HAL
open science

Un modèle d'indexation relationnel pour les graphes conceptuels fondé sur une interprétation logique

Iadh Ounis

To cite this version:

Iadh Ounis. Un modèle d'indexation relationnel pour les graphes conceptuels fondé sur une interprétation logique. Autre [cs.OH]. Université Joseph-Fourier - Grenoble I, 1998. Français. NNT : . tel-00004902

HAL Id: tel-00004902

https://theses.hal.science/tel-00004902

Submitted on 19 Feb 2004

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.

Thèse

présentée par

Iadh Ounis

pour obtenir le grade de Docteur
de l'Université Joseph Fourier
(arrêtés ministériels du 5 juillet 1984 et du 30 Mars 1992)
Spécialité : **Informatique**

Un modèle d'indexation relationnel pour les graphes conceptuels fondé sur une interprétation logique

---

Date de soutenance : 16 Février 1998

Composition du jury :

Président :	Jacques VOIRON
Rapporteurs :	Keith van RIJSBERGEN
	Gianni AMATI
Examinateurs :	Philippe JORRAND
	Yves CHIARAMELLA
	Jean-Pierre CHEVALLET

Thèse préparée au sein du laboratoire
Communication Langagière et Interface Personne-Système – IMAG
Université Joseph Fourier

Je tiens à remercier :

Mr. Jacques Voiron, Professeur à l'Université Joseph Fourier, qui m'a fait l'honneur de présider ce jury de thèse;

Mr. Keith van Rijsbergen, professor at the University of Glasgow, for having accepted to report this thesis. I thank him sincerely for the interest he manifested towards this work, and especially for the theoretical contributions that he brought to the information retrieval domain and constitute the foundation of my thesis;

Mr. Gianni Amati, researcher at the Fondazione Ugo Bordoni, Rome, for having accepted to report this work. I thank him for all the time he spent with the analysis of this thesis, for the passionate discussions we had in Grenoble and Rome, but also for lively e-mail sessions on hot research topics. His useful remarks, as well as his expertise in logic, greatly influenced the final form of this document. For all these reasons, I remain very grateful to him;

Mr. Philippe Jorrand, Directeur de recherche au CNRS, pour son aimable participation à ce jury et pour l'intérêt qu'il a manifesté pour ce travail;

Mr. Yves Chiaramella, Professeur à l'Université Joseph Fourier, et Mr. Jean-Pierre Chevallet, Maître de conférences à l'Université Pierre Mendès-France, qui ont conjointement dirigé ce travail, pour le soutien, leur confiance, la liberté d'action et les encouragements qu'ils m'ont apportés;

Les personnes qui ont contribué d'une façon ou d'une autre à la rédaction de ce document. Je voudrais d'abord remercier Theo Huibers pour la très plaisante et fructueuse collaboration que nous avons mené ensemble. Philippe Mulhem, par sa disponibilité et sa gentillesse, a été d'une grande aide pour la partie expérimentation. Marius Pasca, arrivé très récemment à l'équipe MRIM, s'est aussi beaucoup impliqué dans l'expérimentation, et je le remercie particulièrement du travail qu'il a réalisé lors de l'implantation du prototype. Finalement, mes remerciements vont également à tous les coauteurs de mes articles: Gianni Amati, Jean-Pierre Chevallet, Yves Chiaramella, Theo Huibers et Marius Pasca;

Les membres de l'équipe MRIM, Marie-France pour ses encouragements et son entrain, Catherine, Mourad, Nathalie et Franck qui contribuent activement à l'ambiance qui règne dans cette équipe;

Mr. Michel Simonet, pour les améliorations qu'il a apportées à ce document;

Les résidants du bâtiment B, et en particulier les membres de l'équipe IIHM, grâce à qui les soirées et les week-ends sont moins moroses. Je les remercie surtout pour les soirées Pizzas, si utiles pour continuer à travailler;

Sid-Ahmed pour les encouragements qu'il m'a prodigués lors de la rédaction de cette thèse;

Enfin, je tiens à adresser un grand merci à mes parents, pour leur aide et leur soutien tant apprécié pendant toutes ces années, et à mes tantes pour leur présence et leur affection.

Résumé

L'idée d'établir des relations entre des objets et de les représenter dans la base de connaissances d'un système informatique est le propre de toute approche en Intelligence Artificielle. Cependant, la plupart des formalismes de représentation de connaissances n'exploitent pas toute la richesse de la sémantique de ces relations, ni le comportement qui leur est associé. En recherche d'informations, les traitements de ces relations ne sont guère mieux élaborés et l'impact de leur prise en compte lors de la phase de correspondance n'a jamais été établi, même s'il reste vrai que de nombreuses approches tiennent compte de leur présence dans le document et tentent ainsi de les représenter lors du processus d'indexation. Pourtant la recherche de documents structurés ou complexes exige plus que jamais, outre un langage d'indexation robuste et expressif, la prise en charge de la sémantique des relations ainsi que leurs propriétés. À travers une étude des nouvelles exigences auxquelles la recherche d'informations d'aujourd'hui doit répondre, nous proposons un modèle d'indexation relationnel pour les documents. L'approche consiste à considérer qu'un terme d'indexation est fondé sur des concepts complexes où les connecteurs sémantiques sont vus comme des opérateurs, ou des relations permettant de construire des expressions nouvelles représentant des concepts nouveaux ou des situations nouvelles. Le modèle proposé ne se contente pas de représenter les relations, mais permet aussi d'offrir un cadre général précisant les principes généraux de manipulation de ces relations et la prise en compte de leurs propriétés dans un processus de recherche fondé sur une approche logique. Le modèle proposé comporte deux composantes: le langage de représentation des informations, permettant une approche d'indexation relationnelle, et les règles de dérivation qui, reprenant ce langage, permettent de diriger le processus de correspondance.

Nous utilisons la théorie des situations comme langage de représentation et un système de dérivation de pertinence, reposant sur une axiomatisation de la notion de correspondance entre les documents et la requête pour la prise en compte des relations. Une caractéristique intéressante de ce modèle est qu'il conduit à étendre certains formalismes de représentation de connaissances par des notions utiles en recherche d'informations. Les limitations de la famille des logiques terminologiques, utilisée par ailleurs comme base formelle de l'approche d'indexation relationnelle proposée, peuvent ainsi être surmontées. Cependant, la complexité des traitements associés à cette famille de logiques empêche de les utiliser comme un modèle opérationnel. Nous proposons alors le formalisme des graphes conceptuels comme un bon compromis entre la complexité des démonstrateurs de théorèmes et la simplicité des approches algébriques. Ce formalisme est alors vu, à travers une interprétation logique adéquate, comme une implantation d'une logique terminologique étendue et du modèle d'indexation. Notre approche a été implantée sur une plateforme de gestion de graphes conceptuels, réalisée sur le système de gestion de base de données à objets O_2. Le prototype Relief résultant de notre expérimentation a été testé sur une collection d'images et a démontré l'applicabilité et le bien-fondé de notre approche.

Mots-clés: recherche d'informations, indexation relationnelle, graphes conceptuels, interprétation logique.

Abstract

The idea of establishing relations between objects and representing them in the knowledge base of an information system is a feature of all approaches in Artificial Intelligence. However, the majority of knowledge representation formalisms do not fully exploit the semantic richness of these relations and their associated behavior. In information retrieval, the processing of these relations is not at all better performed and the impact of their taking into account at the time of the matching phase has never been established though it is true that many approaches take into account their presence in the document and tend to represent them as part of the indexing process. Nevertheless, retrieval of structured and complex documents makes it necessary, more than ever, to use the relation semantics and properties, next to a solid and expressive indexation language. By use of a study of the new demands to which information retrieval has to answer nowadays, we propose a relational indexing model for documents. The approach consists of considering that an indexing term is based on complex concepts where semantic connectors are considered to be operators, or relations that allow to build new expressions representing new concepts or situations. The proposed model is not limited to relation representation, but also offers a general framework that mentions the general principle for the handling of these relations and the taking into account of their properties in a retrieval process based on a logical approach.

The model proposed here is made up of two components: the representation language allowing for a relational indexing approach, and inference rules that make it possible to direct the correspondence process by use of the language. We use the situation theory as representation language and a relevance derivation system which is based on the axiomatisation of the notion of correspondence between documents and the query to take into account the relations. An interesting feature of this model is that it leads to extensions of some knowledge representation formalisms, with useful notions in information retrieval. The limitations of the terminological logics, otherwise used as the formal basis of the proposed relational indexing approach, can be thus overcome. At the same time, the complexity of the treatments associated to this family of logics does not allow to use them as an operational model. Consequently, we propose the conceptual graph formalism as a good compromise between the complexity of theorem provers and the simplicity of algebraic approaches. This formalism is thus considered, by means of an adequate logical interpretation, as being an implementation of an extended terminological logic and of the indexing model. Our approach was implemented on a conceptual graph handling platform realised on top of the OO-DBMS O_2. The resulting prototype Relief was tested on an image collection. The experimentation proved the practicability and well-foundation of the approach.

Mots-clés: information retrieval, relational indexing, conceptual graphs, logical interpretation.

Table des matières

1 Introduction **1**
1.1 Les Défis de la RI . . . 3
1.2 Vers une Indexation Relationelle . . . 6
1.3 Contenu de la Thèse . . . 9
1.4 Organisation de la Thèse . . . 11

2 Pourquoi les Relations? **13**
2.1 Les Descripteurs en Recherche d'Informations . . . 14
2.1.1 Les Problèmes de l'Indexation par Mots-Clés . . . 15
2.1.2 Les Approches Linguistiques . . . 16
2.1.3 Utilisation d'un Thésaurus . . . 17
2.1.4 L'Indexation Relationnelle . . . 18
2.1.5 Vers des Systèmes de Recherche Conceptuels . . . 21
2.2 Concrétisation d'une Approche Relationnelle . . . 22
2.2.1 Objectivisme et Primitives . . . 22
2.2.2 Quels Types de Relations Sémantiques Choisir . . . 24
2.2.3 Difficultés d'une Indexation Automatique . . . 26
2.2.4 Les Systèmes de Recherche d'Informations Conceptuels . . . 29
2.3 Raisonner sur les Relations . . . 33
2.3.1 L'Approche de Maron . . . 34
2.3.2 Les Réseaux Sémantiques . . . 35
2.3.3 L'Approche de Rime . . . 39
2.3.4 Conclusion . . . 44
2.4 L'utilité des Relations en RI . . . 45
2.4.1 Les Applications Hypermédias . . . 45
2.4.2 La Recherche d'Images . . . 48
2.4.3 La Recherche de Documents Structurés . . . 51
2.4.4 La Recherche de Documents Multimédias . . . 52
2.4.5 Quels Raisonnements sur les Relations en RI? . . . 59
2.5 Discussion et Orientation de la Thèse . . . 61

3 Le Modèle d'Indexation Relationnel **65**
3.1 Présentation de l'Idée du Modèle . . . 66
3.1.1 Hypothèses et Généralités . . . 66
3.1.2 La Théorie des Situations . . . 72
3.2 Un Modèle Relationnel pour la RI . . . 80
3.2.1 Le langage d'Indexation . . . 81

3.2.2 La Fonction de Description χ . . . 82
3.2.3 Un Modèle Préférentiel pour la RI . . . 86
3.3 Utilisation des Propriétés des Relations . . . 90
3.3.1 Dérivation de Connaissances Implicites . . . 90
3.3.2 Comment Gérer les Relations et leurs Propriétés en RI . . . 102
3.3.3 Extension de la Relation d'Endiguement . . . 105
3.3.4 Les Relations de Préclusion . . . 107
3.4 Un Modèle Relationnel de Dérivation de Pertinence . . . 108
3.4.1 Les règles et Axiomes du Système . . . 109
3.4.2 Autres Règles de Dérivation . . . 112
3.4.3 Complétude et Correction . . . 114
3.4.4 Décidabilité . . . 114
3.4.5 Exemple de Dérivation de Pertinence . . . 115
3.5 Extensions Possibles du Modèle . . . 116
3.5.1 Introduction de l'Incertitude . . . 118
3.5.2 Utilisation des Labels de Règles . . . 128
3.5.3 Un Système Multifacette . . . 131
3.6 Conclusion et Résumé . . . 135

4 Une Vision Relationnelle des Graphes Conceptuels 139
4.1 Introduction . . . 139
4.1.1 Pourquoi les Graphes Conceptuels? . . . 140
4.1.2 Notions de Base . . . 142
4.1.3 Les Problèmes du Formalisme . . . 146
4.2 Le Modèle des Graphes Conceptuels et la RI . . . 150
4.2.1 Lien avec la Théorie des Situations . . . 152
4.2.2 Les Propriétés du Modèle . . . 156
4.2.3 Conclusions sur le Modèle . . . 160
4.3 Un Canon Adapté à une Approche Relationnelle . . . 162
4.3.1 Le Treillis des Types de Concepts $\mathcal{T}_c$ . . . 163
4.3.2 Le Treillis des Types de Relations $\mathcal{T}_r$ . . . 164
4.3.3 L'ensemble $\mathcal{M}$ des Marqueurs . . . 167
4.3.4 La Relation de Conformité *Conf* . . . 168
4.3.5 L'ensemble $\mathcal{S}_r$ des Graphes Signatures . . . 170
4.3.6 L'ensemble des Connaissances $\mathcal{K}_s$ . . . 172
4.3.7 L'ensemble d'Endiguement $\mathcal{E}$ . . . 175
4.4 Un Modèle de *GCs* Propre à la RI . . . 181
4.4.1 Graphe Conceptuel Bien Formé . . . 181
4.4.2 Compléments de Définitions . . . 183
4.4.3 Les Graphes Isomorphes . . . 184
4.4.4 Le Langage d'Indexation Relationnelle . . . 186
4.4.5 Méta-Connaissances et Règles de Dérivation . . . 189
4.4.6 Un Modèle Relationnel pour les Graphes Conceptuels . . . 197
4.4.7 La Couverture d'un Graphe Conceptuel . . . 201
4.5 Une Interprétation Logique pour les *CGs* . . . 205
4.5.1 Extension Universelle de l'Opérateur Φ . . . 206
4.5.2 Adéquation et Complétude de ϕ . . . 209

4.5.3 Discussion sur l'Utilisation des Quantificateurs dans les GCs 217
4.5.4 Quels Quantificateurs pour la Recherche d'Informations? 223
4.5.5 Conclusion: Quel(s) Algorithme(s) pour la Projection? 223
4.6 Une Approche Algébrique pour les Graphes Conceptuels 225
4.6.1 Logique Vs. Algèbre 225
4.6.2 Algèbre et Systèmes d'Informations 227
4.6.3 Motivations d'une Approche Fichier Inverse 230
4.6.4 Extension Relationnelle du Fichier Inverse 232
4.6.5 Utilisation d'un Fichier Inverse en *GCs* 239
4.7 Résumé et Conclusion 255

5 Application à un Corpus d'Images 257
5.1 Introduction 257
5.2 Le système Relief 258
5.2.1 Conception Générale du Système 258
5.2.2 De la Plate-forme PGGC vers la Plate-forme ROGER 260
5.2.3 Fonctionnalités et Architecture du Système Relief 265
5.3 Indexation du Corpus 268
5.3.1 Présentation de la Collection 268
5.3.2 Conversion en Graphes Conceptuels 270
5.3.3 Le Modèle d'Indexation 272
5.4 Évaluation des Requêtes 274
5.4.1 Interface d'Interrogation 275
5.4.2 Visualisation des Réponses 276
5.5 Analyse des Résultats 278
5.5.1 Les Performances du Système 278
5.5.2 Résultats sur les Requêtes 280
5.5.3 Évaluation Qualitative 283
5.6 Conclusion 286

6 Conclusion 291
6.1 Conclusions et Apports 291
6.2 Perspectives 293

A Notations OMT 295

B La Collection Test 297

C Les Requêtes FERMI 299

D Le Canon 301
D.1 Les Types de Concepts 301
D.2 Les graphes signatures 302

Index thématique i

Index des citations iv

Table des figures

1.1 Problématique classique de la RI 4

1.2 Exemples d'index pour "*une opacité tissulaire au niveau du poumon*" 8

2.1 Une détection d'une structure d'une expression d'index pour le titre "The Elimination of Special Functions from Differential Equations (exemple tiré de [Bru93])" . . 21

2.2 Transitivité de la préposition "EST" 28

2.3 Une Description de la relation juridique *usefruct* 29

2.4 Exemple de déduction de propriétés 37

2.5 La transitivité du lien IS-A n'est pas toujours valide 37

2.6 Une arborescence RIME pour la phrase "Une hypertrophie de densité tissulaire du lobe droit de la théroïde" 40

2.7 Translation des Arborescences de RIME en LTs 43

2.8 Transitivité de la relation "*Composé-par*" 47

2.9 Les relations "compose" et "couvre" sont transitives 50

2.10 Des inférences valides ne peuvent pas être implantées avec les LTs 57

3.1 Exemple d'imbrication de Relons 82

3.2 Exemple de relation de préférence: $\chi'(d) \prec_R \chi(d)$ 86

3.3 Un document à-propos d'une "*opacité alvéolaire au niveau du lobe droit*" implique que cette maladie touche également le "*lobe*" 93

3.4 Exemple de relation d'endiguement dans RIME. (tiré de [Nie90, page 155]) 98

3.5 Utilisation de la relation d'endiguement pour la dérivation des connaissances implicites 99

3.6 Chemins de transformation d'un index de document 120

3.7 Multiplicité des chemins à certitude maximale 123

3.8 Exemple de transformation d'index 129

3.9 Exemple de transformation de facettes 134

4.1 Un graphe simple pour “Tom poursuit Jerry” 143
4.2 En grisé est représenté la projection de h sur g. h peut être interprété comme suit:“il existe une activité et un sport localisés sur le même lieu possédant un lac”. 145
4.3 Les graphes g et h vérifient $g \leq h$ et $h \leq g$, mais $h \neq g$ 148
4.4 Ces deux graphes sont logiquement équivalent ($\phi(g) = \phi(h)$), cependant ils ne sont pas équivalents par la relation de spécialisation. De plus $g \neq h$. 148
4.5 L'opérateur ϕ n'est pas intuitif. Les dérivations logiques sont opposées aux dérivations sur les graphes. Nous avons $g \vdash_{Copie+Restriction} g'$ alors que $\phi(g') \supset \phi(g)$ 150
4.6 Ici $g \equiv h$ mais $g \neq h$. Ces deux graphes ont cependant la même translation en termes de situations 155
4.7 Les deux graphes g et h sont équivalents. Ils n'ont cependant pas la même translation en termes de situations. 156
4.8 Treillis des Marqueurs 168
4.9 Un graphe signature pour la relation de localité *Loc* 171
4.10 Le sens des arcs de la relation “*entre*” est conforme à la propriété (3) 171
4.11 Le treillis des types de concepts $\mathcal{T}_{Cspe}$ 172
4.12 Le treillis des types de Relations $\mathcal{T}_{Rspe}$ 173
4.13 Les notions de Contexte positif et de contexte négatif 175
4.14 Un lien de coréférence sur le concept [PAYS] 177
4.15 Représentation de la propriété de transitivité 179
4.16 Représentation de la propriété de composition 179
4.17 Représentation de la propriété de symétrie 179
4.18 Applicabilité de la condition du contexte 181
4.19 En Grisé les graphes isomorphes à $g_{\Sigma_{Conduit}}$ 185
4.20 Deux graphes E-isomorphes 187
4.21 Base de Connaissances Minimale 190
4.22 Deux bases de connaissances minimales 192
4.23 Exemple de dérivation de graphes 193
4.24 Exemple de dérivation de graphes 194
4.25 Application de la propriété de spécialisation 196
4.26 Modèle étendu des Graphes Conceptuels 200
4.27 Un système logique à base de graphes conceptuels 201
4.28 L'ensemble $\{g_1, g_2, g_3\}$ constitue une couverture de g 202
4.29 Couverture relationnelle du graphe index g 205
4.30 Le sens des implications est opposé à celui des dérivations 209

4.31 Copie d'un arch 210
4.32 La restriction d'un graphe conceptuel ($S_1 \leq C_1$) 211
4.33 Jointure de deux graphes conceptuels 212
4.34 Simplification de la relation R 213
4.35 Un sens de dérivation plus intuitif 213
4.36 Détermination de la forme standard des graphes 215
4.37 Exemple de graphes conceptuels 216
4.38 Un exemple de graphe conceptuel existentiellement marqué 219
4.39 Le treillis des types de Relations 233
4.40 Fichier Inverse $\mathcal{FI}$ de l'exemple 234
4.41 Le treillis des types de Concepts 237
4.42 Extension à base du treillis des relations 237
4.43 Description sommaire du fonctionnement de l'algorithme de projection 241
4.44 Pourquoi les témoins? 243
4.45 Les treillis des types de concepts et de relations 248
4.46 Un corpus de cinq documents 249

5.1 Conception générale du Système Relief 258
5.2 Extension de la Plate-forme PGGC 261
5.3 Transformation linéaire d'un graphe conceptuel 262
5.4 Modélisation détaillée des graphes conceptuels 264
5.5 La Plate-forme ROGER 265
5.6 Fonctionnalités du système Relief 267
5.7 Architecture du système Relief 268
5.8 Vue spatiale et physique de l'image 271
5.9 Vue structurelle de l'image 272
5.10 Description originelle de l'image 273
5.11 Interface de formulation de requêtes 276
5.12 Interface pour la visualisation des réponses 277
5.13 Temps estimé pour l'évaluation d'une requête 282
5.14 Histogramme comparatif des temps de réponses 283
5.15 Importance des propriétés des relations 285
5.16 Quelques mesures de Rappel/Précision 286
5.17 Comparaison des Résultats qualitatifs 287

5.18 Chargement des Données 288

A.1 Notations de la méthode OMT 295

B.1 Image 4448 de la collection test 297

B.2 Exemple d'image indexée 298

Liste des tableaux

2.1 Détection des “bons” index 14
2.2 Fréquences de certaines relations 18
2.3 Les catégories des relations [Far80a] 19
2.4 Table de connecteurs tirée de [Bru93] 20

3.1 Classification des Prépositions de la Langue Anglaise 103

4.1 Syntaxe algébrique Vs. Syntaxe logique 226
4.2 Une structure de fichier inverse où le document d_1 est indexé par la conjonction Système $\wedge$ Recherche $\wedge$ Information 229
4.3 Table d'accélération pour la relation *Profession* 235
4.4 Table d'accélération pour la relation *Assis-sur* 237
4.5 Fichier Inverse après analyse des archs du premier index 249
4.6 Fichier inverse après analyse des archs des deux premiers documents 250
4.7 Fichier inverse final des graphes index du corpus 251
4.8 Une nouvelle ligne prenant en compte la propriété $R_0 \leq_\sigma R_2$ 251
4.9 Table d'accélération partielle pour la relation R_1 (analyse de la première ligne du fichier inverse) 252
4.10 Table d'accélération partielle pour R_1 (analyse des trois premières lignes du fichier inverse) 252
4.11 Table d'accélération pour la relation R_1 252
4.12 Table d'accélération pour la relation R_0 253
4.13 Table d'accélération pour la relation R_3 253

5.1 Chargement des Index 278
5.2 Table comparative des temps de réponse 281

C.1 Requêtes de la collection test 299

D.1 Extrait du treillis des types de concepts 301

Chapitre 1

Introduction

La recherche d'informations (RI) occupe une place importante dans les systèmes d'informations. En effet, s'il est important de savoir modéliser l'information, il est également nécessaire de pouvoir y accéder facilement. L'augmentation du nombre des documents au niveau des entreprises et des institutions ainsi que la multiplication des documents électroniques nécessite la mise en place d'un système sophistiqué de recherche d'informations. À titre indicatif, environ deux millions de livres et d'articles académiques, scientifiques, médicaux et socio-économiques apparaissent chaque année au niveau de la communauté européenne [dB85]. Dans les années à venir, l'information et en particulier le document est amenée à jouer un rôle important dans l'amélioration de la productivité et la compétitivité des entreprises [Les89]. Un accès efficace à toute cette masse d'informations sera d'autant plus important qu'elles contiennent généralement des données critiques essentielles pour la survie d'une organisation. Le domaine de recherche d'informations prétend apporter des solutions à un tel challenge. La problématique de la recherche d'informations peut ainsi être vue comme la satisfaction d'un besoin en informations d'un utilisateur, qui est exprimé par une requête, par rapport à un ensemble de documents appelé *collection* ou *corpus*.

Cette problématique a beaucoup évolué ces dernières années, en grande partie à cause de l'augmentation du volume d'informations électroniques. Avec l'avènement de l'Internet (également connu sous le nom de World Wide Web) et d'autres hypermédias, nous assistons à une transformation du concept même de document. Celui-ci comporte maintenant des images, des sons, voire même des animations; c'est l'ère du multimédia et des documents structurés. L'accroissement de la quantité d'informations touche pratiquement tous les domaines, rendant de plus en plus nécessaire une exploitation efficace de leurs contenus. La disponiblité de nombreux butineurs sur l'Internet, comme Netscape ou Internet Explorer pour n'en citer que les plus célèbres, même si elle rend possible la recherche, l'accès et la visualisation de ces informations, ne constitue néanmoins pas une solution fiable pour exploiter toute l'étendue des informations disponibles. Ainsi, plusieurs informations restent inutilisables, ou parfois ne peuvent être effectivement lues puisqu'il n'y a pas de moyen efficace permettant leur recherche. À titre d'exemple, une requête adressée par un utilisateur à un outil de recherche sur Internet comme *Lycos*[1] peut engendrer des dizaines de pages de réponses référant des documents susceptibles de contenir une réponse pertinente à la requête. Ainsi, le nombre des réponses est souvent trop important pour être utile. Au bout d'une consultation de deux ou de trois pages de réponses, l'utilisateur finit généralement par se lasser et abandonne la consultation. Pourtant, il peut arriver que l'information correspondant le mieux à sa requête se trouve sur l'une

1. http://lycos.cs.cmu.edu/

des dernières pages renvoyées[2].

Cette augmentation, autant en volume qu'en richesse, a influencé les méthodologies de construction et d'évaluation des systèmes de recherche d'informations. Pour preuve, nous sommes passés des premières collections test[3] à quelques milliers de documents (la CACM ou la collection Cranfield [Cle91]) aux collections de l'ordre du Giga-octet comme par exemple TREC [Jon95] contenant approximativement un million de documents. Ce changement d'échelle incite à la remise en cause de certaines fonctionnalités d'un système de recherche d'informations:

- étant donné la taille importante des corpus, il est souvent fort probable que la liste des documents pertinents soit assez longue. Une attention particulière devra être donnée à la précision des réponses fournies par le système, autrement dit le système devra privilégier la qualité des réponses au détriment de la quantité. On parle alors de systèmes de recherche d'informations orientés vers la précision des réponses.
- les techniques traditionnelles de recherche d'informations se contentent généralement de renvoyer à l'utilisateur une liste de références ou de pointeurs sur les documents. Pourtant, ce dernier s'attend souvent à ce que le système lui retourne directement les passages (les parties de document) contenant l'information recherchée.
- les documents peuvent contenir des informations de nature différente (textes, images, graphiques, etc.). De plus, ils peuvent être structurés selon un schéma donné. Par conséquent, il est essentiel de tenir compte de ces aspects afin d'améliorer les performances qualitatives du processus de recherche. Par exemple, l'aspect structurel du document permet d'inclure dans la réponse, les éléments de structure les plus pertinents par rapport à la requête.

Malheureusement, les systèmes de recherche d'informations actuellement opérationnels n'ont pu suivre l'évolution rapide de ces nouveaux besoins. Leur principal défaut concerne la manière dont ils représentent l'information contenue dans les documents du corpus: souvent les langages de représentation d'informations utilisés sont assez pauvres et ne peuvent remplir les nouvelles propriétés citées ci-dessus qu'un système de recherche d'informations est censé posséder. Les quelques rares tentatives s'étant intéressées au problème sont restées à l'état embryonnaire et s'appuient sur des techniques trop *ad hoc* pour être généralisées. Les travaux de recherche ont surtout porté sur la modélisation de la correspondance entre la requête et les documents du corpus [vR86, Nie90, Hui96], en supposant résolus les problèmes de représentation des index.

La description des documents en vue de leur recherche reste pourtant un problème central. Plus les documents sont complexes, plus l'utilisation d'un langage d'indexation robuste et expressif s'avère nécessaire. Ce langage permet la description fidèle et juste du contenu sémantique de la collection, permettant ainsi au processus de recherche de gagner en efficacité et en précision. En effet, les algorithmes de recherche les plus complets et performants ne peuvent compenser une représentation approximative, pauvre ou partielle des informations contenues dans le corpus. Comme l'a signalé Blair [Bla90], la meilleure façon de savoir ce qu'il faut représenter en recherche d'informations consiste à étudier les activités auxquelles un système de recherche d'informations est destiné. Ainsi pour déterminer le langage qui devra être attribué à la représentation des documents

2. Selon une statistique récente donnée dans le dernier SIGIR'97, moins de 20% des utilisateurs vont au-delà de la première page de réponses.

3. Il s'agit d'un ensemble homogène de documents pour lequel des requêtes et le jugement de pertinence associé ont été définis (voir section suivante).

du corpus, il faudra comprendre ces activités et souligner leur contexte d'utilisation. Les nouvelles exigences énumérées ci-dessus devront donc être au coeur du choix de ce langage.

Il est généralement admis que l'approche classique décrivant les documents du corpus par des listes de mots-clés est insuffisante à mesure que les corpus subissent une augmentation en informations autant en quantité qu'en richesse. Ces approches, vu les problèmes de paraphrasage et d'ambiguïté qui en résultent, ne peuvent pas engendrer des systèmes orientés vers la précision des réponses. En outre, la prise en compte de l'aspect structurel exige l'introduction de certaines relations de composition indiquant l'arborescence sous-jacente à cette structure. C'est le cas du système IOTA [Ker84] où la structure est principalement utilisée pour donner un meilleur contrôle sur la précision et le rappel. La recherche d'images impose aussi la prise en charge des informations géométriques décrivant les objets spatiaux associés à l'image et des relations spatiales décrivant leurs positions relatives. Le modèle de recherche d'images EMIR^2 [Mec95c] énumère ainsi douze relations spatiales utiles pour le processus de recherche.

Ces points rendent nécessaires le passage de la représentation «plate» à base de mots-clés vers des structures sémantiques complexes impliquant plusieurs concepts et relations. Le recours aux formalismes de représentation de connaissances, connus pour leur précision et leur richesse d'expressivité, permet de répondre à un tel besoin. Cependant, si les concepts et les traitements qui peuvent leur être associés sont bien étudiés et pris en compte par ces formalismes, il ce n'est pas le cas pour les relations. Les caractéristiques et propriétés de ces relations ainsi que les opérations qui peuvent leur être appliquées ne sont généralement pas spécifiées. Il suffit d'observer le nombre d'opérateurs de manipulation de relations (ou rôles) dans la famille des logiques terminologiques [Neb90a] et le comparer à celui des opérateurs de manipulation de concepts.

En recherche d'informations, autant le formalisme utilisé pour l'indexation est riche, autant les structures qu'il engendre méritent un traitement plus approfondi lors du processus de correspondance. Cela s'applique aussi bien aux concepts qu'aux relations. Dans les systèmes avancés actuels comme Elen [Che92] ou MIRTL [MSST93], le traitement de ces relations n'est guère élaboré et leur impact sur le processus de recherche n'a jamais été cerné. Pourtant, étant donné la richesse de leur sémantique et les connaissances implicites qu'elles véhiculent, les relations peuvent fortement influencer les décisions de pertinence.

