'extrémité.

Définition 3 Un chemin (X_1, X_2, \dots, X_p) est un circuit si $X_1 \equiv X_p$.

2. L'algorithme de Roy-Warshall-Nolin permet alors de déterminer la matrice $M_1(i, j)$ de dimension $n \times n$, telle que $M_1(i, j) = 1$ si i est relié à j par au moins un chemin ; $M_1(i, j) = 0$ sinon.

Cet algorithme est très simple et rapide (en $\mathcal{O}(n^3)$). Pour plus de précisions voir [Roy 70].

Il suffit alors d'utiliser le fait que dans les boucles, et seulement dans les boucles, la relation \mathcal{P} est réflexive et symétrique pour retrouver les circuits (les boucles) :

- par réflexivité, si $M_1(i, i) = 1$ alors le sommet i est dans un circuit (une boucle de règles).
- par symétrie, tout j tel que $M_1(i, j) = M_1(j, i) = 1$ est relié à i par un chemin dont il est origine et un chemin dont il est extrémité ; i et j sont alors dans le même circuit (la même boucle de règles).

A partir des objets du domaine impliqués dans une boucle, on trouve les règles formant la boucle : ce sont les règles dont une des prémisses et la conclusion font partie des sus-dits objets.

Finalement la relation \mathcal{P} permet d'ordonner les règles en dehors des boucles, d'ordonner les règles dans les boucles, d'ordonner les règles par rapport aux boucles et d'ordonner les boucles les unes par rapport aux autres. On peut ainsi, avec \mathcal{P} et l'algorithme de Roy-Warshall-Nolin ordonner toutes les règles.

Application et résultats de l'ordonnement

Il n'y a pas de règle allant des *diagnostics* vers les *signes* ou plus généralement d'un objet vers un objet de *classe* inférieure. Les règles vont donc être utilisées dans l'ordre normal du raisonnement : des *signes* vers les *syndromes*, puis vers les *diagnostics*.

L'existence de boucles implique que dans chaque règle d'une boucle il existe une prémisses de *classe* égale à la *classe* de la conclusion. Cette *classe* peut alors être associée à la boucle. Ce qui conduit à distinguer au plus trois types de boucles : une boucle sur les *signes*, où n'interviennent que des *signes* ; une boucle sur les *syndromes*, où interviennent des *syndromes* et parfois des *signes* ; une boucle sur les *diagnostics*, où interviennent des *diagnostics* et parfois des *syndromes* ou des *signes*. On va donc chercher un ordonnancement a priori des règles, comme celui décrit figure 4.1, avec une ou plusieurs boucles de chaque type.

En fait dans l'application au domaine des douleurs aiguës de l'abdomen (et ce sera le cas dans d'autres domaines) on ne trouve en pratique qu'une seule boucle importante sur les *diagnostics*, et placée en dernière position du classement des règles et boucles (cf. figure 4.1).

Cette boucle était attendue au vu des dérivations retenues pour les relations décrivant la base des connaissances (cf. 4.1 ch. 3 et 5 ch. 3). Mais d'un point de vue plus profond cette boucle est la seule qui avait de bonnes raisons d'"exister". En effet, fondamentalement les boucles sont indispensables pour exprimer des rapports symétriques entre objets d'une même *classe*. Or il y avait peu de raisons de décrire de telles relations entre *signes* ou *syndromes*. En revanche, en décrivant des hiérarchies¹ dans la *classe* des *diagnostics*², on obtient des boucles dans cette *classe*.

Nous développerons plus l'analyse de l'ordonnement dans l'application au domaine des douleurs aiguës de l'abdomen au chapitre 5 page 107.

¹Les hiérarchies, comme *locaconstituent*, sont les seules relations à dérivations ascendantes — des fils vers le père — et descendantes — du père vers les fils.

²Il y a des semblants de hiérarchies entre les *syndromes* et les *signes* par l'intermédiaire des primitives *signale* et *signale-tous* mais elles ne servent pas à décrire des rapports symétriques, seulement à définir des intermédiaires de raisonnement. En particulier elles ne se traduisent pas par des règles allant du *syndrome* vers les *signes* — du père vers les fils — mais seulement des *signes* vers le *syndrome*, dans l'ordre normal du raisonnement.

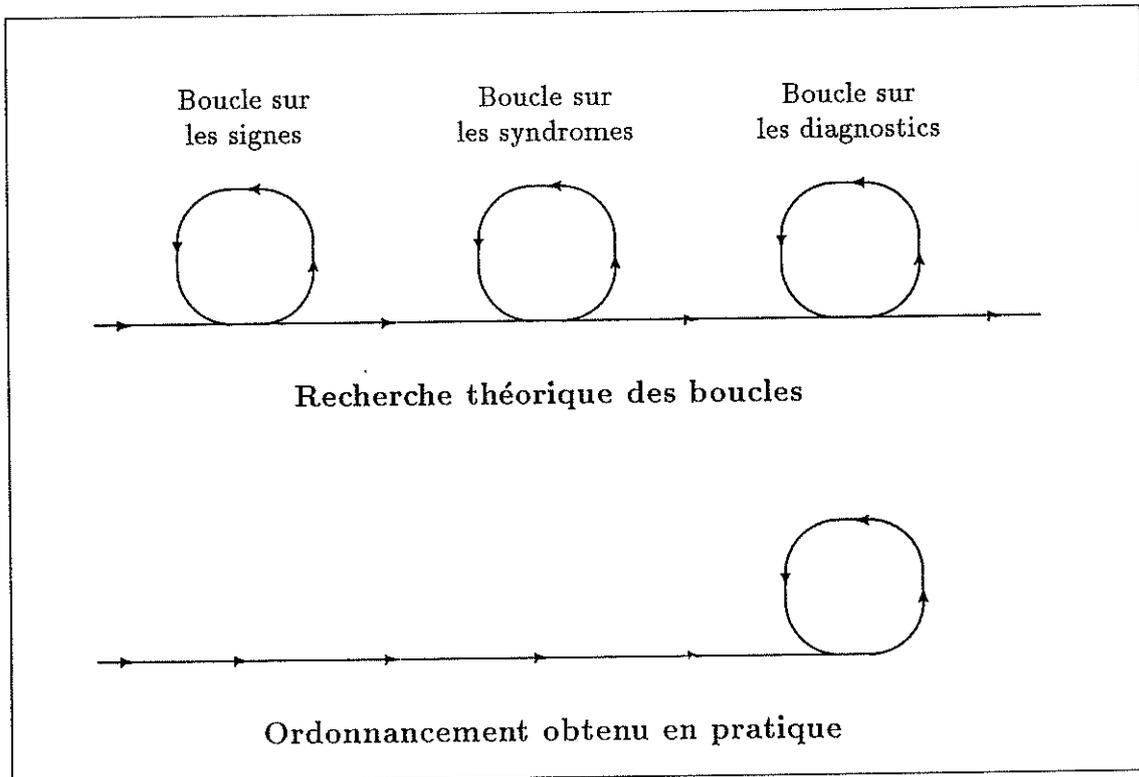


Figure 4.1: L'ordonnement des règles. Théorie et pratique.

Utilité de l'ordonnement

L'ordonnement a pour principale fonction d'accélérer l'exécution du moteur. Pour cela les règles sont classées. Ce classement permet d'éviter le cycle standard d'unification des règles jusqu'à ce qu'aucune ne soit plus applicable. Ainsi le principal intérêt de l'ordonnement est la rapidité d'exécution du moteur qu'il autorise : en dehors des boucles où un traitement spécial est requis (cf. 2.5 ch. 4), les règles ne sont unifiées qu'une seule fois.

L'ordonnement s'avère ainsi nécessaire et suffisant pour une interprétation des règles de production avec parcours exhaustif de la connaissance ; nécessaire dans la mesure où unifier une règle R_1 avant une règle R_2 alors que $R_2 \mathcal{P} R_1$ peut conduire à des résultats incorrects ; suffisant au sens où il permet un parcours optimum de la connaissance dans la mesure où chaque règle n'est unifiée qu'une fois (en dehors des boucles) [Gascuel 81].

On peut donner un exemple de règles où l'ordonnement est nécessaire. Soient les 3 règles :

$$A \Rightarrow Y \quad (1)$$

$$(X \wedge Y) \Rightarrow (\text{non } Z) \quad (2)$$

$$(X \wedge (\text{faux-ou-inconnu } Y)) \Rightarrow Z \quad (3)$$

et les données

A vrai
X vrai
Y inconnu.

Plaçons-nous d'abord dans le cas du système LÉZARD où les règles ne sont unifiées qu'une seule fois.

Si les règles sont ordonnées, elles se présentent dans l'ordre où elles sont écrites ici et "A \Rightarrow Y" sera la première déclenchée, Y sera déclaré *vrai* et Z sera déclaré *faux*.

Si les règles ne sont pas ordonnées "A \Rightarrow Y" peut être la dernière règle examinée ; Y restera alors d'abord *inconnu* et Z sera déclaré *vrai* avant que Y ne soit déclaré finalement *vrai*.

Ainsi dans le premier cas Z est déclaré *faux*, et dans le second cas Z est déclaré *vrai*.

S'il n'y a pas eu ordonnancement des règles on n'est pas dans la même situation que pour LÉZARD.

Il y aura peut être plusieurs balayages de la base des règles comme cela se fait souvent : si les règles se présentent avec "A \Rightarrow Y" en dernière position, le moteur trouvera d'abord "Z *vrai*" ; si le moteur rebalaye ensuite la base de règles, il cherchera à exécuter la seconde règle et il aboutira à une contradiction entre "Z *vrai*" et "Z *faux*". Celle-ci pourra être résolue à l'avantage de l'une ou l'autre des deux propositions ou pas résolue du tout.

En tout état de cause des résultats incorrects sont à craindre.

1.3 Le calcul des propositions et le modus ponens dans Lézard

Le calcul des propositions est un langage formel qui permet de formuler des expressions dans la logique des propositions à partir de propositions et de connectives propositionnelles comme \wedge , \vee , \equiv , ... Il permet aussi de produire de nouvelles propositions, à partir de propositions, et à l'aide de règles d'inférences comme le *modus ponens* ou le *modus tollens* [Nilsson 80, Manna & Waldinger 85].

Le calcul des propositions est donc l'outil utilisé dans LÉZARD pour faire des inférences. Dans LÉZARD les règles de production sont utilisées, en chaîne avant, depuis les prémisses (reliées par la connective \wedge), vers la conclusion en appliquant le *modus ponens*.

En dehors des fonctions décrites précédemment, la compilation des règles a permis la simplification des mécanismes mis en jeu dans le calcul des propositions. En particulier, l'algorithme standard qui consiste à examiner si toutes les prémisses d'une règle sont vérifiées avant d'affirmer la conclusion a pu être modifié pour accélérer l'application du *modus ponens*.

En effet, en associant à chaque objet du domaine les règles dont il est prémisses avec le prédicat dont il est le sujet, on peut incrémenter ou décrémenter un marqueur associé à la règle qui donne le nombre de ses prémisses vérifiées. Une règle est applicable quand toutes ses

prémises sont vérifiées, c'est-à-dire ici quand son marqueur associé est égal à son nombre de prémisses.

L'existence des marqueurs permet ainsi d'accélérer les inférences : à chaque nouveau fait, qu'il soit fourni au départ par les données ou déduit par l'application d'une règle de production, les règles dont il est prémisses voient leurs marqueurs associés automatiquement modifiés et l'application du *modus ponens* ne dépend plus que du test sur l'égalité de deux entiers : Le temps gagné sur le test du *modus ponens* est notable.

Terminons en remarquant que la compilation des règles de production ne modifie en rien la simplicité de leur expression. Ordonner les règles, générer des boucles ne sont que la traduction de méta-connaissances dont le rôle est de simplifier la tâche du moteur d'inférences.

2 Le fonctionnement du moteur

2.1 Préliminaires

Le moteur d'inférences définit le mode de raisonnement du système. Il est très facile de changer de moteur dans la mesure où la syntaxe des connaissances est extrêmement simple et où le moteur d'inférences est totalement indépendant des autres parties du système.

Nous avons donc construit un système qui est très modulaire. Il sépare nettement mode de raisonnement et sémantique des primitives. On peut modifier l'un sans modifier l'autre, ce qui nous a permis d'essayer différents moteurs d'inférences. Nous sommes ainsi partis d'un moteur classique que nous avons amélioré. Environ une dizaine de solutions différentes ont été testées sans difficulté.

2.2 Entrée des données

Nous appelons fait initial toute donnée fournie par l'utilisateur au début de l'exécution du programme.

Les faits initiaux nécessaires au moteur d'inférences peuvent être entrés interactivement par l'expert ou l'utilisateur. Ils peuvent aussi être, et c'est la procédure normale, tirés d'une base de données. Les faits initiaux sont des couples (OBSERVATION, valeur de vérité) ; la valeur de vérité prenant sa valeur dans {*vrai*, *faux*} ; Ce qui correspond au fait que "OBSERVATION *présent*" est *vrai* ou *faux* (cf. 3.2.1 ch. 2). Les observations répertoriées dans le domaine et qui ne sont pas fournies par les données prennent la valeur de vérité *par défaut* qui leur a été attribuée pendant la phase de traduction. Les faits initiaux ne sont pas modifiés pendant toute l'exécution du moteur d'inférences.

2.3 Initialisation

A partir des données, on initialise donc la base des faits du moteur d'inférences. C'est à dire qu'on lui fournit des couples (OBSERVATION, valeur de vérité) correspondant à toutes les observations possibles.

Le moteur peut alors incrémenter le marqueur associé à chaque règle (cf. 1.3 ch. 4). Comme le raisonnement est non additif (cf. 2.4.1 ch. 4), les règles concluant sur les observations dans le même sens ou dans un autre que la valeur de vérité associée à l'observation peuvent être abandonnées et considérées comme déjà appliquées ; ces dernières règles sont a priori inexistantes dans la mesure où l'on a vu que le raisonnement se déroulait des *signes* vers les *diagnostics* et pas l'inverse. Mais il a été choisi de s'assurer de leur inopérance si par hasard il en existait quelques unes entre les *signes* ou les *syndromes* : ainsi il se peut que l'on fournisse un *syndrome* en données ; dans ce cas il faut abandonner (pour ce seul cas évidemment) toutes les règles qui concluent sur ce *syndrome* via les *signes*, quel que soit la valeur de vérité associée à la conclusion.

2.4 Gestion de l'incertain

Comme nous l'avons vu au chapitre 1 l'incertain recouvre deux types de connaissances : (i) les connaissances sémiologiques et des connaissances dérivées de l'interprétation du modèle dont l'incertitude est gérée par des critères de type *quantitatif* et (ii) les hypothèses que l'on peut faire à partir du modèle, via les relations causales et dont la gestion est *qualitative*.

La gestion des hypothèses est implémentée dans le système LÉZARD dans le cadre d'une *logique non-monotone* : un objet peut être déclaré *vrai* puis *faux* au cours du raisonnement.

Nous allons décrire comment est organisée la gestion des hypothèses dans LÉZARD, puis nous verrons comment le système s'appuie dessus pour "faire" du raisonnement incertain, quelles heuristiques il utilise pour choisir les "bons" raisonnements, résoudre les conflits dans le cadre d'un raisonnement *qualitatif*. Un raisonnement *quantitatif* résolvant l'incertitude est implémenté en parallèle pour gérer les connaissances incertaines qui sous-tendent les raisonnements.

2.4.1 Gestion des hypothèses

Le raisonnement du système LÉZARD est principalement *qualitatif*. C'est à dire que les critères déductifs qu'il utilise sont *qualitatifs* et non numériques. Le fondement de ce raisonnement est proche de celui d'un TMS.

Un TMS est l'outil qui permet de maintenir la cohérence de la base des faits d'un système

²La figure 4.2 page 87 est une reproduction de l'écran qui se présente à l'utilisateur quand il demande la liste des observations faites sur un malade.

déductif. Il fait cela au travers des *justifications* : la *justification* d'un fait consiste en d'autres faits et/ou hypothèses. Quand de nouvelles informations provoquent une contradiction, le TMS révisé les hypothèses faites et en informe le système déductif. L'incomplétude des informations est prise en compte dans les hypothèses. La création d'hypothèses par le TMS permet alors au système déductif de travailler dans le cadre d'une logique par défaut ; la révision de ces dernières se fait dans le cadre d'une *logique non monotone*.

Une des caractéristique essentiel d'un TMS (celui de Doyle en particulier [Doyle 79]) est de faire une distinction très nette entre un module déductif, chargé de faire les inférences et le TMS chargé de tester la cohérence des faits et de faire des hypothèses.

Dans le système LÉZARD il n'y a pas de séparation entre un module déductif et un TMS mais la gestion des hypothèses suit le même principe que dans un TMS : le système appuie chaque assertion sur une *justification* associée et, après une déduction, les *justifications* sont modifiées pour en tenir compte. Comme dans un TMS, ces *justifications* permettent de détecter les incohérences des hypothèses effectuées.

A l'inverse du TMS le raisonnement par défaut n'est pas pris en compte au niveau des hypothèses mais au niveau du TRADUCTEUR où les *signes* ont été associés à des valeurs *par défaut* (cf. 3.2.1 ch. 2) et où des règles *par défaut* ont été générées (cf. 4.1 ch. 3).

De plus, dans LÉZARD, Les hypothèses se font sur les règles : le raisonnement supportant une conclusion C est l'arbre *ET* des règles déclenchées qui amènent à C. Il est associé à cette conclusion C et en est la justification. Il est nécessaire de faire des hypothèses quand apparaît un conflit entre deux raisonnements. Les hypothèses que l'on fait portent sur le choix des raisonnements : choisir entre deux hypothèses revient à résoudre un conflit entre deux raisonnements.

Nous allons donc décrire les propriétés et fonctions du système déductif implémenté dans LÉZARD. Nous allons donc voir ce qu'est un raisonnement dans le système LÉZARD et ce que doit permettre de mémoriser le système déductif pour autoriser un bon déroulement de ce raisonnement.

2.4.1.a Un raisonnement

En général, on considère que le raisonnement qui supporte une conclusion C est l'arbre *ET/OU* des règles déclenchées qui amènent à C. Ici on considère une simplification de ce cas général. Une conclusion C est supportée par un unique arbre *ET*, sous-arbre de l'arbre *ET/OU* complet. La limitation qu'entraîne ce choix est que l'on ne tient pas compte de l'effet additif. Lorsque deux règles donnent une même conclusion C on ne retient que l'une des deux. On ne considère pas que la somme de ces deux règles donne plus de certitude sur C que la *meilleure* d'entre elles.

L'intérêt de ce choix est qu'il est simplificateur. Il est implicite dans la plupart des logiques et en particulier dans la logique classique. Il n'est pas présent dans les logiques pondérées utilisées

dans MYCIN [Shortliffe 76] ou SAM [Gascuel 81]. Dans ces systèmes l'additivité n'apporte pas que des avantages ; ils sont en effet particulièrement sensibles à la redondance (une même information peut être comptée deux fois).

En revanche, en négligeant l'additivité on perd la possibilité de renforcer un raisonnement en tenant compte d'un autre, et par la même on perd des possibilités d'explication ; l'explication fournie par le raisonnement retenu n'étant pas forcément meilleure que celle fournie par le raisonnement négligé ou par les deux raisonnements concomitants.

Les raisonnements sont globalement cohérents. Cela signifie qu'à tout moment et dans tous les raisonnements retenus, une valeur de vérité unique est assignée à une proposition, et elle est supportée par un et un seul raisonnement.

Les faits initiaux qui permettent un raisonnement \mathcal{R} (les feuilles de l'arbre ET) seront appelés par la suite la *base* de \mathcal{R} .

2.4.1.b Les justifications du raisonnement

Les *justifications* du système déductif sont ici une liste attachée à un objet contenant trois types d'informations :

1. les *signes* sur lesquels s'appuient le raisonnement. Ces *signes* sont regroupés en sous-listes en fonction de l'arbre ET du raisonnement. Ils sont dans une même sous-liste s'ils sont les prémisses d'une même règle du raisonnement : ce sont les feuilles de l'arbre ET ; ils sont la *base* du raisonnement. Si les *signes* prémisses d'une règle sont inclus dans l'ensemble des *signes* prémisses d'une autre règle ils sont omis. Cela revient à dire que l'on privilégie les règles basées sur un plus grand nombre de *signes*.
2. une liste contenant tous les objets intermédiaires du raisonnement, c'est à dire tous les objets auxquels le raisonnement a affecté une valeur de vérité et qui ne sont pas des faits initiaux.
3. une *fiabilité* calculée comme indiquée en 2.4.2.c ch. 2. Cette *fiabilité*, qui est donc la *fiabilité* de la conclusion de la dernière règle du raisonnement, dépend des objets compris dans tout le raisonnement. Cette *fiabilité* définit aussi la *fiabilité* du raisonnement.

Par exemple, un raisonnement \mathcal{R}_1 bâti sur les règles :

R_1 (*incontestable*) : (VEHICULE \wedge 4-ROUES) \Rightarrow (*non* MOTO)

R_2 (*incontestable*) : (VEHICULE \wedge POIDS > 3500KG) \Rightarrow (*non* VOITURE)

R_2 (*incontestable*) : (VEHICULE \wedge POIDS > 3500KG \wedge NBRE-PASS > 9) \Rightarrow (*non* CAMION)

$$R_4 \text{ (fiable)} : (\text{VEHICULE} \wedge \text{MOTEUR} \wedge (\text{non MOTO}) \wedge (\text{non CAMION}) \wedge (\text{non VOITURE})) \\ \Rightarrow \text{AUTOCAR}$$

permettra d'attacher à l'objet AUTOCAR la liste de *justification* :

$$(((\text{VEHICULE 4-ROUES}) (\text{VEHICULE POIDS} > 3500\text{KG NBRE-PASS} > 9) (\text{MOTEUR})) \\ ((\text{non MOTO}) (\text{non VOITURE}) (\text{non CAMION})) \\ \text{fiable})$$

2.4.2 Critères de choix du meilleur raisonnement

Pour le système, un raisonnement supportant une conclusion *C* est un unique arbre *ET* (cf. 2.4.1.a). Le choix d'un raisonnement se fait donc en sélectionnant dans l'arbre *ET/OU* de tous les raisonnements possibles, celui qui est le *meilleur*.

Quand faut-il faire un choix entre deux raisonnements ? Il faut faire un choix quand il y a un conflit. Quand y a-t-il conflit ? En fait il existe deux cas de figure. Il y a conflit entre deux raisonnements lorsqu'ils sont contradictoires, l'un concluant à *C* et l'autre à (*non C*). Il y a aussi conflit, mais à un degré moindre, lorsque deux raisonnements donnent la même conclusion *C*. Le problème qui se pose alors est de choisir lequel de ces deux raisonnements supportera *C*. Ce dernier problème se pose dans notre cas de déductions où on ne tient pas compte de l'effet additif.

Nous allons donc, ici, étudier quels critères permettent de départager deux raisonnements concurrents, qu'ils soient contradictoires ou pas. Ces critères fondés sur les informations fournies par les *justifications* du système déductif permettent alors la déduction et le choix du *meilleur* raisonnement. Par ailleurs, la plupart des critères utilisés sont *heuristiques*. Ils ont donc demandé une approche pragmatique, un "réglage" pour aboutir à une efficacité satisfaisante, en particulier dans leur ordre d'application. Nous verrons donc comment ils sont utilisées et dans quel ordre, pour résoudre les conflits. Finalement, on notera que ces critères font souvent appel à une notion de spécificité déjà développée dans des travaux de 1985 [Gascuel et al. 85] (cf. 4 ch. 2) mais que cette notion a été ensuite étendue et précisée dans des travaux plus récents comme [Moinard 87]. Dans ces travaux, il est fait état de la notion de spécificité et des principes qu'il faut y rajouter (il en donne cinq) pour l'étendre sans la quitter. Certains d'entre eux sont utilisés dans notre système, d'autres nous sont plus spécifiques (!).

2.4.2.a Utiliser les exceptions

Nous avons vu (cf. 4 ch. 2) que l'on pouvait donner des exceptions dans le raisonnement. Nous allons donner à nouveau les deux façons retenues dans pour donner des exceptions au

ystème et voir ici à quoi ceci correspond du point de vue des justifications et de la spécificité du raisonnement.

Donner une exception avec une règle shunt

le problème qui se pose est : si la règle $R_1 : "A \Rightarrow S_1"$ est généralement vraie et si la règle $R_2 : "S_1 \Rightarrow C"$ est généralement vraie on ne peut dire que la déduction de A à C est généralement vraie. Il faut ici donner au système l'exception avec une règle shunt " $A \Rightarrow (non C)$ ".

Ici, nous choisissons manifestement le raisonnement le plus spécifique.

Donner une exception à l'aide d'une règle contradictoire

L'idée est de nier une règle générale " $s \Rightarrow C$ " en reprenant ses prémisses et en ajoutant une nouvelle prémisses " $(s \wedge s') \Rightarrow (non C)$ " qui spécifie les cas où la règle est fautive. Les effets de l'exception peuvent être directs comme ci-dessus ou indirects. Dans ce cas la suite de règles " $s \Rightarrow S_1 \dots \Rightarrow C$ " est contredite par la suite de règles " $(s \wedge s') \Rightarrow S'_1 \dots \Rightarrow (non C)$ ".

Considérons par exemple les quatre règles suivantes pour voir ce que cela donne au niveau des justifications :

$R_1 : (\text{VEHICULE} \wedge \text{MOTEUR} \wedge \text{4-ROUES}) \Rightarrow \text{VOITURE}$
 $R_2 : (\text{VEHICULE} \wedge \text{MOTEUR} \wedge \text{4-ROUES} \wedge \text{POIDS} > 3500\text{KG}) \Rightarrow \text{CAMION}$
 $R_3 : \text{VOITURE} \Rightarrow (\text{non CAMION})$
 $R_4 : \text{CAMION} \Rightarrow (\text{non VOITURE})$

$S \equiv (\text{VEHICULE} \wedge \text{MOTEUR} \wedge \text{4-ROUES})$
 $S' \equiv \text{POIDS} > 3500\text{KG}$
 C est soit VOITURE soit CAMION
 R_2 est une exception de R_1 .

Ici, nous donnons en fait explicitement une exception à une règle de défaut R_1 . Il n'y a plus de problème de choix de spécificité du raisonnement. L'exception est fournie dans la règle et doit donc être spécifiée clairement. Cette méthode est critiquable car il faut modifier ou créer une règle quand on trouve une exception mais il n'y pas de raison de se priver de cette façon de faire.

2.4.2.b Retenir les raisonnements les plus simples

Retenir les raisonnements les plus simples vis à vis des objets intermédiaires

Considérons le cas de figure suivant : sur la *base* de la situation initiale s le système peut produire un raisonnement \mathcal{R}_1 qui conclut à C . Sur la *base* de la situation initiale $(s \wedge s')$ le système peut produire un raisonnement \mathcal{R}_2 qui conclut à $(\text{non } C)$.