Nous proposons un modèle d'indexation relationnel dont la richesse permettra de répondre aux points énumérés ci-dessus, et où l'accent sera mis sur les relations et leur impact sur le processus de recherche. Le modèle proposé généralise ainsi une technique classique en recherche d'informations qui consiste à explorer les liens sémantiques pouvant exister entre les concepts (thésaurus) à une approche plus globale tenant compte des relations sémantiques intervenant dans les termes d'indexation. La notion de terme d'indexation est fondée ici sur des concepts complexes où les connecteurs sémantiques sont vus comme des relations. Dans la suite de ce chapitre, nous présentons succinctement les défis auxquels la recherche d'informations est aujourd'hui confrontée, puis nous fixons notre vocabulaire, avant d'exposer les grandes lignes de notre proposition.

1.1 Les Défis de la RI

Le rôle d'un système de recherche d'informations est de permettre l'accès aux documents par leur contenu sémantique: l'utilisateur exprime son besoin d'informations en indiquant le contenu qu'il souhaite observer dans les documents retrouvés. Pour ce faire, les systèmes sont fondés sur

un modèle formel de correspondance. La figure 1.1 est une adaptation d'un classique repris dans de nombreux textes traitant de la recherche d'informations [SM83, vR79]. Elle résume ainsi la problématique de cette dernière en l'adoption d'un couple de modèles, à savoir l'*indexation* et l'*interrogation*. Ce dernier est adapté à la prise en compte du besoin formulé par l'utilisateur grâce à un langage de requête, l'autre à même de caractériser les informations décrivant les documents du corpus. Le tout est gouverné par une *fonction de correspondance* dont la finalité est de *décider* si un document du corpus est *pertinent* à la requête de l'utilisateur. La donnée des modèles d'indexation et d'interrogation utilisés ainsi que la fonction de correspondance associée définit un *modèle de recherche d'informations*. Les modèles proposés pour la recherche d'informations sont très nombreux. Des modèles classiques tels que le modèle Booléen, le modèle vectoriel, le modèle probabiliste, ... ([vR79, SM83]), nous sommes passés à des modèles plus formels tels que le modèle sémantique de Croft [Cro85] et le modèle logique de van Rijsbergen [vR86].

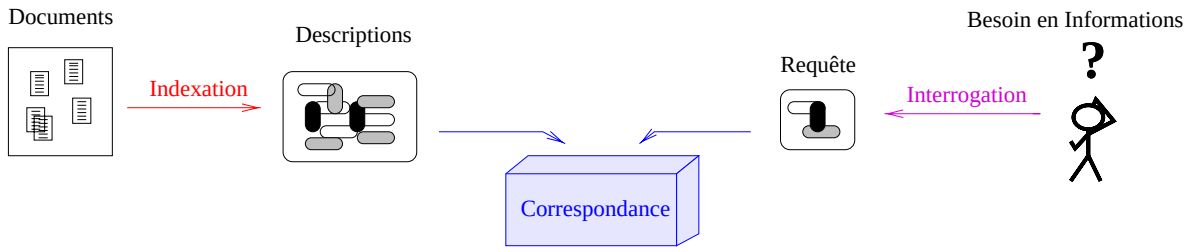


Figure 1.1. Problématique classique de la RI

Au niveau opérationnel, un système de recherche d'informations est évalué en observant ses réponses par rapport à ce que l'utilisateur souhaite retrouver. Les mesures classiquement utilisées à cet effet sont les mesures de *rappel* et de *précision*, comprises dans l'intervalle [0,1]:

$$\text{Rappel} = \frac{\textit{Nb de documents pertinents retrouvés}}{\textit{Nb de documents pertinents}}$$

$$\text{Précision} = \frac{\textit{Nb de documents pertinents retrouvés}}{\textit{Nb de documents retrouvés}}$$

Le "rappel" mesure la capacité du système à retrouver tous les documents pertinents alors que la précision estime la capacité du système à ne retrouver que des documents pertinents. Le but ultime d'une bonne indexation est de produire des termes d'indexation faisant converger ces mesures vers 1. Cependant, ces deux mesures s'avèrent antinomiques: lorsque le rappel est bon, la précision donne de mauvais résultats, et réciproquement. Intuitivement, cette dualité se comprend par le fait qu'un bon rappel induit une description plus large (générique) des documents afin de n'oublier aucun document intéressant. Par ailleurs, si la description est trop large, d'autres documents répondent à la requête, défavorisant ainsi la précision. Réciproquement, la précision nécessite des termes d'indexation plus spécifiques.

Généralement, deux autres mesures sont couramment citées dans la littérature. Il s'agit des taux de *silence* et de *bruit*. Le silence est le complémentaire du rappel, c'est-à-dire le nombre de documents pertinents non retrouvés par rapport à tous les documents pertinents. Quant au bruit, il

constitue le complémentaire de la précision, c'est-à-dire la proportion de documents non pertinents donnés en réponse par rapport au total des documents renvoyés à l'utilisateur.

Toutefois, si les quatre paramètres précédents permettent de conclure qu'un système donné est meilleur qu'un autre, il n'existe aucun moyen de connaître les tenants et les aboutissants d'une telle conclusion [Nie90, Hui96]. Autrement dit, si ces paramètres évaluent les résultats obtenus par un système, ils ne permettent pas d'expliquer et d'analyser pourquoi un système peut établir une décision de pertinence là où un autre système échoue. Ainsi, il est impératif de connaître, pour chaque système, les mécanismes qui sous-tendent ses décisions de pertinence. Ces mécanismes connus, il sera possible de prévoir son comportement et de remédier le cas échéant à ses limites.

La rapidité a toujours été un facteur important pour le succès et l'acceptation des systèmes de recherche d'informations [Bro95]. Si un système est trop lent, il n'est pas envisageable de l'utiliser en recherche d'informations, indépendamment de la qualité de ses réponses en termes de précision et de rappel. La récente augmentation des volumes d'informations dans les collections de documents suggère que ce critère deviendra de plus en plus important. Plus les collections deviennent grandes en volume, plus le coût de recherche sera élevé.

Cet accroissement dans les volumes d'informations impose également la conception de systèmes orientés vers la précision des réponses. Le bruit est pour ainsi dire intolérable quand les collections comptent des millions de documents. En effet, la tâche de l'utilisateur devient très ardue puisqu'il devra retrouver l'information qui l'intéresse dans une masse trop importante de documents non pertinents. Seuls des langages d'indexation expressifs permettent d'améliorer les résultats à ce niveau.

La quête d'un langage idéal pour l'indexation permettant un compromis entre la richesse d'expression et l'efficacité des traitements est un problème ouvert en recherche d'informations. Cependant, plusieurs travaux théoriques [LB85, LB87] ont abouti à la conclusion qu'il existait une opposition entre richesse d'expression et efficacité des calculs. C'est le cas par exemple des logiques terminologiques [BS85, PS84, PSOKK$^+$90, Neb90a], connues pour leur sémantique clairement définie et leur puissance d'expression, mais dont la complexité des inférences constitue un écueil pour toute utilisation efficace.

À l'inverse, les systèmes basés sur les modèles classiques les plus connus comme le modèle Booléen, le modèle vectoriel [Sal71, SM83], ou le modèle probabiliste [vR79, Rob77, Fuh92], réputés pour leur rapidité, semblent atteindre leur potentiel maximum en termes de qualité des réponses. Les extensions qui leur sont appliquées se contentent généralement d'utiliser des nouveaux algorithmes de correspondance basés sur des technologies calculatoires plus avancées. Le résultat n'est de ce fait que des variantes plus ou moins élaborées des modèles d'origine, sauf qu'aucune amélioration sensible n'a été réellement observée sur des cas de corpus réels [Lal96].

Un tel constat d'échec renforce le fait que les meilleurs algorithmes de correspondance ne peuvent compenser des représentations approximatives des documents et des requêtes. L'intérêt d'avoir des algorithmes rapides et polynomiaux est discutable quand les réponses fournies sont inutilisables, voire incorrectes. C'est le cas par exemple des moteurs de recherche sur le Web, rapides mais fournissant un nombre de réponses trop important pour être utilisable. Ces moteurs se basent sur des modèles classiques de recherche d'informations munis de techniques calculatoires efficaces. Van Rijsbergen était plus catégorique en portant des réserves sur les modèles purement calculatoires, émettant en 1986 la conclusion suivante [vR86][pp194]:

> "I have reluctantly concluded that the fundamental basis of all the previous work [in

information retrieval] is wrong ... and any attempts to achieve further improvements [based on statistical techniques] are waste of time."

La pauvreté de la représentation des documents est la principale raison évoquée pour expliquer l'échec relatif des approches classiques en recherche d'informations [Bla90, Lal96, OP97a]. Pour des petites collections, en moyenne 10% des mauvaises réponses du système sont imputées à une représentation hâtive des documents [Bla90]. Si ce taux reste acceptable pour une collection de 1000 documents par exemple, il sera très vite inacceptable dès lors qu'il s'agit de travailler sur des corpus contenant des millions de documents, situation tout à fait raisonnable et courante de nos jours avec l'avènement des autoroutes de l'informations. Il apparaît ainsi impératif d'utiliser un langage d'indexation suffisamment riche et précis pour diminuer ce taux d'erreur et atténuer son impact.

De plus, plusieurs travaux de recherche [CC92, MSST93, Lal96] montrent que la puissance d'expression du langage d'indexation est l'un des éléments essentiels que doivent posséder les systèmes dès lors que l'on aborde des nouveaux domaines d'application comme la recherche des données multimédias (image, vidéo, son, etc.).

La recherche d'informations est donc confrontée à un double défi. D'une part, l'utilisation d'un langage expressif s'avère nécessaire pour l'amélioration des systèmes et pour leur adaptation à des applications variées. D'autre part, le coût d'évaluation des requêtes devra être raisonnable, c'est-à-dire que les fonctions de correspondance utilisées devront se baser sur des algorithmes polynomiaux.

Notre travail s'inscrit autour de ces deux axes: puissance d'expression et rapidité des traitements. La puissance d'expression nous permettra de décrire efficacement des documents multimédias complexes. Nous pourrons ainsi nous attaquer à un éventail de plus en plus large d'applications. La rapidité des traitements permet d'envisager l'utilisation du système sur de très grands corpus avec des coûts raisonnables.

1.2 Vers une Indexation Relationelle

Le processus d'*indexation* est donc une composante importante d'un système de recherche d'informations. Il est mis en oeuvre afin d'extraire préalablement une représentation homogène du contenu sémantique des documents du corpus. Un tel processus est rendu nécessaire par les larges volumes de données présents dans le corpus, et dont l'évaluation au moment de l'interrogation serait inefficace. La phase d'indexation est une fonction d'une importance capitale puisque les représentations internes qu'elle engendre peuvent influencer considérablement l'efficacité du système.

Typiquement, l'indexation classique consiste à sélectionner une liste de mots-clés (ou de termes [4]) extraits du texte [vR79]. Les termes sont *lemmatisés*. Cette lemmatisation est une opération indispensable permettant de représenter, et donc par la suite d'identifier de manière univoque un même terme exprimé sous des formes différentes. Ainsi, des mots comme "indexation", "indexer", "indexé" et "indexées" sont représentés par une unique racine "index", appelée *lemme*. Le plus souvent, il s'agit simplement d'enlever les suffixes parmi une liste connue. Par exemple, un document traitant de *la recherche d'informations sur les relations* est, dans un tel processus, indexé par

4. Certains auteurs parlent plutôt de *descripteurs [Bla90, Bru93, Lar94], notion que nous préférons puisqu'elle est plus proche d'un des objectifs de l'indexation à savoir la description du contenu des documents.*

les termes d'indexation *"recherche"*, *"informa"* et *"rela"*, et sera retrouvé à l'interrogation par la conjonction *"recherche"* ET *"information"* ET *"relations"*[5].

Une telle approche d'indexation a pour principal inconvénient la non prise en charge des relations qui peuvent exister entre les termes d'indexation au niveau des documents. Ainsi, une fois la tâche d'indexation finie, toute connaissance sur les liens sémantiques qui unissent les mots-clés est perdue. Par exemple, les deux mots-clés Garagiste et Voiture peuvent être combinés pour décrire des phrases comme *les Garagistes* **entretiennent** *les Voitures* ou comme *le Garagiste* **a réparé** *la Voiture*.

La représentation des relations au niveau de l'index permet aussi d'éviter certaines ambiguïtés pouvant altérer la précision des réponses fournies par le système. Dans une approche à base de mots-clés, un document comme *La politique de rigueur du gouvernement*, indexé par les descripteurs politique, rigueur et gouvernement, pourra être confondu avec un autre document portant sur *La rigueur du gouvernement dans sa politique* puisque ce dernier sera indexé par les même descripteurs. Pourtant, dans le premier cas il est question plutôt d'une austérité économique, alors que dans le second document il s'agit de s'intéresser à la rigueur du gouvernement dans sa politique globale, c'est-à-dire dans tous les domaines de sa compétence (social, affaires étrangères, économie, etc.).

Pour pallier à cette représentation trop pauvre, une solution possible consiste à utiliser des termes d'indexation plus riches contenant des relations sémantiques. Ainsi, dans le système RIME [Ber88, Ber90, Nie90], un formalisme proche des dépendances conceptuelles de Schank [Sch72] a été utilisé comme langage d'indexation. Il s'agit des *arborescences sémantiques* dont le but est d'expliciter les relations sémantiques entre les divers concepts. D'autres formalismes de représentation de connaissances ont été utilisés pour la modélisation des fonctionnalités de plusieurs systèmes de recherche d'informations. Par exemple, le logique terminologique MIRTL [MSST93, Seb94], les graphes conceptuels [Khe95, Mec95c] et la théorie des situations [Lal96]. Le choix de ces formalismes de représentation a été motivé par leur capacité à représenter des termes d'indexation riches et complexes, et par le fait qu'ils offrent la possibilité de représenter des relations entre objets.

La figure 1.2 montre cinq possibilités d'indexation pour une même expression: par des mots-clés (après lemmatisation), par des groupes nominaux, par des termes terminologiques, par des graphes conceptuels ou par des arborescences sémantiques. Les mots-clés sont extraits du texte et sont ensuite lemmatisés, les groupes nominaux suivent dans cet exemple la syntaxe proposée par Palmer [Pal90]. Le terme principal ici est "opacité", les termes qui lui sont dépendants dans l'expression à indexer étant indiqués entre parenthèses. Des relations syntaxiques peuvent être matérialisées par des mots liens (principalement des prépositions); Dans cet exemple, ce sont "à-det" et "de-det". L'arborescence sémantique présentée ici est celle du système RIME. L'arbre donné en exemple s'interprète comme suit: l'"opacité" a pour valeur "tissulaire" et porte sur un "poumon". La même expression peut être représentée en un graphe conceptuel dans le système ELEN [Che92] ou par un terme de la logique terminologique MIRTL.

Lorsque les termes engendrés par le processus d'indexation sont des concepts complexes où les connecteurs sont vus comme des relations sémantiques, nous les dénommons "*structure sémantique*". Une relation sémantique tient compte de la sémantique des concepts qu'elle relie et spécifie dans quelle mesure ils peuvent être reliés, de façon indépendante de la langue et de sa syntaxe. Par exemple, les concepts HUMAIN et LIQUIDE peuvent être lié par la relation *boire*, mais ne peuvent l'être par la relation *manger*.

Definition 1.2.1 *Une* **structure sémantique** *est une représentation synthétique d'une infor-*

5. Il est aussi question d'appliquer le même processus de lemmatisation à la requête lors de la phase d'interrogation.

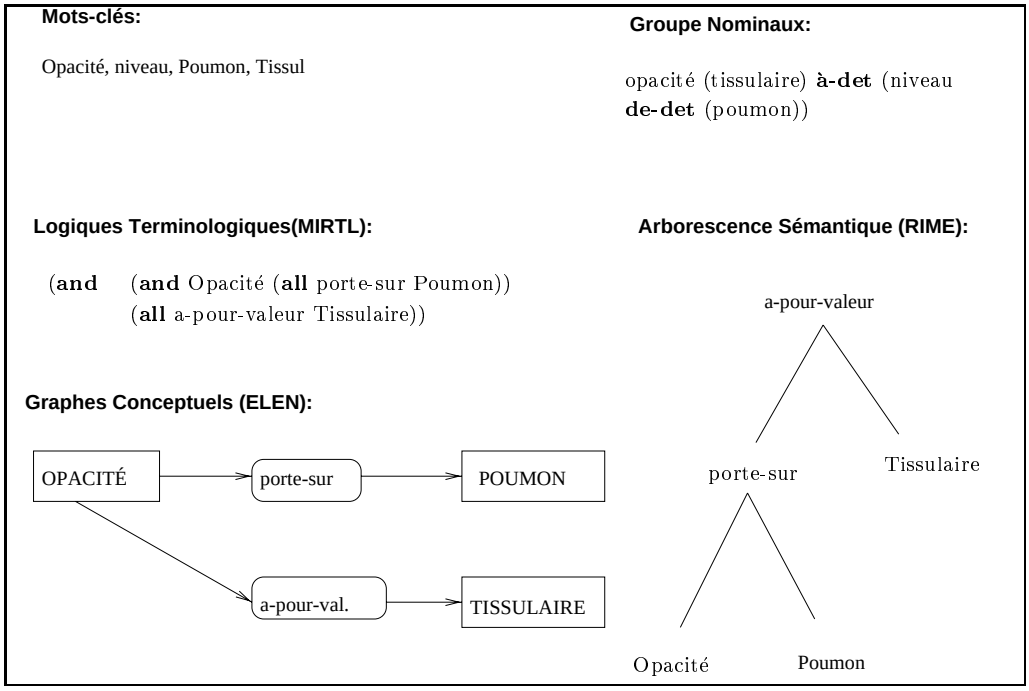


Figure 1.2. Exemples d'index pour "*une opacité tissulaire au niveau du poumon*"

mation contenue dans le document, ayant pour forme une expression "e_i rel *e_j", où* rel *est une relation sémantique et e_i, e_j sont des concepts simples ou des structures sémantiques.*

Les arborescences sémantiques de RIME, les graphes conceptuels et les termes des logiques terminologiques de la figure 1.2 constituent des structures sémantiques. À l'opposé, les relations dans les groupes nominaux matérialisent un lien syntaxique dans la langue naturelle.

En général, l'indexation peut être vue comme un processus de détermination de l'information jugée représentative du contenu d'un document. Quand l'information est représentée sous forme de structure sémantiques, alors nous parlons d'indexation relationnelle.

Definition 1.2.2 *L'***indexation relationnelle** *est le processus qui permet d'engendrer des structures sémantiques à partir de l'analyse du document.*

Un *modèle d'indexation* décrit comment représenter les index (langage d'indexation) et comment les extraire à partir des documents (indexation) [Par96]. Quand l'indexation est relationnelle, le modèle d'indexation qui en découle est dit *relationnel*.

L'adoption d'une indexation relationnelle engendre des structures sémantiques complexes véhiculant de nombreuses informations implicites. La pleine utilisation de ces structures lors du processus de recherche nécessite des traitements précis, mais également un ensemble de connaissances sur le domaine de l'application permettant la déduction de ces informations implicites. Les coûts de traitements relativement élevés associés aux langages expressifs s'expliquent principalement par ces deux considérations.

Au niveau des approches classiques d'indexation, l'évaluation d'une requête par rapport aux documents du corpus peut être fondée sur une comparaison stricte des documents et de la requête, c'est-à-dire sur l'identification des mots-clés de la requête avec ceux des documents, ou bien sur une comparaison plus souple, prenant en compte la sémantique de la requête et celle des documents. Le second cas se traduit le plus souvent par l'utilisation d'un *thésaurus* mentionnant les relations sémantiques liant certains des mots-clés. Les relations les plus souvent considérées sont les relations d'*équivalence*, les relations de *spécialisation/généralisation* et les relations de *voisinage*. Le thésaurus est apparié dans ce cas à la base de connaissances du système.

Si, dans la pratique, une telle démarche semble être suffisante quand les termes d'indexation correspondent à des mots-clés, la description du contenu des documents par des structures sémantiques impose la prise en compte d'autres connaissances sur le domaine. Ainsi, les propriétés des relations présentes dans ces structures doivent être modélisées au niveau de la base de connaissances si l'on souhaite détecter toutes les informations implicites contenues dans les index. La transitivité, la symétrie, ou encore les caractéristiques sémantiques des relations utilisées doivent être explicitement mentionnées dans la base de connaissances. Seulement, il est évident que plus ces propriétés sont finement identifiées, plus les raisonnements sur ces structures seront subtils et complexes.

Très peu de formalismes de représentation de connaissances permettent un raisonnement efficace sur les relations. Les logiques terminologiques les plus expressives autorisent la prise en compte d'une très faible proportion de propriétés de relations. Les raisonnements sont d'ailleurs d'une complexité exponentielle [LB84, Neb90b]. Dans le modèle de base des graphes conceptuels [Sow84], les propriétés des relations ne sont pas traitées.

Si l'on admet l'hypothèse que l'utilisation des structures sémantiques est nécessaire pour l'obtention de systèmes orientés vers la précision des réponses, nous devons trouver une réponse aux trois questions suivantes:

- Quels types de connaissances prendre en compte?
- Comment les traitements de ces connaissances peuvent-ils s'effectuer?
- Comment faire pour que ces traitements soient d'un coût raisonnable?

La réponse à toutes ces questions devra se faire en tenant compte des nouvelles exigences de la recherche d'informations, à savoir la prise en compte des données multimédias, la recherche de documents structurés, les hypermédias et autres. À travers une étude de toutes ces nouvelles applications, nous avons cherché à cerner des éléments de réponse à ces questions.

1.3 Contenu de la Thèse

Nous défendons ici la thèse qu'en plus des termes simples il est nécessaire de représenter les *relations* entre ces termes, surtout dans le cas des documents multimédias. Nous montrons comment les connaissances sur les caractéristiques syntaxiques et sémantiques de ces relations permettent d'améliorer les résultats de la fonction de correspondance tout en autorisant une approche orientée vers la précision des réponses.

Nous présentons alors un cadre général permettant l'étude et l'analyse des caractéristiques de ces relations indépendamment d'un formalisme particulier. Il s'agit d'un modèle d'indexation

relationnel précisant les principes généraux de manipulation de ces relations et la prise en compte de leurs propriétés dans un processus de recherche. Nous montrons alors que ces propriétés sont reliées à des décisions de pertinence. Un raisonnement sur les relations revient ainsi à l'établissement de jugements de pertinence. L'idée défendue ici est que l'analyse des décisions de pertinence d'un système peut se faire à la base de ces relations et leurs propriétés. Notre cadre général est basé sur une approche logique, du type de celle définie dans le cadre du modèle logique, où la notion de pertinence est vue comme une inférence logique.

Le modèle logique a été introduit par van Rijsbergen [vR86] comme un modèle unificateur ou un *méta-modèle*. Il est possible d'exprimer les autres modèles classiques de recherche en instanciant ce modèle [Nie90]. Le modèle logique est fondé sur la logique. Cette dernière présente tous les outils nécessaires à la modélisation des diverses fonctions d'un système de recherche d'informations, outre le fait qu'elle fournit un modèle de représentation de l'information plus précis que ceux généralement utilisés par les approches classiques de recherche d'informations. Une approche logique permet aussi d'expliquer le processus de recherche et de déterminer pourquoi une telle réponse a été jugée pertinente par le système. Une évaluation théorique du système de recherche adjacent est ainsi possible.

Dans un système de recherche d'informations fondé sur cette approche, un document peut avoir une sémantique formelle décrite à travers un ensemble d'axiomes. Chaque axiome décrit ou caractérise une partie du contenu de ce document. D'autre part, en logique, un modèle est une interprétation dans laquelle tous les axiomes sont vérifiés. Dans ce contexte, un document d forme un modèle de ses axiomes associés $A(d)$. On note ceci par $d \models A$ et on parle d'inférence sémantique. Dans les théories formelles, une formule bien formée f peut être déduite, ou prouvée, à partir d'un ensemble d'axiomes A par application de ce qu'on appelle les règles d'inférence ou de dérivation. Pour désigner le fait qu'une formule dérive (ou est prouvée à partir) de A, on note $A \vdash f$ et on parle d'inférence syntaxique.

La formulation du modèle logique de Rijsbergen fut délibérément abstraite, puisqu'elle ne spécifiait pas le type d'inférence ou la logique qu'il convient d'utiliser. L'opinion que nous défendons ici est que la notion de pertinence est plutôt liée à une notion de dérivabilité (inférence syntaxique). Les propriétés des relations et leurs caractéristiques sont ainsi présentées à travers un système de dérivation relationnelle, où chaque règle de dérivation correspond à une prise de décision de pertinence. Afin de définir des règles de dérivation concrètes, nous nous sommes appuyés sur des applications récentes de recherche d'informations. Un bon système devra inclure le maximum de ces règles d'inférence.

Dans cette thèse nous proposons ainsi de spécifier les caractéristiques d'une "bonne" logique pour la recherche d'informations. Nous définissons en effet le degré d'expressivité que doit fournir cette logique et les règles de dérivation nécessaires à l'établissement des décisions de pertinence qui s'imposent. Ces règles s'appuient sur un ensemble de connaissances dont nous nous proposons de spécifier le contenu. En particulier, nous montrons que les relations et leurs propriétés sont à la base de toute décision de pertinence. Notre logique est une instanciation *adéquate* du modèle de Rijsbergen. Ainsi, ses caractéristiques devront être supportées par n'importe quel formalisme opérationnel dédié à l'implantation d'un système de recherche orienté vers la précision des réponses et fondé sur une approche logique.

En général, il est difficile d'établir qu'une logique est plus adéquate qu'une autre. En effet, pour une comparaison objective, il faudrait spécifier les fonctionnalités nécessaires pour une bonne recherche. Si nous nous autorisons à qualifier ainsi notre logique, c'est que nous avons identifié ces

fonctionnalités, à travers une étude des nouvelles exigences de la recherche d'informations.

Cette logique constitue un modèle pour la recherche d'informations puisqu'elle spécifie les décisions de pertinence à prendre et peut guider la construction d'un système opérationnel. Si nous avons préféré dans cette thèse parler de *modèle d'indexation relationnel*, c'est pour mettre en relief l'importance des relations et leurs propriétés dans la logique proposée.

Des formalismes comme les logiques terminologiques ou les graphes conceptuels peuvent être étendus pour inclure les caractéristiques de cette logique. Cependant, afin d'aboutir à des systèmes opérationnels, il est nécessaire d'avoir des algorithmes polynomiaux et décidables. L'expérience a ainsi montré qu'une implantation à base de démonstrateurs de preuves n'est pas efficace dans un contexte de recherche d'informations. Nous soutenons ici l'idée qu'une bonne interprétation logique du formalisme sous-jacent conduit à une implantation algébrique efficace.

Dans cette thèse, nous nous sommes intéressés au formalisme des graphes conceptuels. Le modèle sur lequel se base ce formalisme a été étendu de manière à ce qu'il puisse s'accommoder d'une indexation relationnelle, où la représentation des propriétés des relations et leur gestion sont nécessaires. L'application de notre démarche à ce formalisme est censée discuter les potentialités et les limites de ce formalisme en termes de décisions de pertinence. Pour une implantation efficace, nous associons une interprétation logique à ce formalisme. Cette interprétation élimine les ambiguïtés du modèle de base et autorise une recherche rapide sur le corpus.

Grâce à cette application, nous définissons un modèle d'indexation relationnel pour les graphes conceptuels. Un des avantages de ce modèle est qu'il présente une extension d'un formalisme expressif, tout en maintenant une complexité polynomiale des traitements.

1.4 Organisation de la Thèse

La suite de cette thèse est organisée comme suit. Le chapitre 2 discute la notion de *relation* dans un modèle d'indexation. Il présente les raisons pour lesquelles il convient d'opter pour une indexation relationnelle. L'information est vue ici comme une structure sémantique constituée d'un ensemble de concepts complexes où les connecteurs sémantiques sont vus comme des relations. Des applications récentes en recherche d'informations montrent combien il est utile d'avoir un mécanisme pour la gestion des relations et de leurs propriétés. Ce chapitre montre les possibilités et les limitations des formalismes de représentation de connaissances dans leur capacité à explorer les propriétés des relations et leur sémantique.

Le chapitre 3 propose un modèle d'indexation relationnel pour la recherche d'informations. Ce modèle se base sur la théorie des situations comme langage de représentation, et sur un système de dérivation de pertinence qui, reprenant ce langage, spécifie les décisions de pertinence qu'il est nécessaire d'établir. Ce système permet principalement de guider le processus de correspondance en inférant des nouvelles informations implicites dans les documents à partir des propriétés des relations et leur sémantique. Le système de dérivation est étendu par des règles par défaut permettant de raisonner sur les informations des documents dans un contexte particulier. Ce chapitre décrit enfin comment ce modèle d'indexation peut être enrichi par la notion de *facette*, permettant de représenter le plus fidèlement possible les documents complexes (images, vidéos, documents structurés, ...). Chaque facette correspondra alors à une vue du document selon laquelle il pourra être décrit par le biais d'un ensemble homogène de termes engendrés lors du processus d'indexation relationnelle.

Le modèle proposé pourra être utilisé afin de pallier aux limitations de certains formalismes de représentation de connaissances. Ils pourront être ainsi étendus par un mécanisme de gestion de relations dont les propriétés sont utiles dans un processus de recherche. Au chapitre 4, nous proposons les graphes conceptuels comme un modèle opérationnel pour notre démarche. Nous décrivons l'interprétation logique que nous associons à ce formalisme. Cette interprétation, tout en étant conforme au formalisme originel, permet l'implantation rapide du modèle d'indexation relationnel développé. Enfin, au chapitre 5, nous discutons d'une application de notre modèle opérationnel à un corpus d'images.

Chapitre 2

Pourquoi les Relations?

L'idée d'établir des relations entre des objets et de les représenter dans la base de connaissances d'un système informatique n'est pas récente[1], comme le signale Sabah [Sab88] en se référant à la notion de réseau sémantique en Intelligence Artificielle. Cependant, la plupart des formalismes utilisés pour leur représentation souffrent de certains défauts liés à l'absence d'une méthodologie précisant les principes généraux de manipulation de ces relations et le comportement qui leur est associé. L'effort en Intelligence Artificielle s'est surtout porté sur les concepts et leurs propriétés.

En recherche d'informations, les traitements de ces relations ne sont guère plus élaborés. Les techniques préconisées par les systèmes ne tirent pas vraiment profit des connaissances et des propriétés implicitement véhiculées par ces relations. L'impact de leur prise en compte lors de la phase de correspondance n'a jamais été formellement défini. À l'image de l'Intelligence Artificielle, les approches classiques en recherche d'informations se contentent d'explorer les propriétés sémantiques des termes d'indexation (mots-clés) décrites à l'aide d'un *thésaurus* et se soucient peu de celles relatives aux relations.

Pour plusieurs raisons, il arrive que nous ayons aussi besoin d'expliciter des propriétés sur les relations établies entre certains objets. Le but de ce chapitre est de cerner les cas de recherche d'informations où il est souhaitable de spécifier les caractéristiques des relations et leur comportement sémantique, de manière à ce que la fonction de correspondance puisse les exploiter. Notre démarche consiste donc à présenter ces cas et dégager ce que doit permettre un système comme inférences sur les relations.

Nous introduisons d'abord les raisons pour lesquelles l'utilisation des approches classiques d'indexation ne peut conduire à la conception de systèmes orientés vers la précision des réponses. Nous présentons ensuite les travaux s'étant intéressés à l'introduction de nouvelles approches, visant essentiellement la représentation des relations dans l'index. Nous montrons ensuite l'importance d'avoir des mécanismes d'inférence sur ces relations. Ces dernières sont enfin examinées du point de vue de la recherche d'informations d'aujourd'hui. Cet examen montre l'importance de raisonner sur les relations en vue d'une recherche efficace et performante. Nous concluons enfin sur la nécessité d'une approche théorique générale pour la représentation et le raisonnement sur les relations.