Reportons nous en 2.4.2.a. Si $(s \wedge s')$ permet de déclencher une règle qui amène la conclusion $(\text{non } C)$ alors cette règle est une exception des règles utilisées par \mathcal{R}_1 . Dans ce cas \mathcal{R}_2 est *meilleur* que \mathcal{R}_1 et la conclusion est $(\text{non } C)$. En revanche, $(s \wedge s')$ peut servir à déclencher une règle qui amène la conclusion $(\text{non } C)$ en s'appuyant sur des raisonnements intermédiaires suffisants pour conclure sur C . Dans ce cas \mathcal{R}_2 est un raisonnement parasite du système de règles. \mathcal{R}_1 est *meilleur* que \mathcal{R}_2 et la conclusion est C .

Considérons par exemple les quatre règles :

- $$\begin{aligned} \mathcal{R}_1 &: (\text{VEHICULE} \wedge \text{MOTEUR} \wedge (\text{non MOTO}) \wedge (\text{non CAMION})) \Rightarrow \text{VOITURE} \\ \mathcal{R}_2 &: (\text{VEHICULE} \wedge \text{3-ROUES}) \Rightarrow (\text{non VOITURE}) \\ \mathcal{R}_3 &: (\text{VEHICULE} \wedge \text{3-ROUES}) \Rightarrow (\text{non CAMION}) \\ \mathcal{R}_4 &: (\text{VEHICULE} \wedge \text{3-ROUES}) \Rightarrow (\text{non MOTO}) \end{aligned}$$

Chacune de ces règles est généralement vraie. Si la situation initiale est $(\text{VEHICULE} \wedge \text{MOTEUR} \wedge \text{3-ROUES})$ il se produit un conflit sur VOITURE. En prenant $s \equiv (\text{VEHICULE} \wedge \text{3-ROUES})$ on peut, sur la *base* de s conclure à (non VOITURE) . En prenant $s' \equiv \text{MOTEUR}$, on peut, sur la *base* de $(s \wedge s')$ conclure à VOITURE. La procédure de résolution décrite ci-dessus permet de décider finalement (non voiture) : en effet, il s'agit d'une décision "raisonnable" puisque pour conclure à VOITURE il faut utiliser, entre autres, des arguments qui sont suffisants pour conclure à (non VOITURE) .

Ainsi, dans cet exemple, le raisonnement \mathcal{R}_2 conclut sur $(\text{non } C)$ (ici VOITURE) mais sa *base* est suffisante pour conclure sur C (ici (non VOITURE)), ce que fait le raisonnement \mathcal{R}_1 . Le raisonnement \mathcal{R}_1 qui conclut directement sur C est donc plus simple et doit donc être retenu comme le *meilleur* raisonnement.

Cet exemple ne marche que parce que la première règle possède des exceptions : les tricycles à moteur n'y figurent pas. Il faut toutefois se rappeler que les procédures de résolution sont nécessaires parce que l'on utilise un ensemble de règles qui n'a pas toutes les qualités requises par la logique classique. De plus la procédure de résolution choisie conduit précisément, dans notre exemple, à ne pas tenir compte de la première règle.

Nous sommes ici dans le cadre des raisonnements plus spécifiques mais nous ne nous servons pas de la spécificité comme critère de choix mais d'un principe comme il en est décrits dans [Moinard 87].

Une telle procédure, comme la précédente (cf. 2.4.2.a) n'est efficace que si toutes les exceptions nécessaires au système sont répertoriées. Elles le seront dans les domaines où elles sont assez rares. Si elles sont très fréquentes, c'est à dire si $(S \wedge S')$ a généralement une toute autre signification que S , une telle procédure est de peu d'intérêt.

Retenir les raisonnements les plus simples au niveau des signes

Un raisonnement \mathcal{R}_1 dont la *base* est comprise dans la *base* d'un raisonnement \mathcal{R}_2 est *meilleur* que ce dernier.

Par exemple, regardons le raisonnement \mathcal{R}_1 qui avait conclu à AUTOCAR dans la définition des *justifications* utilisées par le système déductif (cf. 2.4.1.b). Sa liste de *justification* nous donne sa *base* :

$$((\text{VEHICULE 4-ROUES}) (\text{VEHICULE POIDS} > 3500\text{KG NBRE-PASS} > 9) (\text{MOTEUR}))$$

Le raisonnement \mathcal{R}_2 bâti sur les règles R_5 (*incontestable*) : $(\text{VEHICULE} \wedge \text{NBRE-PASS} > 9) \Rightarrow (\text{non MOTO})$

R_6 (*incontestable*) : $(\text{VEHICULE} \wedge \text{NBRE-PASS} > 9) \Rightarrow (\text{non VOITURE})$

R_7 (*incontestable*) : $(\text{VEHICULE} \wedge \text{NBRE-PASS} > 9) \wedge \Rightarrow (\text{non CAMION})$

R_4 (*fiable*) : $(\text{VEHICULE} \wedge \text{MOTEUR} \wedge (\text{non MOTO}) \wedge (\text{non CAMION}) \wedge (\text{non VOITURE}))$

$\Rightarrow \text{AUTOCAR}$

aura la même liste de *justification* que \mathcal{R}_1 sauf que sa *base* sera :

$$((\text{VEHICULE NBRE-PASS} > 9) (\text{MOTEUR}))$$

\mathcal{R}_2 conclut comme \mathcal{R}_1 sur AUTOCAR mais il s'appuie sur moins de *signes* ; il est donc *meilleur* que \mathcal{R}_1 .

Dans ce cas on peut avoir l'impression qu'on privilégie le raisonnement le moins spécifique. Ce n'est pas tout à fait le cas : on privilégie le raisonnement basé sur le moins de *signe* mais il n'y a plus de notion d'exception attachée au *signe*. On ne privilégie donc pas un raisonnement moins spécifique, on privilégie un raisonnement plus simple.

2.4.2.c Choisir le raisonnement le plus fiable

Comme nous l'avons vu en 2.4.1.b les *justifications* incluent une *fabilité*. Cette *fabilité* est calculée selon un algorithme simplifié de propagations de certitude.

Dans LÉZARD, le raisonnement *quantitatif* est une version simplifiée d'un algorithme de propagation de coefficients de certitudes de type MYCIN. Bien que ces coefficients ne soient pas numériques puisqu'il s'agit de *fiabilité* discrète prise dans l'ensemble $\{\textit{incontestable}, \textit{fiable}, \textit{peu-fiable}\}$, leur signification est proche et leur traitement apparenté.

La contrainte que doit vérifier toute formule de propagation des coefficients de certitude est que la certitude de la conclusion d'une règle doit être inférieure ou égale à la plus faible des certitudes des prémisses et de la règle. Elle doit même être égale si tous les prémisses sont *incontestables*, et si la règle l'est. Il en est de même si la règle ou l'une des prémisses est *peu-fiable*. La seule difficulté apparaît lorsque la règle est *fiable* et la conjonction des prémisses aussi. La conclusion peut alors être *fiable* ou *peu-fiable*. Pour choisir entre ces deux possibilités il faudrait faire intervenir des calculs numériques (prenant par exemple en compte le nombre de prémisses *fiables*) ; ce qui revient à dire qu'on se rapprocherait de coefficients véritablement numériques. Or nous verrons que le raisonnement *quantitatif* n'est utilisé qu'en dernier ressort par le moteur et qu'il joue effectivement un rôle assez mineur (cf. 2.4.3). Nous nous sommes donc contentés de la formule de propagation la plus simple possible :

- retenir pour la *fiabilité* de la conclusion d'une règle la *fiabilité* la plus faible d'entre les prémisses et la règle elle-même³.

Selon les *signes* sur lesquels il s'appuie et les règles utilisées, le raisonnement qui conclut sur un objet n'a pas la même *fiabilité*, la *fiabilité* attachée à la conclusion n'est pas la même.

Lorsque l'on compare deux raisonnements d'un point de vue *quantitatif*, le *meilleur* des deux est celui qui permet d'attacher à l'objet sur lequel concluent les deux raisonnements la meilleure *fiabilité*. La meilleure *fiabilité* dépend de l'ordre de précedence défini entre les *fiabilités* : *peu-fiable* < *fiable* < *incontestable*. Cet ordre est logiquement le même que celui retenu pour les objets (cf. 3.2.3 ch. 2).

2.4.3 Résoudre les conflits : choix du meilleur raisonnement

Pour résoudre les conflits il faut pouvoir comparer les raisonnements. Nous avons donné ci-dessus les méthodes de comparaison qu'utilise le système. Nous allons voir maintenant comment et dans quel ordre il les met en œuvre pour résoudre les conflits.

La procédure générale de résolution s'organise comme suit : Soient \mathcal{R}_1 et \mathcal{R}_2 deux raisonnements conflictuels, on applique dans l'ordre⁴ les implications suivantes. Si \mathcal{R}_1 est une exception de \mathcal{R}_2 (cf. 2.4.2.a), \mathcal{R}_1 est *meilleur* que \mathcal{R}_2 . Si \mathcal{R}_1 est plus simple que \mathcal{R}_2 (cf. 2.4.2.b), \mathcal{R}_1

³Nous avons vu en 3.3.2 ch. 2 que certaines règles, celles ayant trait aux observations faites sur le patient, ont une *fiabilité* qui est déjà celle de leur conclusion. Il faut que l'algorithme puisse en tenir compte. Il est évident qu'avec un processus aussi simple ce problème est éludé — il n'en n'est même pas tenu compte — même si ce n'était pas son but.

est *meilleur* que \mathcal{R}_2 . Si \mathcal{R}_1 est plus fiable que \mathcal{R}_2 (cf. 2.4.2.c), \mathcal{R}_1 est *meilleur* que \mathcal{R}_2 . Cette procédure définit un *ordre partiel* sur les raisonnements qui apportent une conclusion (positive ou négative) sur une proposition donnée. Il ne peut pas y avoir de boucle du type : \mathcal{R}_1 est *meilleur* que \mathcal{R}_2 qui est *meilleur* que \mathcal{R}_3 qui est *meilleur* que \mathcal{R}_1 : et ceci, parce que les critères de choix des *meilleurs* raisonnements sont eux-même ordonnés.

Si le système n'arrive pas à résoudre un conflit entre deux raisonnements \mathcal{R}_1 et \mathcal{R}_2 contradictoires il le met en attente. Il se peut que \mathcal{R}_1 ou \mathcal{R}_2 s'avère erroné à la suite d'une rétractation ou qu'apparaisse un raisonnement \mathcal{R}_3 *meilleur* que chacun d'eux. Un conflit entre raisonnements contradictoires peut donc disparaître au bout d'un certain temps. Si ce n'est pas le cas le système le signale à l'utilisateur. En pratique peu de conflits entre raisonnements contradictoires restent non résolus. Dans notre application du système LÉZARD aux douleurs aiguës de l'abdomen 29 des 965 (c'est à dire $\approx 3\%$) cas ont présenté des conflits non résolus (cf. tableau 5.7 en 6 ch. 5).

En revanche, si le système n'arrive pas à choisir entre deux raisonnements faiblement conflictuels, c'est à dire non contradictoire (proposant la même conclusion), celui qui est déjà en place est considéré comme le *meilleur*.

Le nombre de conflits résolus par les quatre premières procédures (exception et ses dérivées et simplicité du raisonnement) est en moyenne trois fois plus important que le nombre de conflits résolus par la quatrième procédure (fiabilité des raisonnements). Ce chiffre est intéressant. Il justifie le caractère de porte de sortie que nous avons attribué au raisonnement *quantitatif*. Il mesure le nombre "d'accidents sémantiques" ou de "raisonnements parasites" que le système évite. Des systèmes du type MYCIN ou SAM utilisent une procédure de résolution des conflits qui est proche de notre dernière procédure. Ils subissent nécessairement ces nombreux "accidents" et leurs performances s'en trouvent inévitablement diminuées.

On peut même remarquer que dans une phase de mise au point du système il est préférable que le processus de résolution quantitative ne soit pas trop efficace : en effet à partir du moment où le but déclaré du système est de résoudre de manière qualitative les conflits, il est inutile qu'un processus numérique conclut sur tous les cas en suspens, les coefficients de certitude fournissant difficilement des explications recevables par l'utilisateur.

2.5 Gestion des boucles

Nous avons vu à quoi correspondaient les boucles en 1 ch. 4 : chacune d'elles est formée d'un ensemble rangé de règles. Nous allons d'abord voir quel mode de propagation de l'information a été retenu dans les boucles avant d'envisager les problèmes théoriques que cela pose et les réponses qui y sont données en pratique.

⁴En fait l'implémentation du module déductif est telle que la recherche du raisonnement d'exception ou du raisonnement le plus simple est imbriquée. En revanche ces recherches ont lieu avant le test sur la fiabilité du raisonnement.

Propagation de l'information dans les boucles

L'algorithme retenu pour les boucles peut s'aborder par le résultat que l'on attend : on ne doit sortir d'une boucle de règles que si plus aucune inférence n'y est possible. On ne doit donc sortir d'une boucle que lorsqu'elle n'apporte plus d'information. En pratique, la sortie se fait dès qu'un parcours complet de la boucle ne s'accompagne d'aucune inférence. L'algorithme semble ainsi très simple et le test de sortie s'apparente à un test standard de fin d'inférences dans un système expert : quand plus aucune inférence n'est possible on s'arrête et on sort. Ce processus est imposé par le fait que la relation \mathcal{P} , appliquée aux règles d'une boucle, est *réflexive* et que rien ne permet donc de garantir que toutes les inférences possibles ont été faites quand les règles ont été balayées une fois.

Il faut noter que l'application de cet algorithme s'accompagne de certaines contraintes : par exemple, une même information ne doit pas se renforcer à chaque passage sur la règle concluant dessus. Ce n'est pas à l'algorithme de prendre cela en compte mais au processeur d'inférences qui doit savoir reconnaître si une règle a déjà été appliquée et ne doit plus l'être à un moment donné.

Le problème est maintenant de savoir si, quand on est sorti de la boucle, on a fait les "bonnes" inférences ; la réponse semble être "oui, s'il n'y a pas de contradictions". Mais la question de savoir si on n'aurait pas pu aboutir à d'autres conclusions peut se poser. D'autre part des contradictions peuvent être obtenues. Dans ces deux cas on peut se demander si un changement de l'ordre d'exécution des règles n'aurait pas permis d'aboutir à d'autres conclusions. Le rangement des règles étant fixé par la relation \mathcal{P} , seul le point de départ des inférences dans la boucle peut modifier l'ordre d'exécution des règles. La question est donc de savoir si on a défini le bon *point d'entrée* de la boucle.

Le point d'entrée

Changer le *point d'entrée* de la boucle revient, en matière d'inférences, à changer l'ordre d'application des règles. On a vu sur un exemple (cf. 1.2) que cela pouvait changer les conclusions possibles ; dans la boucle, où les règles peuvent être balayées plus d'une fois, cela peut créer des conflits : en reprenant les termes de l'exemple on peut être amené à conclure sur "vrai" et "z faux". Des résultats incorrects sont à craindre.

Dans le système LÉZARD un *point d'entrée* est choisi a priori au moment de la recherche de la boucle par l'algorithme de Roy-Warshall-Nolin. La première règle dont une des prémisses et la conclusion font partie des objets impliqués dans la boucle est le *point d'entrée* (cf. 1.2). Ce choix est donc fait de manière arbitraire dans l'état actuel du système.

On pourrait imaginer des procédures de détermination de un ou plusieurs⁵ *points d'entrée* dans la boucle. Le tout est de savoir si cela est vraiment nécessaire vu le caractère arbitraire

du rangement de la boucle.

En pratique, le choix arbitraire du *point d'entrée* n'a pas posé de problèmes. Dans tous les cas étudiés, soit les conflits obtenus ont été résolus, soit leur analyse a montré qu'ils venaient d'autres causes que du choix du *point d'entrée*.

2.6 Dynamique du raisonnement en chaînage avant

La gestion des boucles est le niveau haut ou "top level" du raisonnement. Ensuite il ne reste plus qu'à dérouler la machine, le système déductif résolvant les conflits à l'aide des *justifications*, jusqu'au diagnostic final ou au conflit non levé. L'exécution du moteur d'inférences sur les données se déroule alors comme suit :

1. Initialisation du moteur avec les données (les observations) via un module d'interface. Les données nécessaires au moteur d'inférences peuvent être entrées interactivement ou tirées d'une base de données. A partir des données, on initialise la base des faits du moteur d'inférences.
2. Balayage de la base de règles dirigé par le module de contrôle, et parmi celles qui ne sont pas déjà appliquées, recherche d'une règle applicable.
3. Test de l'existence de la conclusion dans la base des faits. En cas d'absence, la règle est appliquée et le moteur remet à jour les informations qu'il mémorise sur la base des faits (les justifications). Si la conclusion est déjà dans la base des faits, il y a conflit et les *stratégies* de résolution de conflit sont alors activées.
 - (a) si le conflit est levé, la règle est appliquée ou pas selon le cas et les justifications sont remises à jour.
 - (b) si le conflit n'est pas levé, il est mis en attente sauf s'il concerne des raisonnements contradictoires *incontestables*, auquel cas le contrôleur interrompt l'exécution du moteur et le conflit est présenté à l'utilisateur.
4. Test de fin de balayage et retour éventuel à l'étape 2.
5. affichage des résultats

⁵Il peut être intéressant de choisir plusieurs *points d'entrée* dans une boucle : en cas de conflit non levé dans la boucle ou de résultat final considéré comme inacceptable par le contrôle du système, réexaminer la boucle dans un autre ordre peut être une solution envisageable.

3 Connaissances de contrôle

Tout au long de la description du fonctionnement du moteur d'inférences nous avons, sans toujours l'explicitier, signalé l'utilisation d'*heuristiques* ou *stratégies* permettant le contrôle du processus inférentiel.

Nous avons vu en 2 ch. 2 qu'elles ne sont pas toujours déclaratives. Dans l'état actuel du système, même celles qui pourraient être mises facilement sous forme déclaratives ont été codées directement dans le programme. Il est cependant essentiel d'en connaître le contenu pour comprendre le fonctionnement du système. En 2 ch. 2 nous avons donné deux exemples d'*heuristiques* de contrôle. Nous allons les examiner en remarquant qu'en raison de la structure spécifique du système LÉZARD elles ne sont plus explicites au niveau du moteur d'inférences, mais délocalisées car utilisées par le TRADUCTEUR pour générer certaines règles de production. Nous verrons ensuite les contraintes d'application des *stratégies* mises en œuvre pour départager les raisonnements.

3.1 Les heuristiques générales de contrôle du raisonnement

- La recherche d'un processus physiopathologique unique ; cette *heuristique* a été utilisée par le TRADUCTEUR pour la génération des règles d'*exclusion* et de *négation* (cf. 3.2.2 ch. 3). Au niveau du moteur, cette *heuristique* est uniquement représentée par les règles ainsi générées.
- La focalisation sur les niveaux d'abstraction les plus élevés au sein du modèle physiopathologique ; cette *heuristique* n'est évidemment applicable que si des hiérarchies existent dans le modèle physiopathologique, ce qui est le cas ici. Mais comme l'*heuristique* précédente elle a été en partie reprise à son compte par le TRADUCTEUR dans la mesure où il a généré les règles adéquates à partir des hiérarchies. Il faut tout de même tempérer, non l'utilité de cette *heuristique*, mais plutôt son utilisation : en effet les observations faites sur les malades sont les principaux guides du raisonnement et ont souvent tendance à focaliser l'attention sur des *diagnostics* déjà profonds dans la hiérarchie. C'est surtout au moment où des hypothèses doivent être faites qu'il faut essayer de remonter dans les hiérarchies, ce que tentent de faire les règles générées ; ainsi les relations d'*exclusion* générées sous l'influence de l'*heuristique* précédente le sont au plus haut niveau d'abstraction possible (voir annexe I).
- A l'inverse, la description des hiérarchies et leur traduction en règles de production tend à diriger le moteur d'inférences vers les diagnostics à obtenir : les feuilles des hiérarchies. Le système est aidé en cela, comme nous venons de le voir, par les règles portant sur les observations faites sur un patient ; ces dernières évoquent souvent des *diagnostics* déjà profonds dans la hiérarchie si ce n'est terminaux.

Ces *heuristiques* influent sur la déclaration des connaissances — déclaration des taxonomies par exemple — mais elles n'impliquent pas une démarche spécifique d'avertissement de l'utilisateur ; elles sont ou transparente comme la seconde ou explicite — dans le calcul des exclusions — comme la première.

3.2 Les stratégies de choix des raisonnements

La plupart des critères utilisés pour départager les raisonnements sont *heuristiques*. Nous en avons décrit et implémenté quatre. La plupart font appel à la notion de spécificité étendue par des principes. D'autres pourraient être mis en œuvre surtout au niveau des principes d'extension.

On peut les rappeler en se souvenant qu'ils représentent une réalité moins nette dans la mesure où leur mise en œuvre ne permet pas de les séparer :

- Retenir les raisonnements les plus simples ou s'appuyant sur un minimum de *signes*.
- Choisir le raisonnement le plus fiable

Les *stratégies* ainsi mises en place dans le processus déductif sont d'usage transparent dans la mesure où leur utilisation au niveau du moteur n'a pas à être connue de l'utilisateur. En revanche, la manière de donner les exceptions doit être explicitement spécifiée à l'utilisateur pour une plus grande efficacité du système. Les deux *stratégies* suivantes sont donc aussi des contraintes sur la déclaration des connaissances :

- Donner une exception à l'aide d'une règle contradictoire, qui consiste à déclarer explicitement les exceptions dans les règles.
- Donner une exception avec une règle shunt, qui correspond à privilégier le raisonnement le plus spécifique.

4 Exécution, sortie des résultats, traces

Le système fournit un certain nombre d'interface de développement et d'interfaces avec l'utilisateur qui permettent de mieux apprécier son fonctionnement. On en donne ici quelques exemples : la figure 4.2 page 87 montre comment le système présente la liste des observations faites sur un patient ; la figure 4.3 page 88 et les figures 5.3, 5.4 et 5.5 pages 101, 103 et 104 montrent des exemples de diagnostics effectués par le système ; la figure 4.4 page 89 montre un extrait de trace d'exécution des règles. Dans les figures, I, F, P représentent les *fiabilités* attachées aux diagnostics ; NIL correspond à un objet sur lequel aucun raisonnement n'a conclu et qui n'a donc pas de *fiabilité* associée.

» (PRINT-SIGNES 232)

(LB-TUMEUR . *vrai*)

(SEXE-MASCULIN . *vrai*)

(D-PROV-DIFFUSE . *vrai*)

(DEBUT-PROGRESSIF . *vrai*)

(ARRET-MATIERES . *vrai*)

(ARRET-GAZ . *vrai*)

(ANT-CHIR-ABDOMINALE . *vrai*)

(RESP-ABD-NON-VISIBLE . *vrai*)

(METEORISME-DIFFUS . *vrai*)

(D-PROV-LOC-FIG . *vrai*)

(DEFENSE . *vrai*)

(TYMPANISME . *vrai*)

(AEROCOLIE-IMPORTANTE . *vrai*)

(ASP-GRISAILLE-DIFFUSE . *faux*)

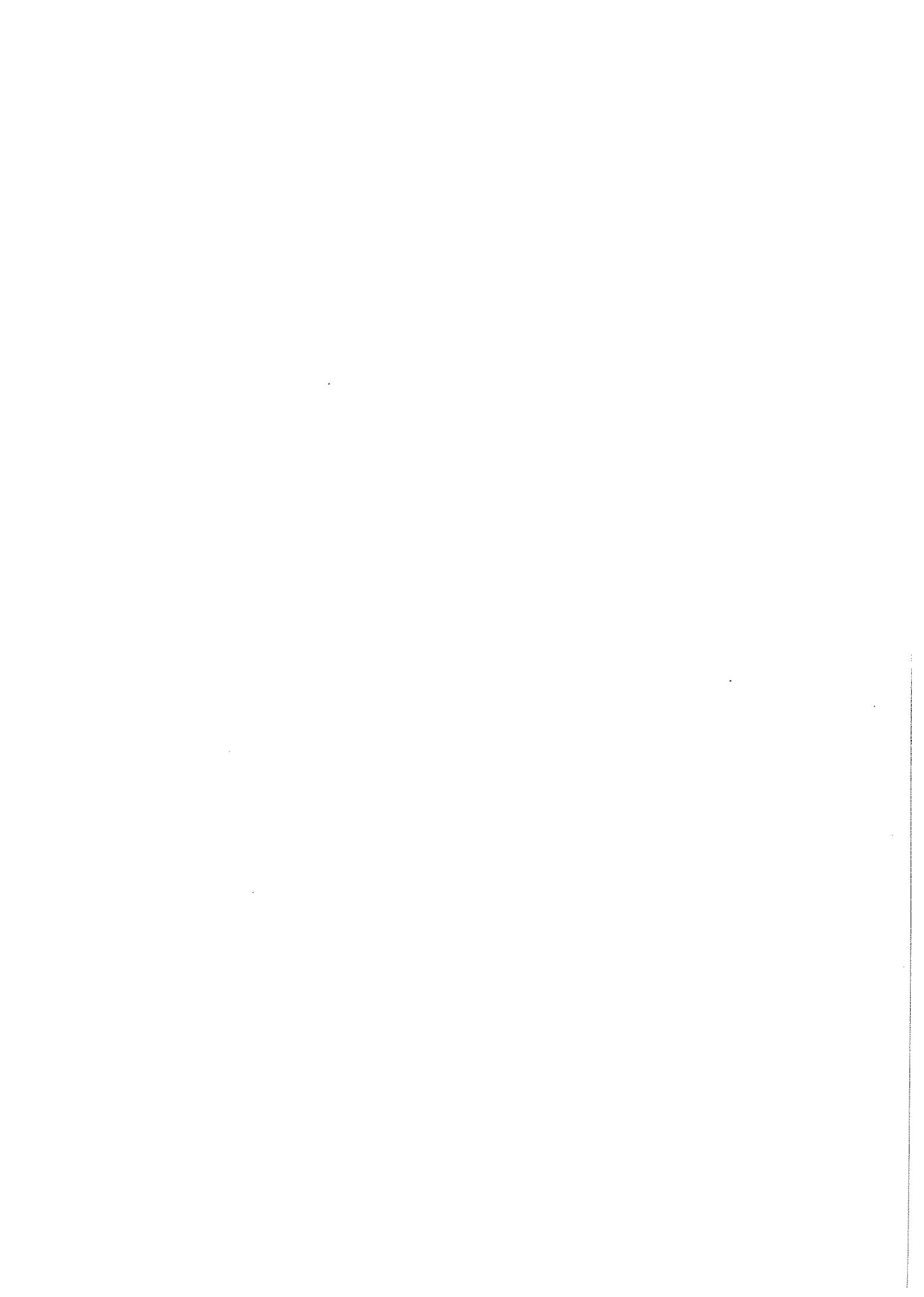
Figure 4.2: La liste des observations faites sur le patient 232.

```
232/965***** DIAGNOSTIC OBTENU *****  
  
CANCER-INT (I) ATT-INTESTIN (I) OCC-ORGANIQUE-SE (P)  
OCC-ORGANIQUE (P) PERITONITE-SE (P) OCCLUSION (P)  
PERITONITE (P) ATT-DIGESTIVE (I) DS (I)  
  
***** NOEUDS INTERMEDIAIRES *****  
  
METEORISME SYNDROME-AUX-1 SIGNES-PERITONITE  
SIGNES-OCCLUSION  
  
**** AUTRES DIAGNOSTICS EVENTUELS ****  
  
OCC-FONCTIONNELLE (NIL)  
  
***** DIAGNOSTIC REEL .... *****  
  
OCC-ORGANIQUE-PR  
  
>>
```

Figure 4.3: Un résultat de l'exécution du moteur (1).

```
=====
J'APPLIQUE R80 (INCONTESTABLE) :
(NON BRULURES-MICTIONNELLES) (F) (NON DYSURIE) (F) (NON POLLAKIURIE) (F)
(NON HEMATURIE) (F) (NON ASP-LITHIASE-AIRE-URINAIRE) (I) (NON CALCUL-UIV) (I)
(NON UIV-DILATATION) (I) (NON UIV-REIN-MUET) (I)
(NON IRRADIATION-DESCENDANTE) (F) (NON IRRADIATION-POST-LOMBAIRE) (P)
      => SIGNES-URINAIRES NON (P)
=====
J'APPLIQUE R203 (INCONTESTABLE) :
(NON ASP-LITHIASE-AIRE-URINAIRE) (I) (NON UIV-ANORMALE) (I)
      => SIGNES-RADIO-C-N NON (I)
=====
J'APPLIQUE R1248 (PEU-FIABLE) :
DEFENSE (P)
      => D-GYNECO-FONC NON (P)
=====
```

Figure 4.4: La trace de quelques règles (1).



Chapitre 5

Evaluation : application

Introduction

Ce cinquième chapitre est la description de l'application du système LÉZARD au domaine des douleurs aiguës de l'abdomen. Il décrit les tenants et les aboutissants de cette application. Une étude comparative critique y est menée avec un système bayésien.

Nous commençons par poser les problèmes rencontrés quand on veut évaluer un système expert et nous proposons des réponses appropriées à notre situation. Nous décrivons ensuite le système bayésien retenu pour être comparé à LÉZARD, et la base des données de l'ARC & AURC, point de départ de cette comparaison.

Nous décrivons ensuite les procédures d'évaluations des deux systèmes en distinguant quatre étapes : la division préliminaire de la base de données initiale en un échantillon d'apprentissage et un échantillon test, la standardisation des réponses, les procédures d'évaluation et enfin une discussion sur la validité des solutions retenues pour la comparaison des performances.

Finalement nous décrivons le système LÉZARD appliqué aux douleurs de l'abdomen et nous discutons les résultats obtenus par les deux systèmes en essayant d'analyser les performances de chacun pour dégager des axes de développement futurs permettant d'accroître les performances de LÉZARD.

1 Evaluation d'un système expert

1.1 Un système pour quelle application ?

Nous nous proposons, dans la définition du projet, de réaliser un système expert permettant de répondre au problème de l'aide à la décision dans le domaine des urgences.

La validation de la base des connaissances et l'évaluation du système concluent un tel travail. Mais il ne s'agit pas de chercher une application à un système expert après l'avoir écrit. Il faut dès le début du projet envisager un domaine d'application.

Nous avons à notre disposition la base de données de l'ARC & AURC. Cette base contient le tableau clinique et le diagnostic de 6732 patients dans le domaine des douleurs aiguës de l'abdomen. Nous avons donc la base des données adéquate pour tester notre système dans le domaine des urgences. Le choix de ce domaine d'application était à partir de là évident.

1.2 Comment valider un système expert ?

Valider les performances d'un prototype de système expert peut se faire sur de nombreuses composantes de ce système :

- les résultats,
- le raisonnement,
- la base des connaissances,
- les heuristiques,
- ...

Nous nous sommes intéressé au plus "parlant", les résultats. C'est de plus le plus accessible et le plus facilement quantifiable. En outre, discuter les résultats et leur origine amène à discuter sur le système en général.

Valider les résultats d'un Système expert c'est les comparer à ceux fournis dans le même domaine par les instruments ou spécialistes habituels. Ici, il était difficile de proposer à des chirurgiens de fournir des conclusions sur des cas de malades qu'ils auraient décrits. Il était encore plus difficile de les faire travailler sur une base de données avec tous les inconvénients que cela peut avoir. Inconvénients traduits par les reproches qu'ils émettent sur le travail qui leur est proposé ou qui transparaissent dans leurs attitudes :

- travail parfois perçu comme une annexion ou une remise en cause de leurs compétences,

- désaccords avec les classifications retenues par ceux qui ont écrit la base,
- critiques sur le concept même du système,
- travail en général long et fastidieux demandant une certaine motivation.

Ces problèmes de validation sont généraux et se présentent souvent quand il s'agit de valider un système expert, et en particulier dans le domaine médical.

La solution retenue a donc été de comparer le système LÉZARD à un système bayésien avec indépendance des signes sur un échantillon représentatif de la base. Le système Bayésien a l'avantage d'avoir des performances comparables à celles d'un expert (et même parfois meilleurs), du point de vue des résultats, tout en étant un système informatique : il est dans la même situation qu'un système expert quand il s'agit d'appréhender les données qui lui sont fournies. Nous ne discuterons pas ici les reproches qui sont faits aux systèmes bayésiens au sujet de leur faibles capacités explicatives et conceptuelles ou de leurs défaillances en cas de diagnostics multiples. Nous nous bornerons à décrire un système bayésien et à l'utiliser pour étalonner les résultats du système LÉZARD.

2 Le système bayésien

Nous allons ici décrire les formules et les hypothèses sur lesquelles s'appuie le système bayésien [De Dombal 76].

Le modèle bayésien

Pour chaque patient, soit $X = (x_1, x_2, \dots, x_p)$ le vecteur des observations de dimension p (ce sont les observations des signes du patient) et $D_i, i = 1, \dots, k$, les k diagnostics considérés. La méthode utilisée consiste à estimer les probabilités a posteriori des diagnostics D_i connaissant les observations $x_j : P(D_i/x_1 \wedge \dots \wedge x_p)$. La règle de décision consiste ensuite à choisir le diagnostic ayant la plus forte probabilité.

Dès que le vecteur des observations est de dimension assez élevée le nombre de probabilités à estimer pour obtenir $P(D_i/X)$ est problématique : prenons par exemple $p = 10$ observations binaires (présent/absent) et $k = 10$ diagnostics. Nous avons 2^p vecteurs X possibles. Et nous avons $k \times 2^p$ probabilités $P(D_i/X)$ à estimer. Cela fait donc, avec les valeurs choisies, $\simeq 10^4$ probabilités à estimer. Avec 30 individus par case il faut finalement un échantillon de $3 \cdot 10^5$ individus.

Pour diminuer la taille de l'échantillon d'apprentissage on utilise le théorème de Bayes et un certain nombre d'hypothèses. Le théorème de Bayes exprime alors que l'on peut estimer la probabilité conditionnelle de la maladie D_i connaissant le vecteur d'observations¹ X par la

formule :

$$P(D_i/X) = \frac{P(X/D_i) \cdot P(D_i)}{P(X)}$$

où $P(D_i)$ est la probabilité a priori pour le diagnostic D_i et $P(X)$ la probabilité a priori du vecteur observations X .

Sous l'hypothèse que les différents diagnostics D_i sont exclusifs et exhaustifs, nous avons

$$P(X) = \sum_{i=1}^k P(X/D_i) \cdot P(D_i)$$

et le calcul de $P(D_i/X)$ ne nécessite plus que l'obtention des $P(X/D_i)$. Malgré ces simplifications, le nombre de probabilités conditionnelles à estimer est toujours le même et d'autres hypothèses sont nécessaires pour diminuer notablement la taille de l'échantillon nécessaire. Usuellement on fait l'hypothèse de l'indépendance conditionnelle des signes².

Le modèle basé sur l'indépendance conditionnelle

Dans le cas où l'on considère que les signes et donc les observations x_j sont indépendants sachant un diagnostic D_i on peut exprimer $P(D_i/X)$ comme étant le produit de l'estimation des probabilités marginales :

$$P(X/D_i) = \prod_{j=1}^p P(x_j/D_i)$$

Pour chaque i et j , $P(x_j/D_i)$ est estimée par la fréquence de la modalité x_j dans l'échantillon de patients souffrant de D_i . En reprenant les nombres de notre exemple nous avons ici besoin d'un tableau de contingence à $2 \times k$ cases pour estimer les $P(x_j/D_i)$ à j fixé. Et avec 30 individus par case cela fait finalement un échantillon de 600 individus³. Sous cette dernière hypothèse on peut enfin estimer $P(D_i/X)$ à partir d'un échantillon de taille raisonnable.

3 La base des données de l'ARC & AURC

3.1 Présentation et caractéristiques

Cette base des données est donc constituée de l'enregistrement des données cliniques de 6732 patients.

¹Le vecteur X doit être complet, c'est à dire que toutes ses composantes doivent être connues.

²Il existe d'autres hypothèses qui permettent de prendre en compte certaines interactions [Séroussi 86], mais nous ne nous y intéresserons pas ici.

³Le même échantillon peut évidemment servir pour l'estimation des probabilités des signes différents de x_j . Il est ainsi suffisant pour estimer $P(D_i/X)$.

La base des données contient environ 20% de données manquantes. De précédents travaux permettent d'envisager que ces omissions sont volontaires et apparaissent, par exemple, quand le médecin, ayant de fortes présomptions pour un diagnostic précis omet certaines étapes logiques de l'investigation. Cela dit, ces omissions ne devraient pas handicaper le système expert si celui-ci a reçu de l'expert les règles de décision qu'il a appliquées dans ces cas particuliers. Dans le cas du système bayésien il n'est pas tenu compte des données manquantes dans les estimations ; ce qui revient à considérer qu'il y a indépendance entre les données manquantes et les diagnostics.

Dans [Séroussi 86] on trouve une description beaucoup plus complète de la base des données avec ses particularités.

3.2 Influence de la base des données sur la base des connaissances

Pour faire une comparaison valable le système LÉZARD doit prendre en compte le même type de diagnostics que le système bayésien et donc que ceux prévus dans la base des données. La représentation des connaissances étant basé sur des taxonomies il était logique d'essayer de faire correspondre les diagnostics terminaux (les feuilles du treillis formé par les taxonomies) avec les diagnostics pris en compte dans la base des données. En particulier il était inutile de représenter un modèle physiopathologique dont les diagnostics terminaux étaient plus précis que les classes décrites dans la base. Nous avons donc essayé d'établir une correspondance avec les 22 diagnostics de la base des données :

APPENDICITE
DOULEUR-NON-SPECIFIQUE
CHOLECYSTITE
OCCLUSION
PERFORATION-ULCERE
HERNIE-ETRANGLEE
PANCREATITE
COLIQUE-NEPHRETIQUE
POUSSEE-ULCERE
SALPINGITE
INFECTION-URINAIRE
KYTE-OVAIRE
SIGMOIDITE
INFARCTUS-MESENTERE
DOULEUR-GYNECO-FONCTIONNELLE
GROSSESSE-EXTRA-UTERINE
ANEVRYSME-DE-L'AORTE
CANCER-SANS-OCCLUSION
PERITONITE-PRIMITIVE
AFFECTION-PULMONAIRE

ABCES FOIE
FAUSSE COUCHE

Malgré cela il n'a pas été possible de retenir les diagnostics correspondants pour LÉZARD. Ainsi dans la base des données on constate que sont répertoriés les classes CANCER-SANS-OCCLUSION et OCCLUSION. Le cancer de l'intestin ayant provoqué une occlusion étant rangé dans la classe OCCLUSION. Ce classement a été retenu pour des raisons de simplification : les classes étant exclusives, chaque patient n'est répertorié que dans une seule d'entre elles. Cela ne pose aucun problème au système bayésien ; au contraire la base est parfaitement adaptée à son mode de discrimination. En revanche, LÉZARD étant un système expert, il doit avoir une description du domaine la plus proche possible de la réalité pour tendre vers une meilleure efficacité ; on y a donc suivi un classement des pathologies plus proche du modèle physiopathologique réel et on a répertorié les diagnostics CANCER-INTESTIN, OCCLUSION, OCCLUSION-ORGANIQUE-PRIMAIRE et OCCLUSION-ORGANIQUE-SECONDAIRE. Un malade sur lequel on fait le diagnostic CANCER-INTESTIN sera considéré comme atteint d'un cancer de l'intestin. Une occlusion y sera associée ou non selon que l'on aura conclut sur OCCLUSION positivement ou négativement.

En raison des problèmes de recouvrement des diagnostics Il a donc fallu adapter les procédures de comparaison entre le système bayésien et le système expert. Mais le problème de dénomination et représentation des diagnostics faussera toujours la comparaison. Nous l'aborderons quand nous examinerons les résultats de la comparaison en 6.

Pour les besoins de la comparaison il a été retenu un échantillon test de 965 patients avec un vecteur de 91 signes pris en compte par le système expert LÉZARD. Il y en avait 147 de disponibles mais l'expert n'en n'a relevé que 91 vraiment significatifs pour le système expert. Ceux qui ont été omis l'ont été en raison de leur manque de fiabilité ou de leur propension à être mal remplis. Certains signes comme les items valués (age, température...) ont subit des transformation destinées à les rendre exploitables dans le cadre d'une logique des propositions (cf. 3.2.1 ch. 2).

Les diagnostics possibles sont donc 22 classes de diagnostics exclusifs dans la base des données et 22 diagnostics terminaux presque exclusifs dans la base des connaissances du système expert⁴.

On peut voir sur les figures 5.1 et 5.2 pages 98 et 99 la représentation du réseau sémantique du domaine. La traduction de ce réseau dans le langage de représentation des connaissances du système LÉZARD est fournie dans l'annexe II.

⁴En raison du calcul des exclusions effectué sur le réseau sémantique des diagnostics, on ne peut garantir l'exclusivité des 22 diagnostics terminaux (cf. 3.2.2 et annexe I). Cependant la structure du réseau est telle que les diagnostics sont en pratique, la plupart du temps, exclusifs : cette exclusivité n'est pas déclarée mais elle est contenue dans les règles de production du système. Pour des données sur un patient manquant de signes *fiabiles* il est possible que l'exclusivité ne soit pas totalement effective. Ceci étant cela n'a aucune incidence sur la procédure de comparaison et les diagnostics peuvent raisonnablement être considérés comme exclusifs.

4 Les procédures d'évaluation

La première condition pour l'évaluation d'un système expert est d'avoir un échantillon suffisamment grand et représentatif des cas sur lesquels on veut faire fonctionner le système [Weiss & Kulikowski 84]. Ce qui est notre cas. Il n'empêche que l'évaluation des systèmes experts n'a pas encore de méthodologie entièrement satisfaisante. La raison essentielle de cet état de fait est la complexité de la forme de leurs résultats, qui consistent non seulement en une ou plusieurs conclusions, éventuellement pondérées, mais en toute rigueur incluent aussi le cheminement (le "raisonnement") qui a été utilisé. En effet, un avantage potentiel des systèmes experts est le raisonnement et la clarté des explications fournies par ceux-ci, à l'opposé des algorithmes numériques. Mais la qualité d'un raisonnement et sa clarté sont des critères subjectifs, dépendant de l'utilisateur. Ils sont difficiles à évaluer : il n'y a pas de procédure formelle permettant de vérifier qu'une conclusion et que le raisonnement suivis sont les meilleurs qui soient. Comparer deux systèmes est, de la même manière subjectif et difficile à automatiser [Gaschnig et al. 83].

Une solution est de présenter à un groupe d'experts un échantillon de cas et de leur demander de formuler leurs réponses. Dans un second temps, les réponses du système expert leur sont présentées, et ils notent de façon subjective la qualité de celles-ci. On compare alors les réponses des experts et celles du système pour apprécier ses performances. L'inconvénient de cette méthode est son prix : il est difficile d'obtenir d'un groupe d'expert un tel travail sur un grand nombre de cas, aussi cette méthode est habituellement utilisée avec un nombre de cas trop faible pour obtenir un résultat précis : l'effectif est en général inférieur à une centaine. Son avantage est qu'elle répond bien au problème posé qui est d'évaluer le système expert dans ses deux caractères essentiels, les performances et la clarté des explications. De plus, et parce qu'elle limitée dans le nombre de cas pris en compte, elle est bien adaptée à l'évaluation des performances du système sur les cas rares, alors que les comparaisons numériques ne s'intéressent qu'aux cas les plus nombreux, les plus simples.

Cette méthode n'a pas été utilisée ici, en partie par indisponibilité des experts. Nous avons donc pris notre parti d'évaluer les différentes méthodes sur un plus grand nombre de cas, ce qui permet tout de même de mieux apprécier la qualité globale de leurs performances. L'inconvénient de cette approche est que nous ne pouvons comparer les méthodes utilisées aux experts humains, mais seulement entre elles. En revanche, on peut essayer d'éviter une certaine dose de subjectivité dans la comparaison des performances des différents systèmes. Nous allons maintenant décrire par le menu la mise en œuvre de notre procédure de comparaison des performances basée sur la qualité des résultats.

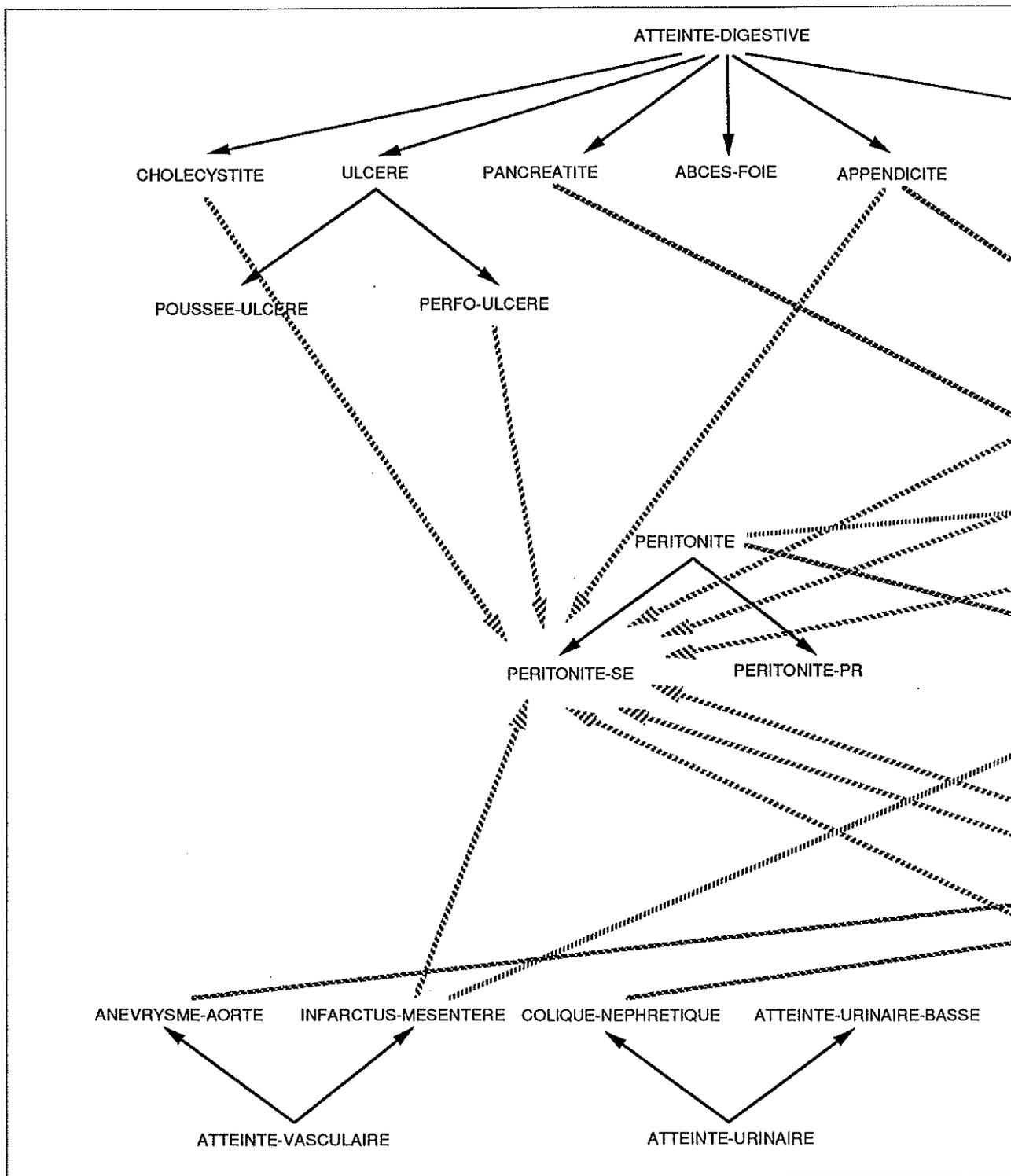


Figure 5.1: Représentation du réseau sémantique. les flèches épaisses interrompues représentent les liens causaux. Les autres flèches les liens taxonomiques.

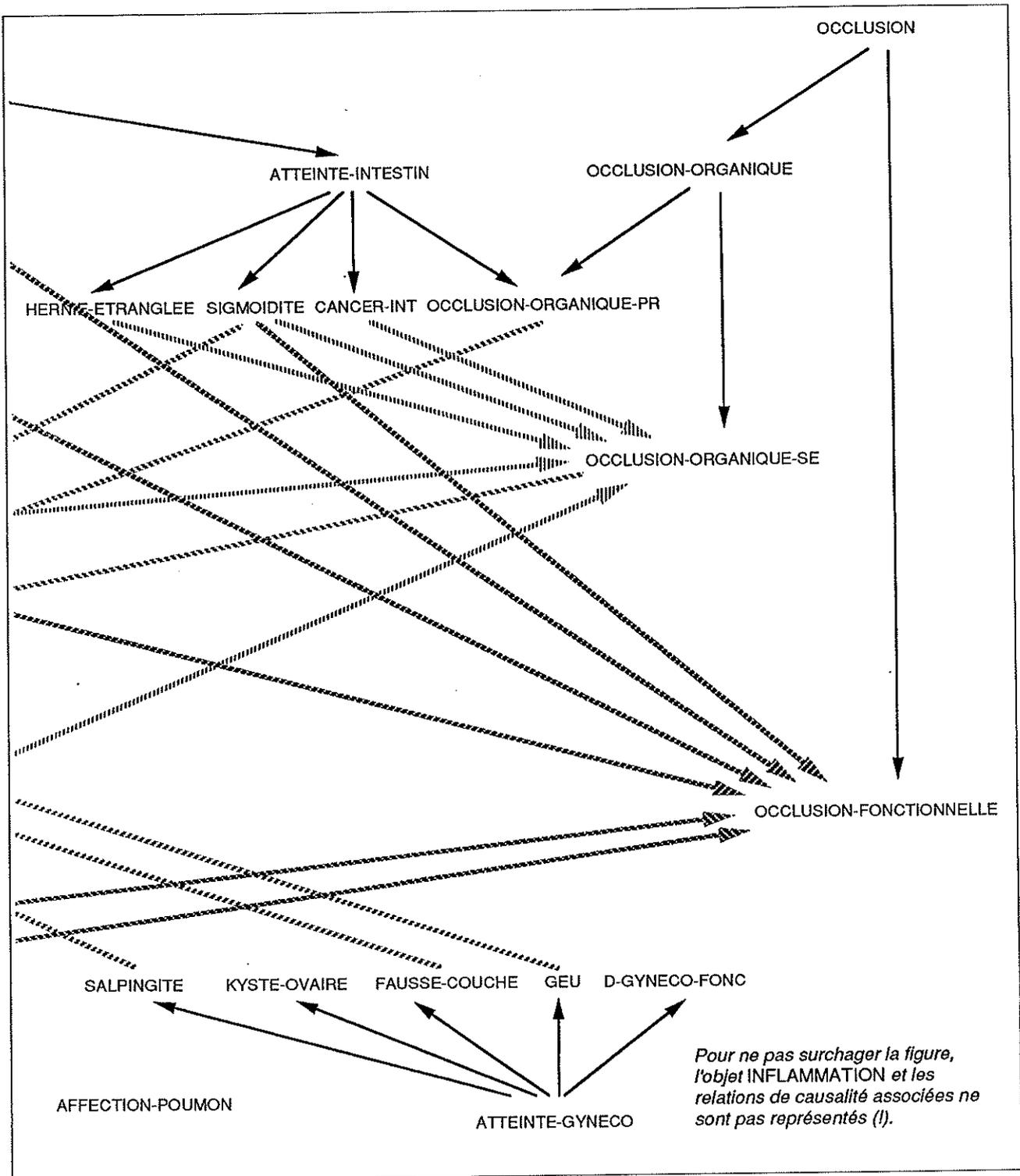


Figure 5.2: Représentation du réseau sémantique, suite.

4.1 Constitution d'un échantillon

La base de données a été divisée en deux parties, la première servant d'échantillon d'apprentissage (5767 patients), et la deuxième d'échantillon test (965 patients).

Pour l'approche bayésienne la totalité de l'échantillon d'apprentissage a servi à créer le système de décision, alors que pour le système LÉZARD la mise au point de la base de connaissances a été effectuée à partir d'un sous-échantillon de 94 cas, tiré de l'échantillon d'apprentissage. Ces 94 patients ont été tirés de l'échantillon d'apprentissage du système bayésien par l'algorithme des nuées dynamiques [Golmard 87] : ces patients sont les plus proches des centres de gravités des classes isolées par l'algorithme. On fabrique ainsi un échantillon où les patients sont représentatifs de toutes les classes présentes dans l'échantillon d'origine.

Pour ce qui est de l'échantillon test, ce sont donc les 965 patients qui ont été retenus pour les deux systèmes.

4.2 Standardisation des réponses

Un des problèmes majeurs de ce travail vient de la différence de la forme des résultats des deux méthodes. Aussi il a été nécessaire de standardiser les types de réponses possibles pour les deux méthodes. Cette standardisation est subjective, inhabituelle, mais indispensable dans le cadre d'une évaluation comparative. Pour les deux systèmes, nous avons divisés la forme des réponses possibles en deux classes : une réponse précise (un diagnostic), plusieurs réponses, ou pas de réponse. La façon dont cette typologie s'effectue est différente selon le système considéré.