1. Elle date de 1956 avec la "Carte du Tendre" de Mademoiselle de Scudery.

2.1 Les Descripteurs en Recherche d'Informations

En recherche d'informations, un terme d'indexation peut prendre plusieurs formes, la forme la plus élémentaire étant le mot-clé ou le descripteur. La disponibilité de nombreux algorithmes efficaces [Sal89], permettant l'extraction automatique des mots-clés dans les textes, explique la large utilisation de cette forme d'indexation particulière dans la communauté de recherche d'informations. Ces algorithmes adoptent tous la même approche, à savoir l'extraction des "*bons*" descripteurs du document. Par *bons descripteurs*, nous désignons les termes fortement représentatifs d'un très petit nombre de documents [Sal71, SM83]. L'hypothèse suivante est ainsi généralement considérée pour l'identification de tels descripteurs:

Hypothèse 1 (Le Bon Descripteur) *La représentativité d'un descripteur par rapport au document est d'autant plus meilleure que cet objet apparaît relativement de façon fréquente dans le document et de façon très restreinte dans les autres documents.*

Comme dans [Sal71], dénotons par FreLoc(t, d) la fréquence d'occurrence du terme t dans le document d. Désignons par FreCor(t, $\mathcal{D}$), la fréquence du terme t dans le corpus des documents $\mathcal{D}$. Selon les valeurs que ces deux fréquences peuvent prendre, un terme d'indexation t sera jugé bon ou pas. Nous distinguons quatre possibilités présentées dans le tableau suivant:

		FreLoc(t, d)	
		grande	petite
FreCor(t, $\mathcal{D}$)	grande	mot vide	Non applicable
	petite	**bon**	Non utilisable

Table 2.1. Détection des "bons" index

La première possibilité (haut gauche) indique que le terme t apparaît fréquemment à la fois dans le document et dans le corpus. Si t est utilisé comme terme d'indexation, alors pratiquement tous les documents seront retournés à l'utilisateur. La discriminance du terme t par rapport au corpus est ainsi très faible. Le plus souvent, il s'agit des termes vides [2], ou des termes trop spécifiques au corpus, comme par exemple "information" dans un corpus sur les systèmes d'informations.

La deuxième possibilité (bas droite) est opposée au cas précédent. Le terme t est un bon discriminant du document par rapport au corpus, mais il n'est pas possible de l'utiliser puisqu'il n'est pas un bon représentant du contenu du document. L'utilisabilité du terme d'indexation est en effet proportionnelle à l'importance du terme dans le document, c'est-à-dire qu'elle dépend de sa fréquence interne dans le document.

Le troisième cas (bas gauche) indique que le terme t est à la fois un bon discriminant du document par rapport au corpus (la fréquence globale FreCor(t, $\mathcal{D}$) est petite) et un bon représentant du contenu du document (la fréquence interne FreLoc(t, d) est grande). Ainsi, à la lumière de l'hypothèse 1, t est un bon terme d'indexation pour d.

2. On parle aussi de mots communs comme "le", "de", "au", etc. D'un point de vue sémantique, ces mots sont dépourvus de sens.

Dans la réalité, il n'est généralement pas possible de rencontrer le quatrième cas (haut droite), surtout dans des corpus homogènes. Les chances de rencontrer un terme dont la fréquence d'apparition globale est très forte alors que sa fréquence interne dans le document est faible sont pratiquement négligeables. Dans tous les cas, ce terme ne sera pas un bon index à cause des mêmes raisons évoquées pour les mots vides.

Il est facile d'affecter un poids w à un terme t. À la base de ce poids, la décision de considérer t comme un bon descripteur ou pas sera prise. La formule adoptée est la mesure classique de *tf.idf* (*tf* pour «term frequency», et *idf* pour «inverse document frequency»):

$$w(t,d) = \mathrm{FreLoc(t,\ d)} \log_2 \frac{\mid \mathcal{D} \mid}{\mathrm{FreCor(t,\ \mathcal{D})}}$$

La formule ci-dessus véhicule bien le sens de l'hypothèse 1. Le poids d'un terme d'indexation est en effet proportionnel à sa fréquence d'occurrence interne dans le document (term frequency), autrement dit à sa représentativité dans le document, et inversement proportionnel à sa fréquence globale dans le corpus (inverse document frequency), permettant la prise en compte de son aptitude à discriminer le document dans le corpus.

L'avantage de l'approche d'indexation à base de descripteurs réside dans le fait qu'elle est simple et efficace. Elle présente toutefois un inconvénient majeur: l'indexation réduit le contenu du document à un ensemble de descripteurs sans exprimer de relations entre ces descripteurs. En effet, à l'intérieur du document, ces termes apparaissent dans un contexte particulier qui n'est plus repris dans l'index. Le descripteur est ainsi arraché de son contexte. Pourtant, ce dernier peut être crucial lors de l'élaboration de la décision de pertinence. Malgré cet inconvénient, l'approche d'indexation par mots-clés reste, et de loin, la plus répandue et la plus utilisée dans la communauté de recherche d'informations.

2.1.1 Les Problèmes de l'Indexation par Mots-Clés

Dans le chapitre précédent, nous avons déjà recensé des exemples où l'indexation par mots-clés pose un certain nombre de problèmes. Les carences de cette forme d'indexation peuvent être expliquées par des arguments linguistiques liés à la notion d'ambiguïté [Lar94]:

- Une ambiguïté due à la polysémie: un mot-clé peut présenter une ambiguïté morphologique ou lexicale. Le mot "ferme" dans un index peut être un verbe, un adjectif, ou un nom. De plus, un mot peut renvoyer à plusieurs contextes. Par exemple, le mot-clé "cellule" peut se référer à plusieurs domaines: "cellule nerveuse", "cellule de prison", "cellule de crise", etc.
- Une polysémie due à l'absence de syntaxe: une indexation par l'expression "*production de pétrole*" est plus riche que celle effectuée par la combinaison des deux mots-clés "production" et "pétrole". Elle distingue mieux le contenu sémantique du document. En effet, la sémantique de cette expression ne peut être correctement exprimée par la conjonction de ces deux mots-clés. Dans le modèle Booléen [vR79, SM83], on assimile les prépositions à des connecteurs logiques lors de la phase d'indexation. La précision des réponses se voit donc altérée, puisque les informations de type syntaxique (les prépositions dans ce cas) peuvent guider la recherche.

Une conséquence directe de ces deux types d'ambiguïtés est que les performances du système en termes de précision et de rappel sont telles qu'il n'est pas possible d'envisager les mots-clés comme

langage d'indexation expressif, précis et robuste. La syntaxe concourt ainsi à la détermination du sens, et ne peut être négligée comme outil d'indexation des textes. Ceci explique le recours à des approches linguistiques visant à aboutir à des descripteurs plus précis.

2.1.2 Les Approches Linguistiques

Ces approches ont en commun le fait qu'elles s'intéressent toutes au statut linguistique du descripteur. La compréhension des textes en langue naturelle est composée de deux modes de signifiance: le mode sémiotique et le mode sémantique [Gue82]. Le mode sémiotique concerne le signe linguistique. Il est en rapport avec les mots de la langue, appelés *lexèmes*, et il doit être reconnu. Le mode sémantique concerne un contenu et il doit être compris. Ce dernier est véhiculé par des parties du discours ou *syntagmes*. Ces syntagmes expriment un contenu et sont donc appariés à des *concepts*.

Les syntagmes sont représentés par les relations qui existent entre les mots de la langue. De ce fait, ils ne peuvent être déterminés qu'après une analyse syntaxique du texte. La syntaxe établit ainsi un lien référentiel entre lexèmes et concepts. La question qui se pose alors est la suivante: "*quel est le statut linguistique du syntagme*". Elle se traduit par la problématique suivante:

> Quelles sont les syntagmes qui participent le plus à la thématique du document et qui peuvent être ainsi utilisés, aussi bien pour la description du contenu des documents que pour l'expression de la requête?

Dans la littérature [Ker84, Pal90, Lar94], le choix s'est souvent porté sur les syntagmes nominaux. Les approches linguistiques en recherche d'informations s'appuient sur les études montrant la relative concordance entre syntagmes nominaux et thèmes d'une part, et syntagmes verbaux et rhèmes d'autres part [Par96]. Le thème indique de quoi on parle, et le rhème ce qu'on en dit. Il a été ainsi admis dans [Pal90], que puisque dans une approche de recherche il est plutôt question des thèmes, les syntagmes nominaux sont mieux à même à décrire le contenu des documents.

Nous verrons dans la suite de ce chapitre que ce choix est assez restrictif. Une bonne description du document passe nécessairement par la prise en charge de certaines relations (des verbes entre autres) particulières qui dépassent la représentation par syntagmes nominaux. Les rhèmes doivent aussi être représentés si l'on veut favoriser une recherche d'informations précise où seuls les documents pertinents doivent être retournés à l'utilisateur. Supposons ainsi qu'un utilisateur pose la requête suivante: "*Je cherche des situations dont lesquelles les Sioux ont été battus par la cavalerie fédérale*". Considérons également le document contenant le texte suivant:"*En 1876, les Sioux ont battu la cavalerie fédérale à Little Big Horn*". Si l'on considère une approche d'indexation à base de syntagmes nominaux (thèmes), le document sera représenté par les termes d'indexation suivants:

- «les Sioux»
- «la cavalerie fédérale»
- «Little Big Horn»

Si l'on applique le même processus à la requête, cette dernière sera représentée par les syntagmes «les Sioux» et « la cavalerie fédérale». Le document sera ainsi retrouvé pour cette requête, même

s'il n'est pas précisément à-propos de ce que cherche l'utilisateur. Ce document sera également pertinent à n'importe quelle autre requête contenant les syntagmes «les Sioux» et « la cavalerie fédérale» reliés par n'importe quel verbe. La prise en compte des syntagmes verbaux permet ainsi d'améliorer les performances qualitatives du système.

2.1.3 Utilisation d'un Thésaurus

Les problèmes liés au vocabulaire d'indexation font que les performances du système de recherche ne sont pas à la mesure des attentes. Plusieurs termes d'indexation peuvent désigner un même objet ou un même concept. Il devient alors difficile de retrouver les documents correspondant à une requête s'ils n'emploient pas les mêmes termes que cette dernière. Plusieurs solutions sont possibles, la plus connue étant l'utilisation d'un thésaurus.

Un thésaurus recense les différents termes d'indexation utilisables dans une application et les relations sémantiques qui les lient. D'une application à une autre, les termes présents dans un thésaurus varient énormément en fonction du domaine traité et du niveau de granularité choisi pour la représentation des termes d'indexation (mots-clés, groupes de mots, etc.). On peut comparer les thésaurus aux réseaux sémantiques. Cependant à l'inverse de ces derniers, les thésaurus se limitent à des liens définissant des *relations lexicales* entre les noeuds [Sab88, SDV95, Bri95]. On distingue les catégories de relations lexicales suivantes:

- les relations d'*analogie*. Elles servent à établir des liens entre synonymes, antonymes ou entre verbes converses. Par exemple, «voiture» et «automobile» sont synonymes, «chaud» et «froid» sont antinomiques et les deux verbes «vendre» et «acheter» sont converses [Fil88].
- les relations de *dérivation*. Elles lient deux termes de catégories différentes dont les sens sont identiques ou dérivables l'un de l'autre. Par exemple, «aimer» et «amour», ou bien «école» et «scolaire».
- les relations de *co-occurrence*. Elles précisent, dans un contexte particulier, des liens idiomatiques entre deux termes. Par exemple, il existe généralement une relation de co-occurrence entre «question» et «poser».
- les relations d'apport *sémantique*. Ces relations indiquent, lorsque deux termes sont en relation, quelles informations sémantiques sont apportées à l'un par rapport à l'autre. Par exemple, «cheval» est un spécifique de «animal», alors que «fruit» est un générique de «pomme».

Les thésaurus peuvent intervenir au moment de l'indexation des documents et lors de l'évaluation des requêtes. Les thésaurus hiérarchiques (c'est-à-dire les thésaurus restreints à des arborescences définies par des relations sémantiques de généralisation) s'avèrent utiles lors de l'indexation afin de classer les documents dans une hiérarchie de concepts [CY92]. Dans tous les cas, les thésaurus permettent l'élargissement de la portée des requêtes, améliorant ainsi les performances du système.

Les thésaurus sont généralement construits de façon manuelle par les spécialistes du domaine de l'application. Cependant, étant donné les coûts élevés d'une telle approche, on se tourne de plus en plus vers des méthodes de construction automatiques [PW91, CL92, Gre92]. Ces dernières se basent sur des mesures statistiques de co-occurrence de termes. Les premières expériences sont encourageantes malgré la difficulté de les appliquer à des grands corpus.

Actuellement, plusieurs thésaurus sont disponibles dans le commerce, le plus connu étant Wordnet [Mil96]. Si nul ne doute que le thésaurus est d'un apport considérable pour une recherche efficace, il n'en reste pas moins que la démarche est non exhaustive puisqu'elle se limite aux seules relations lexicales.

2.1.4 L'Indexation Relationnelle

Les stratégies classiques d'indexation s'intéressent à l'extraction des mots-clés ou des thèmes à partir des documents, puis à leur évaluation selon leur importance dans le document. Le principal inconvénient de cette approche est qu'elle néglige certaines relations importantes entre les mots-clés ou les thèmes. Pourtant, ces relations peuvent être d'une grande utilité lors du processus de correspondance, étant donné les informations qu'elles véhiculent. Ainsi, les relations entre objets jouent un rôle sémantique important. Dans les domaines des bases de données relationnelles [BCHL93] ou de l'ingénierie des systèmes d'informations (cf. Merise [NECH92]), l'importance des relations est largement reconnue.

Forme de la relation	*Nbre. d'Occ*	*fréquence*
$Entity_A$ **on** $Entity_B$	2680	0.0128
$Entity_A$ **in** $Entity_B$	2598	0.0124
$Entity_A$ **of** $Entity_B$	2097	0.0100
$Entity_A$ **for** $Entity_B$	1771	0.0084
the $Entity_A$ **on** the $Entity_B$	190	0.00091
the $Entity_A$ **in** the $Entity_B$	403	0.0019
$Entity_A$ **Without** $Entity_B$	82	0.00039
The paper **deals with/describes** *topic*	29	0.00013

Table 2.2. Fréquences de certaines relations

Dans la collection test de CACM, nous avons noté une forte fréquence d'occurrence de certaines relations. Après analyse, nous avons converti cette collection en une représentation homogène incluant uniquement les champs "titre" et "résumé". Le nombre de mots dans cette collection a été alors calculé. Nous avons ensuite cherché à compter le nombre d'apparitions de certaines séquences de mots incluant des relations. Par exemple, en posant une requête de type "un #paramètre *rel* #paramètre", nous mentionnons le fait que nous cherchons toutes les séquences de la collection commençant par le mot "un" suivi de n'importe quelle chaîne de caractères, elle-même suivie par le mot relation "*rel*" et finissant par n'importe quelle chaîne de caractères. Quelques exemples de ces requêtes et leurs résultats sont donnés dans la table 2.2. Les fréquences sont calculées par rapport au nombre total des mots de la collection.

À la marge de ces résultats, il apparaît clair que certaines relations doivent être prises en compte lors du processus d'indexation. On parle alors d'indexation relationnelle. En recherche d'informations, une telle démarche a été initiée par les travaux de Farradane [Far80a, Far80b]. Ce dernier est le premier à proposer une classification de relations dont la représentation est souhaitable lors de

l'indexation. Une variante récente de l'approche de Farradane a été proposée par Bruza [Bru93].

a) L'Approche de Farradane

L'origine philosophique de cette approche réside dans le fait qu'une grande partie du sens d'une information est contenue dans les relations entre termes. Prenons par exemple la phrase suivante:

Jean **a bu** un verre

Alors que dans l'approche traditionnelle par mots-clés, cette phrase pourrait être indexée par les deux mots-clés John et *verre*, Farradane insiste sur la nécessité de représenter aussi la relation *boire* entre ces deux mots-clés. Dans cet exemple particulier, l'indexation de la phrase avec John et *verre* peut même être considérée comme erronée, puisque naturellement *John* boit plutôt le contenu du verre (de l'eau, du vin, du coke, etc.) et non pas le verre lui-même. Cela confirme encore la remarque que l'approche à base de mots-clés ne conserve pas le contexte d'apparition de ces mots. En l'occurrence, il s'agit ici du mot-clé verre. La représentation de la relation *boire* permet ainsi d'éviter une telle ambiguïté sémantique, puisque nous savons tous qu'on ne boit pas des verres mais seulement le liquide qu'ils contiennent. Un parallèle peut être ainsi dressé avec les modèles conceptuels dans les bases de données. La description d'un document par le seul biais des mots-clés serait comparable à un modèle entité-association sans les relations.

Les performances mitigées des systèmes engendrés par l'utilisation des approches linguistiques ou des thésaurus a motivé la démarche de Farradane. L'échec de ces approches réside dans le fait qu'un texte contient plus de relations que celles considérées par les approches en question. Les autres tentatives d'inclusion des nouveaux types de relations ont été faites de façon trop ad-hoc pour être généralisées. Farradane leur reproche ainsi d'être uniquement applicable dans le domaine pour lequel elles furent conçues.

	Mécanisme Associatif		
Conceptualisation	*Conscience*	*Association Temporaire*	*Association Fixe*
Simultanée	Juxtaposition	Activité propre	Association
Concept non-distincts	Équivalence	Dimensionelle	Appartenance
Concept distincts	Substitution	Action	Dépendance fonctionnelle

Table 2.3. Les catégories des relations [Far80a]

Farradane propose une démarche plus générale où neuf types de relations primitifs sont introduits[3] (voir table 2.3). À travers ces types, n'importe quelle relation peut être classifiée. Par exemple, l'énoncé *un écrivain* **écrit** *un livre* indique une relation de dépendance fonctionnelle entre écrivain et livre. Le choix des neufs classes primitives se base sur une étude des mécanismes de pensée psychologiques[4].

3. Plus exactement neuf catégories de relations.
4. Pour une présentation plus détaillée de l'argumentation de Farradane, voir [Far80a, Far80b].

La généralité de cette démarche peut facilement être vérifiée. Par exemple, les relations de co-occurrence au niveau d'un thésaurus peuvent être exprimées à travers le type de relation *Juxtaposition*[5]. Ainsi, *Juxtaposition*(Commandant, Cousteau), exprime le fait que l'information à-propos de "Commandant" apparaît en présence de l'information à-propos de "Cousteau" dans le corpus.

Les index résultants de cette approche sont sémantiquement plus riches que ceux obtenus par les techniques traditionnelles d'indexation. Ils capturent de façon plus exhaustive le contenu informationnel des documents. L'approche présente toutefois l'inconvénient d'une indexation manuelle. En effet, il n'existe aucune spécification formelle indiquant comment procéder à la description des documents. Ainsi, il est difficile de décider par exemple si un terme est ou non en relation de *Juxtaposition* avec un autre terme. C'est probablement la raison pour laquelle l'indexation relationnelle de Farradane n'est pas utilisée.

b) Les Expressions d'Index

Bruza [Bru93] présente ses travaux comme une variante de l'approche de Farradane. Il introduit ce qu'il appelle les expressions d'index[6]. Une expression d'index consiste en une *structure* contenant un nombre de termes séparés par des connecteurs modélisant les relations entre ces termes. Les termes sont pris à partir d'une liste prédéfinie et correspondent à des syntagmes nominaux. Les connecteurs sont spécifiés dans un ensemble restreint aux prépositions de la langue anglaise et à un connecteur spécial, noté $\circ$, appelé *connecteur nul*. Ce dernier permet d'exprimer des associations de termes dans des phrases comme par exemple *Homme* $\circ$ *Blanc* pour un *homme blanc*. La table 2.4 suivante montre un certain nombre de connecteurs utilisés par Bruza.

Connecteur	*Type de relation*	*Exemples*
of	Possession	Castle of queen
	Action	Pollination of crops
in, on, etc.	Position	Trees in Guardens
with, $\circ$	Association	Assistance with problems
		fruit $\circ$ trees

Table 2.4. Table de connecteurs tirée de [Bru93]

L'avantage de l'approche de Bruza est que le processus d'indexation peut se faire de façon automatique. La technique utilisée s'appuie sur celle proposée par Craven [Cra86]. Il s'agit d'indexer les documents à travers leurs titres, ceux de leurs sections, de leurs sous-sections, de leurs figures, etc. Une fois les mots vides supprimés, comme par exemple "*the*" et "*a*", les parties qui restent sont successivement analysées afin de leur attribuer une interprétation donnant lieu à une structure d'expression d'index. La structure résultante devra correspondre aux interprétations qu'aurait donné la majeure partie des utilisateurs, sémantiquement parlant cela s'entend. L'hypothèse de base pour la détection des structures est de considérer les connecteurs comme des opérateurs ayant une certaine *priorité*. Cette dernière est utilisée pour décider si la structure courante est à approfondir

5. En Anglais *Concurrence*.
6. En anglais *The index expressions*.

ou à élargir. Bruza utilise ainsi deux niveaux de priorité sur les connecteurs. Ils sont basés sur le fait que certains connecteurs lient les termes plus fortement que d'autres. Ceux qui lient le plus les termes dirigent la structure vers la profondeur. Ainsi le connecteur "*from*" a plus de priorité que le connecteur "*of*" selon la conception de Bruza. Un exemple de construction d'une structure d'expression d'index est donné dans la figure 2.1 suivante.

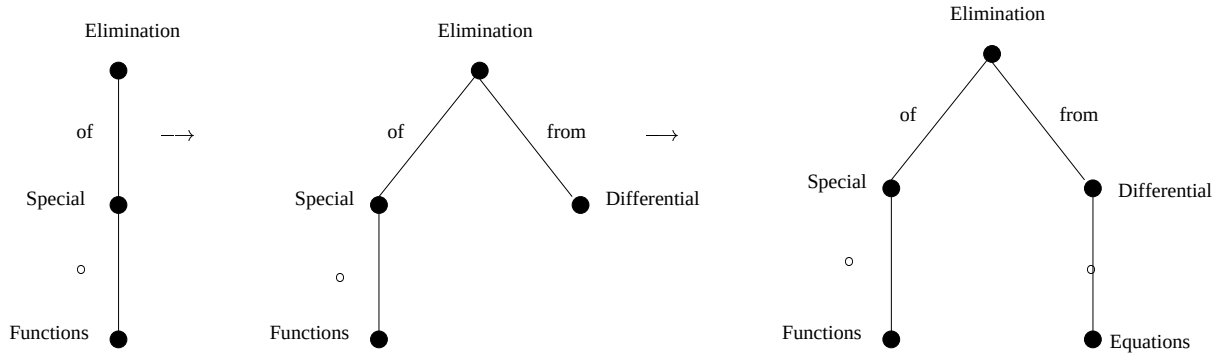


Figure 2.1. Une détection d'une structure d'une expression d'index pour le titre "The Elimination of Special Functions from Differential Equations (exemple tiré de [Bru93])"

Si au niveau de l'indexation, l'application de cette approche semble donner des résultats satisfaisants, son inconvénient est qu'au niveau de la correspondance, il n'est pas possible de détecter la similarité entre une expression d'index comme "*page in a book*" et une autre comme "*page of a book*". Dans cet exemple le connecteur "*in*" dans la première expression a la même sémantique que le connecteur "*of*" dans la seconde. En effet, ils expriment tous les deux une relation de composition *part-of* entre "page" et "book". De même il n'est pas possible de détecter la similarité entre la phrase *Adventures of Tom* et *Tom's Adventures*. Aucune analyse sémantique visant à reconnaître le sens véhiculé par les expressions d'index n'est ainsi établie.

2.1.5 Vers des Systèmes de Recherche Conceptuels

Le grand avantage des mots-clés réside dans leur simplicité. Cela permet d'avoir des algorithmes de recherche très performants. De plus, l'approche par mots-clés peut facilement être appliquée à tout type de document et à n'importe quel domaine. Cependant, une telle représentation est trop pauvre, et doit donc être exclue pour les systèmes où la précision est importante. D'une part, l'hypothèse consistant à dire qu'il existe une bivalence entre les mots-clés et la sémantique est souvent fausse à cause des divers sens possibles que peut avoir un mot-clé. D'autre part, les relations entre termes ne sont pas modélisées.

Les représentations par groupe nominaux ou par expressions d'index possèdent une meilleure précision que les mots-clés, et sont indépendantes du domaine de l'application. La condition qu'il faut respecter ici est de produire des groupements reflétant la compréhension mentale des utilisateurs. Cependant, dans certains cas l'analyse syntaxique est insuffisante sans la prise en compte de la sémantique. Une telle démarche ne permet pas ainsi d'identifier les similarités entre les syntagmes extraits des documents et ceux de la requête.

Afin d'améliorer les performances qualitatives du système et orienter la recherche vers la pré-

cision des réponses, il est nécessaire d'adopter une analyse sémantique plus fine des documents. Cela s'apparente à des problèmes de représentation mettant en scène différents concepts et relations entre ces concepts plutôt que des simples extractions de groupes nominaux. Même l'extension de ces derniers par des verbes ou par des relations de type de celles énumérées par Farradane ne permet une description fidèle du contenu sémantique des documents. Nous privilégions ainsi une indexation du *contenu sémantique* du document. Ce dernier met en scène des concepts et des relations conceptuelles permettant de décrire le plus fidèlement possible le schéma mental qu'on se ferait après la lecture du document. Nous énonçons alors l'hypothèse suivante:

Hypothèse 2 (Un bon terme d'indexation) *Un bon terme d'indexation est fondé sur des concepts complexes où les connecteurs sont vus comme des relations sémantiques permettant de cerner le contenu sémantique du document*

Actuellement, les systèmes dits *conceptuels* [Cro86] permettent une telle approche d'indexation. La représentation interne qu'ils adoptent est indépendante de la langue et repose uniquement sur la sémantique. Dans ces systèmes, un concept désigne une représentation sémantique qui peut se limiter aux concepts simples, mais qui comprend aussi généralement des liens ou des relations entre concepts. La représentation de ces concepts dans la littérature prend diverses formes: frames, graphes conceptuels, formules logiques, arborescences, etc.

Le problème majeur d'une telle démarche est sa lourdeur au niveau de la réalisation. D'une part une indexation automatique est difficile à mettre en oeuvre. D'autres part, il est nécessaire de déterminer une limite au processus de compréhension des textes. Ceci est fixé par le niveau de précision requis par l'application, c'est-à-dire par la granularité et le nombre des primitives conceptuelles utilisées. Dans la section suivante nous montrons les étapes nécessaires à la concrétisation d'une approche relationnelle sémantique.

2.2 Concrétisation d'une Approche Relationnelle

La mise en place d'un processus d'indexation relationnelle permettant l'extraction, à partir des documents du corpus, de termes d'indexation conformes à l'hypothèse 2 impose l'utilisation de taxinomies de concepts et de relations conceptuelles. En effet, le contenu sémantique des documents doit être formulé par rapport à ces taxinomies. Par conséquent, la définition des primitives sémantiques du domaine d'application s'avère nécessaire. La section suivante résume comment sont choisies les primitives. Nous donnons ensuite les types de relations primitives les plus couramment utilisées dans la littérature. Les primitives du domaine de l'application spécifiées, nous montrons la difficulté d'adopter une approche d'indexation automatique.

2.2.1 Objectivisme et Primitives

Les théories basées sur ce que l'on appelle l'*objectivisme* constituent actuellement la majorité des recherches liées à la compréhension de la langue naturelle. Ces théories sont basées sur l'hypothèse que le sens d'une phrase est lié aux conditions de sa véracité. Nous comprenons une phrase quand on connaît les cas pour lesquels elle sera évaluée à vraie. De plus, les conditions de véracité des phrases sont systématiquement liées à celles de ses parties [Mal91]. La *sémantique à valeur de*

vérité, *la sémantique formelle*, et *la sémantique de Montague* désignent toutes des termes utilisés pour substituer la notion d'objectivisme.

Selon la vue objectiviste, la réalité consiste en des objets, des propriétés sur ces objets et des relations qui les lient. Ces entités sont considérées comme les "briques" de la construction de la réalité. Pour obtenir le sens d'une expression, cette dernière devra être décomposée en sous-expressions, et ainsi de suite jusqu'à ce que chaque sous-expression peut être liée à un objet, une propriété ou une relation. Une telle théorie du sens est appelée *théorie compositionnelle*. Le principe de la compositionnalité stipule que le sens d'une phrase est fonction des sens de ses composants [Hir91].

L'approche de Montague est compositionnelle [Mon74], la condition de véracité d'une phrase est fonction des conditions de véracité de ses parties. La théorie des situations adhère également au principe de compositionnalité [Dev91][pp231]:

> "One obvious property of language is that meaning of a larger unit, such as a sentence, is a function of the meanings of the individual parts, the words and phrases that go together to give the sentence."

La mise en oeuvre de cette conception implique que soient choisis un ensemble de traits qui seront utilisés par la suite pour la construction de toutes les représentations manipulées. On parle alors de *primitives sémantiques*. La recherche de ces primitives reste un éternel effort qu'il faut se donner la peine de réaliser. Plusieurs travaux ont tenté de dégager des primitives grâce auxquelles, toutes les autres entités complexes peuvent être définies. Aristotle, par exemple, a défini le concept HOMME en termes des concepts CRÉATURE et ANIMAL, où CRÉATURE et ANIMAL peuvent eux-mêmes être définis par le biais de termes plus primitifs. De cette façon Aristotle prétend que chaque concept peut être réduit en un nombre fixe de primitives. Ces primitives appellées *catégories* correspondent par exemple à une Substance, une Qualité, un Temps, etc. Dès lors, plusieurs tentatives pour retrouver les primitives "ultimes" ont été menées. Elles furent basées sur des motivations philosophiques, linguistiques ou pragmatiques. Le résultat était soit des systèmes avec très peu de primitives, comme par exemple le système de dépendance conceptuelle de Schank [Sch75], soit des systèmes avec une centaine de primitives comme par exemple les 90 primitives du programme de traduction automatique de Wilks [Wil73, Wil75].

Ces primitives à partir desquelles on représente le sens des concepts n'ont toutefois pas de définition rigoureuse. On se contente généralement de leur associer des règles et des propriétés évidentes [Wil87] :

Finitude: le nombre de primitives utilisées doit être bien entendu fini, mais surtout inférieur (et même très inférieur) au nombre de sens de mots à représenter.

Étendu: les primitives doivent permettre des représentations différentes pour des sens différents.

Indépendance: aucune primitive ne doit pouvoir s'exprimer en fonction des autres.

Non réductibilité: aucun ensemble de primitives ne peut être remplacé par un sous-ensemble plus petit.

Même si ces recommandations semblent être claires, l'arbitraire le plus total préside aux choix des primitives. Wilks [Wil87, pp.759] donne la définition suivante:

> "A set of primitives is a reduction device which yields a semantic representation for a

natural language via a translation algorithm and which is not plausibly explicated in terms of, or reducible to, other entities of the same type."

Une vue commune aux approches objectivistes est la suivante:

- Chaque entité est soit primitive soit construite à partir de primitives par certain principes de composition;
- La structure conceptuelle interne est le résultat de l'application de ces principes de composition;
- Les entités sans structure interne constituent des primitives et ont un sens directement saisissable, le sens des autres entités est indirectement déduit par le biais des principes de composition qui les relient à des entités primitives.

Il est intéressant à ce niveau de remarquer le parallèle qui existe entre la façon dont les primitives sont traitées dans le cadre d'une vue objectiviste, et la manière dont une formule en logique de premier ordre est construite à partir des atomes.