4.2.1 Standardisation des réponses du système Lézard

Les résultats du système expert ont une forme très éloignée de celle de la méthode bayésienne puisque LÉZARD n'utilise aucun nombre. Ils se présentent sous la forme de trois listes : une liste des conclusions retenues (diagnostics déclarés *vrais*), une liste des syndromes retenus et une liste des conclusions possibles (déclarées *inconnues*). La seconde liste est intéressante au niveau du raisonnement, pas au niveau d'une évaluation des réponses basée sur les résultats. La dernière liste a un rôle plus ou moins important selon que la première liste est vide de diagnostics terminaux ou pas. Si il y a des diagnostics terminaux dans les conclusions retenues par le système alors la dernière liste n'a que peu d'intérêt par rapport à la première ; elle représente les derniers diagnostics sur lesquels le système n'a rien conclu. Si la première liste est vide de diagnostics terminaux c'est que le système a tout éliminé sauf quelques diagnostics sur lesquels on n'a peu d'informations ; ce sont alors les seules conclusions possibles que retient le système.

Les conclusions peuvent être un des 22 diagnostics précis ou un terme plus général (diagnostic de haut niveau), apparaissant uniquement dans ce système en raison des taxonomies. La contrainte d'exclusivité des diagnostics étant pratiquement effective (cf. 3.2), il ne figure

généralement au plus qu'un diagnostic précis dans la liste des conclusions retenues. En fait il peut n'y avoir aucun diagnostic précis retenu. Dans ce dernier cas il peut y avoir un ou plusieurs diagnostics précis possibles.

Ont donc été considérées comme réponses précises celles comportant un diagnostic précis retenu (cf. figure 4.3 page 88) ou un diagnostic précis possible (cf. figure 5.3 page 101). Les réponses imprécises ou hésitantes sont formées de celles où un terme diagnostic de haut niveau⁵ figure parmi les conclusions retenues avec plusieurs diagnostics terminaux parmi les conclusions possibles (cf. figure 5.4 page 103). De plus, il peut y avoir des conflits sur la valeur de vérité des diagnostics ; conflits qui n'ont pu être levés au cours du raisonnement (cf. figure 5.5 page 104).

```

95/965 ***** DIAGNOSTIC OBTENU *****

INFLAMMATION (I)

***** NOEUDS INTERMEDIAIRES *****

AGE<30

**** AUTRES DIAGNOSTICS EVENTUELS ****

DS (NIL) ATT-DIGESTIVE (NIL) APPENDICITE (NIL)

***** DIAGNOSTIC REEL .... *****

APPENDICITE

>>

```

Figure 5.3: Un résultat de l'exécution du moteur (2).

La classification des malades se fait finalement en essayant de satisfaire l'un des critères suivants pris dans cet ordre :

Critère 1 *Il y a conflit sur un diagnostic*

⁵Ce terme est fonction de la hiérarchie des taxonomies. Dans notre application, par exemple, ATT-DIGESTIVE, ATT-GYNECO, ATT-VASCULAIRE et ATT-URINAIRE sont des diagnostics de haut niveau.

- ce diagnostic est le diagnostic exact alors le système est en conflit dans le vrai (CV)
- ce diagnostic n'est pas le diagnostic exact alors passer aux critères suivants en mémorisant que le système est en conflit (apparition de "C" devant la caractéristique du diagnostic effectué)

Critère 2 *Le système a retenu un seul diagnostic terminal alors c'est son choix et il est vrai ou faux (PV ou PF)*

Critère 3 *Le système a retenu plus d'un diagnostic terminal*

- le bon diagnostic est parmi eux ; alors le système est hésitant mais dans le vrai (HV)
- le bon diagnostic n'est pas parmi eux ; alors le système est hésitant mais dans le faux (HF)

Critère 4 *Aucun diagnostic terminal n'a été retenu mais un seul diagnostic terminal reste possible ; alors c'est le bon choix et il est vrai ou faux (PV ou PF)*

Critère 5 *Aucun diagnostic terminal n'a été retenu et n diagnostics terminaux sont possibles :*

- Si $n > 5$ alors le système est en situation de non-réponse (NR)
- Si $n \leq 5$ et que le bon diagnostic est parmi eux alors le système est hésitant mais dans le vrai (HV)
- Si $n \leq 5$ mais que le bon diagnostic n'est pas parmi eux alors le système est hésitant mais dans le faux (HF)

Critère 6 *Aucun diagnostic terminal n'est ou retenu ou possible ; alors le système considère qu'il n'y a rien de grave et propose une DOULEUR-NON-SPÉCIFIQUE et ce choix est vrai ou faux (PV ou PF)*

4.2.2 Standardisation des réponses du système bayésien

Le système bayésien supposant l'exhaustivité et l'exclusivité des diagnostics il semblait facile de savoir si le résultat était faux ou pas : le système fournissait une liste de diagnostics par ordre de probabilité décroissante. En fait il fallait adapter les critères d'évaluation pour qu'ils soient du même type que ceux de LÉZARD sachant que : (i) il n'y a pas de conflits dans le système bayésien (ii) les critères retenus ne pouvaient être que pragmatique et donc subjectifs. Une optimisation la plus objective possible s'imposait alors. La classification des malades se fait alors comme suit :

```

7/965 ***** DIAGNOSTIC OBTENU *****

PERITONITE-SE (P) INFLAMMATION (I) PERITONITE (P)
ATT-GYNECO (P) DS (P)

***** NOEUDS INTERMEDIAIRES *****

SYNDROME-AUX-3 AGE<30 TEMPERATURE-ELEVEE
SIGNES-PERITONITE SIGNES-GYNECO

**** AUTRES DIAGNOSTICS EVENTUELS ****

SALPINGITE (NIL) GEU (NIL)

***** DIAGNOSTIC REEL .... *****

SALPINGITE

>>

```

Figure 5.4: Un résultat de l'exécution du moteur (3).

- Soit les 5 premiers diagnostics proposés par le système D_1, D_2, D_3, D_4 et D_5 , classés par probabilités décroissantes, respectivement P_1, P_2, P_3, P_4 et P_5 .
- Soient les seuils $S_1 = 0.5, S_2 = 0.2, S_3 = 0.1, S_4 = 0.2$; valeurs données à titre indicatif.

Les critères à essayer de satisfaire, pris dans ce ordre sont :

Critère 1 Si $P_1 - P_2 \geq S_1$ alors le choix du système est D_1 et celui-ci est précis vrai (PF) ou précis faux (PF)

Critère 2 Si $P_{diage\text{exact}} \geq S_2$ et si le diagnostic exact est D_1 alors le choix du système est hésitant mais exact et le système fournit donc une réponse hésitante vraie (HV)

Critère 3 Si $P_{diage\text{exact}} \geq S_2$ et si le diagnostic exact n'est pas D_1 alors :

- Si $P_1 - P_{diage\text{exact}} \leq S_3$ alors le système est hésitant mais dans le vrai (HV)

Il y a un conflit portant sur ATT-URINAIRE
 Ce conflit est cause par une incompatibilite entre
 UIV-DILATATION et MURPHY

Les regles responsables sont

=====

J'APPLIQUE R457 (FIABLE) :.

UIV-ANORMALE (I)
 ⇒ ATT-URINAIRE

alors que

=====

J'APPLIQUE R1266 (FIABLE) :

MURPHY (F)
 ⇒ ATT-URINAIRE NON (F)

***** DIAGNOSTIC REEL *****

COLIQUE-NEPHRETIQUE

»

Figure 5.5: Un résultat de l'exécution du moteur (4).

- Si $S_3 < P_1 - P_{diage\text{exact}} < S_4$ alors le système est en situation de non-réponse (NR)
- Si $S_4 \leq P_1 - P_{diage\text{exact}}$ alors le système est hésitant mais dans le faux (HF)

Critère 4 Si $P_{diage\text{exact}} \leq S_2$ alors le diagnostic exact n'est pas évoqué

- Si $P_1 \geq S_2$ alors le système hésite entre plusieurs résultats dont le diagnostic exact est absent et il est alors en situation hésitante fausse (HF)
- Si $P_1 < S_2$ alors le système est en situation de non-réponse (NR)

4.3 Procédures d'évaluation unifiées

Finalement, pour les deux approches, les résultats se divisent en cinq classes :

- (PV) réponse précise vraie (PV ou CV ou CPV du système expert),
- (HV) réponse hésitante vraie (HV et CHV du système expert),
- (NR) non réponse (NR ou CNR du système expert),
- (HF) réponse hésitante fausse (HF ou CHF du système expert) et
- (PF) réponse précise fausse (PF ou CPF du système expert)

qui permettent ainsi une comparaison des deux systèmes.

4.4 Validité des procédures

Ce type de procédure d'évaluation est une réponse aux problèmes que l'on se pose quand il s'agit de comparer les résultats d'un système de diagnostic avec un autre ou avec un expert. P. Politakis a des problèmes similaires quand il s'agit de comparer les conclusions de SEEK et d'un expert [Politakis & Weiss 84]. Ceux-ci peuvent se résumer en quelques points :

- il existe parfois des liens entre des conclusions qui semblent différentes. Ainsi les relations taxonomiques induisent des liens qui peuvent poser des problèmes comme nous allons le voir dans l'exemple suivant. Ces liens peuvent parfois conduire à une interprétation erronée des conclusions ;
- une interprétation des conclusions de chaque modèle est nécessaire pour pouvoir prendre en compte ces liens ;
- il est alors nécessaire de spécifier les conditions précises d'évaluation des performances.

Le traitement des douleurs non spécifiques (DNS) en est une illustration : ce choix de DNS est pratiquement impossible à faire par le système en raison de la structure des taxonomies et des signes souvent présents sur un patient ; effectivement un patient a toujours un peu de température et donc une quelconque INFLAMMATION ; or, cette affirmation empêche automatiquement de choisir une DNS puisqu'elle est rangée dans les douleurs spécifiques (DS). Donc si le système n'a conclu que sur INFLAMMATION et rien d'autre il est dans la situation du dernier critère (cf. critère 6 page 102) et il est justifié de proposer une DNS.

Dans notre cas et en raison de la disparité de fonctionnement et d'expression des systèmes ces points sont cruciaux. La procédure de comparaison que nous avons mise au point ici en tient compte et y apporte une réponse adaptée. On pourrait penser que ces procédures sont bien subjectives et que d'autres pourraient être adoptées avec plus de profit. Notons cependant quelques points pour répondre à cette critique :

- les résultats du système bayésien sont assez peu sensibles aux modifications des seuils dans une fourchette raisonnable (variation des résultats exacts de moins de 3% pour des variations de seuil de plus de 100%).
- on pourrait penser que l'on a avantagé le système LÉZARD dans l'évaluation des performances au niveau des conflits. Il n'en est rien. C'est parce que au contraire le traitement des conflits au niveau du moteur d'inférences était un peu brutal (et ce pour des besoins d'information à conserver durant le développement du système) que nous avons adopté ce type d'évaluation.

5 Le système Léopard et les douleurs aiguës de l'abdomen

Matériel et logiciel

Le système LÉZARD est écrit en COMMON-LISP et tourne sur VAX-VMS II. Le code tient sur environ 5000 lignes, commentaires compris. Le programme chargé en mémoire utilise environ 500 Koctets.

5.1 Le Traducteur

Tous les mécanismes décrits au chapitre 3 sont utilisés dans le TRADUCTEUR mis en œuvre sur l'application aux douleurs aiguës de l'abdomen à l'exception du processus de généralisation. L'expert n'a effectivement accepté que des règles ayant peu de prémisses et n'a pas trouvé nécessaire de les généraliser.

Nous avons vu en annexe I que l'algorithme de recherche des exclusions est environ en $O(n^2)$. Dans l'application aux douleurs aiguës de l'abdomen cet algorithme tourne en quelques minutes sur VAX 8800, accès aux fichiers compris. L'exécution effective de cet algorithme est suffisamment rapide dans la mesure où elle n'a lieu qu'au moment de la génération des connaissances.

Au niveau du processus d'acquisition, les *connaissances profondes* sont constituées de 153 objets (*signes, syndromes* ou *diagnostics*) reliés par 142 liens. 500 questions environ sont posées à l'expert et contrôlent la génération de 1068 règles de production. Sur ces 1068 règles, 529 sont *incontestables*, 301 *fiables* et 238 *peu-fiables*.

5.2 Le moteur

L'ordonnement des règles

Comme nous l'avons vu en 1.2 ch. 4 page 70 l'ordonnement des règles a produit une seule boucle sur les *diagnostics*, placée en dernière position du classement des règles et boucles (cf. figure 4.1 page 71). Sur les 1068 règles de la base des connaissances, 548 font partie de la boucle et les 529 autres la précèdent. Sur les 548 règles de la boucle, 372 sont *incontestables*, 100 sont *fiabiles* et 76 sont *peu-fiabiles*. L'existence de cette seule boucle était prévisible dans le cadre des relations définissant les connaissances du domaine : en effet, les boucles sont indispensables pour exprimer des rapports symétriques entre objets d'une même *classe* ; en décrivant une unique hiérarchie dans la seule *classe* des *diagnostics*, on obtient une boucle dans cette seule *classe*. De même, il est logique d'observer que la plupart ($\approx 70\%$) des règles *incontestables* sont dans la boucle : la hiérarchie des *diagnostics* génère en effet un grand nombre de règles bouclantes et *incontestables*, alors que les connaissances sémiologiques sont souvent incertaines. En revanche l'existence d'une et une seule boucle placée en dernière position du classement des règles/boucles appelle quelques commentaires.

Dans notre application nous avons obtenu une seule grande boucle parce que les taxonomies ne définissaient qu'une seule hiérarchie, mais au cours d'autres tests sur des taxonomies plus complexes nous avons obtenu plusieurs boucles, toujours sur les *diagnostics*.

Pour ce qui est de l'emplacement de la boucle en dernière position de l'ordonnement des boucles et des règles, rien de particulier ne le justifie sinon que pour notre application il n'a pas été généré de règle entre *diagnostics* sortant du cadre des objets impliqués dans les hiérarchies et donc dans la boucle. Cela pourrait être envisagé et nous l'avons observé dans les tests abordés précédemment où les hiérarchies plus complexes laissaient la place à des règles encadrant les boucles de *diagnostics*.

L'exécution du moteur

Les données fournies au moteur sont les *signes* des malades. Chaque malade présente entre 5 et 20 *signes*. L'exécution du moteur sur un malade dure en moyenne 10 secondes sur VAX-STATION II. 40 à 90 règles sont appliquées durant le diagnostic d'un malade.

6 Résultats

Il faut d'abord noter que sur son échantillon d'apprentissage (94 cas) le système expert a environ 60% de bonnes réponses. Ceci peut paraître peu mais durant la constitution de la base des connaissances un effort a été fait pour ne pas répondre spécifiquement aux cas particuliers posés mais pour être le plus général possible.

Les résultats des deux systèmes sur l'échantillon test de 965 patients sont résumés dans les quatre tableaux suivants : 5.6, 5.7, 5.8 et 5.9 pages 108, 109, 110 et 111.

Diagnostic	PV	HV	HF	PF	NR	Total
APPENDICITE	161	42	25	33	6	267
DNS	43	44	51	91	4	233
CHOLECYSTITE	63	7	10	9	1	90
OCCLUSION	50	13	7	6	1	77
PERFORATION ULCERE	34	4	2	5	0	45
HERNIE ETRANGLEE	25	3	3	5	0	36
PANCREATITE	25	1	3	6	0	35
COLIQUE NEPHRETIQUE	18	1	2	6	1	28
POUSSEE ULCERE	10	4	2	8	1	25
SALPINGITE	8	4	6	6	1	25
INFECTION URINAIRE	4	0	1	12	1	18
KYSTE OVAIRE	0	0	6	12	0	18
SIGMOIDITE	10	1	1	4	0	16
INFARCTUS MESENTERE	2	0	3	5	0	10
D GYNECO FONC	0	1	2	6	0	9
GEU	3	0	2	3	0	8
ANEVRYSME AORTE	4	0	2	1	0	7
CANCER SS OCCLUSION	1	0	3	3	0	7
PERITONITE PRIMITIVE	0	2	0	4	0	6
AFF PULMONAIRE	1	0	1	0	0	2
ABCES FOIE	0	0	1	1	0	2
FAUSSE COUCHE	0	0	0	1	0	1
Totaux	462	127	133	227	16	965
Pourcentages	47.9	13.2	13.8	23.5	1.7	100.0
Caractéristique	Bien classés		Mal classés		Nr	
Pourcentages	61.0%		37.3%		1.7%	

Figure 5.6: Approche bayésienne

Diagnostic	PV	HV	CV	CPV	CPV	CHF	CPF	HF	PF	NR	CNR	Total
APPENDICITE	135	53	0	1	0	1	3	20	47	4	3	267
DNS	30	33	0	1	0	3	19	41	101	4	1	233
CHOLECYSTITE	50	0	5	1	2	0	8	6	16	0	2	90
OCCLUSION	42	2	4	10	2	0	0	5	8	4	0	77
PERFORATION ULCERE	14	2	0	5	5	1	1	3	13	0	1	45
HERNIE ETRANGLEE	11	6	0	0	0	0	5	4	9	0	1	36
PANCREATITE	22	0	7	0	1	0	2	1	2	0	0	35
COLIQUE NEPHRETIQUE	13	0	0	1	2	2	3	1	6	0	0	28
POUSSEE ULCERE	0	2	0	0	0	1	3	3	13	3	0	25
SALPINGITE	2	9	0	0	0	1	4	1	8	0	0	25
INFECTION URINAIRE	2	0	0	0	0	0	0	0	16	0	0	18
KYSTE OVAIRE	0	4	0	0	0	0	0	3	10	1	0	18
SIGMOIDITE	1	1	0	0	0	0	1	3	8	1	1	16
INFARCTUS MESENTERE	0	3	0	0	0	1	1	0	4	1	0	10
D GYNECO FONC	0	2	0	0	0	0	0	2	5	0	0	9
GEU	3	1	0	0	1	0	1	0	2	0	0	8
ANEVRYSME AORTE	3	0	0	0	0	1	0	1	2	0	0	7
CANCER INTESTIN	0	0	0	1	0	0	0	1	5	0	0	7
PERITONITE PRIMITIVE	1	0	0	0	0	0	1	0	2	2	0	6
AFFECTION PULMONAIRE	1	0	0	0	0	0	1	0	0	0	0	2
ABCES FOIE	0	0	0	0	0	0	0	0	2	0	0	2
FAUSSE COUCHE	0	0	0	0	0	0	0	1	0	0	0	1
Totaux	330	118	16	20	13	11	53	96	279	20	9	965
Pourcentages	34.2	12.2	1.7	2.1	1.3	1.1	5.5	9.9	28.9	2.1	0.9	100.0
Caractéristique	Bien classés					Mal classés			Nr tot.			
Pourcentages	51.5%					45.5%			3.0%			

Figure 5.7: Approche système expert

Diagnostic	PV	HV	HF	PF	NR	Total
APPENDICITE	135	54	24	47	7	267
DNS	30	34	63	101	5	233
CHOLECYSTITE	50	8	14	16	2	90
OCCLUSION	42	18	5	8	4	77
PERFORATION ULCERE	14	12	5	13	1	45
HERNIE ETRANGLEE	11	6	9	9	1	36
PANCREATITE	22	8	3	2	0	35
COLIQUE NEPHRETIQUE	13	3	6	6	0	28
POUSSEE ULCERE	0	2	7	13	3	25
SALPINGITE	2	9	6	8	0	25
INFECTION URINAIRE	2	0	0	16	0	18
KYSTE OVAIRE	0	4	3	10	1	18
SIGMOIDITE	1	1	4	8	2	16
INFARCTUS MESENTERE	0	3	2	4	1	10
D GYNECO FONC	0	2	2	5	0	9
GEU	3	2	1	2	0	8
ANEVRYSME AORTE	3	0	2	2	0	7
CANCER INTESTIN	0	1	1	5	0	7
PERITONITE PRIMITIVE	1	0	1	2	2	6
AFF PULMONAIRE	1	0	1	0	0	2
ABCES FOIE	0	0	0	2	0	2
FAUSSE COUCHE	0	0	1	0	0	1
Totaux	264	156	180	305	60	965
Pourcentages	34.2	17.3	16.6	28.9	3.0	100.0
Caractéristique	Bien classés		Mal classés		Nr	
Pourcentages	51.5%		45.5%		3.0%	

Figure 5.8: Approche système expert unifiée

Diagnostic	Occurrences	LÉZARD	Bayes
APPENDICITE	267/965	70.8%	76.0%
DNS	233/965	27.5%	37.3%
CHOLECYSTITE	90/965	64.4%	77.8%
OCCLUSION	77/965	77.9%	81.8%
PERFORATION ULCERE	45/965	57.8%	84.4%
HERNIE ETRANGLEE	36/965	47.2%	77.8%
PANCREATITE	35/965	85.7%	74.3%
COLIQUE NEPHRETIQUE	28/965	57.1%	67.9%
POUSSEE ULCERE	25/965	8.0%	56.0%
SALPINGITE	25/965	44.0%	48.0%
INFECTION URINAIRE	18/965	11.1%	22.2%
KYSTE OVAIRE	18/965	22.2%	0.0%
SIGMOIDITE	16/965	12.5%	68.7%
INFARCTUS MESENTERE	10/965	30.0%	20.0%
D GYNECO FONC	9/965	22.2%	11.1%
GEU	8/965	62.5%	37.5%
ANEVRYSME AORTE	7/965	42.9%	57.1%
CANCER SS OCCLUSION	7/965	14.3%	14.3%
PERITONITE PRIMITIVE	6/965	16.7%	33.3%
AFF PULMONAIRE	2/965	50.0%	50.0%
ABCES FOIE	2/965	0.0%	0.0%
FAUSSE COUCHE	1/965	0.0%	0.0%

Figure 5.9: Comparaison LÉZARD/Bayes des diagnostics bien classés. En gras les meilleurs pourcentages pour chaque diagnostic.

Des études statistiques ont été menées sur les résultats [Golmard 87]. Je ne les développerai pas ici mais j'en donnerai les conclusions :

- Les résultats du système LÉZARD sont globalement inférieurs à ceux de l'approche bayésienne
- mais on voit que pour les diagnostics dont les effectifs sont faiblement représentés les performances de LÉZARD sont meilleures : les tests de différence effectués sur les résultats sont meilleurs pour LÉZARD (de manière statistiquement significative ou non significative selon les diagnostics) pour des diagnostics dont 35 occurrences au plus apparaissent dans l'échantillon de test [Golmard 87]. Ce résultat est logique ; les connaissances de l'expert ne sont pas dépendantes de la base des données, tandis que l'algorithme de Bayes est moins performant sur les diagnostics de faible effectif. Ceci est un résultat important : c'est l'un

des principaux avantages des systèmes experts ; les résultats ne dépendent pas de la taille d'un quelconque échantillon d'apprentissage à partir du moment où l'on a suffisamment d'exemples représentatifs de la diversité d'un diagnostic.

Dans cette optique l'échantillon des 94 patients servant pour l'apprentissage n'a pas fait la part belle aux cas rares. Les 94 patients sont représentatifs des classes de l'échantillon d'apprentissage du système bayésien. Or, pour tirer au mieux parti des capacités du système expert il aurait fallu privilégier les cas rares pour permettre de mieux analyser les conditions de leur diagnostic. C'est dans ce domaine que le système expert est le meilleur, mais un bon choix de l'échantillon d'apprentissage aurait pu le rendre encore plus performant.

On peut essayer d'analyser les causes de ces faibles performances :

- Incomplétude de la base des connaissances. Un gros travail reste à réaliser dessus et il existe de nombreux cas pour lesquels une base des connaissances plus achevée permettrait de rectifier les erreurs de diagnostic.
- Insuffisance du mode de représentation des connaissances. Bien que le système LÉZARD ait déjà de nombreuses connaissances : base des connaissances, heuristiques dans l'acquisition des connaissances, heuristiques dans le contrôle de l'inférence, . . . , elles ne fournissent pas une description complète du domaine. Elle est cependant sans commune mesure avec la procédure bayésienne qui a un mode de représentation des connaissances très faible mais qui fait de l'apprentissage statistique sur un grand nombre de cas (≈ 6000).

Ainsi, devant certains tableaux cliniques, un expert humain formule a priori des observations telles que : "il n'existe aucun signe de gravité", ou au contraire: "je ne connais pas encore le diagnostic, mais c'est une affection très grave". Ces conclusions sont plus proches d'une intuition que d'un raisonnement logique, et donc difficiles à représenter dans une base de connaissances. Les erreurs dues à des insuffisances de représentation de ce type se retrouvent d'ailleurs dans la plupart systèmes experts [Szolovits 88].

- Le diagnostic des DOULEUR-NON-SPECIFIQUE (qui représentent le quart des diagnostics) est difficile à faire pour le système expert. Les patients relevant de ce diagnostic présentent souvent des signes non significatifs : c'est leur accumulation qui éveille l'attention du médecin. A l'inverse il peuvent présenter une accumulation incohérente de signes importants.

C'est en partie un problème de représentation des connaissances et le codage de tableaux cliniques types est une solution envisageable. Mais ce n'est pas la seule : le diagnostic de DOULEUR-NON-SPECIFIQUE est un peu un diagnostic "parasite" dans la mesure où y a été rangé tout les cas pour lesquels il n'avait pas été trouvé de diagnostic au niveau de la base des données. Cela explique pourquoi la procédure bayésienne et LÉZARD ont tous deux des résultats mauvais et comparables sur le diagnostic des DOULEUR-NON-SPECIFIQUE (27,5% de DOULEUR-NON-SPECIFIQUE bien classés dans LÉZARD et 37%

dans le système bayésien ; cf. tableau 5.9 page 111). On peut imaginer des procédures de décision *heuristiques* au niveau du contrôle du moteur qui permettraient de s'affranchir des diagnostics "parasites" tout en n'étant pas spécifique au domaine. De toute façon si on examine les résultats des 2 systèmes hors DOULEUR-NON-SPECIFIQUE, on trouve pour LÉZARD 59,2% de bien classés et pour le système bayésien 68,6% de bien classés. Dans les deux cas cela améliore notablement les performances⁶.

Le problème de la DOULEUR-NON-SPECIFIQUE est spécifiquement discuté dans [Séroussi 86].

- Les problèmes abordés en 3.2 page 95 sur la correspondance des 22 diagnostics fausse la comparaison au désavantage du système LÉZARD. Ainsi, nous avons vu que dans la base de l'ARC & AURC il y a les diagnostics OCCLUSION et CANCER-SANS-OCCLUSION alors que dans la base des connaissances de LÉZARD il y a le diagnostic CANCER-INTESTIN et les diagnostics OCCLUSION-ORGANIQUE-PRIMITIVE et OCCLUSION-ORGANIQUE-SECONDAIRE. Pour des raisons de cohérence nous avons été obligé de faire correspondre CANCER-SANS-OCCLUSION avec CANCER-INTESTIN et OCCLUSION avec OCCLUSION-ORGANIQUE-PRIMITIVE. Dans le cas d'un cancer de l'intestin ayant provoqué une occlusion organique le diagnostic exact sera considéré comme étant OCCLUSION pour la base des données c'est à dire OCCLUSION-ORGANIQUE-PRIMITIVE pour le système expert. Ce dernier peut alors conclure, à juste titre, à CANCER-INTESTIN accompagné d'un processus occlusif comme OCCLUSION-ORGANIQUE-SECONDAIRE : et sa conclusion sera jugée fausse.