2.2.2 Quels Types de Relations Sémantiques Choisir

Dans cette section nous énumérons les types de relations conceptuelles les plus couramment utilisées en recherche d'informations. Cette énumération est issue d'un travail de compilation des travaux présentés dans [Sab88, Jou93]. Elle se base aussi sur de nombreux systèmes conceptuels de recherche d'informations. Il s'avère ainsi que souvent, les relations primitives admettent un des types suivants:

a) La hiérarchie

Le lien de hiérarchie est un lien fondamental dans plusieurs applications. Il est généralement nécessaire de différencier les liens entre classes, des liens entre classes et individus, c'est-à-dire en termes mathématiques, distinguer l'inclusion de l'appartenance:

- l'appartenance d'un individu à une classe ou l'attribution d'une propriété à un objet revient à la création d'un lien hiérarchique, comme par exemple, l'énoncé *Jean est un homme* traduit par: *Sorte-de*(Jean, Homme). Cela signifie que l'individu "*Jean*" appartient à la classe des *hommes* ou encore que le concept HOMME s'applique à *Jean*;
- l'inclusion ou la "subordination d'une classe à une classe" est exprimée dans des énoncés comme *les hommes sont des mammifères* ou encore *les lapins sont des animaux*. Pour le premier énoncé, le lien signifie que l'ensemble des individus auxquels on attribue le concept HOMME est inclus dans l'ensemble des individus auxquels on attribue le concept MAMMIFÈRE.

Les liens de hiérarchie sont caractérisés par la non-symétrie. Le lien d'appartenance est un lien non réflexive et non transitive. Par contre le lien d'inclusion est réflexive, antisymétrique et transitive.

b) Le Repérage

Il s'agit des relations permettant de représenter des schémas de type : une entité X -(une entité repérée)- est repérée par rapport à Y - (une entité repère). Ils regroupent les liens d'identification comme dans l'énoncé *Tunis est la capitale de la Tunisie*, ou encore *Stendhal est l'auteur de rouge et noir*. Le repérage sert parfois à indiquer un sujet ou un objet distinctif pour un verbe ou une propriété donnée. Par exemple, les liens qui existerait entre *briller* et *soleil* ou bien entre *rond* et *ballon*.

c) L'ingrédience

Les relations d'ingrédience expriment les liens de composition. Nous dirons que x est un ingrédient de y, pour signifier que le concept x est une partie du concept global y. Ce type de lien est l'un des plus importants que ce soit en intelligence artificielle ou en recherche d'informations. En intelligence artificielle, il permet de décrire comment sont composés des éléments complexes en fonctions d'autres éléments plus simples, comme par exemple *la plume appartient au stylo*, ou que *le doigt est une partie de la main, laquelle est une partie du bras*. En recherche d'informations, les relations d'ingrédience correspondent aux relations d'agrégation permettant de spécifier pour les besoins de la recherche de documents structurés qu'un *document comporte des chapitres, lesquels sont constitués de sections, les sections contenant elles-mêmes des paragraphes, etc.*

A la différence de l'inclusion, les liens d'ingrédience ne sont pas antisymétriques. Ils vérifient par contre les propriétés de transitivité, de réflexivité et de non-symétrie.

d) La Localisation

Les relations de ce type permettent de préciser les positions des divers éléments. Il ne s'agit pas uniquement de liens spatiaux (*bas-haut, droite-gauche, à-côté, etc,*) comme dans des énoncés de type *le livre est sur la table*, ou *un jardin entoure ma maison*, mais de toute relation pouvant indiquer une localisation donnée. Paradis dans [Par96] mentionne ainsi l'importance de comprendre et d'indexer des phrases comme *nous discutons aux sections 1.1.1 et 2.4.2 de la structure des documents*, où les relations de localisation décrivent des informations de méta-discours donnant plus de détails sur le contenu sémantique et la structure des documents.

Dans la littérature, les relations de localisation occupent une place de choix. Elle sont souvent basées sur les travaux de Allen [All83]. Ce dernier décrit un ensemble complet de relations spatiales dans un espace à une dimension. Ces relations sont utilisées pour former des prédicats primitifs de localisation. Ainsi, les relations spatiales adoptées par Mechkour [Mec95c] pour les besoins de la recherche d'images sont au nombre de 12, et ont pour support les relations de Allen.

e) La Succession

Un certain nombre de connaissances sont définies comme des listes ordonnées (jours de la semaine, lecture des chapitres d'un document, etc.). Elles peuvent être exprimées par des liens de succession. Par exemple, pour les besoins de la recherche de documents textuels structurés, Paradis [Par96] introduit une relation d'ordonnancement permettant de fixer l'ordre des *paragraphes*, des *divisions*, etc. d'un texte. Il introduit également des relations de séquence décrivant l'enchaînement

des mots. Ainsi, dans le groupe de mots "recherche d'informations", la relation de séquence établit que le mot «recherche» précède le mot «d'», qui précède lui-même le mot «informations». Dans les approches hypertextes [Khe95], le sens de parcours des noeuds d'un document est également exprimé par des relations de succession.

Il existe bien entendu plusieurs autres types de relations sémantiques. Par exemple, ceux désignant des équivalences sémantiques, des contrastes, des actions, etc. Chaque type de lien peut être modélisé par plusieurs relations primitives. Souvent dans les applications, il faudra choisir entre confondre ces diverses relations en une seule relation primitive ou en particulariser certaines. Autant le nombre de primitives associées à un type de relation est grand, autant la représentation engendrée est précise. Cependant, le choix final dépendra de l'application concrète à traiter. Il n'est pas ainsi possible de donner de règle générale sur les primitives. Pour les besoins d'une recherche précise et fidèle, nous préférons toutefois avoir le maximum de relations primitives.

2.2.3 Difficultés d'une Indexation Automatique

Il est souvent délicat de traduire des énoncés de la langue naturelle dans un langage formel (ou dans un langage logique) car les indications syntaxiques ne sont pas toujours suffisantes pour déduire l'interprétation sémantique de ces énoncés. Ainsi, les énoncés "*page in a book*" et "*page of a book*" sont similaires et doivent être exprimés par le même index «*part-of*(page, book)», où "*part-of*" est une relation primitive d'ingrédience. De plus, une relation dans un index peut avoir plusieurs sens possibles. Reprenons par exemple, les deux énoncés suivants:

(1) Jean est assis

(2) Jean est intelligent

L'énoncé (1) dénote la *position* dans laquelle se trouve Jean, alors que l'énoncé (2) exprime une *qualité* de l'individu Jean. La préposition *est*[7] peut être ainsi associée à une relation sémantique de type *Position* pour le premier cas, alors qu'elle correspond plutôt à une relation sémantique comme *Qualité* dans le second énoncé.

Ainsi la difficulté réside essentiellement dans la détection automatique des associations sémantiques. Outre une analyse syntaxique, la détection des relations sémantiques dans les phrases exige une analyse sémantique approfondie. Les catégories syntaxiques sont plus "grossières" que les catégories sémantiques. Les deux types sémantiques «Localité» et «Qualité» ne sont pas biunivoquement traduits par des types syntaxiques distincts.

Selon les relations sémantiques détectées dans le texte, les inférences envisagées ne sont évidemment pas les mêmes. Reprenons ainsi l'exemple donné dans [Des90]:

(3) les hommes sont mortels

(4) les hommes sont nombreux

7. Les linguistes parlent plutôt du *copule* "est" [Des87, Des90, Jou93].

Les deux mots “mortels” et “nombreux” ont un même type syntaxique, à savoir qu'ils correspondent tous les deux à un adjectif. Pourtant si le raisonnement :

$$\frac{\text{les hommes sont mortels} \quad \text{les Athéniens sont des hommes}}{\text{les Athéniens sont mortels}}\ \textbf{Transitivité}$$

est valide, un raisonnement similaire consistant à asserter la déduction suivante:

$$\frac{\text{les hommes sont nombreux} \quad \text{les Athéniens sont des hommes}}{\text{les Athéniens sont nombreux}}\ \textbf{Transitivité}$$

n'est pas nécessairement vrai, même si les deux déductions se présentent formellement de façon identique: l'adjectif «mortels» a été simplement remplacé par l'adjectif «nombreux».

Le premier raisonnement est une instanciation du schéma d'inférence (A), le second du schéma (B):

$$(\mathrm{A})\frac{(X \subset Z) \quad (Y \subset X)}{(Y \subset Z)} \qquad (\mathrm{B})\frac{(X \in Z) \quad (Y \subset X)}{(Y \in Z)}$$

Le premier schéma est valide puisque l'inclusion est transitive. Le second est non valide. Les prédicats «sont mortels» et «sont nombreux» ont été interprétés respectivement par une propriété («appartenance à une classe») portant sur des objets individuels et, par une propriété («appartenance à une classe de classe») portant sur des classes.

De la même manière, si l'inférence suivante :

$$\frac{\text{Socrate est Athénien} \quad \text{les Athéniens sont courageux}}{\text{Socrate est courageux}}\ \textbf{Transitivité}$$

est valide, le raisonnement qui consiste à dire que:

$$\frac{\text{Socrate est Athénien} \quad \text{les Athéniens sont nombreux}}{\text{Socrate est nombreux}}\ \textbf{Transitivité}$$

est manifestement incorrect. Les deux raisonnements précédents correspondent aux schémas respectifs (C) et (D) suivants:

$$(\mathrm{C})\frac{(X \in Y) \quad (Y \subset Z)}{(X \in Z)} \qquad (\mathrm{D})\frac{(X \in Y) \quad (Y \in Z)}{(X \in Z)}$$

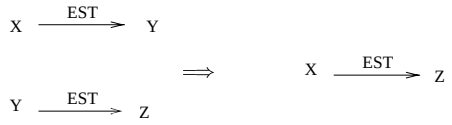


Figure 2.2. Transitivité de la préposition "EST"

Le premier schéma (C) correspond à une relation de composition valide, le second ne l'est pas car, d'une façon générale, l'appartenance n'est pas transitive.

Dans les exemples ci-dessus, il n'y a aucun indice morpho-syntaxique qui permettrait d'accepter (A) et (C) et de rejeter (B) et (D). Il semble que tous les schéma suivent le raisonnement de la figure 2.2 suivante.

Ainsi, adopter une indexation relationnelle automatique des textes revient à résoudre des problèmes sémantiques complexes. La difficulté de réaliser une indexation automatique s'accentue dès lors qu'on considère les corpus actuels, constitués de données multimédias. Ainsi, la distance entre le signal de ces données et leur sémantique est élevée. Par exemple, dans le cas des images, le signal est une matrice de points[8], alors que la sémantique consiste à identifier symboliquement différents éléments de l'image. Ainsi, là où la matrice des points indique du vert et du marron, la sémantique véhiculée correspond à un arbre. Une donnée multimédia véhicule des informations diverses: ainsi une image contient des couleurs, du contraste, des objets (arbre, tronc, etc.), etc. Toutes ces informations participent à la description de l'image, d'où les difficultés rencontrées lors de l'indexation et de l'interrogation des données multimédias.

Pour les documents textuels, les solutions vont des approches globales intégrant des traitements généraux de la langue naturelle à des approches pragmatiques valables pour des domaines restreints. Comme exemple de la première démarche, les systèmes applicatifs [Des90] permettent d'effectuer des raisonnements sur la langue naturelle. Grâce à l'adoption de la grammaire catégorielle [Des90, Jou93, Des87], l'extraction de la sémantique des textes est aisément menée. L'approche adoptée dans le cadre du système RIME [Ber88, Nie90] est plus pragmatique. Elle exploite le vocabulaire médical précis des documents du corpus pour automatiser leur analyse sémantique.

Au niveau des corpus multimédias, les tentatives d'indexation automatique ont surtout porté sur les images [WAL+94, HM96]. L'indexation des images pour leur recherche reste un problème ouvert. La nécessité de combiner des critères symboliques, des critères physiques et de lier ces critères par des relations est évidente. Cependant il n'existe pas à l'heure actuelle une technique permettant d'extraire d'une image l'ensemble de ces composants avec suffisamment de précision. Les travaux actuels s'orientent vers la combinaison et la fusion de modèles provenant de l'analyse d'image et de la vision par ordinateur pour garantir que le procédé d'indexation puisse s'adapter à tout type d'images. Les travaux menés au MIT par Minka et Piccard [MP96] vont dans ce sens. Certaines tentatives d'indexation automatique existent également pour le cas du son [SG92] ou encore pour la vidéo [HM94, HLMS95].

Nous n'avons pas, pour notre part, l'objectif de décrire un processus d'indexation automatique. Nous préférons spécifier quels sont les éléments indispensables à la recherche des documents complexes en posant comme hypothèse qu'un processus permet d'obtenir ces éléments. Ce processus

8. En Anglais, il s'agit des *pixels*.

reste actuellement assisté par un intervenant humain.

2.2.4 Les Systèmes de Recherche d'Informations Conceptuels

Sous le vocable de systèmes de recherche d'informations conceptuels sont regroupés tous les systèmes qui tentent de dépasser les méthodes classiques de recherche d'informations basées sur les mots-clés en intégrant des techniques en provenance de l'intelligence artificielle. Ces systèmes adoptent des représentations bâties sur des concepts et des relations. Les index générés permettent une description en *compréhension* du contenu des documents, et sont conformes à l'hypothèse 2. Ces systèmes se distinguent par la façon dont ils abordent l'indexation des documents:

a) Indexation Manuelle

COREL [BCd86] est un système de recherche d'informations expérimental qui emploie des techniques de l'intelligence artificielle. Il est écrit en Pearl dans un environnement de Vax 11/780. COREL suit une approche conceptuelle et est destiné à une base d'articles sur le code civil de l'état de Louisianna (USA). Le système COREL effectue une indexation à base de frames afin de pallier aux problèmes classiques engendrés par les mots-clés.

Comme cela a été mentionné par ses auteurs, le principal inconvénient de ce système est que l'indexation se fait manuellement. Afin de modéliser le contenu sémantique des documents considérés, deux types d'entités sont utilisés, à savoir des concepts et des relations conceptuelles légales. Les relations lient soit des individus soit des propriétés (des classes)[9]. Les auteurs ont développé un langage utilisant une grammaire à base de frames, pour lequel ils ont défini quelques concepts juridiques de base. La figure 2.3 donne un exemple de représentation dans ce formalisme. Chaque frame est associé à un type correspondant à un groupe de relations ayant des rôles juridiques donnés dans le code civil. Il existe cinq types de frames. Par exemple, la relation *usefruct* appartient au type *SUB-TYPE-LESS-THAN-OWNERSHIP* .

SYSTEM-DESCRIPTION-RELATIONSHIP
relationship name:
usefruct
list of actors:
grantor
usefructuary
naked owner
type name:
SUB-TYPE-LESS-THAN-OWNERSHIP

Figure 2.3. Une Description de la relation juridique *usefruct*

L'interrogation dans COREL est dirigée par des menus. L'utilisateur commence par entrer le type

9. Les relations sur les classes ne sont autres que des généralisations des relations sur les individus.

de frame qui l'intéresse, ou en d'autres termes le type des informations juridiques qui l'intéressent. Selon ce type, le système lui demande de remplir les slots à travers la proposition d'un certain nombre de listes d'options prédéfinies. Une fois le frame entièrement constitué par l'utilisateur (éventuellement en remplissant des slots par des options comme "je ne sais pas"), un ensemble plus spécifique d'objets répondant à la requête utilisateur est identifié, quand cela est possible. À ce stade le système se déplace à un autre menu pour demander son avis à l'utilisateur. L'inconvénient de ce système est que le niveau de détail adopté pour la structure de l'information indexée laisse pensif si on met en regard la quantité de travail requise pour l'indexation et la compréhension des documents.

Elen [Che92] est un système conceptuel de recherche d'informations destiné à la gestion et l'interrogation de logiciels et de leurs documentations associées. Le système Elen utilise le modèle des graphes conceptuels [Sow84] pour représenter les connaissances de la base.

Une des caractéristiques de Elen est de proposer un langage d'indexation et un langage d'interrogation distincts. La volonté de confier les difficultés liées à la manipulation du langage d'indexation au gestionnaire du système uniquement, et permettre à l'utilisateur de s'affranchir de ces contraintes, est à l'origine de cette distinction.

Ainsi, si l'indexation utilise les graphes conceptuels, le langage d'interrogation est basé sur un pseudo-langage graphique qui guide la formation des requêtes et masque à l'utilisateur le structure de graphe conceptuel sous-jacente. La première phase de l'évaluation des requêtes consiste à traduire la structure saisie en un graphe conceptuel. La recherche, quant à elle, est fondée sur l'opération de projection. Afin d'optimiser cette opération, une pré-sélection des documents est effectuée en se basant sur des signatures. Les signatures forment un ordre partiel similaire à celui des graphes conceptuels, ce qui permet de sélectionner rapidement les signatures candidates en positionnant la requête dans cette hiérarchie. Elen a été réalisé dans l'environnement intégré de Xerox, en utilisant le langage orienté objets Loops. Il fut testé sur des descriptions de méthodes Loops et sur des fonctions du noyau de UNIX. Ce système présente l'intérêt de proposer une première adaptation du modèle des graphes conceptuels au domaine de la recherche d'informations.

MIRTL (Multimédia Information Retrieval Terminological Logic) [MSST93] est un système terminologique dédié à la recherche d'informations multimédia. Les documents sont décrits manuellement par des termes terminologiques en utilisant des concepts et des rôles (des relations binaires entre concepts). La requête, quant à elle, est introduite comme un concept. Le processus de recherche repose sur le mécanisme de la subsomption. La complexité de traitement d'un tel mécanisme, surtout dans le cas d'un langage assez expressif, a ramené ses auteurs à sacrifier certains constructeurs de termes. Par conséquent, un nouveau système basé sur une logique terminologique plus restreinte, appelée Mirlog [SS96] fut récemment présenté dans le cadre de la revue finale du projet européen FERMI [Fer94]. Le prototype traite un corpus d'images originellement indexé par des graphes conceptuels [CM97]. Une fonction de translation automatique a permis de faire passer les index dans le format de la logique Mirlog. Les temps de réponse du prototype n'ont pas été pour autant améliorés. Le prototype dans sa version actuelle est implémenté en C dans un environnement UNIX.

b) Indexation Automatique

Il s'agit des systèmes opérant une analyse sémantique pour la compréhension des documents. Ainsi, l'indexation dans le système de recherche médical Rime [Ber88] se fait automatiquement.

Le système RIME utilise un formalisme proche des dépendances conceptuelles de Schank [Sch72]. Le système utilise un treillis de 650 concepts. Ce système est décrit en détail dans la section 2.3.3 (page 39).

MENELAS [MEN94] est un projet européen auquel participe douze équipes de recherche originaires de Belgique, France, Grande-Bretagne et Irlande. Il est également dédié au domaine médical. Tout comme ELEN, il est basé sur les graphes conceptuels. L'indexation est cependant réalisée ici de façon automatique, par le biais d'une grammaire contextuelle et d'une analyse sémantique/pragmatique.

MENELAS propose une application permettant d'analyser les comptes-rendus d'hospitalisation, de les stocker et de pouvoir y accéder afin de les exploiter. Le système est capable de traiter des comptes-rendus exprimés en français, anglais ou néerlandais. Le treillis des types de concepts utilisés regroupe environ 1000 concepts. Une des caractéristiques de ce système est sa modularité, son développement se déroulant simultanément dans plusieurs sites. Il utilise les langages Prolog, Lisp, C et C^{++} sous UNIX. Les requêtes sont introduites par le biais d'une interface graphique comportant différents menus. Cette interface permet principalement d'associer un contexte à une requête (démographique, date d'émission, raison d'admission, etc.) et de sélectionner des concepts par un parcours de treillis. Il est intéressant à ce niveau de soulever le fait que les requêtes ne sont pas saisies sous forme de graphe conceptuel, mais sous forme de conjonctions de disjonctions de concepts.

Le système LASSIE [DBS90] utilise un formalisme dérivé des logiques terminologiques [Neb90a] pour la représentation des index. LASSIE propose une interface où les requêtes peuvent être introduites en langue naturelle. Ces dernières sont alors transformées par un procédé automatique en termes terminologiques. LASSIE a été testé sur un corpus de génie logiciel consistant en une librairie de composants logiciels réutilisables. Plusieurs relations sont extraites automatiquement des codes sources comme par exemple celles relatives aux commandes #define, #include, etc. L'utilisateur, en l'occurrence un analyste programmeur, décrit conceptuellement le composant logiciel qu'il souhaite retrouver dans la base (comportant plus d'un million de lignes de code C). Le système le retrouvera par classification grâce à sa base de connaissances terminologiques constituée après l'indexation.

Le prototype SCISOR (System for Conceptual Information Summarization, Organization and Retrieval) est un système d'analyse de textes en langue naturelle, circonscrit aux dépêches financières du serveur en ligne DOW-JONES, qu'il scrute en cherchant à détecter les fusions/acquisitions d'entreprise. Pour la représentation des documents, il utilise un langage proche de KL-ONE [dB85], le père des logiques terminologiques. Le système SCISOR est implanté en Lisp sur UNIX. L'indexation dans ce système est un processus complexe qui implique plusieurs analyses. Une extraction lexicographique est d'abord effectuée. Le texte est segmenté pour empêcher l'analyse de certaines portions jugées trop complexes ou hors-propos. Un filtre est utilisé pour déterminer le thème du document (fusion, acquisition d'une société, etc.). Le processus d'analyse linguistique pour l'indexation est très poussé. Un analyseur ascendant (bottom-up full parser) et un analyseur descendant partiel (top-down skimming partial parser) sont utilisés. Initialement présenté comme un système de compréhension et de génération de résumé, ce système pourrait à notre avis se révéler intéressant dès lors que le domaine d'application est restreint.

c) Les Limites des Systèmes Conceptuels

L'indexation en compréhension des documents, et ainsi l'adoption d'une approche orientée vers la précision des réponses, exige le développement des systèmes de recherche d'informations conceptuels. Ces systèmes devront manipuler des concepts dont le sens est non ambigu et dont les relations qui les lient sont spécifiées. Cependant, la phase d'indexation s'est révélée rédhibitoire dans les premiers prototypes comme COREL ou encore dans ELEN, du fait de son caractère manuel. Ces derniers ne peuvent donc être utilisés que pour des corpus contenant un ensemble restreint de documents.

Pour remédier à ce problème, certains systèmes conceptuels comme I3R [Cro86], RUBRIC-TOPIC [TAAC87, TAA89], ou TAÏGA [MAD92] procèdent par, ce qu'on appelle, *extraction automatique*. Le procédé ne se réfère pas à une analyse linguistique, mais à un processus de sélection des idées importantes dans un document, par la reconnaissance de certaines expressions ou d'indices linguistiques. Souvent privilégiée parce qu'elle requiert peu de ressources pour être mise en oeuvre, cette approche nécessite toutapercevoirefois une condition pour son utilisation. En effet, le langage employé dans les documents doit contenir suffisamment d'idiomes et d'expressions typiques. Dans le cas contraire, une grande partie des concepts et des relations véhiculés par le texte restent non identifiés simplement parce qu'ils ne sont pas suffisamment indiqués par les indices définies dans le système. Ainsi, l'indexation résultante reste insuffisante et n'avantage pas des bonnes performances qualitatives.

Pour les documents textuels, l'analyse sémantique est celle qui est le mieux à même de permettre une indexation automatique. Le principal inconvénient de cette approche est sa complexité et la difficulté de son implantation. De plus, il n'est pas envisageable de l'appliquer à n'importe quel domaine. Des projets comme CYC [Len95], qui visent une modélisation des connaissances indépendante du domaine, n'ont eu jusqu'à présent que des résultats mitigés.

Paradoxalement, les systèmes conceptuels n'ont pas encore atteint un stade qui leur permet de soutenir la comparaison avec les systèmes traditionnels à base de mots-clés ou de groupes nominaux. Pour s'en convaincre, il suffit de consulter les derniers actes de la conférence SIGIR. Croft [Cro92, p252] résume bien ce constat:

> "It has not been established that a representation based on the meaning of the document text (using case frames for example) is superior (from an IR perspective) to a representation that simply identifies such features as important word stems and noun groups."

Cette suprématie des approches traditionnelles est manifeste si l'on considère le plébiscite involontaire que constitue l'adoption des techniques statistiques par les systèmes conceptuels qui se posent précisément en challenger: I3R, RUBRIC-TOPIC, etc. Au fait, les systèmes conceptuels sont donc amenés à relever non seulement un défi scientifique en démontrant la validité de leur approche (chose qui peut s'avérer facile à notre avis), mais aussi et surtout un défi économique dans le sens où l'apport de l'approche doit être assez significatif pour justifier le remplacement des techniques à base de mots-clés. Les performances relativement discutables des systèmes conceptuels s'expliquent à notre avis par les raisons suivantes:

(1) tout d'abord, les méthodes calculatoires et statistiques sont à un stade de maturité très avancé. Elles sont ainsi pleinement efficaces et opérationnelles. Ceci à la différence des approches

conceptuelles encore au stade de la recherche. Le principe d'une approche conceptuelle pour la recherche d'information ne fut adopté que très récemment et reste donc à l'état embryonnaire que ce soit par la définition des fonctions de correspondance ou pour le classement des réponses fournies;

(2) il est très difficile de juger de la validité des représentations associées aux documents. S'il s'agit d'une indexation manuelle, il arrive que l'indexeur omette des détails du contenu sémantique du document ou qu'il exprime mal certains concepts lors de la représentation. De plus, à présent, il n'est pas possible d'envisager avec suffisamment de précision une indexation automatique des documents;

(3) les fonctions de correspondance utilisées par les approches conceptuelles ne fournissent pas les règles d'inférence nécessaires pour la prise en compte des décisions de pertinence qui s'imposent. Nous verrons par exemple que les propriétés des relations ne sont pas exploitées par ces fonctions. Un tel constat fait en sorte que plusieurs documents pertinents ne soient pas renvoyés à l'utilisateur;

(4) la comparaison des deux approches s'est faite généralement sur des corpus hétérogènes et avec des requêtes se limitant à des mots-clés. Cela avantage bien entendu les indexations à base de mots-clés, puisque les approches sémantiques sont dédiés à des domaines d'expertise.

La difficulté d'indexer les données multimédias, mentionnée dans la section 2.2.3 et l'état d'art actuel des systèmes conceptuels font en sorte qu'on n'est pas près de trouver une solution imminente au point (2) ci-dessus cité. Une telle problématique dépasse le cadre de cette thèse et relève plutôt du domaine du traitement de la langue naturelle, des images et autres médias.

Par contre, nous montrons au niveau de cette thèse qu'il est possible avec une approche conceptuelle d'aboutir à des algorithmes polynomiaux tout en résolvant le point (3). À travers l'implantation à base de graphes conceptuels du prototype Relief [OP97b], nous montrons que nous obtenons des résultats encourageants pour un corpus d'images spécialisé et des requêtes complexes. De plus, par le biais du modèle logique relationnel que nous proposons au chapitre 3, nous espérons apporter une contribution au point (1). Enfin, la comparaison de notre prototype avec le célèbre système SMART [Sal71] montre qu'une approche conceptuelle donne des résultats nettement meilleurs qu'une approche classique dès lors qu'il s'agit d'indexer puis de chercher des données complexes comme les images.

2.3 Raisonner sur les Relations

Toutes les approches décrites au début de ce chapitre s'accordent sur l'importance des relations. Indépendamment du degré d'exhaustivité de ces approches, la construction d'un thésaurus, l'adoption d'une approche linguistique ou encore la conception d'un système de recherche conceptuel, exige la représentation de certaines relations. Cependant, peu d'approches permettent de raisonner sur ces relations et d'exploiter ainsi les propriétés qu'elles présentent.

Plusieurs systèmes ont toutefois été confrontés au besoin de gérer certaines propriétés de relations. Dans la suite, nous décrivons trois approches dans lesquelles ce besoin a été identifié. La première approche est liée à la construction d'un thésaurus pour la recherche d'informations. La

seconde décrit comment les inférences sur les relations sont gérées au niveau des réseaux sémantiques. Enfin, nous présentons l'approche du système RIME. Ce dernier peut être considéré comme un des pionniers des systèmes de recherche conceptuels s'intéressant à la gestion des relations.

2.3.1 L'Approche de Maron

Lorsque les occurrences de deux termes dans les documents du corpus sont statistiquement liées, le fait de considérer que le premier terme est une variante fiable de l'autre terme lors de l'évaluation d'une requête est une hypothèse fréquemment adoptée pour la construction automatique des thésaurus [PW91]. La transitivité de la relation de co-occurrence est souvent assumée dans ce type de construction. Ainsi, nous avons non seulement les relations qui apparaissent explicitement à partir des données de co-occurrence, mais aussi nous pouvons inférer l'existence d'autres relations implicites en supposant la validité de la transitivité. Par exemple, si un terme t_1 co-occurre avec un terme t_2 et que t_2 co-occurre avec t_3, alors nous pouvons inférer, en utilisant la transitivité, qu'il existe une relation sémantique entre le terme t_1 et le terme t_3, même s'il ne co-occurrent pas explicitement dans le corpus. Cette distinction entre les relations explicitement représentées dans le corpus, et les relations qui peuvent être déduites de ces informations explicites rappelle la distinction logique entre l'intension et l'extension d'un terme, d'un symbole ou d'une proposition. L'extension d'un terme est l'ensemble des entités auxquelles il s'applique, alors que l'intension d'un terme est l'ensemble des propriétés qui le définissent [10].

Maron [Mar67] fut le premier à voir le parallèle entre l'intension et l'extension des expressions logiques avec les intensions et les extensions des relations. Il établie que le sens *intentionnel* d'une relation R est l'interprétation de R en termes d'autres relations. Par exemple, la relation *beau-père* existant entre un homme et un autre individu stipule que l'enfant de cet homme est marié à l'autre individu en question. Le sens *extensionnel* d'une relation R est l'interprétation de R en termes de tuples pour lesquels la relation est valide. Dans le cas de la relation *beau-père*, l'extension pourrait être l'ensemble des paires d'individus pour lesquelles la relation est valide. Maron [Mar67, pp.218] donnait ainsi les définitions suivantes:

> "We say that the extension of a relation is the set of ordered pairs for which the relation holds, and the intension of a relation is its definitional equivalent in terms of other relations."

Selon Maron, les extensions des relations sont celles représentées explicitement dans le corpus (il s'agit des paires de termes qui co-occurrent), alors que l'intension d'une relation est une relation qui peut être dérivée ou inférée d'autres relations (de façon extensionnelle ou intentionnelle).

Dans les systèmes automatiques où le thésaurus est représenté comme une matrice, les relations qui ne sont pas explicites dans le corpus peuvent être facilement déterminées. Il s'agit d'appliquer la *fermeture transitive* de la matrice originelle de termes [Bla90]. Ainsi, étant donné un thésaurus de termes représenté par une matrice B, l'algorithme de Warshall [ST81] suivant est appliqué:

10. Allwood et al. [AAD93] donnent la relation suivante entre intension et extension: "An intension is something that relates a linguistic expression [term] to its extension. It determines the extension of a linguistic expression ... an intension is a function: something that for every possible situation or world picks out exactly those objects which make up the extension of a given expression".

Algorithme 1 (Algorithme de Warshall)
Définir une nouvelle matrice $A = B$

```
Pour i = 1 jusqu'à i = n
    Pour tous les j
        Si A(j,i) = 1 alors
            Pour k = 1 jusqu'à k = n
                A(j,k) := A(j,k) + A(i,k)
            FinPour
        FinSi
    FinPour
FinPour
retourner matrice A
```

L'approche de Maron est intéressante dans un sens qu'elle fournit un support théorique pour inférer des relations non explicitement représentées dans le corpus, c'est-à-dire qu'elle permet de raisonner sur les relations et leurs propriétés. Nous pouvons ainsi constater que la définition d'intensionnalité de Maron est celle adoptée pour les rôles en logiques terminologiques [Neb90a]. Cette approche a été appliquée uniquement aux relations sémantiques de co-occurrences, et a l'inconvénient ainsi de ne s'intéresser qu'à la propriété de transitivité. Elle constitue, à notre connaissance, la seule approche en recherche d'informations à s'intéresser aux liens entres relations à travers la notion d'intensionnalité.