De la même manière, pour des raisons de classification imprécise de nombreux diagnostics d'affections ayant provoqué des occlusions sont rangés sous le simple vocable OCCLUSION dans la base de l'ARC & AURC. Identiquement le système LÉZARD peut trouver la bonne affection accompagnée d'un syndrome occlusif et être déclaré en erreur. Ces problèmes apparaissent dans 47 cas sur 965. Il est évident que l'on ne peut affirmer que le système expert a raison dans tous ces cas mais il y en a une proportion non négligeable où il en est ainsi (quelques vérifications nous ont montré des cas correspondant à ces problèmes de classification). Remarquons simplement que ces problèmes touchent près de 5% des patients de l'échantillon test.

- le moteur d'inférences qui comporte déjà des heuristiques de raisonnement, manque encore d'informations de ce type. La reconnaissance et la pertinence de ce genre d'heuristiques est difficile à effectuer et demande de longs tests.

⁶Notons encore que l'on peut améliorer les résultats de LÉZARD sur les diagnostics autres que DOULEUR-NON-SPECIFIQUE (dans une proportion parfois supérieure à 30%) au détriment de ces dernières. Mais les DOULEUR-NON-SPECIFIQUE étant très nombreuses dans la base des données cela fait baisser les résultats globaux. En expérimentation réelle il serait évidemment préférable de maintenir une version de la base des connaissances améliorant les résultats des diagnostics graves au détriment de DOULEUR-NON-SPECIFIQUE.

7 Conclusion et perspectives

L'étude menée a permis le développement et la validation de certaines méthodes sur une base de cas réels. Le prototype fonctionne et même si ses résultats ne sont pas excellents, les raisons de ce déficit peuvent être analysées.

Dans l'hypothèse de développements futurs sur ce prototype, des idées peuvent déjà être retenues :

- sur le plan de la représentation des connaissances des améliorations sont à envisager pour mieux traduire les connaissances du médecin,
- sur le plan de la résolution des conflits un certain nombre d'heuristiques doivent être apprises et mises en œuvre.
- tous les problèmes pratiques qui sont apparus dans la comparaison des deux systèmes du point de vue de la base des données et de la base des connaissances doivent être au maximum résolus, dans le cadre d'une application donnée.

Ces idées seront reprises et développées dans la conclusion de la thèse.

Chapitre 6

Perspectives

Introduction

Ce dernier chapitre dresse un bilan des problèmes abordés dans la thèse.

Nous y abordons les améliorations possibles de l'implémentatiuon de LÉZARD.

Une dernière partie prospective discute, entre autres, des problèmes d'acquisition et de représentation des connaissances.

Nous avons essayé, dans LÉZARD, de traduire une des caractéristiques essentielles de la démarche médicale : raisonner en présence d'informations incomplètes. Cela nous a amené à développer un système qui utilise un certain nombre de connaissances informelles, *heuristiques*, que ce soit au niveau de l'acquisition des connaissances ou au niveau du raisonnement. Les composantes fondamentales de ce système sont les suivantes :

- des *connaissances profondes* qui fournissent un modèle du domaine et donnent le cadre de l'expertise ;
- des *méta-connaissances* qui permettent de guider le processus d'acquisition des *connaissances de surface* à partir des *connaissances profondes*, en contrôlant le dialogue avec l'expert ; cette traduction permet de faire ressortir les connaissances opératoires à partir du modèle *profond* et ainsi d'éliminer les conflits potentiels d'interprétation ;
- des *connaissances de surface* qui sont les règles utilisées par le moteur d'inférences ; elles correspondent à un modèle opérationnel où chaque granule de connaissance est valide ; leur processus d'interprétation est alors simplifié, dans la mesure où il ne dépend que d'une interprétation logique indépendante du domaine ;

- des *stratégies* qui fournissent les informations nécessaires à l'exploitation des *connaissances de surface* par le moteur d'inférences, en particulier dans la résolution de conflits.

Ces caractéristiques, qui donnent le squelette du système, montrent qu'il est utilisable dans de nombreux domaines pour toute tâche de discrimination diagnostique.

Le faible nombre de primitives du langage de description des connaissances ne permet pas de décrire toutes les connaissances médicales qu'un expert peut vouloir exprimer. Il en est de même pour la description des *méta-connaissances* d'acquisition des connaissances et surtout pour les *stratégies* du moteur. Pourtant, les résultats obtenus permettent de valider l'approche retenue et mettent en lumière les développements qui pourraient être apportés au système.

Nous allons d'abord détailler les principales insuffisances apparues dans le système et voir quelles solutions pourraient les pallier avant d'aborder d'un point de vue plus général les perspectives de développement du système, en particulier au niveau des *méta-connaissances*.

1 Les améliorations envisagées

Les connaissances profondes

Le langage de représentation des connaissances manque de primitives ; il ne permet pas d'exprimer, par exemple, le tableau clinique d'une maladie sous la forme habituellement utilisé par les médecins : en allant d'un diagnostic vers les signes qui en sont des conséquences.

On peut alors imaginer créer des primitives qui permettent de répertorier les signes associés à une maladie et non pas les maladies associées à un signe comme le permettent pour l'instant les primitives *evoque(nt)*.

Le traducteur

On peut se poser la question de savoir si le TRADUCTEUR a permis de construire une base de règles cohérentes. Des travaux de ce type ont été décrits dans [Ayel et al. 88]. Dans le cas du système LÉZARD la question est un peu différente. On ne cherche pas une cohérence de la base de règles, mais on essaie au contraire d'exploiter les incohérences que l'on considère comme inévitables dans la description des connaissances. Nous verrons au niveau du moteur que cette exploitation peut être envisagée d'une nouvelle manière.

Les *heuristiques* utilisées dans le TRADUCTEUR pour générer de nouvelles connaissances et la *logique L* doivent faire l'objet d'une réflexion plus approfondie. Il est en effet possible d'utiliser plus formellement la logique *L* pour piloter le questionnaire et pour être plus rigoureux dans les

questions à poser. De même, les *heuristiques* peuvent être revues et leur utilisation contrainte ou étendue.

La compilation des règles de production

Nous avons vu que lors de la compilation des règles, des boucles de règles sont recherchées et qu'à l'intérieur de celles-ci un certain classement est effectué (cf. 1) ; il faut noter que, alors qu'il a sa raison d'être et que son efficacité est évidente en dehors des boucles, ce classement est d'une efficacité douteuse dans les boucles. On peut alors envisager d'améliorer le traitement des règles dans les boucles, principalement pour obtenir une meilleure résolution des contradictions.

Alors que l'on peut effectivement ordonner les règles bouclantes, le problème vient de la manière de gérer la boucle et donc, principalement, des *points d'entrée* dans la boucle. Pour l'instant, le choix d'un *point d'entrée* est aléatoire mais on peut imaginer une procédure de détermination d'un tel point se déroulant en 2 étapes : (1) détermination statique des "bons" candidats pour être *point d'entrée* (2) puis choix dynamique du *point d'entrée* durant l'exécution du moteur d'inférences. Nous nous contenterons de donner ici quelques indications sur ces déterminations statiques et dynamiques :

- **Détermination statique.** Les règles dont toutes les prémisses sont des conclusions de règles qui précèdent la boucle sont des candidats a priori intéressants pour un *point d'entrée*. De telles règles existent car une règle précède une boucle si sa conclusion fait partie des prémisses d'une des règles de la boucle.
- **Détermination dynamique.** Si ces règles sont de bons candidats lors du déclenchement de la boucle par le moteur, il se peut que certaines d'entre elles, voire toutes, ne soient pas déclenchables. Le choix dynamique du *point d'entrée* peut alors s'effectuer comme suit : choix parmi les *points d'entrée* de ceux déclenchables.

Si une contradiction est détectée dans la boucle et ne peut y être levée, il se peut que le choix d'un nouveau *point d'entrée* résolve le problème. Notons que cela impose de mémoriser l'état du système avant d'entrer dans une boucle.

Moteur d'inférences

La principale amélioration envisagée pour le moteur est de lui faire gérer les règles sous forme de "paquets" consistants : on a vu en effet, en 2.4.1 page 74 que le moteur ne faisait pas d'hypothèse à la manière d'un TMS. Pourtant la base des règles est par construction inconsistante et il faudrait pouvoir ne retenir qu'un certain nombre d'observations hypothèses sur lequel appuyer le raisonnement. En se plaçant dans le cadre d'une gestion d'hypothèses proche

des TMS ou des ATMS [de Kleer 86, Cordier 86], la comparaison des raisonnements faits à partir des hypothèses concurrentes permettra d'effectuer des choix. Dans le cadre de ces choix de raisonnement, des développements sur les raisonnements par défaut et leur spécificité ont évidemment leur place [Moinard 87].

Du point de vue des explications, le moteur ne fournit pour l'instant que les traces des règles de production déclenchées. Il est déjà prévu de revenir aux *connaissances profondes* associées aux règles pour générer des explications. On pourra alors avoir des explications aussi *profondes* que le modèle [Swartout 77, Ferrand 84, Reynaud 89].

D'autres améliorations sont déjà envisagées. Certaines sont en cours d'implémentation, d'autres ne sont que des projets. On peut citer :

- Le raisonnement est pour l'instant non-additif, dans la mesure où on ne retient comme support d'une conclusion qu'un arbre *ET* (cf. 2.4.1.a page 75). Nous pouvons, et c'est déjà une fonctionnalité en œuvre, tenir compte de l'arbre *ET/OU* complet et mettre ainsi en place un raisonnement additif. Ceci est intéressant que ce soit du point de vue du raisonnement *qualitatif* ou du point de vue du raisonnement *quantitatif*.
- La partie *quantitative* du raisonnement n'est pas très sophistiquée (cf. 2.4.2.c page 80). On peut se poser la question de l'utilité et de la mise en place d'un calcul à base de coefficients de certitude comme dans MYCIN, ou de fonctions d'appartenance comme dans les ensembles flous de Zadeh [Zadeh 78].
- Dans un contexte où sont autorisées des questions durant l'exécution du moteur, on peut imaginer d'implémenter des stratégies pour obtenir des informations via des arbres d'interrogation (plan de questions/réponses adaptés à chaque maladie hypothétique) [Morice 88].
- Il est des cas où une procédure de choix probabiliste comme celle de Bayes est meilleure qu'un système expert. Cela est vrai pour LÉZARD (cf. le cas des DNS, chapitre 6 page 107) comme pour d'autres systèmes : une méthode probabiliste a souvent des résultats équivalents à ceux d'un expert. On pourrait envisager une coopération entre le système expert et un module probabiliste : l'interaction avec un système d'analyse de la décision *quantitatif*, de type bayésien par exemple, permettra de faire des diagnostics inaccessibles au système expert, même si ses explications ne sont pas recevables.

2 Perspectives

Pour terminer, nous allons aborder deux problèmes plus généraux que ceux développés précédemment : (i) les *méta-connaissances*, leur représentation et leur acquisition et (i) les méthodes d'apprentissage que l'on pourrait mettre en œuvre pour acquérir les *connaissances de surface*.

2.1 Les méta-connaissances

Les *méta-connaissances* d'acquisition des connaissances comme les *stratégies* du moteur sont difficiles à appréhender.

Les *stratégies* de résolution de conflit sont peut-être les plus faciles à acquérir et à représenter : leur rôle est celui de principes donc on a une idée assez claire de ce que l'on recherche et leur représentation qui se fait dans le cadre du langage d'implémentation du système peut être déclarative. En effet, bien qu'on ne puisse peut-être pas proposer un schéma unique de représentation, on peut, en tout état de cause, les encoder chacune dans une fonction du langage d'implémentation du système appellable par le contrôleur. On obtient là une certaine déclarativité des connaissances.

Pour les autres *méta-connaissances* le problème est plus ardu. Nous avons vu que leurs natures sont très différentes les unes des autres. Leur expression n'en est que plus difficile dans le cadre d'un schéma de représentation commun. En fait, il semble qu'il ne soit pas raisonnable d'essayer de les exprimer, soit dans un schéma, soit de façon déclarative ; le problème n'est pas là : là où il y a problème, et nous reprenons une argumentation déjà développée précédemment, c'est dans la difficulté qu'il y a à identifier, acquérir et utiliser ces *méta-connaissances*. Notre démarche a été très pragmatique et il est évident que les solutions retenues sont trop spécifiques. Des travaux sur ces *méta-connaissances*, leur acquisition et leur représentation sont nécessaires.

2.2 Acquisition des Connaissances et apprentissage

Nous avons vu précédemment les méthodes employées dans le système LÉZARD pour acquérir des connaissances (cf. chapitre 3). L'acquisition développée dans le TRADUCTEUR est une forme d'apprentissage, mais cet apprentissage n'est ici pas très automatisé : il nécessite un professeur. On peut se poser le problème d'adapter au système LÉZARD une forme d'acquisition des connaissances sans professeur. Dans ce cas plusieurs questions se posent : Pourquoi vouloir faire de l'apprentissage sans professeur ? Le système LÉZARD se prête-t-il à ce type d'apprentissage ? Quelles connaissances peut-on apprendre et quelles techniques utiliser ?

Ces questions sont liées et la réponse sera globale :

1. On peut vouloir faire de l'apprentissage sans professeur pour deux raisons : premièrement pour améliorer l'acquisition des connaissances ; deuxièmement pour améliorer les performances du système. Ces deux raisons sont liées dans la mesure où si l'on attend une amélioration du processus d'acquisition des connaissances d'un système, on espère bien obtenir de meilleures connaissances et par là améliorer les performances du système.
2. Si on se place du point de vue de l'acquisition des connaissances dans le système LÉZARD, on constate que le TRADUCTEUR, propose parfois un travail un peu fastidieux à l'expert (cf. 5 ch.5) ; on a donc envie d'automatiser en partie ce processus.

3. D'autre part, la structure modulaire du système se prête bien à l'implantation d'un module d'apprentissage.
4. Maintenant, il ne semble pas raisonnable de vouloir acquérir automatiquement l'ensemble de la base des connaissances : le modèle *profond* et les *méta-connaissances heuristiques* doivent toujours être données par un expert. En revanche, l'acquisition et l'amélioration automatique des *connaissances de surface* à partir des *connaissances profondes* semble être un objectif raisonnable¹. On va donc, en suivant les mêmes réflexions qui ont conduit à l'élaboration du TRADUCTEUR, chercher des méthodes sans professeur pour acquérir ou améliorer les *connaissances de surface* à partir des *connaissances profondes*.

Après ces réponses, nous allons décrire une méthode d'apprentissage des *connaissances de surface* à partir des *connaissances profondes* développée par Steels et Van de Velde [Steels & Van de Velde 85] ; puis nous verrons comment LÉZARD se situe de ce point de vue ; nous aborderons enfin le système SEEK [Politakis et Weiss 84]. A partir de là nous pourrons donner quelques perspectives de développement de l'apprentissage dans le système LÉZARD [Charlet 89b].

Apprentissage par l'exemple des connaissances de surface

Steels et Van de Velde proposent d'acquérir les *connaissances de surface* par apprentissage sur des exemples. Partant des *connaissances profondes* et d'un ensemble de cas réels — les données des cas à traiter et les conclusions des experts —, leur programme construit un ensemble de règles de production qui constituent les *connaissances de surface*.

Le processus suivi est alors :

- utiliser les *connaissances de surface* quand elles sont valables ;
- utiliser les *connaissances profondes* quand les règles sont inadéquates ; dans ce dernier cas une nouvelle *connaissance de surface* doit être extraite des *connaissances profondes* pour une meilleure efficacité du système quand le même type de cas se représentera.

Le processus décrit ci-dessus est, en dehors de la spécificité de sa phase d'apprentissage, tout à fait adapté à l'exploitation d'une base des connaissances représentées par des *connaissances profondes* et des *connaissances de surface*, que ce soit en médecine ou dans un autre domaine : en médecine, par exemple, le raisonnement diagnostique a lieu à plusieurs niveaux, des connaissances cliniques de *surface* aux connaissances physiopathologiques *profondes*, et l'expert n'éprouve apparemment aucune difficulté à passer d'un type de connaissances à un autre quand

¹Un "objectif raisonnable" ne signifie pas que l'on ne peut faire de l'apprentissage que sur les *connaissances de surface*. D'autres s'essaient à l'acquisition d'heuristiques ou de *connaissances profondes*. Mais, dans notre cas c'est l'apprentissage des *connaissances de surface* qui est manifestement à développer.

cela est nécessaire [Szolovits et al. 88, Fink 87]. En revanche la mise en œuvre de ce processus n'est pas facile car elle nécessite l'utilisation de deux interpréteurs pour être vraiment efficace : un pour les *connaissances profondes*, un pour les *connaissances de surface*.

L'apprentissage dans LÉZARD

Dans le système LÉZARD, l'acquisition des connaissances est du même type que précédemment dans la mesure où l'on part des *connaissances profondes* pour générer des *connaissances de surface*. En revanche, les conditions d'application sont évidemment différentes. En particulier, les processus de génération de règles sont plus élaborés car, (i) ils sont spécifiques au domaine (tout du moins pour certains) et (ii) l'interprétation des *connaissances profondes* a lieu dans un environnement incertain. A la différence de Steels et Van de Velde qui appliquent directement les *connaissances de surface* générées sur des exemples, elles sont, dans LÉZARD, filtrées par l'expert.

Cette approche s'apparente à l'apprentissage par conseil ("Learning by taking advice") [Cohen & Feigenbaum 82, Mostow 83] : Le conseil est ici le modèle donné par l'expert — une information de "haut niveau". Modèle qui est ensuite traduit en règles de production — une information de "bas niveau".

Seek

Le système SEEK est un système développé initialement pour faire l'amélioration d'une base de règles de production pondérées dans un domaine donné. Ses principales caractéristiques sont :

- utilisation d'informations statistiques sur les performances d'un système expert ;
- utilisation d'une base de cas avec conclusions connues comme principale source d'informations pour le raffinement des règles. Les cas dans lesquels la conclusion est fautive et différente de l'expert permettent de suggérer les règles responsables de ce fait ;
- proposition de suggestions pour modifier les règles mises en cause ;
- système interactif.

SEEK peut être vu comme un système d'apprentissage restreint limité à l'amélioration de règles existantes. Il n'est pas autonome dans la mesure où il propose ses modifications à l'expert.

Perspectives

La structure du système LÉZARD, et du TRADUCTEUR en particulier, a été élaborée en tenant compte des problèmes d'acquisition des connaissances. Ainsi, adapter un processus d'apprentissage automatique sur le module TRADUCTEUR est aisé dans la mesure où sa structure supporte déjà un module d'apprentissage.

Les deux systèmes décrits précédemment fournissent d'intéressants axes de développement pour faire de l'apprentissage dans le système LÉZARD. En particulier, la méthodologie développée dans SEEK permet, par exemple, de modifier la crédibilité (la *fiabilité* pour LÉZARD) des règles générées.

Sans entrer plus dans les détails, notons que les deux systèmes présentés ouvrent des perspectives d'apprentissage tout à fait réelles pour le système LÉZARD.

Références Bibliographiques

1. **A. Barr and E.A. Feigenbaum (eds), 81a**
Search & Representation of Knowledge.
In: A. Barr and E.A. Feigenbaum (eds.), *The Handbook of Artificial Intelligence, vol 1*, W. Kaufmann, Inc., Los Altos, California, 1981; Chapters II & III.
2. **A. Barr and E.A. Feigenbaum (eds), 81b**
Applications-oriented AI Research: Science & medicine.
In: A. Barr and E.A. Feigenbaum (eds.), *The Handbook of Artificial Intelligence, vol 1*, W. Kaufmann, Inc., Los Altos, California, 1981; Chapters VII & VIII.
3. **Ayel M., Chein M., Pipard E., Rousset M.-C., 88**
De la cohérence dans les bases de connaissances.
In: *Actes des journées nationales du PRC-GRECO/IA, Toulouse, Teknea, , 1988;357-384.*
4. **R.K. Bathnagar, L.N. Kanal and J.F. Lemmer 86**
Handling Uncertain Information.
In: L.N. Kanal and J.F. Lemmer (eds.), *Machine Intelligence and Pattern Recognition, vol 1: Uncertainty In Artificial Intelligence*, North-Holland, Amsterdam, 1986;3-26.
5. **Berge, C., 83**
Graphes, Bordas, Paris, 1983.
6. **Beuscart R., Beuscart C., Auvergnat J.-C., Duhamel A., Morice V., 87**
SES : Système Expert Septicémies. Expertise multiple et aide à la décision médicale.
In: *Actes du colloque IA/Santé associé au SITEF, Toulouse, Dubar, Toulouse, 1987;115-126.*
7. **Bonissone P., 87**
Reasoning, Plausible.
In: S.C. Shapiro (ed.), *Encyclopedia of Artificial Intelligence, vol 2*, J. Wiley & Sons, Inc., New York, 1987;854-863..
8. **Bouaud J., 89. K** ; Un langage pour l'Implémentation d'Outils de Représentation des Connaissances. Thèse d'université, université Paris VII, 1989.
9. **Buisson J.C., 87**
TOULMED, un générateur de systèmes experts qui prend en compte l'imprécision et l'in-

certitude des connaissances.

Thèse de doctorat de l'Institut National Polytechnique de Toulouse, 1987.

10. Chandrasekaran B., Mittal S., 83
Deep versus compiled knowledge approaches to diagnostic problem solving.
Int. J. Man-Machine Studies, 19:1983 (425-436).
11. Charlet J., Gascuel O., 86
Anatomie du système LÉZARD: le traducteur.
In: *Proceedings of the 6th International Workshop on Expert Systems and their Applications*, Avignon, Gerfau, Paris, 1986;95-106.
12. Charlet J., Gascuel O., 87
Acquisition des connaissances dans le système LÉZARD.
In: *Actes du colloque IA/Santé associé au SITEF*, Toulouse, Dubar, Toulouse, 1987;105-114.
13. Charlet J., 88
Acquisition de connaissances à partir d'un modèle causal et d'heuristiques.
In: M.H. Hamza (ed.), *Proceedings of the IASTED International Conference*, Genève, Acta Press, Zurich, 1988;67-71.
14. Charlet J., 89a
Acquisition des connaissances & raisonnement dans un système expert de seconde génération. In: *Proceedings of the 9th International Workshop on Expert Systems and their Applications*, Avignon, Gerfau, Paris, 1989;175-191.
15. Charlet J., 89b
Knowledge Acquisition by Causal Model and Meta-knowledge.
In: *Proceedings of the Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Paris, Express-Tirages, Bourg-La-Reine, 1989;212-225..
16. Charniak E., McDermott D., 87
Abduction, Uncertainty and Expert Systems.
In: *Introduction to Artificial Intelligence*, Addison-Wesley Publishing Company, Reading, Massachusetts 1987;Chapter 8.
17. R. Cohen and E.A. Feigenbaum (eds), 82
Learning and Inductive Inference.
In: R. Cohen and E.A. Feigenbaum (eds.), *The Handbook of Artificial Intelligence*, vol 3, W. Kaufmann, Inc., Los Altos, California, 1982;Chapter XIV.
18. Cordier M.-O., 86
Informations incomplètes et contraintes d'intégrité : le moteur d'inférences SHERLOCK.
Thèse de doctorat d'état, Université de Paris-sud, Orsay, 1986.

19. De Dombal F.T., 76
Computer-Aided Diagnosis: a Pratical Proposition?
In: F.T. de Dombal and F. Gremy (eds.), *Decision Making and Medical Care*, North-Holland, Amsterdam, 1976.
20. Doyle J., 79
A Truth Maintenance System.
Artificial Intelligence 12 (231-272) 1979.
21. Dreyfus H.L., 79
What Computers can't do. The Limits of Artificial Intelligence, Harper & Row, Publishers, Inc., New York, 1979.
22. Dubois D., Prade H., 85
Théorie des possibilités, application à la représentation des connaissances en informatique, Masson, Paris, 1985.
23. Falissard, B., 85
Systèmes experts et systèmes formels.
Rapport interne #24, Département IA & Médecine, INSERM U194, Paris, 1985.
24. Ferrand P., 83
SESAM: un système explicatif d'aide médicale.
Thèse de troisième cycle, université d'Orsay, 1983.
25. Ferrand P., 84
SESAM: an Explanatory Medical Aid System.
In: T. O'shea (ed.), *Proceedings of the 6th ECAI, Pisa*, North-Holland, Amsterdam, 1984;13-20.
26. Ferrand P., Levy M., Chirat V., Boisvieux J.-F., 87
SESAM-DIABÈTE : un système expert explicatif d'aide au comportement pour les diabétiques insulinotraités.
In: *Actes du colloque IA/Santé associé au SITEF, Toulouse*, Dubar, Toulouse, 1987;323-330.
27. Fieschi M., 81
Aide à la décision en médecine: le système SPHINX.
Thèse de docteur en médecine, Faculté de médecine de Marseille, 1981.
28. Fieschi M., 84
Intelligence artificielle en médecine. Des systèmes experts, Masson, Paris, 1984.
29. Fink P.K., 87
Control and integration of diverse knowledge in a diagnostic expert system.
In: A. Joshi (ed.), *Proceedings of the 9th IJCAI, Los Angeles*, M. Kaufmann, Inc., Los Altos, California, 1985;426-431.

30. Gaschnig J., Klahr P., Pople H., Shortliffe E., Terry A., 83
In: F. Hayes-roth, D.A. Waterman, D.B. Lenat (eds.), *Building Expert System*, Addison-Wesley Publishing Company, Reading, Massachussetts, 1983;Chapter 8.
31. Gascuel O., 81
Un système expert dans le domaine médical.
Thèse de troisième cycle, université Pierre et Marie Curie, Paris, 1981.
32. Gascuel O., 85
Un système expert pour la réalisation de diagnostics.
Technique et Science Informatiques, 4(4):359-372 (1985).
33. Gascuel O., Charlet J., Genetet S., Mari B., 85
Le système LÉZARD un travail expérimental sur le raisonnement incertain.
In: *Proceedings of the 5th International Workshop on Expert Systems and their Applications*, Avignon, Gerfau, Paris, 1985;.
34. Ghallab M., 88
Compilation de bases de connaissances. In: *Actes des journées nationales du PRC-GRECO/IA*, Toulouse, Teknea, , 1988;231-234.
35. Ginsberg A., Weiss S., Politakis P., 85
Seek2: A Generalised Approach to Automatic Knowledge Base Refinement.
In: A. Joshi (ed.), *Proceedings of the 9th IJCAI*, Los Angeles, M. Kaufmann, Inc., Los Altos, California, 1985;367-374.
36. Golmard J.-L., Rodary M., ARC & AURC Cooperative Group, 85
Weights optimization in a rule-based, expert system : an application to to the diagnosis of acute abdominal pains. In: *Lecture Notes in Medical Informatics*, 28:25-30 (1985).
37. Golmard J.-L., 87
Comparaison de méthodes informatiques d'aide à la décision en médecine.
Rapport SIM #86c01230, INSERM U194, Paris, 1987.
38. Gorry G.A., Silverman H., Pauker S.G. 78
Capturing Clinical Expertise: A computer program that consider the response to digitalis.
American Journal of Medicine, 64:452-460 (1978).
39. Hewitt, C.E., 72
Description et Theoretical Analysis of PLANNER: a Language for Proving Theorems et Manipulating Models in a Robot.
Technical Report TR-258, MIT AI Lab., Cambridge, Massachussetts, 1972.
40. Kayser D., 88
Le raisonnement à profondeur variable. In: *Actes des journées nationales du PRC-GRECO/IA*, Toulouse, Teknea, , 1988;109-136.