2.3.2 Les Réseaux Sémantiques

En intelligence artificielle, les réseaux sémantiques étaient les premiers à s'intéresser à la représentation des liens sémantiques entre les entités du domaine. La définition de réseau sémantique est assez difficile à poser, puisque ce terme désigne tout un ensemble de formalismes graphiques. La définition d'un réseau sémantique est tellement générale qu'elle peut ainsi englober des formalismes aussi divers que les dépendances conceptuelles, les frames, les graphes conceptuels ou les logiques terminologiques; ce qui la rend presque vide de sens [Sch91]. Il se dégage cependant des caractéristiques communes [Sow91]. Un réseau sémantique se présente ainsi comme un graphe orienté, sans circuits, en général connexe, et doublement étiqueté: les sommets sont étiquetés par des concepts, les arcs par des relations binaires entre concepts. Le réseau peut correspondre à un modèle de la mémoire sémantique à long terme d'un locuteur. Pour comprendre certains énoncés ou pour procéder à certaines inférences, le locuteur fait appel à ses connaissances enregistrées dans sa mémoire. L'information qui est portée par un sommet du réseau est définie (i) par le concept lui-même; (ii) par les autres concepts auxquels il est lié et enfin (iii) par la nature sémantique des relations. Les relations EST-UN (en anglais IS-A, rendu parfois par SORTE-DE et POSS «possède») sont des relations typiques des réseaux sémantiques. La grande popularité de ces derniers s'explique par les caractéristiques suivantes:

- la représentation des connaissances «de sens commun» est plus simple et plus naturelle qu'avec le calcul des prédicats;

- la transitivité de certaines relations est supportée, ce qui autorise l'héritage des propriétés d'une entité par ceux qu'elle subsume. Par conséquent, il s'en suit une économie de mémoire de stockage pour la représentation de plusieurs relations;

- la facilité dans la recherche d'informations nécessaire à certains raisonnements et inférences;

La première caractéristique s'explique par le lien entre les réseaux sémantiques et les modèles psychologiques [Des87, Sab88]. Les deux dernières propriétés sont, quant à elles, fondées principalement sur des critères d'économie de traitements informatiques. Toutefois, c'est essentiellement la facilité avec laquelle les déductions sont faites dans les réseaux sémantiques qui explique leur large utilisation. Il est en effet facile de réaliser des raisonnements par composition des relations.

a) Les Inférences dans les Réseaux Sémantiques

Dans l'exemple de la figure 2.4, plutôt que de représenter des relations redondantes, des liens implicites peuvent être retrouvés par simple héritage de propriétés. Plus généralement, les raisonnements réalisés au niveau des réseaux sémantiques peuvent se symboliser par la donnée d'une *table de composition de relations* [Sab88]. Il s'agit d'un ensemble de relations de la forme «$R_1 \circ R_2 = R_3$», exprimant que si les concepts C_1 et C_2 sont liés par une relation R_1 et si C_2 et C_3 sont liés par une relation R_2 alors C_1 et C_3 sont implicitement liés par la relation R_3.

Exemple 1

SORTE-DE $\circ$ *SORTE-DE = SORTE-DE [Transitivité de la relation SORTE-DE]*

SORTE-DE $\circ$ *CONTIENT = CONTIENT [Composition de relations]*

Il est ainsi possible de représenter la propriété de transitivité et de tenir compte des informations implicites que pourrait engendrer la composition de certaines relations sémantiques. En recherche d'informations, une telle caractéristique de raisonnement est très utile pour obtenir des réponses à des questions ayant trait à ces liens implicites. Un graphe représentera la question et une simple comparaison de graphes permettra de répondre à des requêtes comme par exemple *que contiennent les livres?* Un mécanisme plus complexe, faisant intervenir des inférences sera cependant nécessaire pour répondre à des questions dont le graphe associé ne correspond pas directement aux documents. Par exemple, c'est le cas pour une requête comme *Que contiennent les Proceedings?* Il s'agit d'explorer la table de composition de relations puis d'établir les déductions possibles avant de comparer le graphe de la question au réseau des connaissances. Il suffit au fait de saturer le graphe de toutes les déductions.

A l'opposé, la table de composition de relations ne permet pas de modéliser le fait qu'un lien sémantique est symétrique ou que deux relations sont sémantiquement liées. Certaines inférences restent ainsi impossibles à mener comme dans le cas des requêtes de type *Quel objet est à droite du stylo?* Pour obtenir une réponse non vide dans le cas de la figure 2.4, il faudra que le système soit capable de représenter le fait que les liens "A-DROITE" et "A-GAUCHE" sont inverses l'un de l'autre.

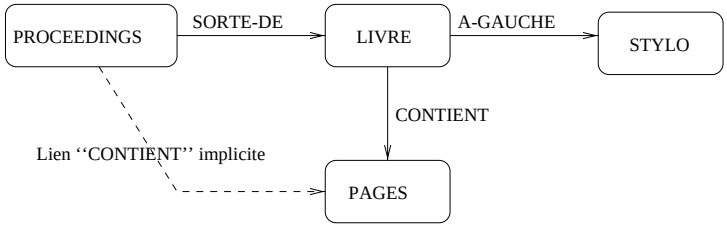


Figure 2.4. Exemple de déduction de propriétés

b) Les limites des Réseaux Sémantiques

Malgré leur capacité à représenter les relations sémantiques tout en présentant des mécanismes d'inférence très utiles, les réseaux sémantiques ne se sont jamais imposés comme un formalisme opérationnel pour la représentation des connaissances. Le principal reproche qui leur a été fait concerne leur sémantique trop floue. Des réflexions d'ordre épistémologiques furent alors menées. Aux questions de Woods [Woo75] ("*what's in a link*"), se sont succédé les réponses de Brachman [Bra77] ("*what's in a concept*"), et un compromis s'est formé autour de la nécessité de préciser les notions fondamentales de ces réseaux. Par exemple, les types et les instances ne sont pas différenciés dans le réseau.

La relation "IS-A" a aussi fait l'objet de nombreuses critiques [Bra83]. Ainsi, il s'est avéré que ce lien n'est pas assez primitif puisqu'il peut avoir plusieurs sens selon le type des concepts qui sont liés, et selon que ces derniers soient génériques (qu'ils s'appliquent à plusieurs individus) ou spécifiques (qu'ils ne s'appliquent qu'à un seul individu). Le lien "IS-A" peut ainsi se référer à des inclusions ou à des appartenances ensemblistes, à des relations de généralisation/spécialisation (e.g. un chien est un animal), ou encore à des relations de description (e.g. lien entre *Roi de France* et *Roi*). On s'est alors longtemps interrogé sur la pertinence de placer la relation "IS-A" dans le réseau au même titre que d'autres relations primitives. D'autant plus, qu'une propriété comme la transitivité n'est pas toujours valide "IS-A", cela dépend entre autres du sens qui lui est attribué:

Exemple 2 *On peut prendre pour exemple les schémas de dérivation erronés de la section 2.2.3 (page 26):*

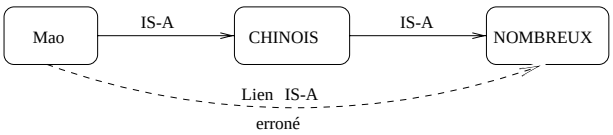


Figure 2.5. La transitivité du lien IS-A n'est pas toujours valide

D'autre part, la quantification dans le réseau n'est pas claire, particulièrement quand les liens impliquent deux concepts génériques. Par exemple, si nous avons "ÉTUDIANT$\rightarrow^{assister}\rightarrow$COURS" dans un réseau sémantique, on ne sait pas si chaque étudiant assiste à un cours différent ou si

tous les étudiants assistent au même cours. D'une façon générale, un lien entre deux entités X et Y dans un réseau sémantique correspond à une relation R(X, Y) qui, selon qu'elle est précédée de $\forall X \forall Y, \exists Y \forall X, \ldots$ pourra donner lieu à 4 interprétations distinctes. Pour résoudre ce problème, Simmons et Bruce [SB71] proposent d'introduire des liens primitifs comme: *some, some-indef, all, no, not-all,* ... en vue de rendre compte de toute combinaison possible de quantificateurs logiques. Dans [Woo91], Woods énumère différentes possibilités de quantifications entre deux concepts et propose des "opérateurs de formation de relations" permettant d'expliciter la quantification.

De nombreux travaux théoriques visent à préciser la sémantique des réseaux. Ces travaux établissent le besoin d'expliciter la sémantique des liens dans le réseau, et de reconnaître à quel niveau de représentation se situe un modèle. Des langages pour la représentation des connaissances comme KL-ONE proposé par Brachman [Bra79, BS85], les dépendances conceptuelles de Schank ou encore les graphes conceptuels de Sowa sont issus de tels travaux. Partant du constat que les réseaux sémantiques ne pouvaient constituer une théorie générale de la représentation des connaissances, Brachman dans le système KL-ONE distingue trois axes dans les représentations sémantiques: le premier selon lequel s'organise la hiérarchie des concepts en classes et sous-classes, le deuxième permet d'expliciter d'autres rapports entre concepts par des liens attributs et le troisième axe précise les rapports entre les éléments de description à l'aide d'un nombre fini de relations primitives. Les travaux de Brachman sur KL-ONE ont donné ensuite lieu à ce qu'on appelle aujourd'hui les logiques terminologiques ou les langages de description. Sowa, lui, conçoit son modèle des graphes conceptuels comme un langage intermédiaire entre les langues naturelles et la logique. Il n'est donc pas surprenant que ce formalisme, dans sa version étendue, soit au moins aussi puissant que les logiques de premiers ordre, permettant de représenter aisément les quantifications.

Paradoxalement, si ces travaux ont permis de préciser la sémantique des réseaux, ils ont eu des répercussions négatives sur l'inférence des connaissances. Ainsi, que ce soit pour le cas des logiques terminologiques ou des graphes conceptuels, il n'est pas possible d'effectuer des raisonnements du type de ceux que nous avons présenté dans la section précédente [Sat96, RDBFPD96]. Par exemple, ces formalismes ne permettent pas de modéliser une propriété comme la transitivité, puisque leurs dérivations se limitent le plus souvent à un raisonnement sur les concepts (spécialisation/généralisation, spécification de concepts, etc.).

En recherche d'informations, nous avons autant besoin d'un modèle de représentation à sémantique claire et précise que d'un mécanisme d'inférence efficace permettant de bien gérer les propriétés des concepts et des relations. Malheureusement, les formalismes de représentation de connaissances d'aujourd'hui vérifient la première condition, mais peu d'entre eux présentent des mécanismes d'inférence adéquats pour les besoins de recherche, principalement, à cause du manque de raisonnement sur les relations. Dans la littérature, il est souvent question d'améliorer la capacité d'inférence de ces formalismes. Par exemple, dans [LR95], le système CARIN munit les logiques terminologiques de règles de dérivation de la logique de premier ordre. Dans la section 2.4, nous donnons des types d'inférence qui ne peuvent pas être modélisés par les logiques terminologiques et qui sont pourtant utiles en recherche d'informations.

Le système RIME est parmi les premiers systèmes basés sur un formalisme sémantique (les dépendances conceptuelles de schank) à souligner l'importance de gérer les relations et d'explorer leurs structures. Dans la section suivante, nous donnons une présentation détaillée de ce système.

2.3.3 L'Approche de RIME

Le système RIME est destiné à des physiciens dont les besoins en informations sont très précis incluant généralement des concepts complexes et structurés. Ces utilisateurs ont ainsi besoin de communiquer avec un système de recherche d'informations pouvant représenter et évaluer de tels concepts. Une conséquence directe est que ce système devra être en mesure de retourner des résultats dont la précision est au niveau de celle utilisée par les physiciens pour les requêtes. En d'autres termes, le système devra être orienté vers la précision des réponses et supporter ainsi un langage d'indexation expressif, précis et robuste. L'idée de base du système RIME est que seule l'utilisation explicite des relations sémantiques dans la construction des index des documents permet de remplir la contrainte de précision. Ceci est bien entendu en contraste avec les indexations classiques à base des mots-clés où les relations entre termes ne sont pas prises en considération.

a) La Représentation des Connaissances dans RIME

Le modèle de représentation interne de RIME a été défini en coopération avec les spécialistes compétents du domaine de la radiologie [Ber88]. Il est issu d'un compromis entre les besoins ultérieurs des utilisateurs et les possibilités de traitements envisageables. La définition du modèle fut inspirée des dépendances conceptuelles de Shank [Sch72]. Elle est basée sur un modèle sémantique utilisant des représentations arborescentes dans lesquelles les noeuds feuilles sont des concepts primitifs du domaine, et les noeuds non-terminaux sont des opérateurs sémantiques. Ces derniers dénotent des relations sémantiques et permettent la construction de concepts complexes. Les concepts primitifs représentent le niveau le plus bas de représentation des documents et des requêtes. Les opérateurs sémantiques permettent l'expression de concepts structurés en combinant soit les concepts primitifs, soit des concepts structurés déjà construits.

Dans RIME, le vocabulaire est précis et très spécialisé. Les connaissances sémantiques sont bien formalisées et déterminées. Ainsi, le langage médical est beaucoup moins ambiguë que la langue naturelle en sa totalité. Il est donc possible d'établir une représentation interne correspondant précisément à la sémantique d'une phrase dans ce langage. De ce fait, nous pouvons totalement exprimer cette sémantique par l'intermédiaire d'une grammaire. Une représentation possible des éléments décrits par cette grammaire est une mise en forme arborescente.

Tous les arbres manipulés sont le résultat d'une phase de compréhension traduisant le langage médical en des structures sémantiques. Dans RIME, chaque concept, élémentaire ou non, possède un type, comme par exemple *lésion*, *organe*, etc. Ces types sont organisés dans une hiérarchie. La grammaire fixe ainsi à la fois la structure du langage et sa sémantique.

Théoriquement un concept d'un type sémantique donné peut toujours être considéré comme étant constitué par d'autres concepts plus fins reliés entre eux par une certaine relation sémantique [Ber88]. Ainsi, le concept "hypertrophie" peut être vu comme une "augmentation" portant sur le "volume". Les termes "augmentation" et "volume" étant des concepts plus fins, et "porte-sur" étant la relation entre eux. La décomposition peut éventuellement se poursuivre de cette manière. Cependant, dans RIME, en conformité avec la vision objectiviste (voir section 2.2.1, une limite a été établie par la définition de certains concepts primitifs, dits *concepts élémentaires* [Ber88] qui ne peuvent jamais être décomposés dans l'application. L'approche adoptée est que des concepts plus fins ne doivent être considérés que si les besoins de l'application l'exigent. À l'inverse de la notion de concept élémentaire, les concepts qui sont composés à partir des concepts élémentaires

mis en relations sémantiques, ou qui peuvent être décomposés en ceux-ci, sont appelés des *concepts complexes*.

Par exemple, le concept complexe suivant est dérivé de la phrase "*Une hypertrophie de densité tissulaire du lobe gauche de la théroïde*":

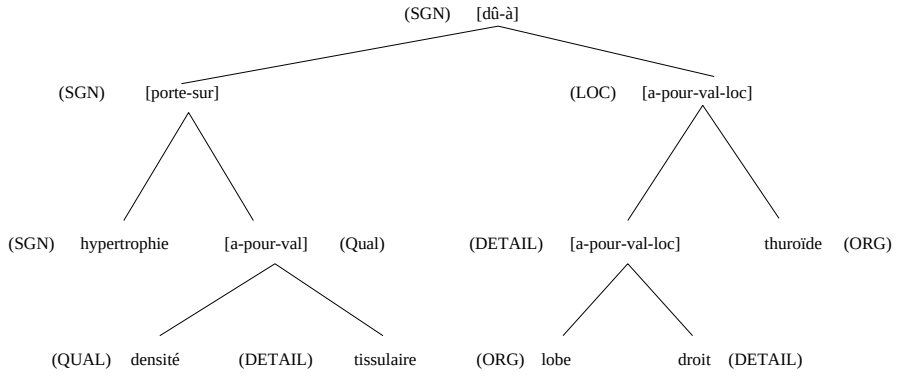


Figure 2.6. Une arborescence RIME pour la phrase "Une hypertrophie de densité tissulaire du lobe droit de la théroïde"

À l'image du modèle des dépendances conceptuelles de Shank, le systèmes RIME inclue un nombre relativement restreint de relations. Ces dernières se comportent comme des opérateurs unaires ou binaires. il s'agit de *relations primitives* regroupées dans les types suivants:

booléen: Ce type regroupe les opérateurs logiques classiques tels que le ET ($\wedge$), le OU ($\vee$) ou le NOT ($\neg$)

sémantique: RIME comporte sept relations sémantiques, à savoir:

i) "*dû-à*" exprimant une relation de causalité
ii) "*per-de-déd*" pour "permet de déduire", exprimant une relation de déduction.
iii) "*montré-par*" exprimant une relation de démonstration.
iv) "*porte-sur*", représentant une relation de localisation.
v) "*en-rel-topo-avec*" pour "en relation topologique avec" , représentant une relation topologique.
vi) "*a-pour-val*" pour "a pour valeur" exprimant une relation d'évaluation.
vii) "*a-pour-val-loc*" pour "a pour valeur locative " exprimant une relation de position précise.

La grammaire de RIME spécifie, entre autres, quels types de concepts peuvent être liés par une relation primitive donnée et quel type de concept cela engendre. Par exemple, la relation sémantique binaire "*porte-sur*" peut lier des concepts de type SGN (pour signe) et LOC(pour localisation) ou des concepts de type SGN et FCT (pour fonction):

$$[porte\text{-}sur](\text{SGN},\text{LOC}) \text{ ou } [porte\text{-}sur](\text{SGN},\text{FCT})$$

Dans les deux cas, le concept construit par le biais de l'application de la relation *porte-sur* est de type SGN.

b) Le Processus de Correspondance dans RIME

L'évaluation d'une requête dans RIME se base sur le modèle logique de Rijsbergen. Cette évaluation détaillée dans [Nie90], ramène le processus de correspondance à la recherche de la plausibilité d' une implication logique entre la requête Q et le contenu des documents D du corpus, par rapport à un ensemble de connaissances K: $R_K(D, Q) = P_K(D \rightarrow Q)$

Quand il n'est pas possible d'inférer directement la requête à partir du document, trois façons d'évaluer $P_K(D \rightarrow Q)$ sont possibles [Nie89, Nie90]. Il s'agit soit de modifier les connaissances K du système, soit la prémisse D, soit la conclusion Q de l'implication. La modification de la prémisse D, à savoir l'arborescence sémantique associée au document, semblait être plus adéquate à concevoir d'un point de vue pratique puisqu'elle évite des transformations superflues et des contrôles supplémentaires d'un coût assez considérable [Nie88].

L'approche par modification du document consiste à transformer le document $w_o = D$ en des documents plus génériques w_n qui sont à comparer avec la requête Q. La valeur de pertinence associée au document initial est celle calculée par l'évaluation de la requête par rapport au dernier document produit par les transformations. La description D étant constituée d'une arborescence valide, les règles de transformations utilisées par RIME étant également valides, il n'est possible de déduire que des document w_n valides.

Typiquement, cette approche est implantée par le biais de la sémantique des mondes possibles de Kripke [Kri63, Kri71]. Le document initial et ses possibles transformations correspondent à des *états concevables* ou encore à des *mondes possibles*. Les transitions entre ces mondes sont structurées en un graphe, par rapport à une relation σ exprimant la certitude du passage d'un monde vers un autre. Si les mondes correspondent à des interprétations logiques données, la valeur de vérité de la formule Q sera fonction du fait qu'elle soit une conséquence sémantique ou pas des formules associées aux mondes possibles. Le document sera jugé pertinent si tous les modèles qui rendent vrais les formules d'un de ses mondes possibles rendent également vraie la formule Q. Une logique modale floue a été utilisée pour obtenir une évaluation continue de la fonction de certitude P, au lieu des valeurs de vérité binaires dans les logiques modales standards.[11]

Le processus de transformation de RIME est basé sur un nombre restreint de règles de dérivation. Ces règles permettent principalement d'exploiter certaines propriétés sémantiques des relations primitives utilisées. Si l'on désigne par A et B des arborescences sémantiques, et par [rel] une relation primitive de RIME, ces règles se limitent à ce qui suit, où $\rightarrow_c$ exprime la plausibilité $c \in]0, 1]$ d'une implication:

- $[rel](A, B) \rightarrow_1 A$ et $[rel](A, B) \rightarrow_1 B$;
- si $a \rightarrow_c b \in K$ et $a = A$ et $b = B$, alors $A \rightarrow_c B$;
- si $A \subseteq B$, alors $A \rightarrow_1 B$, où $\subseteq$ représente l'inclusion d'arbres;
- si $A \rightarrow_c A'$, avec A et A' ayant la même type sémantique, nous avons alors: $[rel](A, B) \rightarrow_c [rel](A', B)$;
- si $C = [rel_1](A, B)$ et $A = [rel_2](E, F)$ et (A, C, E) appartiennent au même type sémantique, alors $[rel_1](A, B) \rightarrow_c [rel_1](E, B)$.

11. Des détails supplémentaires sur le processus de transformation et les fondements théoriques de l'approche se trouvent dans [Nie90].

La plausibilité c sera utile pour calculer la certitude de passage d'un monde vers un autre exprimée par la relation σ.

c) Les Problèmes du Système RIME

Les règles de RIME sont correctes dans le sens où elles correspondent à des cas réels où l'implication entre les arborescences peut formellement être prouvée, avec une plausibilité $0 < c \leq 1$. Le problème ici est que ces règles sont définies en extension. En d'autres termes, il n'est pas possible de prouver que cette extension est *complète* par rapport à un problème général de preuve exprimé en $A \rightarrow_c B$, pour n'importe quelle configuration de A et B.

Ce problème est dû au fait qu'il n'existe pas de sémantique formelle associée à la syntaxe du langage de représentation de RIME. Seuls le sens commun et l'expertise médicale ont été utilisés pour la construction de ce langage. Cela constitue une limitation théorique importante du modèle opérationnel sous-jacent: étant donnée une requête Q, nous ne pouvons pas être certain que tout document impliquant logiquement cette requête sera retrouvé par le système. Seuls les documents satisfaisant les règles de dérivation ci-dessus seront retrouvés. En d'autres termes, il n'est pas prouvable que l'ensemble des règles de dérivation présentées par RIME soit suffisant pour implanter une approche logique à base d'arborescences sémantiques.

La probable incomplétude de RIME est liée à un manque de formalisation: le système de dérivation ne se base sur aucun modèle logique formel, même si par le biais de la sémantique des mondes possibles, il peut être expliqué. Une solution possible consiste à transformer les arborescences de RIME en un formalisme pourvu d'un mécanisme d'inférence complet par rapport à une sémantique claire. Mulhem dans [Mul93] propose de transformer ces arborescences en des graphes conceptuels. Le lien direct de ce formalisme avec la logique de premier ordre explique ce choix. La projection des graphes est alors utilisée comme fonction de correspondance. Cela a donné lieu au prototype PRIME (Prototype de Recherche d'Information MédicalE) dont une présentation se trouve dans [MMFB97].

Pour notre part, nous avons détecté un lien évident entre RIME tel qu'il a été présenté dans [Nie90] et les logiques terminologiques (LTs). D'ailleurs, l'approche suivie par Nie pour l'évaluation de la certitude des données (certitude totale, probable, probablement-imposible, impossible) rappelle la sémantique à quatre valeurs développée par Patel-Schneider [PS89] pour les logiques terminologiques.

Par ailleurs, dans une arborescence de type [Relation](C_1, C_2), le second concept C_2 décrit généralement (via la relation) une localisation, une valeur qualitative, un contexte, ... attaché(e) au premier concept C_1. Le second concept (avec la relation) constitue une description du premier concept. Nie considère que dans une telle formation, le premier concept est un *concept principal* ou un *gouverneur* alors que le second concept est *dépendant* de celui-ci. Par exemple, dans "[porte-sur](*opacité*, *poumon*)", le signe "*opacité*" est décrit par la localisation "*poumon*", formant un signe plus détaillé "*opacité pulmonaire*". On peut alors considérer que le concept principal "*opacité*" représente une sémantique (un phénomène médical) moins détaillée que toute la description ("*opacité pulmonaire*"). Un tel raisonnement est similaire à celui adopté par KL-ONE et les logiques terminologiques qui en résultent: pour la construction d'un terme complexe, on part d'un concept primitif, auquel on rattache un ensemble de propriétés. Ces dernières sont introduites par le biais d'un ensemble de rôles (relations) et d'opérateurs de formation de termes [12]. Dans la figure 2.7 suivante,

12. Un exemple de logique terminologique est présenté dans la section 2.4.4 a) (page 53).

nous montrons une translation possible des arborescences de RIME en des termes terminologiques:

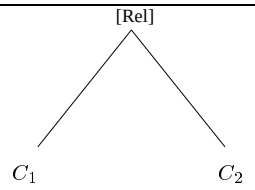
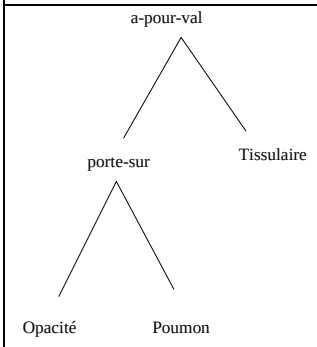
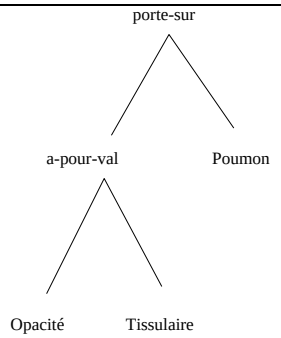
RIME	*Logique Terminologiques*
	$(\mathbf{and} C_1 (\mathbf{all} Rel C_2))$
Exemples	
	(**and** (**and** Opacité (**all** porte-sur Poumon)) (**all** a-pour-val Tissulaire))
	(**and** (**and** Opacité (**all** a-pour-val Tissulaire)) (**all** porte-sur Poumon))

Figure 2.7. Translation des Arborescences de RIME en LTs

Dans cette translation, les opérateurs sémantiques "[Rel]" de RIME correspondent à des rôles de la logique terminologique. Chaque rôle est associé à l'opérateur de formation de concept *all* pour préserver la notion du concept gouverneur. Les opérateurs booléens ET, OU correspondent aux opérateurs de formation de concepts *and* et *or*, respectivement. Enfin, l'opérateur booléen NOT peut être associé à l'opérateur de formation de rôle *not*.

Une fois la translation ainsi réalisée, la relation de subsomption introduite par les logiques terminologiques joue le rôle de la fonction de correspondance. Cette translation est utile à plus d'un titre:

- RIME comporte un nombre assez restreint d'opérateurs de formation de termes. La relation de subsomption dans ce cas est complète et décidable [Neb90a]. Le problème de la probable incomplétude du système peut ainsi être résolu en choisissant cette dernière comme mécanisme

d'inférence. De plus, mis à part l'opérateur de formation de concept *some*, RIME admet la même puissance d'expressivité que MIRLOG [SS96], proposée comme une logique adéquate pour la recherche d'informations.

- Le passage des arborescences vers un formalisme basé sur les logiques terminologiques permet de détecter de façon plus simple les possibles équivalences et similitudes sémantiques entre les divers arbres. Dans RIME, ces similitudes sont détectées grâce à des procédés de normalisation d'arbres. Ces procédés imposent un ordre particulier sur la combinaison des opérateurs. Par exemple, une description via l'opérateur [a-pour-val] est prioritaire par rapport à une description via l'opérateur [porte-sur]. Ainsi, contrairement à la première arborescence de l'exemple de la figure précédente, la seconde arborescence admet une forme normalisée. Sémantiquement ces deux arbres sont équivalents. Avec une représentation en termes terminologiques, de telles équivalences sont automatiquement détectées au niveau de la relation de subsomption, par la biais des deux règles suivantes valide dans toute logique terminologique: (**and** B (**and** C D)) $\equiv$ (**and** B C D) et (**and** A B) $\equiv$ (**and** B A)

- Si le système RIME fournit un cadre intéressant pour représenter les relations et tenir compte de leur structure ainsi que de leur sémantique (via l'introduction des rôles), il n'en reste pas moins que du fait qu'il s'apparente à une logique terminologique, il partage avec celle-ci l'absence d'inférence sur les relations et leurs propriétés. Ainsi, excepté la négation, RIME ne présente pas d'opérateurs particuliers sur les relations sémantiques. Par exemple, il n'est pas possible d'inférer que le document contenant la relation *en-rel-topo-avec*(*poumon*, *thorax*) est pertinent pour la requête *en-rel-topo-avec* (*thorax*, *poumon*), sachant pourtant que *en-rel-topo-avec* est une relation symétrique.

Ainsi, mise à part l'absence d'une sémantique claire, le principal inconvénient de RIME est qu'il ne tienne pas compte des propriétés mathématiques des relations (symétrie, transitivité, etc.). De plus, il n'existe pas de moyens pour exprimer des liens sémantiques entre les relations. Par exemple que telle relation est une spécialisation de l'autre, ou que telle autre relation R_1 est un inverse d'une relation R_2. Ces propriétés peuvent pourtant avoir des incidences importantes sur les performances qualitatives du système, puisqu'elles interviennent dans la phase de correspondance. La translation du système RIME vers une logique terminologique confirme cet inconvénient. Les logiques terminologiques sont justement connues pour présenter très peu d'opérateurs de formation de rôles, et donc un mécanisme limité de raisonnement sur les relations.

2.3.4 Conclusion

Les trois approches décrites dans cette section ont tenté, chacune à sa manière, de présenter un mécanisme d'inférence sur les relations. Il apparaît toutefois clair que ces mécanismes sont insuffisants pour gérer convenablement les relations et toute l'étendue de leurs propriétés. Certaines inférences évidentes sur les relations ne sont pas prises en considération. Le système RIME, les réseaux sémantiques ou encore les formalismes de représentation de connaissances qui en découlent n'offrent que des capacités limitées d'inférence sur les relations. Comme ces formalismes servent généralement de langage d'indexation, il est souhaitable d'identifier les types de raisonnement sur les relations qui leur manquent le plus. Dans la section suivante, nous menons une étude dans laquelle les besoins en recherche d'informations en termes de représentation et de gestion des relations seront identifiés.

2.4 L'utilité des Relations en RI

Nous avons déjà souligné dans l'introduction le besoin de considérer les relations entre termes. Les sections précédentes ont confirmé ce besoin, tout en montrant que la prise en compte de ces relations et leurs caractéristiques est parfois nécessaire. En recherche d'informations, des applications récentes montrent qu'il convient non seulement de représenter les relations mais aussi de traiter leurs propriétés. Dans ce qui suit nous énumérons quelques unes de ces applications. Une attention particulière sera donnée à un modèle de recherche de documents structurés dont l'implantation en logiques terminologiques nécessite une multitudes d'extensions, essentiellement dues à l'abscence de raisonnements sur les rôles dans ces logiques.

2.4.1 Les Applications Hypermédias

Grâce au Web (WWW) et ses outils de navigation comme Netscape ou Internet Explorer, la notion d'hypertexte et d'hypermédia est devenue très familière. Un hypertexte[13] est une représentation non-linéaire d'une information textuelle sous forme d'un graphe de noeuds connectés par des liens. Un noeud représente une portion d'information, et le lien modélise la relation entre deux noeuds. L'utilisation de l'hypertexte se fait principalement par le biais de la navigation dans le graphe. Plusieurs parcours de navigation sont alors possibles. Le modèle le plus simple considère un seul type de lien et des noeuds qui ne peuvent être que des documents complets. Les liens permettent dans ce cas de relier les documents entre eux. Cependant, la plupart des systèmes hypertextes permettent la définition de liens à partir d'une partie d'un document vers des parties d'autres documents. Il est alors question de définir quelle partie du document peut être utilisée comme source ou destination d'un lien. Pour un texte, on peut ainsi utiliser des chapitres, des sections, des paragraphes, etc. De même pour les types de liens, ils peuvent être typés pour modéliser une relation de type particulier entre certains noeuds. Les liens peuvent être statiques, calculés une fois pour toutes et liés aux documents concernés, ou bien calculés dynamiquement. Dans le deuxième cas, les documents destination du lien sont construits lorsque ce lien est activé par l'utilisateur. L'*hypermédia* est une généralisation d'un hypertexte, où l'information peut être constituée de n'importe quel type de média (texte, image, son, graphique, vidéo, etc.).

Malgré une bonne interface, les systèmes hypermédia souffrent du problème de l'orientation. En effet, la multiplicité des parcours qu'un utilisateur peut emprunter dans un gigantesque graphe finit par le désorienter. Une forme de désorientation connue est la distraction. Elle provient du fait qu'à partir de chaque noeud du réseau, l'utilisateur peut être intéressé par d'autres informations qui n'ont pas de relation directe avec le but initial de sa recherche. Grâce à une interrogation par le contenu, propre aux systèmes de recherche d'informations, il est possible d'offrir à l'utilisateur le moyen de se situer à tout moment dans une partie de l'hypermédia conforme à ses objectifs.

Au niveau de la recherche d'informations, le principal avantage à tirer des approches hypermédias est d'utiliser la navigation comme un moyen pour la résolution du problème de l'imprécision des formulations de requêtes. À partir de la réponse du système de recherche, l'utilisateur peut parcourir avec le système hypermédia le graphe des documents et identifier d'autres documents, ne figurant pas dans la réponse, comme pertinents à son besoin. Ces derniers peuvent alors être pris en considération par le système dans la reformulation de la requête.

La complémentarité entre l'approche hypermédia et celle de la recherche d'informations a été

13. Une bonne présentation des hypertextes se trouve dans [Con87].

identifiée [Ago93, AMC95]. Le système hybride résultat présente à l'utilisateur une interface lui permettant d'accéder aux documents, à tout moment de la recherche, soit par des requêtes textuelles soit par navigation dans le graphe d'hypermédia. Plusieurs systèmes ont été bâtis sur une telle démarche. Citons comme exemple, le système MMIR [Dun91] ou encore le système de recherche hypermédia HyperRIME [Khe95].

Pour les besoins de la recherche, les hypermédias considèrent typiquement trois types de relations dans les index des documents:

- les liens de *composition*, c'est-à-dire comment un document est formé à partir de sa structure. En d'autres termes il s'agit de spécifier les noeuds composant un document donné;
- les liens de *succession*, c'est-à-dire comment se succèdent les parties d'un documents en vue de leur lecture;
- les liens *relationnels*, regroupant des liens sémantiques ou des liens de référence entre les parties d'un document ou entre des documents différents. Dans [Khe95], des liens comme «même-sujet», «illustré-par», «se-réfère-à», ... , sont considérés;

Le système MMIR (MultiMedia Information Retriever) est un système hybride combinant l'approche hypermédia avec l'approche système de recherche d'informations pour profiter des avantages de chacune. Dédié à un système hypermédia pour un manuel du code de la route comportant un certain nombre de règles de circulation auxquelles sont associées des images d'illustration, il intègre des liens de *référence* et des relations de *succession*. La structure de l'hypermédia est utilisée pour l'indexation des données non-textuelles (en l'occurrence les images) en exploitant certains liens les reliant aux données textuelles. La composante recherche d'informations de MMIR est basée sur le modèle probabiliste de recherche d'informations [vR79].

Dans [Khe95], Kheirbek présente une approche intégrant un modèle de recherche d'informations avec un modèle hypermédia dans une seule plate-forme, décrite dans le formalisme des graphes conceptuels. Par rapport à MMIR, ce modèle a la particularité d'adresser typiquement une recherche orientée vers la précision des réponses. Le modèle proposé exploite deux types de connaissances du domaine: celles relatives à la structure des documents et celles relatives à leur contenu. Les documents dans ce système sont représentés par une arborescence où les index des sections supérieures sont calculés par combinaison des index de leurs sous-sections. Les composants de base (feuilles de l'arborescence) sont indexés par des graphes conceptuels canoniques [Sow84]. Les noeuds intermédiaires correspondant aux sections supérieures sont alors indexés par des graphes produits par la jointure maximale des graphes indexant leurs composants respectifs.

La recherche s'effectue par application de la fonction d'implication directe $D \rightarrow Q$ sur les sections supérieures du document D (opération *fetch*). Quand une de ces sections est retrouvée, on peut raffiner la réponse en appliquant la fonction d'implication indirecte $Q \rightarrow D$ à ses sous-sections (opération *browse*). Le but poursuivi est de fournir à l'utilisateur la partie du document qui répond spécifiquement à son besoin. L'opération *fetch* est appliquée pour présélectionner dans un premier temps les documents ou les parties des documents qui sont pertinents pour la requête, alors que l'opération *browse* vise à sélectionner la composante la plus spécifique à la requête.

Kheirbek a mentionné l'importance d'exprimer la nature récursive de certains liens structurels comme la relation de composition. Il s'agit par exemple de spécifier que la relation *Composé-de*

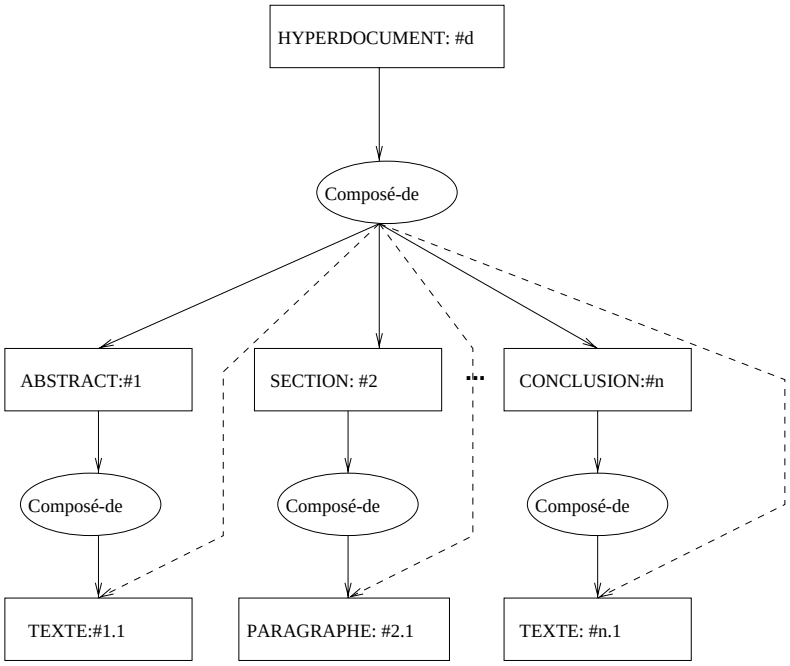


Figure 2.8. Transitivité de la relation "*Composé-par*"

est transitive. Une telle propriété est nécessaire à modéliser quand la requête fait référence à des données implicites contenues dans l'index.

Dans l'exemple de la figure 2.8, les arcs en pointillé représentent des liens implicites engendrés du fait que la relation *Composé-de* est transitive. Lors de l'implantation de son système HyperRIME à base de graphes conceptuels, Kheirbek a été confronté à la nécessité de raisonner sur des propriétés de ce type. Ainsi, si un utilisateur cherche un document contenant (composé d') un texte, il risque fort de ne pas retrouver le document hypermédia D de cette figure. Il n'est pas possible de déduire le lien de composition implicite entre [HYPERDOCUMENT: #1] et [TEXTE:#1.1] par le biais d'un simple raisonnement sur les graphes. En effet, la projection utilisée ici comme fonction de correspondance n'est autre qu'une simple comparaison directe de graphes, ne faisant intervenir que des déductions limitées à la spécialisation de concepts.

D'autres relations structurelles comme *illustré-par*, *référencé-par* ou *commenté-par* sont également transitives: si un document est illustré par un poème, et que ce dernier est illustré par une image particulière, alors il faudra en déduire que le document est illustré par cette image. Pour pallier à cette limitation des graphes conceptuels, et afin de se conformer à son modèle théorique, la solution préconisée par Kheirbek fut de pré-calculer la fermeture transitive de toutes ces relations (voir figure 2.8): toutes les déductions possibles des connaissances implicites sont ainsi explicitement ajoutées au graphe du document, de façon à ce que la projection de la requête tienne compte de cette propriété. Une telle approche n'est pas bien entendue idéale du fait de son caractère *ad-hoc*, et de son coût de représentation.

2.4.2 La Recherche d'Images

Les images ont constitué la première forme de document non-textuel qui a été incluse dans un processus de recherche d'informations. Afin de caractériser les particularités de ce média, plusieurs modèles de recherche d'images ont été proposés. Ces derniers précisent en quoi consiste une bonne représentation de l'image en vue d'une recherche efficace de ce type de média. Un bon résumé des modèles d'images proposés dans la littérature se trouve dans [LKT+95].

Ainsi, le modèle EMIR^2 [Mec95c] est présenté par Mechkour comme un modèle général de représentation des images fixes permettant une description complète de ces dernières. Ce modèle se veut indépendant de tout formalisme de représentation de données ou de connaissances. Il fournit un cadre dans lequel les caractéristiques de l'image fixe peuvent être représentées. L'extraction automatique du contenu sémantique de ces images n'est pas traitée dans ce modèle. L'indexation se faisant à travers un outil semi-automatique, se contentant d'assister l'indexeur [CM97]. L'auteur semble ainsi suivre la remarque de Narasimhalu [Nar95] où les outils de compréhension automatique du contenu des images doivent être remplacés par des outils interactifs.

Dans ce modèle, l'index d'une image devra inclure cinq vues. La première vue a trait au niveau physique de l'image, correspondant à une matrice de points, et au format de l'image. Les quatre autres vues correspondent au niveau de description logique de l'image et reflètent ainsi son contenu sémantique:

La vue structurelle: l'image est vue comme un ensemble d'objets. Les objets de l'image pouvant eux même être composés d'autres objets. La vue structurelle de l'image a la forme d'un graphe orienté ayant pour noeuds les objets de l'images, et pour arcs des *relations de composition structurelle*. Cette vue permet de décomposer l'image en conservant la structure des éléments présents dans l'image. Par exemple, une image peut contenir un arbre et une maison. La maison peut être composée de murs et de toits. Dans ce cas, la vue structurelle comportera un objet image o_{image} composé d'un objet o_1 (l'arbre) et d'un objet o_2 (la maison), qui sera lui-même composé d'un objet o_3 (le mur) et d'un objet o_4 (le toit).

La vue spatiale: cette vue est très importante dans le cas de la recherche d'images. Elle permet de décrire la forme des objets et les relations spatiales qui les lient ensemble. Il s'agit de positionner les objets les uns par rapport aux autres. La vue spatiale d'une image est un graphe dans lequel les noeuds sont les objets spatiaux, et les arcs sont des *relations spatiales*. Les objets spatiaux sont caractérisés par la forme de l'élément dans le plan de l'image: point, segment, polygone, etc. Mechkour distingue 12 relations spatiales élémentaires (Nord, Sud, Couvre, etc.).

La vue perceptive: elle regroupe un certain nombre d'attributs attachés à la perception visuelle des images. Trois attributs sont pris en compte dans EMIR^2: la couleur, la texture et la brillance.

La vue symbolique: la vue symbolique associe une description sémantique à l'image et aux objets de l'image. La description regroupe un ensemble de termes pouvant être reliés ensemble par des *relations sémantiques*. Par exemple, une image dans laquelle apparaît un homme âgé tenant une canne peut être décrite par une description comme: *tenant*(*Homme-Agé*, *Canne*). Cette description est complétée par des attributs formatés associés aux termes.

L'importance des relations dans le modèle d'EMIR^2 est ainsi primordiale. Certaines relations présentent des propriétés dont il faudra tenir compte dans le processus de correspondance. Par exemple, certaines relations spatiales ou structurelles présentent des propriétés mathématiques intéressantes (symétrie, transitivité, inversion, etc..). De plus, plusieurs relations symboliques ou spatiales peuvent être regroupées en classes et sous-classes (comme pour les concepts). Ainsi, la relation *Touche* est un spécifique de *proche*, c'est-à-dire qu'à chaque fois qu'un objet o_1 touche un autre objet o_2, on en déduit que o_1 est proche de o_2. Une requête adressant deux objets qui soient proches devra avoir comme réponse un document dans lequel ces deux objets se touchent. Il est également utile d'avoir une relation de spécialisation entre des relations d'arités différentes. Par exemple, la relation *entre* mentionnant qu'un objet o_1 est "entre" deux autres objets o_2 et o_3 dans le plan de l'image, implique nécessairement l'existence d'une relation *a-droite* entre o_1 et o_2 et *a-gauche* entre o_1 et o_3. On en déduit que la relation spatiale d'arité 3 *entre* est un spécifique des relations binaires *a-droite* et *a-gauche*. Enfin, des relations symboliques peuvent être agencées (composées) pour donner d'autres relations implicitement véhiculées par l'index du document.

Le modèle opérationnel développé par Mechkour est basé sur le formalisme des graphes conceptuels. Les graphes conceptuels présentent l'avantage de pouvoir unifier assez aisément les vues du modèle proposé. La représentation globale de l'image correspond ainsi à un unique graphe conceptuel g (voir figure 2.9). En effet, chaque objet de EMIR^2, qu'il soit structurel, symbolique, perceptif, ou spatial, est représenté par un concept particulier dans le graphe conceptuel g. De la même manière les relations du modèle, à savoir les relations de composition, les relations spatiales et les relations symboliques, sont représentées par des relations entre les concepts du graphe conceptuel g. Les attributs formatés sont quant à eux représentés par des concepts et associés à leurs objets respectifs par une relation prédéfinie.

Malheureusement, les graphes conceptuels manquent de capacités d'inférence et de déduction. Il n'est pas possible, entre autres, d'élaborer des raisonnements sur les propriétés des diverses relations utilisée. Mechkour a ainsi adopté une approche *ad-hoc* de type de celle présentée dans la section précédente, où toutes les relations implicites obtenues par application des propriétés des relations sont ajoutées au graphe (voir figure 2.9). L'approche suivie s'est toutefois contentée de traiter les propriétés mathématiques des relations, ignorant ainsi les possibles liens de hiérarchies entre les diverses relations symboliques utilisées. De même les agencements possibles des relations ne sont pas pris en compte.

Dans [Meg95], un modèle général d'images est également introduit. Tout comme Mechkour, les méthodes d'extraction automatique des connaissances ne sont pas abordées et seul l'aspect modélisation du contenu de l'image est décrit. Une image est alors vue comme un objet à trois niveaux :

Forme : l'image est vue comme une combinaison de régions colorées et sa représentation se définit par le triplet (A, π, F). A est une région de l'image, π est une partition de A c'est-à-dire un ensemble de points, $\pi = \{S_1, \ldots, S_n\}$ et enfin la fonction F associe à chaque point de A une couleur.

Contenu : ce niveau comprend un ensemble d'éléments pour la modélisation sémantique du contenu des images et son organisation. Cette modélisation se compose donc d'entités interreliées et elle repose sur trois mécanismes d'abstraction : la classification des entités (classes décrivant des collections d'objets similaires), leur agrégation (définition des propriétés des classes) et leur spécialisation (taxinomie des classes avec une relation "IsA").

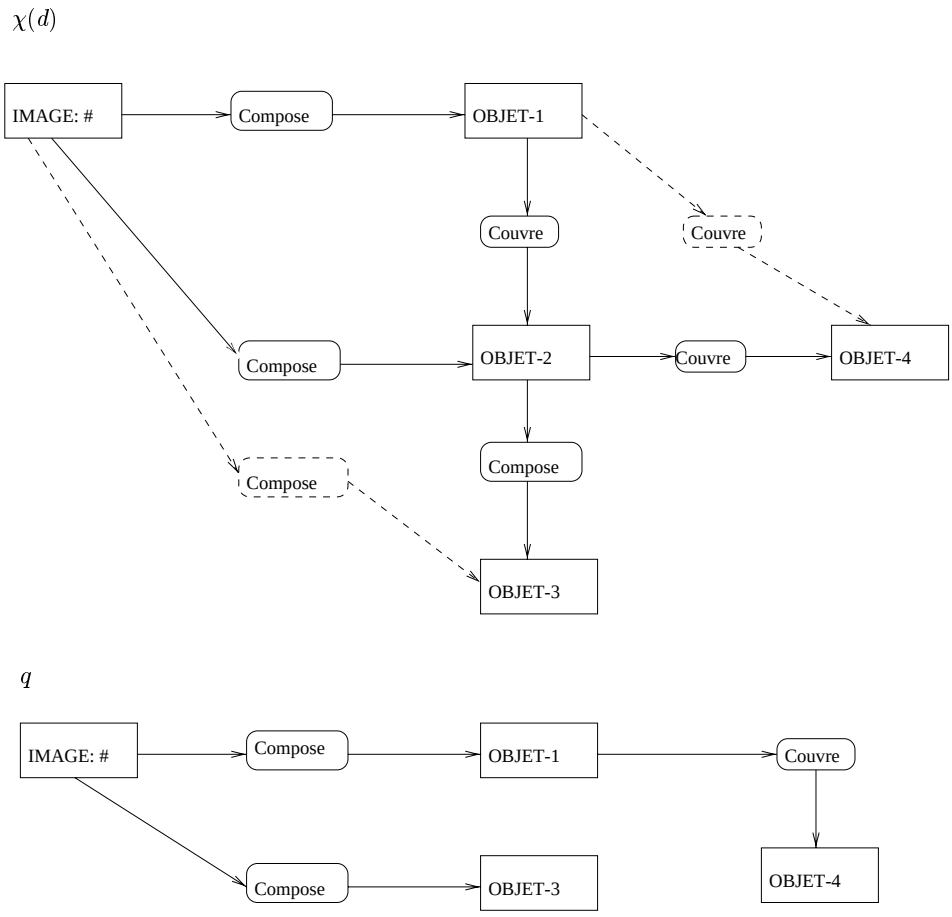


Figure 2.9. Les relations "compose" et "couvre" sont transitives

Liaison : ce niveau établit la jonction entre les deux niveaux précédents. Il s'agit d'une fonction qui associe à des ensembles disjoints de spots des entités du "content level". Il s'agit donc de la construction de la représentation de l'image puisque cette fonction lie les régions physiques de l'image à des entités conceptuelles.

Comme dans le cas d'EMIR^2, un ensemble de relations spatiales est nécessaire à modéliser en vue d'une recherche par le contenu (meets, overlap, etc.). Cependant, contrairement au modèle de Mechkour, où les relations entre les objets de l'image sont traitées au niveau des index, Meghini a choisi de traiter ces relations lors de la correspondance. Ces relations sont toutefois indispensables puisqu'elles peuvent fréquemment intervenir dans la recherche de données spatiales.

De plus, ces relations présentent des propriétés dont il faut tenir compte lors du processus de correspondance. Implanté grâce à la logique terminologique MIRLOG [MS96], le système sous-jacent souffre aussi d'un manque de raisonnement sur les relations imputable à la logique terminologique utilisée. En effet, les logiques terminologiques tout comme les graphes conceptuels ne supportent pas de mécanisme d'inférence sur les relations.

2.4.3 La Recherche de Documents Structurés

Il y a un intérêt accru dernièrement pour la structure que peut présenter les documents [CACS94, CR94, GFS94, Mac90, NBY96]. Pour plusieurs applications travaillant sur le texte intégral, c'est-à-dire sur des textes entiers et non pas seulement sur des résumés ou des extraits, il semble nécessaire de pouvoir au moins reconnaître la structure du texte, de façon à pouvoir distinguer, par exemple, un titre du corps du texte. Pour les systèmes travaillant sur des corpus spécialisés, il est souvent aussi nécessaire d'inclure la structure dans la représentation du document, puisqu'elle est parfois un aide indispensable à la compréhension du document.

Paradis [Par96] définit un modèle d'indexation pour les documents textuels qui tient compte de la structure des documents et d'autres informations complémentaires au discours. Le but du modèle est de spécifier les informations nécessaires permettant une bonne recherche de documents textuels structurés. Comme pour les applications précédentes, le modèle confirme le besoin de représenter certaines relations particulières. Cependant, il pose en plus , et de manière très claire, les propriétés syntaxiques (mathématiques) et sémantiques qui doivent être associées à ces relations. Le modèle proposé comporte deux composantes: le langage de représentation, qui définit de façon conceptuelle les informations du document, y compris les index eux-mêmes, et les règles de dérivation, qui, reprenant ce langage, permettent de déduire un type particulier d'index, à savoir les thèmes. Les thèmes reflètent aussi bien la structure statique du document que le processus dynamique lié à l'interaction avec l'utilisateur. Le contenu d'un document est donné par un ensemble de composants qui sont à leur tour définis en donnant leurs composants respectifs. La structure de base d'un document selon ce modèle est ainsi donné par une relation de *composition*, à laquelle s'ajoute une relation d'ordonnancement qui donne l'ordre, pas forcément linéaire, dans lequel le document doit être parcouru.

Le modèle distingue trois grands types de structure, à chacun correspond un type particulier de relation:

La structure linguistique: elle reflète l'organisation des symboles qui constituent le document textuel. Tout élément qui apparaît dans un document peut être considéré comme un symbole. Les mots, les syntagmes nominaux ou les phrases peuvent être considérés comme des symboles. La position de ces derniers dans le document est déterminée par la structure linguistique. Pour modéliser cette notion de structure linguistique, Paradis considère une relation de *composition* notée *part* permettant de regrouper les éléments textuels, une relation de *séquentialité* notée *séq* donnant l'ordre dans lequel doit être lu le texte, et enfin une relation de *dépendance* notée *dépendant* permettant d'exprimer qu'une information est subordonnée à une autre;

Les relations *part* et *séq* sont antisymétriques, irréflexives et injectives. La relation *dépendant* est antisymétrique et irréflexive.

La structure logique: La structure logique du document textuel reflète l'organisation explicite d'abstractions logiques représentant des parties de document. Il s'agit d'une extension à la structure linguistique, c'est-à-dire qu'elle partage les mêmes relations *part* et *séq* sauf que ces relations s'appliquent ici non pas à des mots, syntagmes, etc., mais à des sections, des chapitres, ... , ou plus généralement à toute entité d'informations contenant suffisamment d'informations pour être compréhensible. Une relation de *référence*, notée *réf* permet au niveau logique le renvoie à d'autres parties du même texte ou d'un autre document.

La structure de discours: Les structures linguistiques et logiques mettent en relation des parties

du document en se basant sur des règles syntaxiques définies a priori: des règles d'organisation des symboles du document textuel dépendant de la langue du document et des règles d'organisation d'entités d'informations dépendant du type de document. Ces deux structures sont indépendantes des idées ou des thèmes du document. Il existe pourtant une organisation de ces idées permettant de rendre le document compréhensible par le lecteur. Cette organisation est reflétée par ce qui est appelée la structure de discours du document. Pour ce faire, Paradis, utilise deux types de relations particuliers. La dominance, notée *domine* et l'intention notée *intention*. La relation d'intention exprime explicitement dans le texte du document qu'une idée est présente dans un segment de discours (Par exemple *Dans le chapitre 3, nous décrivons un modèle relationnel pour la RI*). La relation de dominance est très similaire à *part*, puisqu'elle exprime la composition des idées exprimées par une entité abstraite d'informations contenue dans le document. Elle présente ainsi les même propriétés mathématiques que *part* (antisymmétrie, irréflexivité, et injectivité).

Le modèle de Paradis dispose de son propre langage de représentation et de ses propres règles de déduction permettant de gérer les relations ci-dessus citées. Par exemple, la règle suivante permet de déduire la relation de dominance à partir de la relation de composition de la structure logique:

Règle 1 (Structure de discours)
$(part(\sigma_1, \sigma_2) \wedge division(\sigma_1) \wedge division(\sigma_2)) \Rightarrow domine(\sigma_1, \sigma_2)$

Cependant, outre le fait que ces relations et les règles qui leurs sont associées ne sont valides que pour des documents structurés textuels[14], des critères comme la complétude et la correction du langage par rapport à une sémantique n'ont pas été étudiés. Le but de l'auteur n'étant pas de faire une logique mais de détecter les éléments nécessaires à la description de documents structurés textuels. De plus, de l'aveu même de son auteur, le langage introduit s'apparente plutôt à une ontologie et donc à un méta-langage devant être implanté en un formalisme opérationnel. Du coup les propriétés mathématiques des relations ne sont pas forcément présente dans le modèle opérationnel. Ainsi, le prototype PIF utilisé pour valider le modèle est basé sur les *frames* comme langage de représentation, et ignore donc finalement les contraintes sur les relations.

2.4.4 La Recherche de Documents Multimédias

Il arrive souvent que le document comporte des informations de nature différente. Un document peut ainsi contenir du texte, des images, des graphiques, etc. Il peut être structuré, et peut bien entendu contenir des liens hypermédias. On parle dans ce cas de *document complexe*. Le travail décrit dans [CFM96] est le produit d'une étude menée par l'équipe MRIM pour définir les aspects qui doivent être considérés quand il s'agit de représenter puis de rechercher des documents complexes. Cette étude fournit un modèle formel et précis pour la recherche des documents structurés multimédia[15].

Une des caractéristiques de ce modèle est qu'il se base sur une intégration des résultats des travaux ayant concerné les monomédias, à savoir les modèles développés par Mechkour [Mec95c,

14. Nous verrons dans la section suivante que la recherche de documents structurés multimédia implique la représentation d'autres types de relations.
15. La notion de multimédia dans ce modèle se limite au texte, à l'image et au graphique.

Mec95a] et Meghini [Meg95] pour la recherche d'images, par Paradis [Par95, Par96] pour les documents textuels, par Kheirbek [Khe95] pour les hypermédias et par Mechkour [Mec96] pour les graphiques. Ce modèle est conçu comme des spécifications devant être implantées par tout formalisme dédié à la recherche de documents structurés et multimédia.

Récemment la logique terminologique MIRLOG [SS96] (Multimedia Information Retrieval LOGic) fut présentée comme une contribution originale aux modèles logiques de recherche d'informations, fournissant un cadre général pour le traitement des aspects inférentiels et représentationnels de la recherche d'informations. De plus, ses auteurs prétendent qu'elle est adéquate pour la recherche des documents multimédias hautement structurés et complexes. Cette logique dispose d'une sémantique complète et d'un mécanisme d'inférence décidable, grâce à l'adoption d'un langage de description relativement peu expressif. Dans [OC96a], nous avons essayé d'évaluer cette logique par rapport au modèle multimédia ci-dessus cité. L'objectif de l'étude étant d'analyser dans quelle mesure, la logique MIRLOG est capable d'exprimer les spécifications exigées par le modèle multimédia proposé.

Outre le fait que cette étude nous a permis de cerner les extensions nécessaires à appliquer au langage de représentation de MIRLOG en vue d'une implantation satisfaisante du modèle multimédia, elle a fait ressurgir les limites de cette logique terminologique à raisonner sur les relations (ou les *rôles* selon la terminologie adéquate). Certaines propriétés mathématiques comme la symétrie, la transitivité, etc. ne peuvent d'ailleurs être directement prises en compte par cette logique, même avec l'ajout de tous les opérateurs de formation de rôles dont disposent les logiques terminologiques les plus expressives.

Dans ce qui suit, nous faisons abstraction des difficultés de MIRLOG à représenter des attributs calculés, des méta-ensembles (ensemble d'ensembles) ou à modéliser des comportements de fonctions. Ces dernières sont pourtant utiles dans le contexte de la recherche des documents structurés multimédias. Nous nous limitons dans notre présentation aux éléments du modèle nécessitant un raisonnement sur les relations. Le détail de l'évaluation de MIRLOG par rapport au modèle multimédia proposé, ainsi que les solutions que nous avons adopté pour résoudre les limitations que présente MIRLOG, se trouvent dans [OC96a]. Nous introduisons d'abord succinctement MIRLOG, puis nous mentionnons ensuite les principales difficultés que nous avons rencontré avec cette logique pour l'implantation de certaines caractéristiques du modèle multimédia.

a) La Logique MIRLOG

Les logiques terminologiques[16] forment une famille de langages qui tirent leurs racines de KL-ONE, lequel fut né de la thèse de doctorat de Ronald Brachman et ses travaux sur le niveau "épistémologique" dans les réseaux sémantiques [BS85]. Elles adressent surtout le problème de la structure d'un concept, c'est-à-dire la description de concepts complexes, et de relations (héritage, rôles, attributs, etc.) entre ces concepts. Il est énuméré une douzaine de logiques terminologiques implantées [SOKK+90], pour n'en nommer que quelques-unes qui ont eu un certain succès, citons KRYPTON, KL-TWO, NIKL, KANDOR, LOOM, MESON, et CLASSIC. D'autres logiques ont plutôt une vocation théorique, c'est-à-dire qu'elles ont été définies pour l'étude de leur expressivité et de leur complexité: mentionnons par exemple, $\mathcal{FC}$, $\mathcal{ALC}$, $\mathcal{ALU}$, $\mathcal{NTF}$, etc. Une étude comparative et détaillée sur ces langages se trouve dans [Mac91].

16. Ces formalismes sont indifféremment appelés *Langages de description de concepts* ou *Langage de description basés sur les structures* ou encore *Logiques de description*.

Tout en gardant intact le noyau du langage KL-ONE, ces logiques terminologiques se sont généralement employées à en diminuer le pouvoir d'expression, afin d'améliorer l'efficacité des raisonnements sous-jacents, en particulier l'algorithme de subsomption. Plutôt que des formules, les logiques terminologiques manipulent des descriptions appelées termes. Un terme dénote soit un individu (une constante individuelle), soit une relation unaire (concept), soit une relation binaire (rôle) du domaine de discours.

Pour définir une logique terminologique, on choisit généralement un ensemble d'opérateurs de formation de termes correspondant chacun à des noms de relations, permettant de construire des termes complexes par combinaison de descriptions élémentaires. Il existe des opérateurs de formation de termes distincts pour les concepts et pour les rôles. Une caractéristique intéressante des logiques terminologiques est qu'elles ont une sémantique bien définie pour tous les opérateurs de formation de termes.

Les concepts (dénotés ici par les lettres C et D) de la logique Mirlog sont ceux du langage $\mathcal{ACC}$ [SSS91]. Ils sont construit à partir de concepts et de rôles primitifs, selon la syntaxe suivante:

$$
\begin{array}{rcrcl}
C, D & \rightarrow & \top & | & \text{(concept universel)} \\
 & & \bot & | & \text{(concept absurde)} \\
 & & A & | & \text{(concept primitif)} \\
 & & C \sqcap D & | & \text{(conjonction de concepts)} \\
 & & C \sqcup D & | & \text{(disjonction de concepts)} \\
 & & \neg C & | & \text{(négation de concept)} \\
 & & \forall R.C & | & \text{(quantificateur universel)} \\
 & & \exists R.C & | & \text{(quantificateur existentiel)}
\end{array}
$$

Les rôles dans Mirlog sont soit primitifs, soit des négations:

$$
\begin{array}{rcrcl}
R & \rightarrow & P & | & \text{(rôle primitif)} \\
 & & \neg P & | & \text{(négation de rôle)}
\end{array}
$$

Généralement, on utilise également la notation linéaire (*and* C D) pour $C \sqcap D$ ou (*all* R C) pour $\forall\ R.C$, ou encore (*some* R C) pour $\exists\ R.C$.

Dans la littérature, le fonctionnement des logiques terminologiques peut être décrit à travers deux types de sémantiques: une sémantique dénotationnelle ou une sémantique axiomatique. Par exemple, les logiques LOOM [Mac91], NF* [BS92] ou encore Back [VLS87] sont définies à travers une sémantique dénotationnelle, par contre Classic [AR89] adopte, elle, une sémantique axiomatique.