41. **de Kleer, J., 84**
Choices without Backtracking.
In: *Proceedings of the 4th AAAI, Austin, Texas*, M. Kaufmann, Inc., Los Altos, California, 1984;79-85.
42. **de Kleer, J., 86**
An Assumption-based TMS.
Artificial Intelligence 28 (1986) 127-162.
43. **Klein, D., Finin, T., 87**
What's in a Deep Model ? A Characterization of Knowledge Depth in Intelligence Safety Systems.
In: J. McDermott (ed.), *Proceedings of the 10th IJCAI, Milan*, M. Kaufmann, Inc., Los Altos, California, 1987;559-562.
44. **Konolige K., 79**
An Inference Net Compiler for the PROSPECTOR rule-based Consultation System.
In: *Proceedings of the 6th IJCAI, Tokyo*, M. Kaufmann, Inc., Los Altos, California, 1979;487-489.
45. **Koton P.A., 85**
Empirical and Model-Based Reasoning in Expert Systems.
In: A. Joshi (ed.), *Proceedings of the 9th IJCAI, Los Angeles*, M. Kaufmann, Inc., Los Altos, California, 1985;297-299.
46. **Kulikowski C.A., 87**
Domain knowledge.
In: S.C. Shapiro (ed.), *Encyclopedia of Artificial Intelligence, vol 1*, J. Wiley & Sons, Inc., New York, 1987;251-253..
47. **Kulikowski C.A., Weiss S.M., 82**
Representation of Expert Knowledge for Consultation: the CASNET and EXPERT Projects.
In: P. Szolovits (ed.), *Artificial Intelligence in Medicine*, Westview Press, Inc., Boulder, Colorado, 1982;Chapter 2.
48. **Laurière J.L, 85**
Représentation et utilisation des connaissances.
Technique et Science Informatiques, 1(1/2):25-42/109-133 (1985).
49. **Lenat D.B., 82**
The Nature of Heuristics.
Artificial Intelligence 19 (1982) 189-249.
50. **Levesque H.J., 84**
A Logic of Implicit and Explicit Belief.

- In: *Proceedings of the 4th AAAI, Austin, Texas*, M. Kaufmann, Inc., Los Altos, California, 1984;198-202.
51. Long W., Shapur N., Criscitiello M.G., 82
A knowledge Representation for Reasoning about the Management of Heart Failure.
In: *Proceedings of the Computers in Cardiology Conference, Seattle*, 1982.
 52. Long W., Naimi S., Criscitiello M.G., Pauker S.G., Szolovits P., 84 An Aid to Physiological Reasoning in the Management of Cardiovascular Disease.
In: *Proceedings of the Computers in Cardiology Conference*, 1984.
 53. McCarthy J., 80 Circumscription: a Form of Nonmonotonic Reasoning.
Artificial Intelligence 13 (1980) 27-39.
 54. McDermott D., Doyle J., 80
Nonmonotonic Logic I.
Artificial Intelligence 13 (1980) 41-72.
 55. Manna Z., Waldinger R., 85 *The Logical Basis for Computer Programming*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1985.
 56. Michalski R.S., 83
A theory and methodology of inductive learning.
In: R.S. Michalski, J.G. Carbonell and T.M. Mitchell (eds.), *Machine Learning, An Artificial Intelligence Approach*, Tioga Publishing Company, Palo Alto, California, 1983;??-??.
 57. Moinard Y., 87
Donner la préférence au défaut le plus spécifique. In: *Actes du 6ième congrès RFIA, Antibes*, Dunod, Paris, 1987;1123-1132.
 58. Morice V., Degoulet P., Jeunemaitre X., Chatellier G., Devriès C., Hemdaoui A., Gascuel O., Boisvieux J.-F., Menard J., 86
Réalisation et évaluation d'un système expert en hypertension artérielle.
In: *Proceedings of the 6th International Workshop on Expert Systems and their Applications, Avignon*, Gerfau, Paris, 1986;561-570.
 59. Morice V., 88
SAGA : un générateur temporel de plans pour la recherche diagnostique.
Thèse de doctorat d'état, Université de Paris 6, 1988.
 60. Mostow D.J., 83
Machine Transformation of Advice into a Search Procedure.
In: R.S. Michalski, J.G. Carbonell and T.M. Mitchell (eds.), *Machine Learning, An Artificial Intelligence Approach*, Tioga Publishing Company, Palo Alto, California, 1983;367-403.

61. Nilsson N.J., 80 *Principles of Artificial Intelligence*, M. Kaufmann, Inc., Los Altos, California, 1980.
62. Patil R.S., 81
Causal representation of patient illness for electrolyte and acid-base diagnostic.
Thesis, Artificial Intelligence Memo AIM-267, MIT AI Lab., Cambridge, Massachusetts, 1981.
63. Patil R.S., Szolovits P., Schwartz W.B., 82
Modeling Knowledge of the Patient in Acid-Base and Electrolyte Disorders.
In: P. Szolovits (ed.), *Artificial Intelligence in Medicine*, Westview Press, Inc., Boulder, Colorado, 1982;Chapter 6.
64. Patil R.S., 87
A case Study on Evolution of System Building Expertise: Medical Diagnosis.
In: W.E.L. Grimson and R.S. Patil (eds.), *AI in the 1980's and Beyond: An MIT survey*, MIT Press, Cambridge, Mass., 1987;75-108.
65. Pauker S., Gorry G., Kassirer J., Schwartz S., 76
Towards the Simulation of Clinical Recognition: Taking a Present Illness by Computer.
American Journal of Medicine, 60:981-996 (1976).
66. Politakis P., Weiss S.M., 84
Using Empirical Analysis to Refine Expert System Knowledge Bases.
Artificial Intelligence 22 (1984) 23-48.
67. Pople, H., 82
Heuristics Methods for Imposing Structure on Ill-Structured Problems: the Structuring of Medical Diagnosis.
In: P. Szolovits (ed.), *Artificial Intelligence in Medicine*, Westview Press, Inc., Boulder, Colorado, 1982;Chapter 5.
68. Reiter R., 80 A Logic For Default Reasoning.
Artificial Intelligence 13 (1980) 81-132.
69. Sembugamoorthy, V., Chandrasekaran, B., 86
Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems.
In: J.L. Kolodner and C.L. Riesbeck (eds.), *Experience, Memory, and Reasoning*, Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, 1986;47-73.
70. Rastier, F., 87
Sur la sémantique des réseaux. *Quaderni di Semantica* VIII (1987) 115-131.
71. Reynaud, C., 89
Une aide à l'acquisition des connaissancesde surface à partir des connaissances profondes.

In: *Proceedings of the 9th International Workshop on Expert Systems and their Applications*, Avignon, Gerfau, Paris, 1989;207-222.

72. Roy, B., 70
Algèbre moderne et théorie des graphes, T1 & T2, Dunod, Paris, 1970.
73. Séroussi B., 86
 Computer-Aided Diagnosis of Acute Abdominal Pain when Taking into Account Interactions.
Methods of Information in Medicine, 25:194-198 (1986).
74. Shafer G. 76
A Mathematical Theory Of Evidence, Princeton University Press, , 1976.
75. Shortliffe E.H., 76
Computer-Based Medical Consultations: MYCIN, Elsevier, New York, 1976.
76. Sombé L., 89
 Raisonnements sur des informations incomplètes en intelligence artificielle. Comparaison de formalismes à partir d'un exemple. *Revue d'Intelligence Artificielle*, 2(3/4):9-210 (1989).
77. Steels L., 85
 Second Generation Expert Systems.
 In: *Future Generation Computer Systems*, North-Holland, Amsterdam, 1985.
78. Steels L., Van de Velde W., 85
 Learning in Second Generation Expert System.
 In: J. S. Kowalik (ed.), *Knowledge Based Problem Solving*, Prentice-Hall, Inc., New Jersey, 1985.
79. Steels L., 88
 The deepening of expert systems.
 In: R. Vichnevetsky, P. Borne and J. Vignes (eds.), *Proceedings of the 12th World Congress on Scientific Computation (IMACS)*, Paris, Gerfidn, Villeneuve d'Ascq, 1988;323-326.
80. Swartout W., 77
 A Digitalis therapy advisor with explanations.
 In: *Proceedings of the 5th IJCAI*, Cambridge, M. Kaufmann, Inc., Los Altos, California. 1977;819-825.
81. Szolovits P., Pauker S.G., 78
 Categorical and Probabilistic Reasoning in Medical Diagnosis.
Artificial Intelligence 11 (1978) 115-144.
82. Szolovits P., 82
 Artificial Intelligence and Medicine.
 In: P. Szolovits (ed.), *Artificial Intelligence in Medicine*, Westview Press, Inc., Boulder, Colorado, 1982;Chapter 1.

83. Szolovits P., Kassirer J.P., Long W.J., Moskowitz A.J., Pauker S.G., Patil R.S., Wellman M.P., 86
An Artificial Intelligence Approach to Clinical Decision Making.
Research Report, MIT AI Lab., Cambridge, Massachusetts, 1985.
84. Szolovits P., Patil R.S., Schwartz W.B., 88
Artificial Intelligence in Medical Diagnostic.
Annals of Internal Medicine 108 (1988) 80-87.
85. Touretzki D.S., 84
Implicit ordering of defaults in inheritance systems.
In: *Proceedings of the 4th AAAI, Austin, Texas*, M. Kaufmann, Inc., Los Altos, California, 1984;322-325.
86. Van de Velde W., 86
Learning Heuristics in Second Generation Expert Systems.
In: *Proceedings of the 6th International Workshop on Expert Systems and their Applications, Avignon*, Gerfau, Paris, 1986;883-899.
87. Weiss S.M., Kulikowski C.A., 82
A Practical Guide to Desining Expert Systems, Rowman & Allanheld Publishers, Totowa, New Jersey, 1984.
88. Zadeh L.A., 78
Fuzzy sets as a basis for a theory of possibility.
Fuzzy Sets and Systems, 1(1):3-28 (1978).

Annexe I

Calcul des exclusions et des négations

Dans cette annexe, nous précisons la manière dont les relations d'*exclusion* et de *négation* introduites au chapitre 3 sont obtenues par le système LÉZARD. Nous nous appuyerons sur l'exemple volontairement simplifié déjà introduit en 3.2 ch.1 (cf. figure I.1).

Reprenons rapidement le but de ce calcul : sur la base de l'"hypothèse du processus unique" le système interprète le modèle et propose un certain nombre d'*exclusions mutuelles* ou au contraire de *compatibilités logiques* entre diagnostics. Le but final étant dual, (1) générer des règles d'*exclusions mutuelles* cohérentes avec le modèle physiopathologique, (2) contrôler ce modèle. Les règles de *compatibilité* servent d'étape intermédiaire et permettent certaines validations en interaction avec l'utilisateur. Les relations mises en jeu au niveau du modèle sont les taxonomies (*locaconstituent*, *locaconstituent-n*, *processconstituent*, *processconstituent-n*) et les relations causales (*peut-causer*, *peut-causer-n*). Elles forment le réseau sémantique du domaine ; ce sont sur elles qu'est basé le calcul des *exclusions*. Les relations sémiologiques d'évocation (*evoque*, *evoquent*) sont, de plus, mises en jeu dans la génération des relations de *négation*. Les relations générées vont être alors :

1. Des relations de *compatibilité logique* comme :

APPENDICITE et OCCLUSION *sont-compatibles*

2. Des relations d'*exclusions mutuelles* décrites par la primitive *sexcluent* comme :

ATTEINTE-DIGESTIVE et ATTEINTE-URINAIRE *sexcluent*

3. Des relations d'*exclusions mutuelles conditionnelles* comme :

ATTEINTE-VASCULAIRE et OCCLUSION-ORGANIQUE *sexcluent-en-l-absence-de* INFARCTUS-MESENTERE

Des relations de *négation* vont être par ailleurs générées :

4. Des relation de *négation* comme :

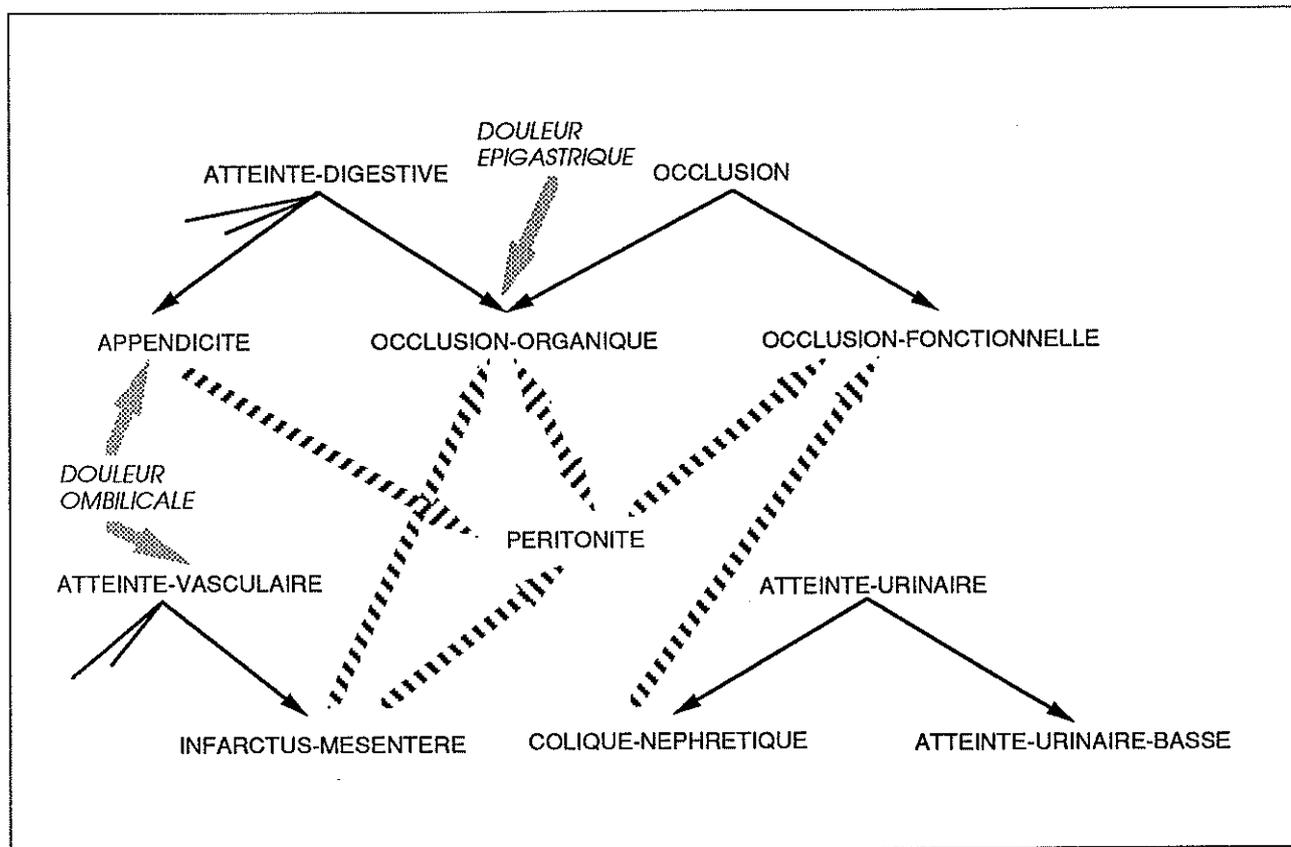


Figure I.1: Représentation du réseau sémantique. Les flèches épaisses interrompues représentent les liens causaux, les grisées les relations d'évocation, les noires les liens taxonomiques.

DOULEUR-OMBILICALE *nie* ATTEINTE-URINAIRE

5. Des relation de *négation conditionnelle* comme :

DOULEUR-EPIGASTRIQUE *nie* ATTEINTE-VASCULAIRE
en-l-absence-de INFARCTUS-MESENTERE

Remarquons que ces relations sont générées à partir du réseau vu d'un point de vue statique, avant toute inférence : aucune supposition n'est faite sur la possibilité d'assigner une valeur de vérité *vrai* ou *faux* à l'objet de manière dynamique (durant le raisonnement).

1 Vocabulaire et définitions préliminaires

Avant d'expliciter le principe de calcul nous allons introduire et rappeler quelques définitions dans lesquelles nous noterons " $X_1 \text{ :::> } X_2$ " pour exprimer que " X_1 peut causer X_2 ". Rappelons que, comme nous l'avons vu au niveau de la figure 2.1 page 28, les relations taxonomiques (les relations causales aussi mais ce ne sont pas elles qui nous intéressent ici) définissent un diagnostic "père" et des diagnostics "fils".

Définitions concernant les relations taxonomiques

Définition préliminaire 1 *Un diagnostic X_2 est un "frère" d'un diagnostic X_1 s'ils ont le même père dans le cadre d'une taxonomie.*

Ainsi dans l'exemple de la figure I.1 : APPENDICITE est un frère de OCCLUSION-ORGANIQUE.

Définition préliminaire 2 *Un diagnostic X_2 est un "descendant" d'un diagnostic X_1 si X_2 est le fils de X_1 dans une relation taxonomique ou le fils d'un descendant de X_1 .*

Par exemple : APPENDICITE est un descendant de ATTEINTE-DIGESTIVE.

Définition préliminaire 3 *Un diagnostic X_1 est un "ascendant" d'un diagnostic X_2 si X_2 est un descendant de X_1 dans une taxonomie.*

Définition 4 *Deux diagnostics ont un "lien hiérarchique" si l'un est descendant de l'autre (cf. définition 3).*

Par exemple : ATTEINTE-URINAIRE et COLIQUE-NEPHRETIQUE ont un lien hiérarchique.

Définition 5 *Deux diagnostics X_1 et X_2 ont un "descendant commun" s'il n'ont pas de lien hiérarchique et qu'il existe un troisième diagnostic Y qui est descendant de X_1 et de X_2 .*

Par exemple : ATTEINTE-DIGESTIVE et OCCLUSION ont un descendant commun qui est OCCLUSION-ORGANIQUE.

Définitions concernant les principales compatibilités

Définition 6 *Un chemin causal entre des diagnostics X_1 et X_2 est un chemin de relations causales — décrites par les primitives peut-causer ou peut-causer-n — orienté de X_1 vers X_2 . Ce chemin peut être réduit à une seule relation causale entre X_1 et X_2 .*

Par exemple, dans le cadre du réseau sémantique donné dans la figure I.1, il y a un chemin causal réduit à une seule relation causale entre INFARCTUS-MESENTERE et PERITONITE :

INFARCTUS-MESENTERE :::> PERITONITE.

De même, entre APPENDICITE et OCCLUSION-FONCTIONNELLE il y a un chemin causal :

APPENDICITE :::> PERITONITE :::> OCCLUSION-FONCTIONNELLE

Définition annexe 7 *Une chemin causal entre des diagnostics X_1 et X_2 est "coupé" quand un diagnostic intermédiaire (ni X_1 , ni X_2) du chemin devient faux (un chemin causal réduit à une seule relation causale ne peut donc être coupé).*

Ainsi dans l'exemple précédent, quand PERITONITE devient faux.

Définition 8 *Deux diagnostics X_1 et X_2 ont une cause commune Y quand il y a un chemin causal entre Y et X_1 et entre Y et X_2 .*

Par exemple, dans le cadre du réseau sémantique donné dans la figure I.1 : PERITONITE est une cause commune à OCCLUSION-ORGANIQUE et OCCLUSION-FONCTIONNELLE :

OCCLUSION-ORGANIQUE <::: PERITONITE :::> OCCLUSION-FONCTIONNELLE

Définition annexe 9 *Deux diagnostics X_1 et X_2 n'ont plus de cause commune Y quand, ou bien cette cause devient fautive, ou bien le chemin causal entre Y et X_1 est coupé, ou bien le chemin causal entre Y et X_2 est coupé.*

Ainsi dans l'exemple précédent, quand PERITONITE devient faux.

Définition 10 *Deux diagnostics X_1 et X_2 ont une "relation de compatibilité causale" s'ils satisfont un des critères suivants :*

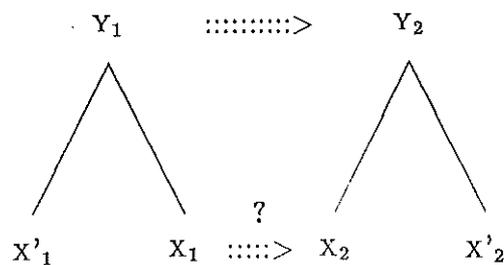
- X_1 et X_2 ou X_2 et X_1 sont reliés par un chemin causal (cf. définition 6)

- X_1 et X_2 ont une cause commune (cf. définition 8)
- au moins un descendant de X_1 et un descendant de X_2 ont une relation de compatibilité causale
- au moins un ascendant de X_1 et un ascendant de X_2 ont une relation de compatibilité causale

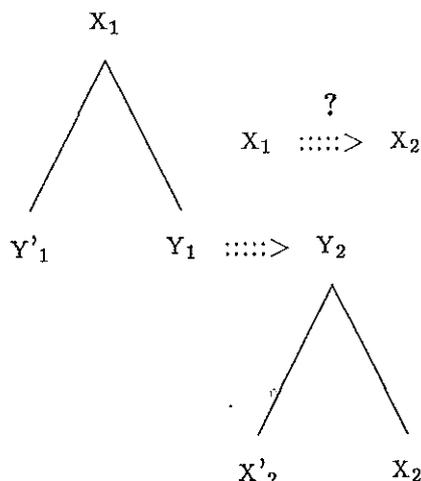
Une relation de compatibilité causale entre X_1 et X_2 est alors la succession des diagnostics qui vont de X_1 à X_2 via les relations causales et les filiations des relations taxonomiques. La relation de compatibilité causale n'est pas orientée, elle décrit simplement toutes les possibilités de "passer" de X_1 à X_2 : chacune d'elles est un "lien causal".

Par exemple, ATTEINTE-VASCULAIRE et ATTEINTE-DIGESTIVE ont 2 liens causaux, (ATTEINTE-VASCULAIRE, INFARCTUS-MESENTERE, OCCLUSION-ORGANIQUE, ATTEINTE-DIGESTIVE) et (ATTEINTE-VASCULAIRE, INFARCTUS-MESENTERE, PERITONITE, OCCLUSION-ORGANIQUE, ATTEINTE-DIGESTIVE).

Le dernier critère de la définition 10 sur les ascendants est trop contraignant dans la mesure où une relation causale entre des ascendants n'implique pas une relation de compatibilité causale entre X_1 et X_2 . En effet, si X_1 ou X_2 ont des frères, respectivement X'_1 et X'_2 , la relation de compatibilité causale peut très bien se reporter sur ces derniers (cf. figure ci-dessous). Mais il y a entre X_1 et X_2 , comme entre X'_1 et X'_2 , une relation de compatibilité causale possible et nous préférons supposer un trop grand nombre de compatibilités que de générer des *exclusions* abusives, quitte à demander confirmation à l'utilisateur sur leur validité.



De même, on aurait pu imaginer pour compléter la définition 10 un critère comme "un ascendant ou un descendant de X_1 et un ascendant ou un descendant de X_2 ont une relation de compatibilité causale". Mais, encore une fois, il vaut mieux être trop contraignant sans compter que la mise en œuvre de ce critère aurait posé des problèmes de temps calcul.



Définition annexe 11 Un lien causal entre deux diagnostics X_1 et X_2 est "rompu" quand un des diagnostics du lien, autre que X_1 ou X_2 , devient faux.

Une relation de compatibilité causale entre deux diagnostics X_1 et X_2 est "annulée" quand tous les liens causaux reliant X_1 et X_2 sont rompus.

2 Le calcul des exclusions

2.1 Définition des compatibilités et des exclusions

Définition 12 Deux diagnostics X et Y sont "trivialement logiquement compatibles" s'ils satisfont un des critères suivants :

- X et Y sont reliés par au moins un chemin causal réduit à une seule relation causale ($X \text{ } \text{:::} > \text{ } Y$ ou $Y \text{ } \text{:::} > \text{ } X$),
- deux de leurs descendants respectifs sont reliés par au moins un chemin causal réduit à une seule relation causale,
- X et Y ont un lien hiérarchique (cf. définition 4),
- X et Y ont un descendant commun (cf. définition 5).

La définition suivante sur les diagnostics *non-trivialement logiquement compatibles* se différencie de la première sur les diagnostics *trivialement logiquement compatibles* dans la mesure où dans le premier cas on définit des *compatibilités logiques* évidentes, apparaissant clairement au moment de la description du modèle physiopathologique alors que dans le second cas on définit des *compatibilités logiques* moins directes, sous-entendues dans le modèle et qu'il est bon d'exprimer pour contrôler ce dernier via l'expert.

Définition 13 Deux diagnostics X et Y sont *non-trivialement logiquement compatibles* s'ils satisfont les deux critères suivants :

- X et Y ne sont pas trivialement logiquement compatibles (cf. définition 12).
- X et Y ont au moins une relation de compatibilité causale (cf. définition 10).

Par exemple, deux diagnostics qui ne sont pas *trivialement logiquement compatibles* mais qui ont une cause commune sont *non-trivialement logiquement compatibles*.

Définition 14 Deux diagnostics X et Y sont *logiquement incompatibles* s'ils satisfont les 3 critères suivants :

- X et Y n'ont pas de lien hiérarchique (cf. définition 4),
- X et Y n'ont pas de descendant commun (cf. définition 5),
- X et Y n'ont pas de relation de compatibilité causale (cf. définition 10).

Les deux diagnostics X et Y sont aussi dits "exclusifs".

Définition 15 Deux diagnostics X et Y sont *conditionnellement logiquement incompatibles* si d'une part :

- X et Y n'ont pas de lien hiérarchique (cf. définition 3),
- X et Y n'ont pas de descendant commun (cf. définition 4) ;

et d'autre part :

- X et Y ont une relation de compatibilité causale (cf. définition 10),
- et cette relation de compatibilité causale peut être annulée (cf. définition 11).

Les diagnostics qui sont à la base de l'annulation de la relation de compatibilité causale donc de la rupture simultanée des liens causaux sont les conditions d'incompatibilité logique conditionnelle. Les deux diagnostics X et Y sont aussi dits *conditionnellement exclusifs*.

Le dernier critère sur la rupture simultanée des liens causaux est statique dans la mesure où on ne s'occupe que de la topologie du modèle pour vérifier si ces liens peuvent être simultanément rompus (en particulier, si une relation de compatibilité causale est réduit à une seule relation causale il ne peut être coupé et les 2 diagnostics ne peuvent être *conditionnellement logiquement*

incompatibles). Aucune supposition n'est faite quant à la possibilité dynamique (durant le raisonnement) que les liens puissent être simultanément rompus par négations des diagnostics mis en jeu. Les *exclusions mutuelles conditionnelles* générant des questions, ce sera à l'expert d'évaluer ces relations et leur utilité et de les accepter ou non.

Notons que pour les raisons d'utilité développées en 3.3.2 les *exclusions mutuelles conditionnelles* générées ne le sont que pour un nombre n de liens causaux ≤ 2 . En effet, plus n est grand plus les conditions d'*exclusions conditionnelles* à satisfaire sont nombreuses. Les règles correspondantes à ces exclusions deviennent rapidement irrecevables en raison de leur trop grand nombre de prémisses.

2.2 Rapports entre compatibilité et incompatibilité

On peut se poser la question des rapports entre les *compatibilités logiques* et les *incompatibilités logiques* en particulier conditionnelles. Montrons tout d'abord que les *compatibilités logiques* et les *incompatibilités logiques* appliquées à l'ensemble des objets du réseau sémantique pris deux à deux forment une partition de ce dernier. A partir des définitions 12, 13, 14 et 15, on définit les prédicats binaires A, A', B, C, P et Q :

$A(X,Y) \equiv X$ et Y sont reliés par au moins un chemin causal réduit à une seule relation causale

$A'(X,Y) \equiv X$ et Y sont reliés par des chemins causaux dont aucun n'est réduit à une seule relation causale $\equiv \neg A(X,Y)$

$B(X,Y) \equiv$ deux des descendants respectifs de X et Y sont reliés par au moins un chemin causal réduit à une seule relation causale $\equiv \exists X'$ fils de X et $\exists Y'$ fils de Y tel que $A(X',Y')$

$P(X,Y) \equiv X$ et Y ont un lien hiérarchique

$Q(X,Y) \equiv X$ et Y ont un descendant commun

$C(X,Y) \equiv X$ et Y ont une relation de compatibilité causale

Soit \mathcal{E} l'ensemble des couples d'objets.

- Soit \mathcal{E}_1 l'ensemble des couples d'objets vérifiant la *compatibilité logique triviale* :
 $(X,Y) \in \mathcal{E}_1 \equiv A(X,Y) \vee B(X,Y) \vee P(X,Y) \vee Q(X,Y)$ (1).
- Soit \mathcal{E}_2 l'ensemble des couples d'objets vérifiant la *compatibilité logique non triviale* ;
 $(X,Y) \in \mathcal{E}_2 \equiv \neg(A(X,Y) \vee B(X,Y) \vee P(X,Y) \vee Q(X,Y)) \wedge C(X,Y)$ (2)
- Soit \mathcal{E}_3 l'ensemble des couples d'objets vérifiant l'*incompatibilité logique* ;
 $(X,Y) \in \mathcal{E}_3 \equiv \neg P(X,Y) \wedge \neg Q(X,Y) \wedge \neg C(X,Y)$ (3).
- Soit \mathcal{E}_4 l'ensemble des couples d'objets vérifiant l'*incompatibilité logique conditionnelle* ;
 $(X,Y) \in \mathcal{E}_4 \equiv \neg P(X,Y) \wedge \neg Q(X,Y) \wedge C(X,Y) \wedge \neg A(X,Y)$ (4). la présence de $\neg A(X,Y)$

dans la formule (4) s'explique par le fait que, si X et Y sont reliés par des chemins causaux dont aucun n'est réduit à une seule relation causale, on peut rompre simultanément tous ces chemins

- de plus, par la définition 10, $A(X,Y)$ implique $C(X,Y)$ et $B(X,Y)$ implique $C(X,Y)$.
Soit $\neg A(X,Y) \vee C(X,Y)$ (5) et $\neg B(X,Y) \vee C(X,Y)$ (6).

Pour simplifier l'écriture, nous omettrons par la suite les arguments X et Y.

On peut montrer que $\mathcal{E}_1 \cup \mathcal{E}_2$ est le complémentaire de \mathcal{E}_3 dans \mathcal{E} :

Les couples de $\mathcal{E}_1 \cup \mathcal{E}_2$ vérifient

$$(A \vee B \vee P \vee Q) \vee (\neg(A \vee B \vee P \vee Q) \wedge C) \equiv$$

$$(A \vee B \vee P \vee Q) \vee C \equiv$$

$$A \vee B \vee P \vee Q \vee C$$

avec $(\neg A \vee C)$ et $(\neg B \vee C)$ on trouve finalement $P \vee Q \vee C$

Les couples de \mathcal{E}_3 vérifient $\neg P \wedge \neg Q \wedge \neg C$

qui est la négation de la formule précédente. CQFD

Or par définition \mathcal{E}_1 et \mathcal{E}_2 sont disjoints, donc $\mathcal{E}_1 \mathcal{E}_2 \mathcal{E}_3$ forment une partition de \mathcal{E} . On peut montrer aussi que les couples qui vérifient la *compatibilité logique non triviale* vérifient aussi l'*incompatibilité logique conditionnelle*.

Il faut donc vérifier que (2) implique (4), c'est à dire que

$\neg(\neg(A \vee B \vee P \vee Q) \wedge C) \vee (\neg P \wedge \neg Q \wedge C \wedge \neg A)$ est vrai. Or

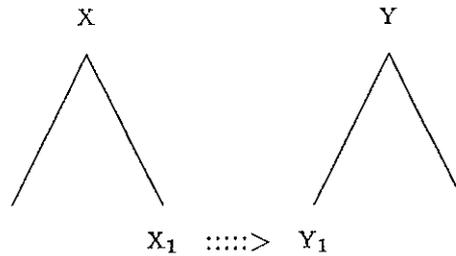
$$\neg(\neg(A \vee B \vee P \vee Q) \wedge C) \vee (\neg P \wedge \neg Q \wedge C \wedge \neg A) \equiv$$

$$\neg(\neg A \wedge \neg B \wedge \neg P \wedge \neg Q \wedge C) \vee (\neg P \wedge \neg Q \wedge C \wedge \neg A) \equiv$$

$$B \vee \neg(\neg P \wedge \neg Q \wedge C \wedge \neg A) \vee (\neg P \wedge \neg Q \wedge C \wedge \neg A) \text{ qui est toujours vrai. CQFD.}$$

Donc $\mathcal{E}_2 \subset \mathcal{E}_4$. En revanche on peut montrer facilement que les couples qui vérifient la *compatibilité logique triviale* ne vérifient pas toujours l'*incompatibilité logique conditionnelle*.

Il suffit de trouver d'abord un couple qui vérifie (1) et (4) :



(X, Y) vérifie B donc $\in \mathcal{E}_1$;

(X, Y) vérifie B donc C et de manière évidente vérifie $\neg P$, $\neg Q$ et $\neg A$ donc $\in \mathcal{E}_4$.

Mais on peut trouver un contre-exemple où un couple (X, Y) vérifie (1) mais pas (4) :

Soit (X, Y) impliqués dans une relation causale $(X \text{ :::> } Y)$.

Ce couple vérifie A, donc $\in \mathcal{E}_1$ mais $\notin \mathcal{E}_4$. CQFD.

Donc $\mathcal{E}_1 \not\subset \mathcal{E}_4$ mais $\exists (X, Y) \in \mathcal{E}_1$ tels que $(X, Y) \in \mathcal{E}_4$ donc $\mathcal{E}_1 \cap \mathcal{E}_4 \neq \emptyset$. Or $\mathcal{E}_2 \subset \mathcal{E}_4$ donc $\mathcal{E}_2 \not\subset \mathcal{E}_4$.
On peut voir sur la figure I.2 le résumé des résultats obtenus.

Le principal intérêt de ces développements est de montrer qu'on ne peut pas faire les recherches des *compatibilités* et *incompatibilités* dans n'importe quel ordre. En effet, si on veut générer des règles exactes d'*exclusion* en nombre restreint et en posant le minimum de questions à l'expert il faut tenir compte des résultats obtenus :

- en générant des *exclusions* à partir des *compatibilités logiques non triviales* (donc en modifiant l'ensemble des couples vérifiant la *compatibilité logique*) on modifie directement l'ensemble des couples vérifiant l'*exclusions mutuelles conditionnelles*. Cela signifie qu'il faut générer en priorité les *compatibilités logiques* et les *exclusions* quelles peuvent amener afin de pouvoir les répercuter dans le calcul des *incompatibilités logiques conditionnelles*.
- Par ailleurs, si on a généré des *exclusions* via les *compatibilités logiques* il faut vérifier quelle est leur interaction avec les *exclusions mutuelles*. Ici encore, il faut s'occuper en priorité de la génération des *compatibilités logiques* et des *exclusions* qu'elles peuvent amener afin de pouvoir les répercuter dans le calcul des *exclusions mutuelles*.

Finalement, l'ordre des calculs est le suivant : calcul des *compatibilités logiques* et exploitation des résultats, puis calcul des *incompatibilités logiques* et des *incompatibilités logiques conditionnelles* ; ces deux derniers calculs pouvant être conduits simultanément. Nous verrons précisément en 2.5 comment a lieu l'exploitation des *compatibilités logiques*, en interaction avec l'expert.

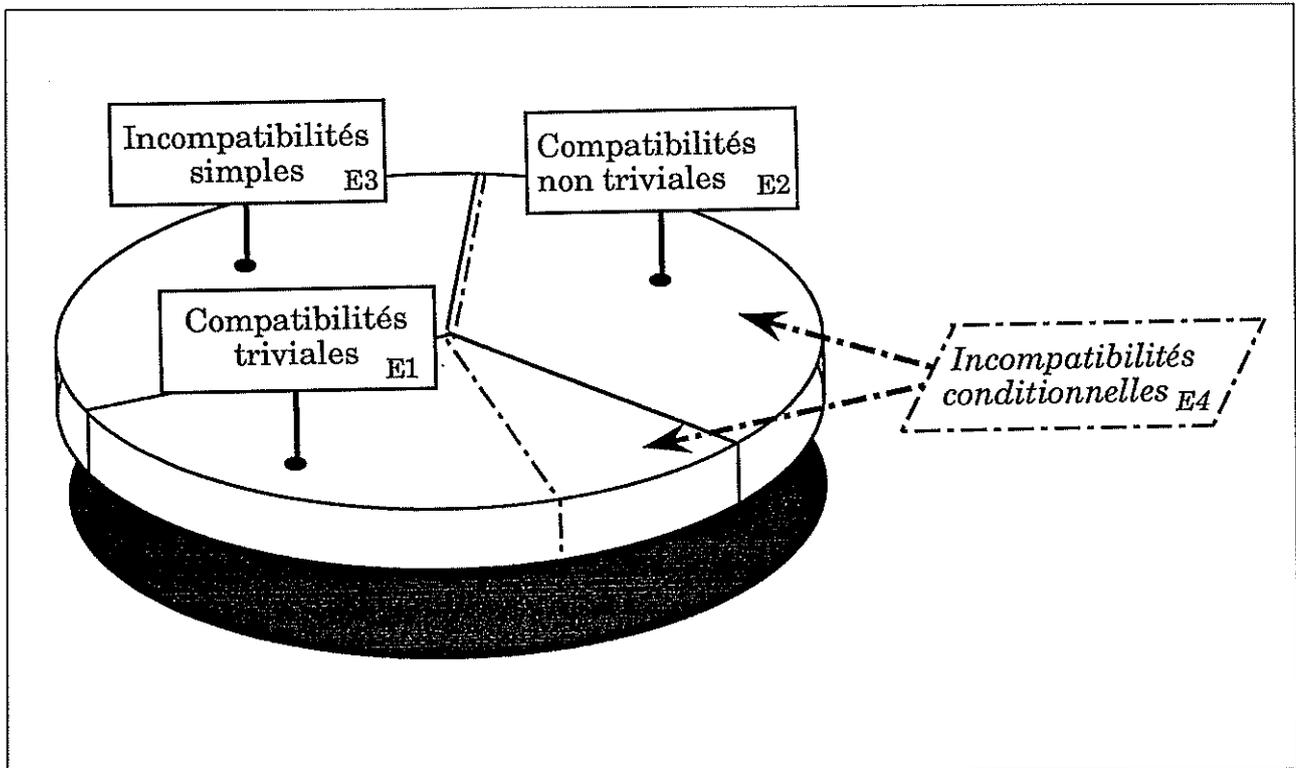


Figure I.2: Rapport entre les compatibilités et les incompatibilités.

2.3 Les algorithmes utilisés

Nous allons ici donner une description sommaire de l'algorithme utilisé pour la recherche des *exclusions mutuelles*. Nous n'étudierons ni le cas des *exclusions mutuelles conditionnelles* qui n'en n'est qu'une extension, ni celui des *compatibilités logiques* dont l'algorithme est analogue.

Une caractéristique importante de l'algorithme est de générer les relations d'*exclusion* au plus haut niveau d'abstraction possible ; le nombre de relations générées est en conséquence réduit au minimum. Par exemple la relation :

ATTEINTE-DIGESTIVE et ATTEINTE-URINAIRE *sexcluent*

est préférée aux deux relations :

ATTEINTE-DIGESTIVE et COLIQUE-NEPHRETIQUE *sexcluent*
 ATTEINTE-DIGESTIVE et ATTEINTE-URINAIRE-BASSE *sexcluent*.

L'algorithme de recherche des exclusions mutuelles

1. Classement du réseau sémantique correspondant au modèle pour que les taxonomies soient ordonnées des plus générales aux plus spécifiques.

Cet ordre est partiel. Le rangement présuppose de plus une certaine structure des taxonomies pour pouvoir fonctionner : ainsi il ne peut accepter des situations où deux objets (tous des diagnostics dans le réseau) sont à la fois père et fils l'un de l'autre et plus généralement les situations où deux objets sont reliés entre eux par des liens "familliaux" multiples comme "descendant et frère". En fait il est adapté à une structure de treillis (arborescence dans laquelle un fils peut avoir plusieurs pères) et on imagine mal une description hiérarchique d'un processus physiopathologique qui ne puisse s'adapter à cette contrainte.

2. Recherche des *exclusions* entre les objets frères dans le cadre d'une taxonomie.

Le classement précédent permet de commencer les recherches par les *exclusions* les plus générales. La découverte d'une *exclusion* à un niveau de généralité permet d'éviter la recherche d'*exclusions* plus spécifiques (des *exclusions* où les objets mis en jeu sont respectivement fils des objets de l'*exclusion* plus générale). La recherche d'*exclusion* possible entre deux objets frères X_1 et X_2 se déroule comme suit :

(a) vérification des présupposés à la recherche :

- si X_1 et X_2 n'ont pas de lien hiérarchique (cf. définition 4)
- si X_1 et X_2 n'ont pas de fils commun (cf. définition 5)
- s'il n'y a pas d'*exclusion* plus générale déjà générée à ce pas-ci

(b) alors lancement de la recherche :

- s'il n'y a pas de relation de compatibilité causale entre X_1 et X_2 alors on génère la relation d'*exclusion* entre X_1 et X_2 .

Cette première partie de l'algorithme s'est intéressée aux *exclusions* les plus intuitives et les plus fréquentes : celles entre objets frères dans une taxonomie ; elles sont plus faciles à appréhender car elles portent sur des objets de même niveau d'abstraction ; ce sont celles-là que donne effectivement l'expert quand on les lui demande dans un système comme SAM.

Il n'empêche que pour être efficace, le système a intérêt à répertorier toutes les *exclusions*. Dans la suite de l'algorithme on va donc chercher les *exclusions* entre objets non frères, en vérifiant évidemment que parmi les *exclusions* déjà générées entre objets frères il n'y en a pas de plus générales que celles que l'on cherche à générer.

3. Classement des objets du réseau du plus général au plus spécifique.

Ce rangement est du même type que celui développé au pas 1 mais ce sont ici les objets et non les taxonomies qu'il s'agit de ranger. Cet ordre est évidemment partiel mais ce qui

importe c'est qu'un père ne soit pas pris en compte avant un fils dans le processus de recherche suivant.

4. Recherche des *exclusions* entre les objets ainsi classés pris deux à deux (pour peu qu'ils ne soient pas frères). Cette recherche a lieu des objets les plus généraux aux objets les plus spécifiques pour réduire les recherches inutiles. La recherche d'*exclusion* possible entre deux objets X_1 et X_2 se déroule comme suit :

(a) vérification des présupposés à la recherche :

- si X_1 et X_2 ne sont pas frères (cf. définition 1)
- s'il n'y a pas d'*exclusion* plus générale déjà découverte lors de la recherche faite au pas 2
- si X_1 et X_2 n'ont pas de lien hiérarchique (cf. définition 4)
- si X_1 et X_2 n'ont pas de fils commun (cf. définition 5)
- s'il n'y a pas d'*exclusion* plus générale déjà générée parmi les couples d'objets non frères explorés précédemment à ce pas-ci.

(b) alors lancement de la recherche :

- s'il n'y a pas de relation de compatibilité causale entre X_1 et X_2 alors on génère la relation d'*exclusion* entre X_1 et X_2 .

5. Elimination d'*exclusions* spécifiques lorsque des *exclusions* plus générales existent. Ce cas peut se produire puisque les phases précédentes de l'algorithme ne vérifient pas qu'une exclusion entre deux objets frères n'est pas plus spécifique qu'une exclusion entre deux objets non frères.

Cet algorithme est en $O(n^2)$.

2.4 Exemple de résultats

Les *exclusions mutuelles* générées selon l'algorithme précédent sur l'exemple de la figure I.1 sont :

ATTEINTE-DIGESTIVE et ATTEINTE-URINAIRE *sexcluent*
 ATTEINTE-VASCULAIRE et ATTEINTE-URINAIRE *sexcluent*
 OCCLUSION et ATTEINTE-URINAIRE-BASSE *sexcluent*
 COLIQUE-NEPHRETIQUE et ATTEINTE-URINAIRE-BASSE *sexcluent*
 OCCLUSION-ORGANIQUE et ATTEINTE-URINAIRE *sexcluent*
 PERITONITE et ATTEINTE-URINAIRE *sexcluent*
 ATTEINTE-VASCULAIRE et APPENDICITE *sexcluent*.

On retrouve des *exclusions* entre frères comme "COLIQUE-NEPHRETIQUE et ATTEINTE-URINAIRE-BASSE *sexcluent*" ou entre non frères comme "PERITONITE et ATTEINTE-URINAIRE *sexcluent*".

Les *exclusions mutuelles conditionnelles* générées sont :

APPENDICITE et OCCLUSION *sexcluent-en-l-absence-de* PERITONITE
 ATTEINTE-VASCULAIRE et OCCLUSION-ORGANIQUE *sexcluent-en-l-absence-de* INFARCTUS-MESENTERE
 ATTEINTE-VASCULAIRE et OCCLUSION-FONCTIONNELLE *sexcluent-en-l-absence-de* PERITONITE ou INFARCTUS-MESENTERE.
 ATTEINTE-VASCULAIRE et PÉRITONITE *sexcluent-en-l-absence-de* INFARCTUS-MESENTERE.

Notons que "ATTEINTE-VASCULAIRE et OCCLUSION *sexcluent-en-l-absence-de...*" n'est pas générée car elle correspond à la rupture de 3 chemins causaux :

- (ATTEINTE-VASCULAIRE, INFARCTUS-MESENTERE, OCCLUSION-ORGANIQUE, OCCLUSION)
- (ATTEINTE-VASCULAIRE, INFARCTUS-MESENTERE, PERITONITE, OCCLUSION-ORGANIQUE, OCCLUSION)
- (ATTEINTE-VASCULAIRE, INFARCTUS-MESENTERE, PERITONITE, OCCLUSION-FONCTIONNELLE, OCCLUSION)

Enfin on obtient les relations de *compatibilité logique non-triviale* suivantes :

APPENDICITE et OCCLUSION sont compatibles
 ATTEINTE-VASCULAIRE et OCCLUSION-FONCTIONNELLE sont compatibles
 OCCLUSION-ORGANIQUE et OCCLUSION-FONCTIONNELLE sont compatibles
 ATTEINTE-DIGESTIVE et OCCLUSION-FONCTIONNELLE sont compatibles.

2.5 Interactions avec l'expert

Nous avons déjà abordé en 3.2.4 et 3.3.2 ch.3 les problèmes d'utilité des *exclusions* générées. Nous allons ici nous attacher aux *compatibilités logiques* et essayer d'illustrer notre propos sur l'exemple de la figure I.1.

Nous avons vu que les *compatibilités logiques* permettent à la fois un contrôle du modèle physiopathologique et la génération de nouvelles connaissances.

Elles permettent de contrôler le modèle physiopathologique dans la mesure où elles génèrent des relations de compatibilité entre les diagnostics cohérents vis à vis du modèle. Si une de ces relations apparaît fautive à l'expert cela peut être ponctuel mais aussi remettre en cause le

modèle lui-même. Dans ce dernier cas il faut évidemment arrêter le processus de traduction des *connaissances profondes* et modifier les connaissances.

Elles permettent aussi de générer des exclusions qui échappent à l'algorithme en raison d'une extension abusive des relations de causalité : en effet, dans l'hypothèse du processus unique (cf. 3.2.1 ch.3) ce sont les relations de cause à effet qui entrent en jeu et 2 objets reliés par une relation de cause à effet peuvent faire partie du même processus physiopathologique. Par extension, 2 objets reliés par un chemin causal (cf. définition 6) peuvent faire partie du même processus physiopathologique, et cette hypothèse est appliquée dans les algorithmes de calcul. Mais elle peut être fautive. Par exemple si on a "A peut-causer B" et "B peut-causer C" on peut ou non avoir "A peut-causer C".

Ainsi, dans notre réseau simplifié, on a "OCCLUSION-ORGANIQUE peut-causer PERITONITE" et "PERITONITE peut-causer OCCLUSION-FONCTIONNELLE", mais la relation "OCCLUSION-ORGANIQUE peut-causer OCCLUSION-FONCTIONNELLE" est inexacte, comme l'indique d'ailleurs la sémantique des termes utilisés. Mais cette information est inconnue au niveau de l'algorithme. Cela nécessite une question à l'expert. Le système va alors proposer la *compatibilité logique* à l'expert qui la refusera. Moyennant quoi le système générera l'*exclusion* entre l'OCCLUSION-ORGANIQUE et l'OCCLUSION-FONCTIONNELLE, et il faudra en tenir compte par la suite.

3 Le calcul des négations

3.1 Définition des négations

Définition 16 *Un diagnostic est nié par un signe s'il est logiquement incompatible avec tous les diagnostics évoqués par ce signe (cf. définition 14).*

Définition 17 *Un diagnostic est nié par un signe conditionnellement à l'absence de certains diagnostics s'il y a incompatibilité logique conditionnelle avec des diagnostics évoqués par ce signe (cf. définition 15) et incompatibilité logique avec les autres diagnostics évoqués par le signe s'il en reste (cf. définition 14). Les diagnostics qui sont à la base de l'annulation de la relation de compatibilité causale sont les conditions de négation.*

Notons que pour les raisons d'utilité déjà développées les *négations conditionnelles* générées ne le sont que si les liens causaux mis en cause dans la relation de compatibilité causale sont au plus au nombre de 2. En effet, au-dessus de ce nombre on obtient des relations générant des règles irrecevables car ayant trop de prémisses.

3.2 Exemple de résultat

Considérons le réseau de la figure I.1. Des *négations* vont être générées à partir des deux relations d'évocation suivantes :

DOULEUR-EPIGASTRIQUE *evoque* OCCLUSION-ORGANIQUE
DOULEUR-OMBILICALE *evoque* ATTEINTE-VASCULAIRE ou APPENDICITE

Les *négations* générées sont :

DOULEUR-EPIGASTRIQUE *nie* ATTEINTE-URINAIRE et
OCCLUSION-FONCTIONNELLE

car "OCCLUSION-ORGANIQUE et ATTEINTE-URINAIRE *sexcluent*"
et "OCCLUSION-ORGANIQUE et OCCLUSION-FONCTIONNELLE *sex-*
cluent"

DOULEUR-OMBILICALE *nie* ATTEINTE-URINAIRE

car "ATTEINTE-VASCULAIRE et ATTEINTE-URINAIRE *sexcluent*"
et "ATTEINTE-DIGESTIVE et ATTEINTE-URINAIRE *sexcluent*".

Les *négations conditionnelles* générées sont :

DOULEUR-EPIGASTRIQUE *nie* ATTEINTE-VASCULAIRE
en-l-absence-de INFARCTUS-MESENTERE

car ATTEINTE-VASCULAIRE et OCCLUSION-ORGANIQUE *sexcluent-en-l-*
absence-de INFARCTUS-MESENTERE

DOULEUR-EPIGASTRIQUE *nie* APPENDICITE
en-l-absence-de PERITONITE

car APPENDICITE et OCCLUSION *sexcluent-en-l-absence-de* PERITONITE

DOULEUR-OMBILICALE *nie* OCCLUSION-FONCTIONNELLE
en-l-absence-de PERITONITE

car APPENDICITE et OCCLUSION *sexcluent-en-l-absence-de* PERITONITE
et ATTEINTE-VASCULAIRE et OCCLUSION-FONCTIONNELLE *sexcluent-*
en-l-absence-de
PERITONITE ou INFARCTUS-MESENTERE).

“DOULEUR-OMBILICALE *nie* OCCLUSION-FONCTIONNELLE *en-l-absence-de...*” et “DOULEUR-
OMBILICALE *nie* OCCLUSION-ORGANIQUE *en-l-absence-de...*” n’ont pas été générées car elles
mettent en jeu plus de 2 chemins causaux.

Annexe II

La base des connaissances de LéZard

Nous donnons ici la base des connaissances de LÉZARD dans le domaine des douleurs aiguës de l'abdomen.