Dans Mirlog, les concepts et les rôles sont interprétés par la biais d'une sémantique dénotationnelle. Les concepts sont interprétés comme des sous-ensembles d'individus du domaine d'application, alors que les rôles sont interprétés comme des relations binaires entre des concepts. Plus précisément, une interprétation $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ consiste en un domaine d'interprétation (domaine d'application) $\Delta^{\mathcal{I}}$, et d'une fonction d'interprétation $.^{\mathcal{I}}$ associant à chaque concept un sous-ensemble de $\Delta^{\mathcal{I}}$ et à

chaque rôle un sous-ensemble de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. La fonction d'interprétation $.^{\mathcal{I}}$ est étendue aux concepts complexes de MIRLOG comme suit:

$$
\begin{array}{lcl}
\top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
(\neg C)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \Leftrightarrow C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} & = & \{p_1 \in \Delta^{\mathcal{I}} \mid \exists p_2, (p_1, p_2) \in R^{\mathcal{I}} \text{ et } p_2 \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} & = & \{p_1 \in \Delta^{\mathcal{I}} \mid \forall p_2, (p_1, p_2) \in R^{\mathcal{I}} \text{ implique } p_2 \in C^{\mathcal{I}}\}
\end{array}
$$

MIRLOG permet d'énoncer des "*assertions d'instances*", grâce auxquelles il est possible d'affirmer qu'une constante individuelle a est une instance d'un concept C, notée $C(a)$, ou qu'une paire d'individus est une instance d'un rôle donné, noté $R(a, b)$. Ces assertions sont dites primitives (dénotées par α), alors que les autres formules d'assertions (dénotées par γ et δ) peuvent être formées selon la syntaxe suivante:

$$
\begin{array}{rcrcl}
\gamma, \delta & \rightarrow & \alpha & \mid & \text{(assertion)} \\
 & & \gamma \wedge \delta & \mid & \text{(conjonction d'assertions)} \\
 & & \gamma \vee \delta & \mid & \text{(disjonction d'assertions)} \\
 & & !\gamma & \mid & \text{(négation d'une assertion)}
\end{array}
$$

Toutes les logiques terminologiques permettent de définir des relations sémantiques entre termes. Ces derniers sont organisés en une taxinomie basée sur la relation de subsomption. Informellement, un terme t_1 subsume un terme t_2 si, et seulement si, t_1 est plus général que t_2 dans la taxinomie ($t_2 \leq t_1$). Ainsi, le terme t_2 est subsumé par le terme t_1, dénoté par $t_2 \sqsubseteq t_1$ si, et seulement si, pour chaque interprétation $\mathcal{I}$, $t_2^{\mathcal{I}} \subseteq t_1^{\mathcal{I}}$. Par ailleurs, deux termes t_1 et t_2 sont dits équivalents et on écrit $t_1 \doteq t_2$ si, et seulement si $t_1 \sqsubseteq t_2$ et $t_2 \sqsubseteq t_1$.

La notion de subsomption est fortement liée à la notion logique de validité. Il est donc normal qu'elle soit utilisée pour la formalisation de la notion de pertinence dans un modèle logique de recherche d'informations: un document D, considéré comme une constante individuelle, est dit pertinent pour une requête Q, représentée comme un concept, si et seulement si, Q subsume D. Autrement dit, en considérant qu'un document D est une constante individuelle, ce dernier sera déclaré pertinent s'il appartient à la dénotation de la requête Q représentée, elle, par un concept. Par rapport au domaine de la recherche d'informations, l'utilisation de la subsomption est assez classique en soi. Cela revient à associer à la relation de pertinence logique (implication), une notion de généricité.

Signalons enfin que MIRLOG adopte une sémantique à quatre valeurs associée à une approche basée sur l'hypothèse des mondes fermés. Ces deux faits n'ont pas de conséquence directe sur l'expressivité de MIRLOG mais uniquement sur sa façon d'établir la relation de subsomption. Nous rappelons en effet que les logiques terminologiques pertinentes [17] [Bel77, Lev84] adoptent la sémantique à quatre valeurs afin d'éviter les paradoxes de l'implication logique. De plus, il a été prouvé

17. Relevant Terminological Logics.

que ces logiques ont généralement un meilleur comportement calculatoire que leurs analogues basées sur la sémantique standard à deux valeurs. La sémantique à quatre valeurs permet de limiter les cas de subsomption à ceux dans lesquels cette relation peut être calculée avec une complexité polynomiale [PS89].

b) Les Problèmes Soulevés

Il a été récemment mentionné dans la littérature [LR95, Hor97a, Hor97b] que les logiques terminologiques ne permettent pas de supporter la transitivité de certains rôles. Ainsi, exceptée la transitivité de la relation sémantique IS-A associée à la taxinomie des termes, les logiques terminologiques ne permettent pas l'inférence d'une relation de subsomption de la forme suivante:

Exemple 3 part $\circ$ part $\sqsubseteq$ part

La relation de composition *part* est en effet généralement transitive. Dans de nombreux cas de figure, comme par exemple la description de l'anatomie humaine ou la représentation de structures physiques complexes [Hor97b, Sat95, Sat96], cette relation est essentielle. Sa transitivité devra être prise en considération en vue d'assurer la complétude de la relation de subsomption.

De plus, les inférences résultants de l'agencement possible de certains rôles, et faisant ainsi intervenir des connaissances du domaine, ne sont pas pris en compte par la relation de subsomption. Essentiellement à cause de l'impossibilité de représenter ces connaissances avec les langages terminologiques:

Exemple 4 ULCÈRE $\sqcap$ $\exists$aPourLocation.ESTOMAC $\sqsubseteq$ $\exists$aPourLocation.PAROI-ESTOMAC

Dans cet exemple, la relation de subsomption asserte que tous les "ulcères d'estomac" ont pour location la "paroi de l'estomac", *sans* nécessiter la spécification de "$\exists$*aPourLocation*.PAROI-ESTOMAC" dans la définition d'un ulcère d'estomac. Un utilisateur posant une requête à-propos des *maladies pouvant affecter la paroi d'estomac* devra recevoir parmi les réponses du système, des documents à-propos des *ulcères affectant l'estomac*. Il s'agit de la représentation d'un fait logique ou encore d'une connaissance universelle: les ulcères affectent la paroi de l'estomac. Les seules assertions d'instances ne permettent pas d'introduire ce fait dans la base de connaissances du système. L'exemple 4 peut être schématisé par la figure suivante:

Dans le cas du modèle des documents multimédias, plusieurs relations sont définies comme étant transitives. Leur implantation avec MIRLOG pose donc un problème qui doit être résolu si l'on veut avoir des bonnes performances qualitatives du système. La non prise en compte de cette propriété peut en effet engendrer des cas de silence. Les données implicites obtenues par le biais de cette propriété peuvent concourir à faire valider l'implication entre le document contenant le rôle transitif et la requête.

Ainsi, le modèle multimédia intègre, entre autres, une *structure logique* représentant les composants du document comme des objets typés et introduisant une relation de composition pour les combiner. La structure logique joue un rôle important dans le modèle multimédia proposé puisqu'elle implante l'organisation explicite du contenu sémantique des document. Les propriétés qu'elle définie sur les relations doivent être représentées lors de son implantation dans un formalisme opérationnel. D'une manière générale, un objet typé dans le modèle est représenté par un concept de

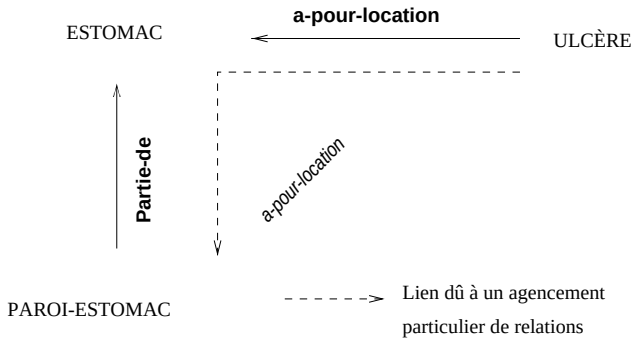


Figure 2.10. Des inférences valides ne peuvent pas être implantées avec les LTs

MIRLOG, alors qu'une relation correspond à un rôle de la dite logique. Pour rendre compte de cette structure logique, le modèle utilise trois relations particulières:

la relation $\preceq_{tst}$ définie un ordre partiel sur des types d'objets structurés. Ces derniers correspondent à des *niveaux d'abstraction* utilisés pour organiser la structure logique des documents, simplifiant ainsi leur compréhension et leur manipulation. Chaque niveau d'abstraction est représenté par un concept primitif dans MIRLOG. Les abstractions peuvent correspondre, par exemple, à des concepts comme BOOK, CHAPTER, SECTION, SUB-SECTION, etc. Un concept complexe, noté $TYPE_{ST}$, est formé par application de l'opérateur de disjonction de concepts $\sqcup$ sur les concepts primitifs associés aux abstractions. La relation $\preceq_{tst}$ peut être implantée dans MIRLOG par un rôle primitif *TypSTructure* effectuant des liens entre des individus du concept $TYPE_{ST}$. Dans le modèle, cette relation est définie comme étant réflexive, asymétrique et transitive.

la relation de composition $\preceq_{str}$ lie des objets structurels et définie ainsi leur hiérarchie de composition. Les objets structurels, dénotés par OS dans le modèle, correspondent aux entités d'informations représentant les différentes parties d'un document. Ils peuvent être représentés dans MIRLOG par un objet primitif noté OBJECT. Chaque objet structurel os_i d'un document est représenté par une unique constante individuelle de OBJECT. La relation $\preceq_{str}$ peut être ainsi implantée par un rôle primitif, nommé *aggregate*, définie sur des éléments de OBJECT:

$$\forall os_i, os_j \in OS,\ os_i \preceq_{str} os_j \equiv$$

$$aggregate(os_i, os_j) \wedge \text{OBJECT}(o_i) \wedge \text{OBJECT}(o_j)$$

Dans le modèle, cette relation est aussi définie comme étant réflexive, asymétrique et transitive.

la relation de séquentialité $\prec_{\mathbf{seq}}$: définie une séquence linéaire sur les objets structurels de type OBJECT. Elle correspond à un ordre standard linéaire permettant d'accéder aux composants des documents. Afin de représenter cette relation dans MIRLOG, nous introduisons un rôle

primitif nommé *IsA-Sequence-Of* dont les instances doivent vérifier que $\prec_{\mathbf{seq}}$ correspond à un ordre partiel strict sur les objets structurels:

$$\forall os_i, os_j \in OS,\ os_i \preceq_{seq} os_j \equiv$$

$$\mathit{IsA\text{-}Sequence\text{-}Of}(os_i, os_j) \wedge \textsc{Object}(o_i) \wedge \textsc{Object}(o_j)$$

À ce stade, nous remarquons qu'en plus de la transitivité, des propriétés comme la symétrie, l'antisymétrie ou encore la réflexivité doivent pouvoir être modélisées par les logiques terminologiques. Il s'agit d'assurer la complétude de la subsomption vis-à-vis des données implicites de la base de connaissances. Au fait, ces propriétés mathématiques peuvent être vues comme des contraintes sur les rôles primitifs concernés. On peut donc imaginer qu'elles peuvent être représentées par le biais de l'application de certains opérateurs de formation de rôles. Cependant, excepté l'opérateur de négation, MIRLOG n'autorise aucune opération sur les rôles primitifs. Par conséquent, ces contraintes (réflexivité, symétrie, etc.) ne peuvent pas être formellement définies par des rôles construits. Il n'est pas ainsi possible de construire un rôle complexe à partir d'un autre primitif en ajoutant des contraintes introduites par le biais d'opérateurs de formation de rôles.

Cette absence d'opérateurs de formation de rôles, permettant de représenter proprement des connaissances sur les relations, n'est pas le propre de MIRLOG, mais de toutes les logiques terminologiques citées dans la littérature. Elle s'explique par des considérations pratiques où la présence d'un grand nombre d'opérateurs de formation de termes a des répercussions négatives directes sur la complexité des traitements [LB84]. Ainsi, on compte une petite dizaine d'opérateurs de formation de rôles mentionnés dans la littérature[18]. Ce nombre est très petit comparé à celui des opérateurs de formation de concepts. Les opérateurs de rôles les plus utilisés étant l'opérateur de conjonction *and*, la négation *not*, la composition *compose* et l'inverse *inverse*. Ce dernier, présent dans MIRTL [MSST93] la première version de MIRLOG, a été abandonner pour diminuer la complexité de la relation de subsomption.

Ainsi, pour représenter les contraintes mathématiques sur les rôles, une solution consiste à introduire explicitement les faits dérivables comme des assertions d'instances dans la base. Par exemple, dans le cas de la relation *Aggregate* ci-dessus introduite, nous établissons les assertions suivantes:

- $\forall os_i$, si $\textsc{Object}(os_i)$ est une assertion d'instance dans la base de connaissances Σ de MIRLOG, alors $\mathit{Aggregate}(os_i, os_i)$ est aussi introduite comme une assertion d'instance dans Σ. Cela permet de représenter explicitement le fait que la relation *Aggregate* est réflexive.

- si $\mathit{Aggregate}(os_i, os_j)$ et $\mathit{Aggregate}(os_j, os_k)$ sont deux assertions dans la base de connaissances Σ de MIRLOG, alors $\mathit{Aggregate}(os_i, os_k)$ est également ajoutée comme une assertion d'instance dans Σ. La transitivité de la relation *Aggregate* est ainsi représentée.

- si $\mathit{Aggregate}(os_i, os_j)$ est une assertion, alors $!\mathit{Aggregate}(os_j, os_i)$ est aussi une assertion d'instance valide dans la base de connaissances. Il s'agit ici de représenter la propriété d'asymétrie.

18. Un bon recensement de ces opérateurs se trouve dans [HKNP94] à travers la présentation de CTL (Common Terminological Logic).

Le même raisonnement est aussi appliqué pour les deux autres relations de la structure logique. Ainsi, un document indexé par $Aggregate(os_i, os_j)$ et $Aggregate(os_j, os_k)$ ne sera retrouvé par une requête contenant $Aggregate(os_i, os_k)$, que si ce fait dérivable est explicitement ajouté à l'index du document. Un telle approche est, bien entendu, très coûteuse en termes de stockage, surtout quand les documents contiennent un grand nombre d'objets structurels. Une autre solution consiste à coupler les langages terminologiques avec un formalisme permettant l'inférence sur les propriétés mathématiques des rôles. Par exemple, des règles telles que celles apparaissant dans le contexte des systèmes experts ont été utilisées [LR95].

Par ailleurs, l'implantation des recommandations du modèle multimédia exige l'utilisation de l'opérateur de formation de rôles *inverse* (ou $R^{\sim}$ comme notation algébrique), absent de la logique MIRLOG pour les raisons déjà évoquées. Par exemple, $Desc_{str}(os_i)$ désigne dans le modèle l'ensemble de tous les objets structurels composants de os_i. Cet ensemble est défini comme suit:

$$\forall os_i \in OS, Desc_{str}(os_i) = \{os_j \in OS \mid os_i \neq os_j \ \& \ os_i \ \preceq_{str} \ os_j\}$$

$Desc_{str}(os_i)$ est représenté dans MIRLOG par un concept complexe nommé $DESCSTR_{osi}$ construit à partir du concept primitif OBJECT et du rôle *Aggregate*:

$$DESCSTR_{osi} \doteq (\text{OBJECT} \sqcap \neg(\{os_i\}) \sqcap (\forall Aggregate^{\sim}.(\text{OBJECT} \sqcap \{os_i\}))$$

où $Aggregate^{\sim}$ dénote l'ensemble contenant l'inverse des paires d'individus dénotées par le rôle *Aggregate*. En l'absence de l'opérateur d'inversion, l'introduction d'un nouveau rôle *Is-Aggregated-By* contenant les inverses des paires dénotées par *Aggregate* s'avère ainsi nécessaire. Malgré le fait qu'une telle approche permet de simuler l'opérateur d'inversion, nous ne pensons pas cependant qu'elle soit très adaptée. Elle suppose en effet l'introduction de nombreux rôles dont la création n'est ni théoriquement fondée, ni requise par le modèle.

Pour finir ce bref résumé de l'évaluation de MIRLOG signalons que, tout comme les autres logiques terminologiques, elle ne permet pas la représentation de relations n-aires. Cependant, le modèle multimédia adopte des relations tertiaires pour la représentation de certains attributs (comme auteur, date, etc.). Ces relations sont surtout destinées à faciliter les traitements dans le modèle. Par exemple, $HasValue(os, Attribute, value)$ est un prédicat permettant d'assigner à chaque attribut d'un objet structurel sa propre valeur. Afin de représenter ce prédicat dans le langage de MIRLOG, nous avons besoins de deux rôles primitifs, qui sont plus lourds à gérer que ce soit du point de vue du stockage ou en termes de complexité de traitements. Au fait, la représentation des différents attributs des documents par cette logique terminologique exige la création d'un grand nombre de rôles, rendant plutôt durs et complexes les traitements qui leurs sont associés. Avec des relations n-aires, le nombre des rôles utilisés diminuera et un meilleur comportement opérationnel sera attendu. James Schmolze [Sch89], avait ainsi proposé l'adoption d'une extension, grâce à laquelle les logiques terminologiques peuvent raisonner sur des termes n-aires.

2.4.5 Quels Raisonnements sur les Relations en RI?

L'analyse des modèles de recherche d'informations d'aujourd'hui que nous avons effectué dans la section précédente montre que la tendance actuelle semble être vers l'emploie de divers types de relations, et la spécification des propriétés syntaxiques et sémantiques que ces dernières présentent.

Cependant peu de systèmes (pour ne pas dire aucun) implantant ces modèles fournissent une fonction de correspondance permettant de tenir compte de ces propriétés et des connaissances implicites qu'elles génèrent.

Ainsi, certaines mauvaises performances peuvent être expliquées par les limites de la prise en compte des propriétés des relations dans la fonction de correspondance. Souvent, ces dernières ne sont pas complètes, c'est-à-dire qu'elles ne font pas dériver toutes les expressions valides dans le système. Certaines requêtes restent ainsi sans réponses malgré l'existence de documents pertinents.

Les systèmes que nous avons présenté dans la section précédente tentent tous de remédier à ce problème par l'ajout direct des informations implicites dérivées de ces propriétés de relations. Seulement cet ajout se fait de manière partielle et *ad hoc*, où seules certaines propriétés triviales comme la transitivité et la symétrie sont traitées. Nous proposons ainsi un cadre plus général spécifiant les connaissances sur les relations que doit représenter un système de recherche d'informations. Ces connaissances seront ensuite traitées par un mécanisme d'inférence particulier qui rectifie les limites des fonctions de correspondance classiques.

La plupart des fonctions de correspondance utilisées exploitent un thésaurus où les relations d'égalité (synonymie, définition, etc.), les relations hiérarchiques (relation de spécialisation / généralisation, appartenance), et les relations structurelles (composés/composants) [GB95] sont spécifiées. Or ces types de relations dans le thésaurus ne suffisent pas à représenter toutes les connaissances du domaine de l'application. En particulier, quand les index sont formés de structures sémantiques incluant un grand nombre d'objets et de relations entre ces objets, il est nécessaire de spécifier comment ces relations peuvent être combinées, ou quelles caractéristiques syntaxiques elles présentent (transitivité, symétrie, etc). Cela permettra de générer des connaissances implicites dans les index pouvant améliorer les performances qualitatives du système.

Au fait, nous ne renonçons pas en pratique aux techniques utilisant des thésaurus sur les objets, mais nous les incluons dans un raisonnement général sur les relations. Les connaissances sur les objets reviennent ainsi à des connaissances sur les arguments des relations. De plus, des applications que nous avons étudié dans la section précédente se dégagent les trois catégories de propriétés de relations suivantes:

Propriétés Mathématiques: Que ce soit pour la recherche des documents structurés ou des documents hypermédias, il est nécessaire de spécifier les caractéristiques mathématiques des relations d'agrégation. De plus, pour la recherche d'images, les relations spatiales présentent des caractéristiques mathématiques nécessaires à considérer dans les fonctions de correspondance.

Agencements de relations: Les noeuds des documents hypermédias peuvent être décrits par des structures complexes incluant plusieurs relations. Certaines relations peuvent être combinées ensemble pour générer d'autres relations implicites pouvant être référencées par les requêtes des utilisateurs. De même, la description des images par des structures sémantiques implique la prise en compte de ces agencements en vue d'une recherche précise et efficace.

Liens sémantiques entre les relations: Comme pour les objets, les relations peuvent être liées par des liens de spécialisation/généralisation, de synonymie, etc.. À l'image d'un thésaurus, la prise en compte de ces liens permettra de réduire le taux du silence lors du processus de recherche.

Les propriétés des relations doivent être incluses dans la base de connaissances du système de recherche. De plus la fonction de correspondance du système devra présenter des mécanismes de dérivation adéquats permettant leur prise en compte. Dans le chapitre suivant, nous présentons un modèle général permettant la représentation et le traitement de ces relations dans le processus de recherche.

2.5 Discussion et Orientation de la Thèse

Il en ressort de l'étude menée dans ce chapitre que non seulement la notion de relation est importante à représenter en vue d'une recherche précise et performante, mais que ses caractéristiques et propriétés doivent pouvoir être tenues en compte lors de la phase de correspondance. Si un consensus semble se dégager autour de la première condition, à en juger par le nombre important des systèmes de recherche conceptuels qu'on rencontre dans la littérature, l'utilisation des propriétés des relations et l'évaluation de leur impact sur la fonction de correspondance n'ont jamais été traitées que d'une façon *ad-hoc*. Il s'agit le plus souvent d'approches, traitant un sous-ensemble de propriétés, pour des tâches très spécifiques et sur un formalisme particulier. C'est le cas par exemple de la majeure partie des modèles de recherche que nous avons présenté dans la section précédente. Ces derniers ont été souvent confrontés à la nécessité d'inclure certaines propriétés de relations dans le but d'aboutir à des fonctions de correspondance ayant un minimum de complétude.

Nous pensons donc qu'une théorie générale permettant l'analyse des propriétés des relations est nécessaire pour la recherche d'informations. Nous croyons en effet que la notion de pertinence est fortement liée aux propriétés des relations et leur comportement sémantique. Le raisonnement sur les relations inclue nécessairement un raisonnement sur les concepts, et sur les individus, et permet ainsi au pire d'aboutir aux mêmes performances des systèmes conceptuels actuels. Les concepts et les individus n'étant autre que des arguments de relations. L'avantage de considérer toutes les propriétés des relations et d'analyser leur comportement est d'aboutir à des systèmes à raisonnements complets et efficaces. Modéliser les propriétés des relations et les inclure dans le processus de recherche permettra ainsi d'améliorer les performances qualitatives des systèmes dont nous disposons. Si nous nous sommes donnés autant de mal à détecter les propriétés des relations dont ont besoin les applications de recherche d'informations d'aujourd'hui, c'est que nous pensons qu'une bonne théorie devra tout d'abord provenir d'observations réelles. Nous partageons ainsi la définition que donna Hawking [Haw88] aux "*bonnes théories*":

> "A theory is good if it satisfies two requirements: it must accurately describe a large class of observations on the basis of a model that contains only a few arbitrary elements, and it must make definite predictions about the results of future observations."

Une conséquence directe de cette définition est qu'une bonne théorie de recherche d'informations T devra s'acquitter des décisions de pertinence de façon cognitivement acceptable. Les travaux de Farradane [Far80a] ont montré l'importance des relations dans le raisonnement psychologique humain. Les caractéristiques des relations jouent ainsi un rôle important dans l'acceptabilité des décisions de pertinence du système. Nous argumentons ici le fait que si nous voulons construire un *bon* système de recherche d'information ou un système plus précis et plus performant, il faut que les caractéristiques des relations soient modélisées et incluses dans le processus de décision de pertinence. En termes de théorie, une propriété P considérée comme importante (ou *bonne*)

en recherche d'informations devra être couverte par la théorie T associée. De plus, les modèles de recherche d'informations existants doivent pouvoir être formalisés avec cette théorie. Nous pensons qu'à chaque propriété de relation, correspond une décision de pertinence. La théorie doit donc inclure les propriétés des relations. De plus nous pensons que les systèmes à base de raisonnements sur les concepts (les approches classiques utilisant les mots-clés comprises) peuvent être formalisés à l'intérieur d'une théorie s'appuyant sur la modélisation des relations et leur comportement.

Si une bonne théorie de recherche d'informations est définie, alors elle servira de base à la détermination des caractéristiques des modèles existants. Ces caractéristiques connues, des extensions au modèle seront possibles à proposer. Elles reposent sur les propriétés manquantes dans le modèle et requises par la théorie. Nous pensons que la logique constitue un cadre adéquat pour la définition d'une telle théorie. Dans ce cas, les propriétés peuvent être modélisées par des axiomes ou des règles, correspondant chacun à une décision de pertinence.

En proposant un modèle logique pour la recherche d'informations [vR86], Van Rijsbergen suggère qu'une logique conditionnelle non-classique doit être utilisée pour la définition de la notion de pertinence. Cette dernière est alors vue comme une inférence logique. Dans ce modèle, la logique en question n'a pas été spécifiée. De nombreux travaux ont été alors menés pour trouver cette *seule et unique logique* de recherche d'informations. Par exemple, la logique modale [Nie90, Nie89, CCO95], les graphes conceptuels [Che92], les logiques terminologiques [MSST93, Seb94, SS96], Datalog [Fuh95], l'imaging [CvR95a, CvR95b], la logique abductive [MT94] et la théorie des situations [Lal96, LvR92], pour n'en nommer que quelques-unes.

Une autre utilisation de la logique en recherche d'informations est proposée par Chiaramella & Chevallet dans [CC92], et par Huibers dans [Hui96]. La logique est vue ici comme un outil pour l'analyse des caractéristiques des modèles de recherche d'informations. C'est l'approche que nous adopterons dans cette thèse. Nous suivons toutefois une démarche proche de celle menée par Huibers. Ce dernier propose un cadre général pour l'analyse et la comparaison des modèles de recherche d'informations. Cependant, Huibers ne donne pas de définition *a priori* à la notion de pertinence. La notion de pertinence même si elle reste liée à une inférence logique, n'a pas de sémantique spécifiée[19]. Dans notre cas, la notion de pertinence est liée aux propriétés des relations et leur comportement. De plus, à l'opposé de Huibers, nous présentons une logique pour la recherche d'informations, à proprement parler, c'est-à-dire une logique complète vis-à-vis d'une sémantique précise permettant d'expliquer le processus de recherche. Nous considérons que cette logique peut être vue comme l'instanciation "idéale" du modèle logique de Van Rijsbergen. Plus précisément, nous considérons que les décisions de pertinence de cette logique doivent être supportées par n'importe quelle autre logique se déclarant être une bonne instanciation du modèle logique, et dédiée aux nouvelles applications de la RI. En d'autres termes, tout en proposant une logique idéale pour la recherche d'informations d'aujourd'hui, nous restons à un niveau pragmatique et opérationnel. Nous nous rapprochons ainsi de l'idée formulée par Cooper [Coo94]:

> "[...] Obviously, theory-guidedness is a good thing if the theory leads to promising retrieval rules to try out. Good theories have inferential power, and inferential power can help minimize empirical floundering. However, having to stay within the constraints of a strict theoretical formalism can also impose costs and penalities. The true extent

19. Huibers se contente en effet de recenser toutes les décisions de pertinence adoptées par les systèmes de recherche d'informations opérationnels, y compris les systèmes conceptuels. Ces derniers n'étant pas munis de fonction de correspondance incluant les propriétés des relations, il est clair que notre théorie se distingue par sa généralité et sa spécifité à ce niveau.

of these costs is not always fully recognized."

Notre approche dans la suite de cette thèse tente en effet de concilier une approche fondamentalement théorique, où la notion d'inférence en recherche d'informations est liée aux relations et leurs propriétés, avec une autre pragmatique dont le but est d'étendre un formalisme opérationnel par les capacités de raisonnements d'une bonne logique en recherche d'informations, puis de l'implanter.

Par souci de neutralité envers les logiques et les formalismes cités dans ce chapitre, nous considérons la théorie des situations [Dev91] comme formalisme de représentation de notre logique. Cette dernière sera appelée *Modèle d'Indexation relationnel* afin de mettre l'emphase sur l'importance des relations et leurs traitements dans notre approche. Une fois cette logique définie, nous pouvons étudier dans quelle mesure, les formalismes opérationnels existants peuvent-ils s'accommoder à moindre coût des caractéristiques de la logique développée.

Chapitre 3

Le Modèle d'Indexation Relationnel

Comme cela a été évoqué dans le chapitre précédent, une recherche précise et efficace nécessite le besoin de représenter aussi bien les relations que leur propriétés. Le but de ce chapitre est de présenter un modèle relationnel permettant d'inclure les caractéristiques syntaxiques et sémantiques des relations dans les décisions des pertinence. Nous proposons de baser ce modèle sur un système logique utilisant la théorie des situations comme langage de représentation.

L'approche présentée dans ce chapitre est conforme à celle spécifiée dans le modèle logique de recherche d'informations [vR86]. Elle diffère cependant de celle-ci, ou des autres démarches logiques adoptées dans [CC92, Lal96], par le fait que nous ne partons pas d'une description d'un document pour aboutir à la description de la requête via une certaine déduction logique. Nous partons d'un ensemble d'axiomes assertant qu'une information est pertinente à une autre, et établissons à travers l'utilisation des propriétés des relations un ensemble de pas de déductions permettant de conclure qu'un document est pertinent à la requête. Comme nous l'avons conclu dans le chapitre précédent, notre approche est similaire à celle adoptée dans [Hui96], à ceci près que nous argumentons le fait que les décisions de pertinence (les pas de déductions) correspondent nécessairement à des propriétés de relations. Par *propriétés de relations*, nous incluons leurs propriétés mathématiques, leurs liens sémantiques avec d'autres relations (relations de spécialisation/généralisation, préclusion, etc.), leurs compositions possibles avec d'autres relations et les raisonnements sur leurs arguments (raisonnements sur les concepts et les individus).

D'un point de vue logique, notre démarche s'appuie sur les travaux de Kraus, Lehman & Magidor [KLM90]. Ces derniers présentent un cadre général dans lequel les systèmes logiques non-monotones peuvent être étudiés et leurs caractéristiques spécifiées. L'étude repose sur la définition des propriétés que devrait supporter un système logique non-monotone. La théorie qu'ils développent est considérée actuellement comme un standard pour la caractérisation des relations d'inférence non-monotones. Dans notre cas, on s'intéresse à un standard caractérisant la relation de pertinence logique, où les déductions s'appuient sur les propriétés des relations des index.

Nous présentons d'abord les grandes lignes du modèle relationnel que nous proposons, et établissons un certain nombre d'hypothèses utiles pour toute approche logique en recherche d'informations. Nous expliquons ensuite les raisons pour lesquelles la théorie des situations a été choisie comme langage de représentation. En conformité avec cette dernière, nous donnons une terminologie propre à notre approche. Nous formulons ensuite des postulats décrivant les propriétés des relations telles qu'elles sont utilisées en recherche d'informations. Notre démarche s'appuiera sur les exemples des

applications présentées dans le chapitre précédent. Nous introduisons alors un modèle logique permettant de refléter chacun de ces postulats par une décision de pertinence. Nous concluons enfin par des extensions possibles au modèle proposé permettant d'abord de l'enrichir par des raisonnements incertains sur les propriétés des relations puis par un raisonnement multifacette. Dans ce dernier cas, les documents peuvent être indexés selon différents points de vue.

3.1 Présentation de l'Idée du Modèle

3.1.1 Hypothèses et Généralités

L'un des fondements d'un modèle logique pour la recherche d'informations est de considérer que l'ensemble des documents d et des requêtes q sont respectivement décrits par des formules logiques $\chi(d)$ et $\chi(q)$ [1]. Dans ce cas, la correspondance est basée sur une notion d'inférence logique ($\chi(d) \rightarrow \chi(q)$). L'idée de départ revient à Cooper [Coo71], qui fut le premier à la formaliser, en introduisant la notion de pertinence logique. Cooper distingue deux aspects de la pertinence:

L'utilité, notion purement pragmatique, associée au fait de savoir si les documents retrouvés par le système sont utiles pour l'utilisateur.

La pertinence logique, notion décrivant si un document retrouvé admet un thème en rapport avec la requête formulée par l'utilisateur.