En premier lieu nous avons les relations décrivant le modèle physiopathologique :

DNS DS INFLAMMATION PROCESSCONSTITUENT PERE-ABSTRAIT ;
ATT-DIGESTIVE ATT-VASCULAIRE ATT-URINAIRE ATT-GYNECO
AFFECTION-POUMON LOCACONSTITUENT DS ;
PERITONITE OCCLUSION PROCESSCONSTITUENT-N DS ;
PERITONITE-PR PERITONITE-SE PROCESSCONSTITUENT PERITONITE ;
OCC-FONCTIONNELLE OCC-ORGANIQUE PROCESSCONSTITUENT OCCLUSION ;
OCC-ORGANIQUE-PR OCC-ORGANIQUE-SE PROCESSCONSTITUENT OCC-ORGANIQUE ;

ATT-INTESTIN CHOLECYSTITE ULCERE APPENDICITE ABCES-FOIE PANCREATITE
LOCACONSTITUENT ATT-DIGESTIVE ;
POUSSEE-ULCERE PERFO-ULCERE PROCESSCONSTITUENT ULCERE ;
CANCER-INT OCC-ORGANIQUE-PR HERNIE-ETRANGLEE SIGMOIDITE
LOCACONSTITUENT ATT-INTESTIN ;
ANEVRYSME-AORTE INFARCTUS-MESENTERE LOCACONSTITUENT ATT-VASCULAIRE ;
COLIQUE-NEPHRETIQUE INFECTION-URINAIRE LOCACONSTITUENT ATT-URINAIRE ;
SALPINGITE KYSTE-OVAIRE GEU FAUSSE-COUCHE
D-GYNECO-FONC LOCACONSTITUENT ATT-GYNECO ;

OCC-ORGANIQUE-SE OCC-ORGANIQUE-PR PERFO-ULCERE APPENDICITE CHOLECYSTITE
SIGMOIDITE GEU FAUSSE-COUCHE SALPINGITE INFARCTUS-MESENTERE
PEUT-CAUSER PERITONITE-SE ;

ANEVRYSME-AORTE COLIQUE-NEPHRETIQUE APPENDICITE PANCREATITE

SIGMOIDITE PERITONITE-SE PERITONITE-PR PEUT-CAUSER-N OCC-FONCTIONNELLE ;
 INFARCTUS-MESENTERE HERNIE-ETRANGLEE SIGMOIDITE PERITONITE CANCER-INT
 PEUT-CAUSER-N OCC-ORGANIQUE-SE ;

PERITONITE-SE AFFECTION-POUMON PANCREATITE APPENDICITE CHOLECYSTITE SIGMOI-
 DITE
 CANCER-INT SALPINGITE GEU FAUSSE-COUCHE KYSTE-OVAIRE D-GYNECO-FONC
 INFECTION-URINAIRE ABCES-FOIE ANEVRYSMO-AORTE INFARCTUS-MESENTERE PERITONITE-
 PR
 COLIQUE-NEPHRETIQUE ANEVRYSMO-AORTE DNS PEUT-CAUSER-N INFLAMMATION ;

Ensuite nous trouvons les relations sémiologiques :

SIGNES-URINAIRES SIGNALE BRULURES-MICTIONNELLES DYSURIE POLLAKIURIE HEMATURIE
 ASP-LITHIASE-AIRE-URINAIRE CALCUL-UIV UIV-DILATATION UIV-REIN-MUET
 IRRADIATION-DESCENDANTE IRRADIATION-POST-LOMBAIRE ;
 SIGNES-URINAIRES CONSTANT-DANS ATT-URINAIRE ;
 SIGNES-GYNECO SIGNALE PERTES-VAGINALES RETARD-REGLES GROSSESSE TV-MASSE
 TV-DOULEUR-DROITE TV-DOULEUR-GAUCHE TV-DOULEUR-DOUGLAS TV-SANG
 VOLUME-UTERIN-AUGMENTE DOULEUR-MOBILISATION-UTERUS PHASE-OVULATOIRE ;
 SIGNES-GYNECO CONSTANT-DANS ATT-GYNECO ;
 SIGNES-OCCLUSION SIGNALE ARRET-MATIERES ARRET-GAZ ;
 SIGNES-OCCLUSION CONSTANT-DANS OCCLUSION HERNIE-ETRANGLEE ;
 SIGNES-PERITONITE SIGNALE CONTRACTURE RESP-ABD-NON-VISIBLE TR-DOULEUR-DOUGLAS ;

SIGNES-PERITONITE CONSTANT-DANS PERITONITE ;
 SIGNES-CHOLECYSTITE SIGNALE D-PROV-LOC-HYPOC-D D-PROV-HAUTE MURPHY
 CHOL-IV-LITHIASE CHOL-IV-VESICULE-NON-VISIBLE CHOL-IV-VBP-NON-VISIBLE
 ASP-LITHIASE-AIRE-BILIAIRE ;
 SIGNES-CHOLECYSTITE CONSTANT-DANS CHOLECYSTITE ;
 SIGNES-VASCULAIRES SIGNALE MASSE-ABD-BATTANTE MASSE-ABD-NON-BATTANTE
 SOUFFLE-ABDOMINAL ;
 SIGNES-VASCULAIRES CONSTANT-DANS ANEVRYSMO-AORTE ;
 TEMPERATURE-ELEVEE SIGNALE $TEMP > 37.5 - \leq 38.5$ TEMPERATURE $> 38.5 - \leq 39.5$ TEM-
 PERATURE > 39.5 ;
 TEMPERATURE-ELEVEE CONSTANT-DANS INFLAMMATION INFECTION-URINAIRE SIGMOIDITE
 SALPINGITE ABCES-FOIE ;
 TEMPERATURE-ELEVEE EVOQUE INFLAMMATION ;
 TEMPERATURE > 38.5 SIGNALE TEMPERATURE $> 38.5 - \leq 39.5$ TEMPERATURE > 39.5 ;
 TEMPERATURE > 38.5 IMPLIQUE INFLAMMATION ;

SEXE-MASCULIN IMPLIQUENT NON ATT-GYNECO ;
 AGE<30 SIGNALE AGE<15 AGE>=15-<30 ;
 AGE<30 IMPLIQUE NON CANCER-INT ;

D-PROV-LOC-HYPOC-D EVOQUE

ABCES-FOIE CHOLECYSTITE CANCER-INT OCC-ORGANIQUE-PR ATT-URINAIRE ;
 D-PROV-LOC-HYPOC-G EVOQUE CANCER-INT OCC-ORGANIQUE-PR ATT-URINAIRE ;
 D-PROV-LOC-EPIGASTRIQUE EVOQUE

ABCES-FOIE ULCERE CHOLECYSTITE PANCREATITE CANCER-INT OCC-ORGANIQUE ;
 D-PROV-LOC-EPIGASTRIQUE CONSTANT-DANS ULCERE ;

D-PROV-HAUTE EVOQUE

ABCES-FOIE ULCERE CHOLECYSTITE PANCREATITE ATT-INTESTIN ATT-VASCULAIRE ;
 D-PROV-DROITE EVOQUE CANCER-INT OCC-ORGANIQUE APPENDICITE ATT-URINAIRE ;
 D-PROV-GAUCHE EVOQUE ATT-INTESTIN ATT-URINAIRE ;

D-PROV-LOC-FID EVOQUE APPENDICITE INFECTION-URINAIRE ATT-GYNECO CANCER-INT
 OCC-ORGANIQUE ATT-VASCULAIRE ;

D-PROV-LOC-FID ATT-DIGESTIVE IMPLIQUENT APPENDICITE ;

D-PROV-LOC-FIG EVOQUE

ATT-INTESTIN INFECTION-URINAIRE ATT-GYNECO ATT-VASCULAIRE ;

D-PROV-LOC-HYPOG EVOQUE APPENDICITE ATT-INTESTIN ATT-URINAIRE ATT-GYNECO ;

D-PROV-BASSE EVOQUE APPENDICITE ATT-INTESTIN ATT-URINAIRE ATT-GYNECO PERFO-
 ULCERE ;

D-PROV-DIFFUSE EVOQUE ATT-INTESTIN PANCREATITE ATT-VASCULAIRE PERITONITE DNS ;

D-PROV-LOC-OMBILICALE EVOQUE

ATT-INTESTIN PANCREATITE ATT-VASCULAIRE APPENDICITE ;

IRRADIATION-POST-LOMBAIRE EVOQUE

CHOLECYSTITE ATT-URINAIRE PANCREATITE ANEVRYSMES-AORTE ABCES-FOIE ;

IRRADIATION-SCAPULAIRE EVOQUE

CHOLECYSTITE ABCES-FOIE ATT-GYNECO AFFECTION-POUMON ;

IRRADIATION-DESCENDANTE EVOQUE ATT-URINAIRE ;

DEBUT-BRUTAL EVOQUE PANCREATITE PERITONITE ATT-VASCULAIRE COLIQUE-NEPHRETIQUE

GEU KYSTE-OVAIRE OCC-ORGANIQUE-PR ;

DEBUT-PROGRESSIF EVOQUE OCC-ORGANIQUE-PR APPENDICITE CHOLECYSTITE SIGMOIDITE

SALPINGITE CANCER-INT GEU FAUSSE-COUCHE KYSTE-OVAIRE DNS ;

EXACERBATION-MARCHE EVOQUE INFLAMMATION ;

EXACERBATION-INSPIRATION EVOQUE CHOLECYSTITE ABCES-FOIE ;

EXACERBATION-ALIMENTATION EVOQUE CHOLECYSTITE ABCES-FOIE PANCREATITE OCCLUSION ;

DS-SOULAGEE-ALIMENTATION EVOQUE POUSSEE-ULCERE ;
DS-SOULAGEE-ANTEFLEXION EVOQUE PANCREATITE ;

ARRET-MATIERES ARRET-GAZ EVOQUENT OCCLUSION ;

S-URINAIRES SIGNALE BRULURES-MICTIONNELLES DYSURIE POLLAKIURIE ;
S-URINAIRES EVOQUE ATT-URINAIRE ATT-GYNECO APPENDICITE SIGMOIDITE ;
HEMATURIE IMPLIQUENT ATT-URINAIRE ;

PERTES-VAGINALES EVOQUE ATT-GYNECO ;
RETARD-REGLES EVOQUE GEU ;
GROSSESSE EVOQUE GEU FAUSSE-COUCHE ;
GROSSESSE CONSTANT-DANS FAUSSE-COUCHE ;
GROSSESSE TV-MASSE IMPLIQUENT GEU ;
PERTES-VAGINALES TV-MASSE EVOQUENT SALPINGITE GEU KYSTE-OVAIRE ;
PHASE-OVULATOIRE EVOQUE D-GYNECO-FONC ;

ANT-DOULEURS-SEMBLABLES EVOQUE
APPENDICITE POUSSEE-ULCERE CHOLECYSTITE PANCREATITE D-GYNECO-FONC ;
ANT-CHIR-ABDOMINALE EVOQUE OCC-ORGANIQUE-PR ;

FACT-DECL-MEDICAMENTEUX EVOQUE ULCERE PANCREATITE ;
FACT-DECL-ALCOOLO-ALIM EVOQUE CHOLECYSTITE ULCERE PANCREATITE ;

HEMATOCRITE<35 SIGNALE HEMATOCRITE<15 HEMATOCRITE>=15-<=35 ;
HEMATOCRITE<15 EVOQUE GEU KYSTE-OVAIRE ATT-VASCULAIRE ;
LANGUE-SABURRALE EVOQUE INFLAMMATION ;
LANGUE-SABURRALE IMPLIQUENT NON ATT-GYNECO ;
LANGUE-SABURRALE IMPLIQUENT NON ATT-URINAIRE ;
LANGUE-SABURRALE IMPLIQUENT NON ATT-VASCULAIRE ;
LANGUE-SABURRALE IMPLIQUENT NON AFFECTION-POUMON ;

RESP-ABD-NON-VISIBLE EVOQUE PERITONITE OCCLUSION ;
ONDUL-PERISTALTQUES EVOQUE OCCLUSION ;

METEORISME SIGNALE METEORISME-LOCALISE METEORISME-DIFFUS ;
 METEORISME EVOQUE OCCLUSION ;
 DOULEUR-DECOMPRESSION EVOQUE APPENDICITE ;
 DEFENSE EVOQUE
 APPENDICITE SIGMOIDITE SALPINGITE CHOLECYSTITE PANCREATITE GEU PERITONITE ;
 DEFENSE D-PROV-LOC-FIG IMPLIQUENT SIGMOIDITE ;
 MURPHY EVOQUE CHOLECYSTITE ;
 CONTRACTURE EVOQUE PERITONITE ;

MASSE-ABD-NON-BATTANTE EVOQUE OCCLUSION CHOLECYSTITE CANCER-INT ;
 MASSE-ABD-BATTANTE EVOQUE ANEVRYSMES-AORTE ;

HERNIE-IRRREDUCTIBLE EVOQUE HERNIE-ETRANGLEE ;
 HERNIE-IRRREDUCTIBLE OCCLUSION IMPLIQUENT HERNIE-ETRANGLEE ;

CONTENU-GASTR-FECALOIDE IMPLIQUENT OCC-ORGANIQUE ;

TYMPANISME EVOQUE OCCLUSION ;

BHA-DIMINUES BHA-AUGMENTES EVOQUE OCCLUSION ;
 SOUFFLE-ABDOMINAL EVOQUE ATT-VASCULAIRE ;

TV-DOULEUR-DROITE TR-DOULEUR-DROITE
 EVOQUE KYSTE-OVAIRE GEU SALPINGITE APPENDICITE PERITONITE ;
 TV-DOULEUR-GAUCHE TR-DOULEUR-GAUCHE
 EVOQUE KYSTE-OVAIRE GEU SALPINGITE SIGMOIDITE PERITONITE ;
 TV-DOULEUR-DOUGLAS TR-DOULEUR-DOUGLAS EVOQUE SALPINGITE PERITONITE
 GEU KYSTE-OVAIRE FAUSSE-COUCHE ;
 TV-MASSE TR-MASSE EVOQUE SIGMOIDITE SALPINGITE KYSTE-OVAIRE GEU OCCLUSION ;
 TV-SANG EVOQUE FAUSSE-COUCHE KYSTE-OVAIRE GEU ;
 TR-SANG EVOQUE ATT-INTESTIN OCC-ORGANIQUE ;
 VOLUME-UTERIN-AUGMENTE IMPLIQUENT FAUSSE-COUCHE ;
 VOLUME-UTERIN-AUGMENTE CONSTANT-DANS FAUSSE-COUCHE ;
 DOULEUR-MOBILISATION-UTERUS CONSTANT-DANS FAUSSE-COUCHE ;
 DOULEUR-MOBILISATION-UTERUS EVOQUE SALPINGITE FAUSSE-COUCHE KYSTE-OVAIRE ;
 TV-SANG GROSSESSE DOULEUR-MOBILISATION-UTERUS IMPLIQUENT FAUSSE-COUCHE ;
 TV-DOULEUR-DROITE TV-DOULEUR-GAUCHE EVOQUENT SALPINGITE ;

LEUCOCYTES > 15000 IMPLIQUENT INFLAMMATION ;
 BILIRUBINE-AUGM EVOQUE CHOLECYSTITE ;
 TRANSAMINASES-AUGM EVOQUE ABCES-FOIE ;
 AMYLASE-AUGM EVOQUE PANCREATITE ;
 AMYLASE-AUGM SIGNALE HYPERAMYLASEMIE HYPERAMYLASURIE ;

NIVEAU-COLON EVOQUE OCC-ORGANIQUE ;
 AEROILEIE-IMPORTANTE AEROCOLIE-IMPORTANTE NIVEAU-GRELE EVOQUE OCCLUSION ;
 AEROILEIE-IMPORTANTE ANT-CHIR-ABDOMINALE EVOQUENT OCC-ORGANIQUE-PR ;
 NIVEAU-GRELE ANT-CHIR-ABDOMINALE EVOQUENT OCC-ORGANIQUE-PR ;
 ASP-GRISAILLE-DIFFUSE EVOQUE INFARCTUS-MESENTERE PERITONITE ;
 NON PERITONITE NON OCCLUSION IMPLIQUENT NON INFARCTUS-MESENTERE ;
 PNEUMOPERITOINE EVOQUE PERFO-ULCERE SIGMOIDITE PERITONITE ;
 PNEUMOPERITOINE D-PROV-LOC-EPIGASTRIQUE NON D-PROV-LOC-FIG NON D-PROV-BASSE
 IMPLIQUENT PERFO-ULCERE ;
 PNEUMOPERITOINE D-PROV-HAUTE NON D-PROV-LOC-FIG NON D-PROV-BASSE
 IMPLIQUENT PERFO-ULCERE ;
 PNEUMOPERITOINE D-PROV-LOC-FIG NON D-PROV-LOC-EPIGASTRIQUE NON D-PROV-HAUTE
 IMPLIQUENT SIGMOIDITE ;
 PNEUMOPERITOINE D-PROV-BASSE NON D-PROV-LOC-EPIGASTRIQUE NON D-PROV-HAUTE
 IMPLIQUENT SIGMOIDITE ;
 PERITONITE IMPLIQUE PNEUMOPERITOINE ;
 ASP-LITHIASE-AIRE-BILIAIRE EVOQUE CHOLECYSTITE ;
 ASP-LITHIASE-AIRE-URINAIRE EVOQUE ATT-URINAIRE ;

UIV-ANORMALE EVOQUE ATT-URINAIRE ;
 UIV-ANORMALE SIGNALE CALCUL-UIV UIV-DILATATION UIV-REIN-MUET ;
 CALCUL-UIV IMPLIQUENT COLIQUE-NEPHRETIQUE ;
 SIGNES-RADIO-C-N SIGNALE ASP-LITHIASE-AIRE-URINAIRE UIV-ANORMALE ;
 SIGNES-RADIO-C-N CONSTANT-DANS COLIQUE-NEPHRETIQUE ;

LB-ARRET-COLON IMPLIQUENT OCC-ORGANIQUE ;
 LB-TUMEUR IMPLIQUENT CANCER-INT ;

CHOL-IV-ANORMALE EVOQUE CHOLECYSTITE ;
 CHOL-IV-ANORMALE SIGNALE CHOL-IV-LITHIASE CHOL-IV-VESICULE-NON-VISIBLE
 CHOL-IV-VBP-NON-VISIBLE ;

RP-OPACITE-PARACHYMENTEUSE EVOQUE AFFECTION-POUMON ;
RP-OPACITE-PARACHYMENTEUSE IMPLIQUENT AFFECTION-POUMON ;
RP-EP-PLEURAL EVOQUE PANCREATITE AFFECTION-POUMON ABCES-FOIE ;

Table des matières

Remerciements	iii
Contenu de la thèse	v
Préface	1
1 Motivations	1
2 Lézard	2
3 Présentation des chapitres de la thèse	2
1 Introduction	5
1 Les approches de l'incertitude	6
1.1 Qualité de l'information	8
1.2 Incomplétude des informations	9
1.3 Fiabilité des inférences	10
2 Objectifs du projet	10
2.1 Caractéristiques du système	10
2.2 Choix du domaine	12
3 Le système Lézard.	12
3.1 Choix des approches	12
3.2 Structure du système Lézard	12
2 La représentation des connaissances	17
1 Problématique	17
2 Les connaissances médicales	19
3 Le langage de représentation des connaissances	21
3.1 Les restrictions du langage	21
3.2 La description d'un objet	22
3.2.1 Définition et valeurs de vérité d'un objet	22
3.2.2 Classe d'un objet	26
3.2.3 Fiabilité d'un objet	26
3.3 Les relations	27
3.3.1 La signification des relations	28

3.3.2	Fiabilité d'une relation versus fiabilité d'une règle	31
4	Les exceptions	32
4.1	Introduction	32
4.2	Donner une exception avec une règle shunt	33
4.3	Donner une exception à l'aide d'une règle contradictoire	34
4.4	Conséquences	34
5	Interprétation du modèle physiopathologique	34
3	Le traducteur	37
1	Les connaissances profondes et les connaissances de surface	37
1.1	Définitions	38
1.1.1	Les connaissances de surface	38
1.1.2	Les connaissances profondes	39
1.1.3	Relativité de ces deux concepts	40
2	Acquisition des connaissances	40
2.1	Acquisition des connaissances dans le système Lézard	41
2.2	Avantages et inconvénients de l'approche développée	42
3	Les mécanismes d'interprétation des connaissances	43
3.1	Les connaissances implicites ou explicites	43
3.1.1	Problématique dans le cadre de la dérivation formelle	43
3.1.2	Croyance implicite/Croyance explicite	44
3.2	La compatibilité/incompatibilité entre diagnostics	45
3.2.1	L'hypothèse du processus unique	46
3.2.2	Compatibilité et exclusions mutuelles	47
3.2.3	Exclusions et négations	49
3.2.4	Utilité et discussion	49
3.3	La généralisation des règles	50
3.3.1	Utilité des connaissances	50
3.3.2	Généralisation	50
3.4	L'interprétation des liens causaux	51
3.4.1	Transformation de la relation causale en implication	51
3.4.2	Abduction	52
4	Traduction ; dérivation formelle des relations	52
4.1	Dérivation de chaque primitive	55
4.2	Création de nouvelles primitives	59
5	Description d'un processus d'interprétation complet	59
6	Les méta-connaissances du traducteur	62
4	Le moteur d'inférences	65
1	La compilation des règles de production	65
1.1	Ordonnancer les règles de production	66

1.2	La recherche des boucles dans LéZard	69
1.3	Le calcul des propositions et le modus ponens dans LéZard	72
2	Le fonctionnement du moteur	73
2.1	Préliminaires	73
2.2	Entrée des données	73
2.3	Initialisation	74
2.4	Gestion de l'incertain	74
2.4.1	Gestion des hypothèses	74
2.4.1.a	Un raisonnement	75
2.4.1.b	Les justifications du raisonnement	76
2.4.2	Critères de choix du meilleur raisonnement	77
2.4.2.a	Utiliser les exceptions	77
2.4.2.b	Retenir les raisonnements les plus simples	79
2.4.2.c	Choisir le raisonnement le plus fiable	80
2.4.3	Résoudre les conflits : choix du meilleur raisonnement	81
2.5	Gestion des boucles	82
2.6	Dynamique du raisonnement en chaînage avant	84
3	Connaissances de contrôle	85
3.1	Les heuristiques générales de contrôle du raisonnement	85
3.2	Les stratégies de choix des raisonnements	86
4	Exécution, sortie des résultats, traces	86
5	Evaluation : application	91
1	Evaluation d'un système expert	92
1.1	Un système pour quelle application ?	92
1.2	Comment valider un système expert ?	92
2	Le système bayésien	93
3	La base des données de l'ARC & AURC	94
3.1	Présentation et caractéristiques	94
3.2	Influence de la base des données sur la base des connaissances	95
4	Les procédures d'évaluation	97
4.1	Constitution d'un échantillon	100
4.2	Standardisation des réponses	100
4.2.1	Standardisation des réponses du système LéZard	100
4.2.2	Standardisation des réponses du système bayésien	102
4.3	Procédures d'évaluation unifiées	105
4.4	Validité des procédures	105
5	Le système LéZard et les douleurs aiguës de l'abdomen	106
5.1	Le Traducteur	106
5.2	Le moteur	107

6	Résultats	107
7	Conclusion et perspectives	114
6	Perspectives	115
1	Les améliorations envisagées	116
2	Perspectives	118
2.1	Les méta-connaissances	119
2.2	Acquisition des Connaissances et apprentissage	119
	Références Bibliographiques	123
Annexe I	Calcul des exclusions et des négations	133
1	Vocabulaire et définitions préliminaires	135
2	Le calcul des exclusions	138
2.1	Définition des compatibilités et des exclusions	138
2.2	Rapports entre compatibilité et incompatibilité	140
2.3	Les algorithmes utilisés	143
2.4	Exemple de résultats	145
2.5	Interactions avec l'expert	146
3	Le calcul des négations	147
3.1	Définition des négations	147
3.2	Exemple de résultat	148
Annexe II	La base des connaissances de Léopard	151

Liste des figures

1.1	L'architecture du système LÉZARD.	13
1.2	Représentation du réseau sémantique. Les flèches épaisses interrompues représentent les liens causaux, les grisées les relations d'évocation, les noires les liens taxonomiques.	15
2.1	relations reliant un ensemble d'objets fils F_i à un objet père P	28
3.1	Acquisition des connaissances dans le système LÉZARD.	42
3.2	Prise en compte des connaissances incertaines dans le cadre de la logique L.	46
3.3	Dialogue entre le système Lézard et l'expert au sujet des relations d' <i>exclusions mutuelles</i> et de <i>compatibilité logique</i> générées.	48
3.4	relations reliant un ensemble d'objets fils F_i à un objet père P et optionnellement un ensemble d'objets B_j	53
3.5	relations reliant deux objets A_1 et A_2 et optionnellement un ensemble d'objets B_j	54
3.6	60
4.1	L'ordonnement des règles. Théorie et pratique.	71
4.2	La liste des observations faites sur le patient 232.	87
4.3	Un résultat de l'exécution du moteur (1).	88
4.4	La trace de quelques règles (1).	89
5.1	Représentation du réseau sémantique. les flèches épaisses interrompues représentent les liens causaux. Les autres flèches les liens taxonomiques.	98
5.2	Représentation du réseau sémantique, suite.	99
5.3	Un résultat de l'exécution du moteur (2).	101
5.4	Un résultat de l'exécution du moteur (3).	103
5.5	Un résultat de l'exécution du moteur (4).	104
5.6	Approche bayésienne	108
5.7	Approche système expert	109
5.8	Approche système expert unifiée	110
5.9	Comparaison LÉZARD/Bayes des diagnostics bien classés. En gras les meilleurs pourcentages pour chaque diagnostic.	111

- I.1 Représentation du réseau sémantique. Les flèches épaisses interrompues représentent les liens causaux, les grisées les relations d'évocation, les noires les liens taxonomiques. 134
- I.2 Rapport entre les compatibilités et les incompatibilités. 143



LEZARD : acquisition des connaissances et gestion de l'incertitude dans un système expert de seconde génération.

Résumé :

La plupart des systèmes experts médicaux utilisent des modes de représentation des connaissances évolués, autorisant la description de modèles physiopathologiques complexes. Les connaissances ainsi déclarées sont appelées connaissances profondes (*deep knowledge*). Pourtant, quelle que soit leur complexité, de tels modèles ne fournissent pas toujours les connaissances nécessaires à une tâche diagnostique. Des connaissances empiriques ou de surface (*shallow knowledge*) doivent nécessairement être intégrées au système.

Nous proposons ici une approche qui permet l'acquisition des connaissances de surface à partir des connaissances profondes : ces dernières ont un contenu sémantique complexe qui autorise plusieurs interprétations logiques valides. La traduction de ces connaissances profondes en connaissances de surface oblige à ne retenir que les interprétations sémantiquement valides. Ce découplage permet ensuite d'avoir une procédure d'interprétation simplifiée dans la mesure où elle ne dépend plus que du cadre logique des connaissances de surface.

Dans notre système, l'expert est interrogé durant le processus d'acquisition des connaissances de surface à partir des connaissances profondes. Des métaconnaissances heuristiques permettent de guider son interrogatoire. L'information fournie par les connaissances profondes est ainsi complétée de manière très précise au niveau des connaissances de surface très granulaires ; cette complétion permet de résoudre de façon satisfaisante les problèmes d'incertitude, concourant à la construction d'une base de connaissances opératoire.

Cette base est ensuite utilisée par un moteur d'inférences de type "chaînage avant" fonctionnant dans le cadre d'une logique non monotone. Ce moteur s'appuie sur un système de maintien de la cohérence fournissant les informations nécessaires à la discrimination des raisonnements possibles. Des stratégies heuristiques sont alors employées pour choisir le bon raisonnement.

Le système essentiel LÉZARD, dans lequel ces mécanismes sont mis en œuvre, est appliqué au domaine des douleurs aiguës de l'abdomen.

Mots clés :

Intelligence artificielle, Système expert, diagnostic médical, acquisition des connaissances, heuristiques, connaissances profondes, connaissances de surface.