La différence entre ces deux aspects est facile à saisir. En effet, un document logiquement pertinent pour un besoin en informations donné, peut être jugé inutile par l'utilisateur, vraisemblablement parce qu'il n'a pas confiance dans l'exactitude de l'information que contient ce document (eg. document trop ancien, auteur inconnu, etc.). Cette thèse adressera uniquement le concept de la pertinence logique telle qu'elle a été définie par Cooper [2]:

> A stored sentence is logically relevant to a representation of an information need if and only if it is a member of some minimal premise set of stored sentences for some component statement of that need.

Cette définition fait ainsi allusion à la conséquence logique. Si la requête est une conséquence logique du document, alors ce document est jugé pertinent pour la requête. Une telle définition sous-tend que la requête est une représentation parfaite du besoin en informations et que nous pouvons exactement représenter le contenu des documents du corpus. Cependant, la réalité montre combien une telle hypothèse est erronée. Une des questions adressée par van Rijsbergen dans [vR86] a trait à la façon dont le système doit se comporter quand la formule "$\chi(d) \rightarrow \chi(q)$" n'est pas directement établie. Les documents étant rarement décrits de façon exhaustive, il est fort probable que la requête n'a pu être inférée à cause de cette représentation partielle. Ainsi l'ajout d'autres données sous la forme de formules logiques peut éventuellement rendre l'inférence possible (ou plausible).

1. La requête q peut être formulée dans un langage de requête quelconque. Le système devra alors la transformer en une formule logique $\chi(q)$ conforme à la représentation interne du système. Les documents étant bien entendu indexés conformément à cette même représentation interne.

2. La problématique liée à la notion d'utilité est examinée par Denos dans une autre thèse de l'équipe [Den97].

L'évaluation de la pertinence est fonction du plus petit ajout qui rendrait l'inférence possible. Cette idée fut exprimée dans le principe d'incertitude de Rijsbergen [vR86]:

> "Given any two *sentences* x and y, a measure of the uncertainty of y→x relative to a given *data set* is determined by the *minimal extend* to which we have to add information to the data set in order to establish the truth of y→x"

Ce principe ne spécifie ni la logique à choisir pour la modélisation de la recherche d'informations, ni le type d'inférence à utiliser pour prouver que y→x:

(1) $y \rightarrow x$ est vraie dans une interprétation particulière de la logique choisie; i.e, $y \supset x$.

(2) x est une conséquence logique de y dans la logique choisie; i.e, $y \models x$.

(3) $y \rightarrow x$ est valide dans la logique choisie; i.e, $\models y \supset x$.

(4) x est dérivable (ou peut être prouvée) de y dans la logique choisie; i.e, $y \vdash x$.

(5) $y \rightarrow x$ est un théorème de la logique choisie; i.e, $\vdash y \supset x$.

Ces notions ont des significations différentes en logique, et il paraît ainsi adéquat de préciser, chaque fois que l'on construit un modèle logique de recherche d'informations, le sens adopté pour y→x. Comme nous l'avons mentionné dans [OC96b], il n'est pas facile de prouver qu'un type d'inférence particulier est plus adéquat qu'un autre dans une approche logique de recherche. Nous avons cependant énuméré dans [Oun96], un certain nombre de paramètres qui laissent à penser que les interprétations (4) et (5) ci-dessus introduites sont plus intéressantes du point de vue de la recherche d'informations.

En effet, nous croyons qu'en recherche d'informations nous sommes moins concernés par les valeurs de vérités que peuvent prendre des entités tels que les documents ou les requêtes. Par contre nous sommes plus intéressés par l'information véhiculée par ces entités et par le comportement inférentiel du système de recherche d'informations. Ceci est en parfaite harmonie avec l'aspect syntaxique d'une logique[3] (dérivabilité et théorèmes) et donc avec la théorie des preuves. Cette dernière fait abstraction du sens et des valeurs de vérité des symboles manipulés, s'intéressant exclusivement à leur forme et à la manière de les démontrer. Plus précisément, nous pensons à la dérivabilité comme interprétation adéquate pour l'implication en recherche d'informations, ceci pour pouvoir utiliser un ensemble supplémentaire de formules représentant ainsi les connaissances du système. Nous partageons donc le point de vue formulée par Mounia Lalmas et Keith van Rijsbergen dans [LvR93], concernant l'opportunité de considérer des modèles logiques à valeurs de vérité en recherche d'informations:

> "A non truth-based logic is more appropriate because a model of an IR system is more concerned with information content rather than truth-values"

3. Une logique peut généralement être considérée selon deux points de vue. Le premier a trait a l'aspect syntaxique de la logique, il examine des propriétés (et des relations), des expressions et des formules en faisant abstraction de leur signification. Le second point de vue est sémantique, il traite des propriétés et relations des expressions qui reposent sur des rapports entre signifiant et signifié. De plus, comme nous le verrons dans le chapitre suivant, une logique peut également être associée à une algèbre.

Nous formulons ainsi, l'hypothèse suivante:

Hypothèse 3 (La Pertinence est une Dérivation) *Nous faisons l'hypothèse que pour qu'un document d soit pertinent à une requête $\chi(q)$, il existe une chaîne de dérivation ayant pour début la description du document $\chi(d)$ et pour fin la requête $\chi(q)$.*

Une sémantique adéquate et non ambiguë reste cependant nécessaire pour permettre de refléter le fonctionnement du mécanisme de dérivation et d'expliquer éventuellement son comportement. Ainsi, un des résultats connus dans le domaine de la logique est le suivant: si l'ensemble des axiomes et des règles d'inférence de la logique est strictement complet et correct vis-à-vis de sa sémantique, alors la dérivabilité ($\vdash$) coïncide avec la conséquence logique ($\models$) [GG90, Eps90, Eps94]. Nous rappelons qu'intuitivement, on dira que la *complétude* d'une logique garantit que les règles de raisonnements formalisées par cette logique suffisent pour démontrer toutes les propriétés vraies exprimables dans son langage. Autrement dit, le système est dit *complet* s'il vérifie la condition que toutes les tautologies (les formules toujours vraies) sont engendrées par le biais de ses règles et axiomes: Si A est une tautologie alors A est un théorème (si $\models A$ alors $\vdash A$). L'adéquation exprime le fait que le système n'engendre rien d'autre que les tautologies. Si cette condition est satisfaite, le système logique (avec ses axiomes et règles) est dit *correct* ou *adéquat*. On dira alors que les règles de déductions syntaxiques préservent les principes de l'interprétation: si $\vdash A$ alors $\models A$. Plusieurs sens du mot complétude sont à distinguer [GG90]. Ainsi, nous dirons qu'un système est strictement complet si, et seulement si, à toute relation de conséquence entre une proposition A et un ensemble de propositions , correspond une relation de dérivabilité, autrement dit: si , $\models A$ alors , $\vdash A$. Si l'ensemble , est vide, nous dirons que le système est *faiblement complet* ou *complet*[4].

Dans [GG90], les théorèmes suivants sont démontrés: si le processus inférentiel (ensemble des axiomes et des règles d'inférence) de la logique est correct et (faiblement) complet par respect à sa sémantique, la notion de théorème coïncide avec celle de la validité. En plus, si la logique est adéquate et fortement complète, nous verrons aussi la dérivabilité qui coïncide avec la conséquence logique. Il est toujours intéressant de démontrer qu'une logique est complète mais il est des cas où c'est crucial. Ce sont les cas où la satisfaisabilité n'est pas décidable, c'est-à-dire où on ne dispose pas d'une procédure effective, analogue à la méthode des tables de vérités pour le calcul des propositions, permettant de vérifier après un nombre fini d'opérations si une formule admet un modèle ou pas. Le calcul des prédicats, par exemple, n'admet pas une telle procédure de décision pour la satisfaisabilité [GG90]. Par conséquent, sa complétude garantit que l'on puisse vérifier qu'une formule est valide en démontrant qu'elle est un théorème.

En recherche d'informations, si la complétude de la logique est prouvée, cela revient au même de prouver que la requête $\chi(q)$ est une conséquence logique de la description du document $\chi(d)$ ou de démontrer que $\chi(d) \rightarrow \chi(q)$ est une tautologie. Notre hypothèse sera ainsi amplement justifiée, puisque tout ce qu'on fait avec la dérivabilité a son équivalent (peut se justifier) avec la conséquence logique. Prouver la complétude nous permettra d'affirmer que tout ce qu'on dérive à partir de nos règles et axiomes est valide (a un sens) par rapport à une sémantique. Or pour traiter l'aspect sémantique, il est nécessaire de spécifier la notion d'*interprétation* dans le cas de la recherche d'informations. En effet, l'interprétation d'une expression dans un index de document ne peut être celle classique, à savoir "la proposition dénote un fait qui est vrai dans cette interprétation" car un index n'a rien à voir avec les valeurs de vérité du contenu des documents.

4. Dans la littérature, la complétude faible est généralement utilisée comme la définition standard de la complétude [GG90].

La seule façon raisonnable pour interpréter une expression d'indexation avec une valeur de vérité est de considérer cette dernière comme associée au fait que "*l'expression est une bonne représentation du contenu du document*". Par le fait qu'une expression est une *bonne représentation* du document, nous voulons signifier que l'information véhiculée par cette expression est présente dans le document. Par exemple, dans le cas d'un document traitant des *aventures de Tom & Jerry dans un parc public*, l'expression "*Tom poursuit Jerry dans un parc public*" est une bonne représentation de ce dernier, ou en termes de recherche d'informations, ce fait est présent dans le document. Par contre, l'information représentée par l'expression "*Romeo aime Juliette*" n'est pas véhiculée par ce document, elle ne peut être donc une bonne représentation du document en question. S'il existe un procédé d'indexation automatique des documents, nous considérons que tout ce qu'extrait ce procédé d'un document, comme expressions d'indexation, est une bonne représentation du document[5].

Définition 3.1.1 (Interprétation des Expressions d'Indexation) *Pour un document d, nous considérons que son index $\chi(d)$ est exprimé par un ensemble d'expressions $\{e_i\}$ construite à partir d'un langage $\mathcal{L}$: si $e_i \in \chi(d)$ alors e_i est assignée à vrai et est considérée comme une bonne représentation du document.*

Nous préférons adopter un raisonnement basé sur un modèle de *monde ouvert*, c'est-à-dire un monde où toutes les expressions vraies ne sont pas forcément représentées. Essentiellement, cela a pour conséquence qu'une base de connaissances est seulement certaine pour les faits qu'elle contient ou qu'elle peut dériver. Tout ce qui échappe à son contenu est simplement considéré comme *inconnu*. Une telle démarche est très importante en recherche d'informations, car souvent il n'est pas possible de déduire directement si un document contient une information donnée ou pas. Ceci peut être éventuellement démontré quand des nouvelles informations deviennent disponibles. Quand une information est absente dans un document, cela ne veut en aucun cas dire qu'elle est fausse, on considère seulement qu'à ce stade nous manquons de connaissances. Il s'agit d'une approche exactement opposée à celle qui est généralement considérée dans le modèle booléen classique, basé sur ce qu'on appelle la supposition des mondes fermés (close world assumption).

Le principe d'incertitude de Rijsbergen pourrait alors être décrit par la sémantique des mondes possibles [Nie90]. Quand une requête $\chi(q)$ ne peut être directement dérivée d'un index $\chi(d)$, alors soit ce dernier devra être élargi par des nouvelles expressions, soit certaines expressions de l'index doivent être modifiées. Par rapport à la définition 3.1.1 précédente, cette évolution est sémantiquement interprétée comme la précision de nouvelles valeurs de vérité pour certaines expressions du langage $\mathcal{L}$. Dans le cas d'un ajout d'expressions, certaines valeurs de vérité passent de la valeur *inconnu* vers une valeur précise *faux* ou *vrai*. Par contre, la modification d'une expression de l'index du document revient à faire changer la valeur de vérité de cette dernière.

L'analogie avec la sémantique de Kripke [Kri63] apparaît en ces termes: imaginons que les documents correspondent à des mondes. Sur un document, nous appliquons un nombre de changements, ces changements correspondent à une évolution de ce monde de son état initial, vers un autre état, appelé alors *un monde possible* du premier. Ainsi, quand la correspondance entre un document et une requête n'est pas établie de façon directe, l'approche consiste à transformer l'état initial de l'index vers des interprétations qui lui sont reliées et pouvant dériver la requête. Chaque transformation a comme résultat une nouvelle *interprétation logique* des termes d'indexation, la-

5. Si l'indexation est manuelle, il revient à l'indexeur de décider ce qu'est une bonne représentation du document. La décision sera différente d'un individu à un autre, selon leur perception du contenu du document.

quelle correspond à un monde possible. Le processus prend fin quand l'information recherchée est trouvée dans un monde possible ou quand il n'est plus possible d'appliquer des transformations. L'aspect intuitif de la sémantique des mondes possibles et sa forte liaison avec la problématique de la recherche d'informations, justifient le fait que de nombreuses approches aient recours à cette sémantique pour modéliser une approche logique en recherche d'informations. L'idée initiée par Nie [Nie90], fut aussi adoptée dans [AK92, SvR93, vRL96, CvR95a, Seb94]. Reste à savoir quels types de transformations nous appliquons à un monde dans notre modèle.

Comme nous nous intéressons aux relations et à leurs propriétés, nous considérons que leur prise en compte nous permettra d'évoluer d'un monde à un autre. Les relations peuvent avoir plusieurs types différents. Par exemple, elles peuvent exhiber un relation de position (dans, sur. ...), une action (imprime, supprime, ...), un relation d'association (avec, pour, ...), une relation logique (et, ou, ...), etc. Elles agissent comme des connecteurs d'expressions ayant un certain nombre de propriétés et agissant selon des comportements spécifiques. Généralement, des relations de même type présentent des propriétés communes et des comportements semblables. Nous nous proposons d'établir ces propriétés et ces comportements à travers un ensemble d'axiomes et de règles. Le système de dérivation ainsi construit devra être complet vis-à-vis de la sémantique adoptée. Les relations, correspondant ainsi à des connecteurs, doivent sémantiquement opérer comme des fonctions des valeurs de vérité des constituants qu'elles lient. Cela veut dire que la valeur de vérité (toujours par rapport à la définition 3.1.1) d'une expression complexe est déterminée par sa forme et les propriétés sémantiques de ses parties[6]. Par exemple, si l'expression *Tom est-un Chat et Jerry est-une Souris* est vraie dans une interprétation (un index ou un index élargi), alors chacune de ses parties doivent être vraies. Similairement, si les deux expressions *Tom est-un Chat* et *Jerry est-une Souris* sont assignées vraies dans une interprétation, alors l'expression *Tom est-un Chat et Jerry est-une Souris* doit être également vraie dans cette interprétation.

Dans ce qui suit, nous présentons de façon intuitive l'idée générale du modèle qui sera présenté tout au long de ce chapitre. Il s'agit surtout de montrer comment des informations sur les relations entre termes (ou mots en général) peuvent faire améliorer le processus de correspondance.

a) Les Connecteurs dans un Langage

La structure mathématique des langages fut largement étudiée dans la littérature durant les 25 dernières années. Ces diverses études montrent que tous les discours, toutes les phrases d'un langage sont des séquences de parties (les mots) qui apparaissent dans des combinaisons variées; cette observation est commune à tous les langages. Une deuxième constatation stipule le fait que toutes les combinaisons de mots n'appartiennent pas forcément à l'ensemble des phrases. Généralement, la première tâche auquelle on s'attaque lors de la construction d'un langage quelconque consiste à répartir les expressions du langages en classes et à préciser les lois *a priori* qui régissent les agencements d'une expression d'une de ces classes avec les expressions de telle ou telle autre classe. Ces lois constituent la grammaire du langage en question et servent à différentier les séquences d'expressions qui appartiennent au langage, par opposition à celles qui n'y appartiennent pas.

Ces agencements s'effectuent par le biais d'un ensemble d'opérateurs représentant des entités du langage qui permettent de construire des expressions nouvelles décrivant des objets nouveaux ou des situations nouvelles. Il n'y aurait pas de langage sans ces opérateurs formateurs d'expressions [Des90]. Engendrer des expressions bien formées revient à faire agir les opérateurs sur des

6. Propriété connue en logique sous le nom de *l'hypothèse de Fregean* [Eps90].

opérandes d'un type approprié constituant des expressions d'une classe particulière. Il est donc possible de considérer la syntaxe comme un système d'opérations agissant sur des mots, et dont les résultats sont précisément les expressions, c'est-à-dire les séquence de mots qui font partie du langage. Cependant, cette syntaxe impose un certain nombre de restrictions, surtout dans le choix des opérandes qui doivent avoir des types spéciaux, dans les inter-relations d'opérateurs particuliers, et dans la création de sous-domaines spéciaux d'opérandes auxquels seules certaines expressions spéciales s'appliquent. Par exemple, en algèbre, l'opération de division n'est appliquable qu'à des objets non nuls. Dans des langages de programmation évolué comme ADA ou PASCAL, nous manipulons différents types d'expressions, l'évolution d'un opérateur arithmétique sera différente selon que ses opérandes sont des nombres entiers, des nombres réels, des tableaux, etc. Certains opérateurs ne s'appliquent qu'à des expressions logiques, ce sont les opérateurs booléens. La notion de type est introduite pour considérer différents types d'expressions, c'est-à-dire des classes hétérogène d'expressions.

Les opérateurs de formation d'expressions établissent, sous certaines contraintes, des liens entre expressions pour former des expressions plus complexes. Ils se comportent donc comme des *connecteurs* ou des *relations* conduisant à des combinaisons de mots plus complexes et plus structurées. Les connecteurs dont dispose la langue sont extrêmement nombreux. Ceux qui ont donné lieu au plus grand nombre de travaux sont les connecteurs logiques, comme la conjonction (et) ou la disjonction (ou), très présents dans les formalismes de représentation de connaissances et évidemment dans les diverses logiques [Car83]. Ces connecteurs présentent une sémantique et un comportement qui leurs sont propres. Pour comprendre la signification de l'information véhiculée par les expressions qui en découlent, il faudra tenir compte de ces connecteurs.

Un langage d'indexation, surtout lorsqu'il est complexe, n'échappe pas à cette règle. Les expressions d'index sont généralement construites de façon récursive par l'application d'un connecteur sur des expressions simples (des mots-clés) ou complexes. Elles sont donc de la forme "$e_i\, c\, e_j$" où c désigne un connecteur donné, et e_i, e_j sont deux expressions quelconques. Les expressions contiennent ainsi un ensemble de connecteurs présentant une sémantique particulière. Cette dernière pourra influencer le processus de correspondance de part les informations implicites qu'elle véhicule. Ainsi au départ, une requête ne pouvant être directement dérivée de la description du document, pourra le devenir grâce à l'ajout de ces informations implicites.

Ce que nous appelons *connaissances d'un système de recherche d'information* correspond donc à un ensemble de caractéristiques sémantiques définies sur les relations. L'utilisation des ces connaissances sur un monde (représentant l'index du document) conduit le monde vers son monde possible. Dans notre cas, l'ensemble des connaissances sur la sémantique des connecteurs, leurs propriétés et leurs comportements sera représenté par des axiomes et des règles de dérivation.

b) Un système de Dérivation sur les Connecteurs

Le système de dérivation contient les règles et les axiomes ayant trait aux caractéristiques des connecteurs, comme par exemple leur propriétés syntaxiques et sémantiques, leur comportement quand ils sont combinés avec d'autres connecteurs, etc.

Exemple 5 *Nous pouvons considérer que le système de dérivation puisse regrouper des règles de ce type:*

Conjonction (CJ)		**Transitivité (TV)**	**Distributivité (DV)**	**Position (PS)**
$\dfrac{i \text{ et } j}{i}$	$\dfrac{i \text{ et } j}{j}$	$\dfrac{i \text{ connecteur}_1\ j \quad j \text{ connecteur}_2\ k}{i \text{ connecteur}_1\ j \text{ connecteur}_2\ k}$	$\dfrac{i \text{ de } j \text{ et } k}{i \text{ de } j \text{ et } i \text{ de } k}$	$\dfrac{i \text{ poursuit } j \text{ dans } k}{i \text{ dans } k \text{ et } j \text{ dans } k}$

Supposons que le corpus contient un document indexé par $\chi(d) = \{$*Aventures de Tom et Jerry, Tom poursuit Jerry dans un Parc Public, Échappé de Jerry* $\}$. *Supposons aussi qu'un utilisateur est entrain de chercher tous les documents à-propos des* Aventures de Tom dans un Parc Public. *Si l'on se réfère uniquement à l'index, la requête ne s'y trouve pas incluse. Par contre en utilisant les règles d'inférence ci-dessus, nous pouvons prouver que* $\chi(d) \vdash \chi(q)$*:*

$$\dfrac{\dfrac{\dfrac{\text{Tom poursuit Jerry dans un Parc Publique}}{\text{Tom dans un Parc Publique et Jerry dans un Parc Publique}}\,PS}{\text{Tom dans un Parc Publique}}\,CJ \qquad \dfrac{\dfrac{\text{Aventures de Tom et Jerry}}{\text{Aventures de Tom et Aventures de Jerry}}\,DV}{\text{Aventures de Tom}}\,CJ}{\text{Aventures de Tom dans un Parc Publique}}\,TV$$

Les règles de dérivation de cet exemple ne sont pas valides pour tous les connecteurs[7]. Il faudra donc spécifier à travers les connaissances du système ceux pour lesquels elles s'appliquent. Dans la suite nous nous proposons d'établir un système de dérivation logique qui puisse modéliser ce fait. Il sera basé sur la théorie des situations. Pour mieux souligner l'importance de l'aspect sémantique dans les dérivations, nous préférons parler de *relation* plutôt que de connecteur.

3.1.2 La Théorie des Situations

L'objectif de la théorie des situations [Bar89, BE90, BGPT91, BP83] est le développement d'un cadre général mathématique pour l'étude de l'information, laquelle est alors aussi fondamentale que l'énergie en physique. La construction de cette théorie doit se faire d'une manière abstraite, étant donnée que la définition de l'information elle-même est encore difficile à cerner. Cela peut être comparé aux travaux des mathématiciens sur le concept du nombre: bien qu'il ne soit pas possible de donner une définition précise à ce terme, les mathématiciens continuent à travailler sur les nombres, puisqu'ils savent comment les utiliser. Nous donnons dans la section suivante les raisons qui justifient le choix de cette théorie comme base de notre modèle d'indexation relationnel.

a) Pourquoi la Théorie des Situations ?

Le choix de la théorie des situations comme outil de représentation et de recherche d'informations est largement discuté dans les travaux de Lalmas, de Rijsbergen et de Huibers [Lal96, LvR92, LvR93, vRL96, HLvR96]. Notre démarche dans cette thèse est similaire à celle proposée par Huibers [Hui96], où la théorie des situations est vue comme un cadre logique neutre pour étudier et analyser les propriétés théoriques des mécanismes de recherche. Nous avons été ainsi confronté au dilemme de choisir entre la construction d'un nouveau langage de représentation permettant de supporter une approche d'indexation relationnelle et de raisonner sur les structures qu'elle génère,

7. Cette constatation rejoint ce que nous avons déjà mentionné dans la section 2.2.3, page 26, où les schémas de dérivation dépendent de la sémantique et des propriétés du connecteur utilisé.

ou utiliser un langage existant et l'étendre de façon à ce qu'il tienne compte des relations et leurs propriétés dans les traitements qu'il établie. Nous avons préféré la deuxième option car notre but n'est pas de construire un n^{eme} formalisme opérationnel pour la recherche d'informations, mais plutôt de spécifier les caractéristiques que doit présenter un tel formalisme au niveau de l'expressivité et des traitements, pour une recherche précise et efficace.

Outre sa neutralité envers les formalismes opérationnels déjà existants, du fait qu'elle constitue un méta-langage pour la représentation de l'information, notre choix de la théorie des situations s'est renforcé des deux considérations suivantes:

- L'importance de la notion de relation dans cette théorie: En effet, l'unité de base d'information dans cette théorie est l'*infon*. Les infons mettent en relation des objets en leur assignant une polarité, qui indique si la relation en question est vérifiée pour ces objets ou pas. L'infon pourra être ainsi considéré comme une structure sémantique intéressante pour une approche relationnelle.

- Il n'existe pas de distinction entre syntaxe et sémantique dans la théorie des situations. La syntaxe utilisée véhicule l'information contenue. Ainsi, les opérations syntaxiques sur les infons constituent en même temps des raisonnements sémantiques. Ceci est à l'inverse de la plupart des formalismes, où l'information est représentée syntaxiquement de manière indépendante de la signification (sémantique). À cette syntaxe est rattachée une sémantique, permettant de rendre possible la modélisation du contenu informationnel, puis l'application des diverses opérations.

Le deuxième point ci-dessus énuméré nous permet d'étendre cette théorie par des opérations syntaxiques sans appréhender un changement ou une extension de sémantique quelconques. Dans d'autres formalismes comme les logiques terminologiques, l'ajout d'une opération syntaxique doit s'accompagner d'une sémantique adéquate.

Au fait, la théorie des situations, à l'image des formalismes de représentation de connaissances cités dans le chapitre précédent, ne présente pas d'opérations ou d'inférence spéciales sur les relations. Elle peut être ainsi considérée, à juste titre, comme un cadre neutre pour l'étude de l'impact de la prise en compte des relations sur les décisions de pertinence. Nous faisons alors munir la théorie des situations par un mécanisme de raisonnement sur les relations. Le résultat à la fin est une véritable logique avec des axiomes et des règles d'inférence permettant de raisonner sur les relations et d'exploiter leurs caractéristiques. Toutefois, notre objectif principal n'est pas de proposer cette logique comme une instanciation à proprement parler du modèle de Rijsbergen, mais plutôt de l'utiliser comme un cadre formel permettant de déterminer ce qui manque à un formalisme ou à une logique donnée comme règles d'inférence et comme capacité de raisonnement sur les relations.

De plus, la théorie des situations est conforme aux hypothèses que nous avons formulé au début de ce chapitre. La théorie des situations privilégie ainsi la notion d'information plutôt que la valeur de vérité. Bien que la notion d'information soit omniprésente en recherche d'informations, sa modélisation a tardé et son impact a été longtemps ignoré. Ceci s'explique par le fait qu'aucune des théories de l'information connues à l'époque n'avait un comportement logique ou ne pouvait fournir la moindre base d'un processus de recherche. En effet, il n'était question que d'une modélisation restrictive de l'information basée souvent sur des notions probabilistes (Shannon-Weaver [SW64], Carnap [Car66], Hintikka [HS70]). La notion d'information n'a jamais été ainsi globalement et formellement définie.

Par ailleurs, en théorie des situations, le concept de valeur de vérité est secondaire. Le jugement de ce qui est vrai ou faux se fait à la base de l'information disponible et on s'intéresse à ce qui rend une information vraie ou fausse, plutôt qu'à sa simple valeur de vérité. Ainsi, par rapport aux modèles classiques de la recherche d'informations, l'accent est mis sur ce que le document véhicule comme information, et non pas sur le fait que cette information soit présente (vraie) ou pas. De plus, dans une approche logique classique, le paradoxe de l'implication classique [Nie90, pp28–29] (e.g.,"*tous les hommes sont égaux devant la loi, mais il y a des hommes au-dessus de la loi*") peut engendrer des contradictions et des déductions inacceptables. La théorie des situations permet de représenter le contenu du document sans se soucier de la valeur de vérité de ce dernier. Enfin, comme toute théorie d'informations, la théorie des situations présente l'avantage de traiter les informations incomplètes ou partielles. Ceci est ainsi conforme au principe du monde ouvert que nous avons mentionné au début du chapitre.

b) Quelques Notions de Base

Les Infons

L'unité de base d'information dans la théorie des situations est l'*infon*. Il constitue l'information primitive du domaine d'application. Les infons mettent en relation des objets en leur assignant une *polarité* qui indique si la relation en question est vérifiée pour ces objets.

De façon plus formelle, Devlin [Dev91] introduit la notion d'infon comme suit:

Définition 3.1.2 (Infon) *Un infon est une structure* $\langle\langle \mathrm{R}, a_1, \ldots, a_n; i\rangle\rangle$ *qui représente l'information que la relation* R *existe (si* $i = 1$*) ou n'existe pas (si* $i = 0$*) entre les objets* $a_1, ..., a_n$*.*

Les objets selon la conception de Devlin incluent des *individus*, comme par exemple "Tom", "Grenoble", des *emplacements locaux*, comme "jardin", "balcon", des *aspects d'ordre temporel* comme par exemple "10h", "hier", etc, des *types* comme par exemple PERSONNE, TABLE et des *paramètres* pouvant instancier les objets ci-dessus énumérés. Les paramètres sont dénotés par $\dot{\mathrm{p}}$,$\dot{\mathrm{q}}$, $\dot{\mathrm{s}}$, etc. Les relations correspondent à des propriétés uniformes qui maintiennent, ou lient des objets. Si la *polarité* i est égale à 1, l'infon est dit *positif*, autrement il est considéré comme *négatif*.

Exemple 6 *L'infon* $\langle\langle$Parler*, Jean, Alice, 8am, balcon; 1*$\rangle\rangle$ *représente l'information qu'une personne "Alice" était entrain de parler à une autre personne "Jean" vers 8 heures du matin dans un balcon.*

En recherche d'informations, les objets de la définition ci-dessus peuvent, par exemple, correspondre à des mots-clés, à des groupes nominaux ou à des concepts. Quant aux relations, il peut s'agir d'opérateurs sémantiques [Ber88], de relations conceptuelles [Che92] ou encore de connecteurs linguistiques [Pal90, Bru93].

La Relation de Support

Les situations sont à un niveau de granularité supérieur à celui des infons. Les situations sont vues comme des descriptions partielles du monde réel. Les descriptions incluent des faits extraits de ce monde réel. Ces faits sont représentés par des infons. Les infons ne sont ainsi autre que des

propriétés inhérentes à une situation donnée. L'information n'est pas véhiculée par la valeur de vérité des infons, mais plutôt par la valeur de vérité de la proposition *l'infon ϕ est inhérent à la situation S*[8] [Dev90]. La notion de *inhérent à*, est souvent désignée par la *relation de support*, notée $\models$. Par exemple, étant donné un infon ϕ et une situation S, la proposition $S \models \phi$ signifie que l'item d'information ϕ est inhérent à la situation S ou, en d'autres termes, que la situation S supporte l'infon ϕ.

Il est évident qu'il est très difficile dans la réalité de décider si oui ou non, une situation donnée supporte un infon particulier. De plus, dans la réalité, il est difficile de représenter tous les infons supportés par une situation. Par exemple, dans une situation donnée, un être humain n'est pas capable de percevoir tous les détails qui l'entourent à vue d'oeil. Afin de créer une théorie mathématique, la théorie des situations distingue deux types de situations, à savoir les *situation réelles* et *les situations abstraites*. Une situation réelle désigne une partie réelle du monde avec tous ses détails et ses spécificités. Une situation abstraite est par contre une construction mathématique, formée par un ensemble d'infons. La relation de support, pourra être ainsi facilement modélisée, en se basant sur la notion d'appartenance ensembliste:

Définition 3.1.3 (Relation de Support) *La relation binaire* "supporte" *entre une situation abstraite S et un infon ϕ, notée $\models$, est définie par:*

$$S \models \phi \; ssi \; \phi \in S$$

On dit alors que la situation S supporte ϕ (on dit aussi que ϕ est valable dans la situation S).

Exemple 7 *Considérons un document d contenant l'information suivante: "Les Sioux ont battu la cavalerie du Général Custer à Little Big Horn en 1876.". L'information véhiculée par ce document peut être modélisée par l'infon* $\langle\langle$battre,*Sioux,Cavalerie du Général Custer*; 1$\rangle\rangle$. *Ce document supporte ainsi l'infon en question. La relation* "battre" *dénote une relation particulière entre les termes «Sioux» et «Cavalerie du Général Custer». De telles relations sont utilisées pour fournir des informations contextuelles pouvant guider le processus de correspondance en recherche d'informations. Un index possible du document d ci-dessus pourrait être la situation D:* $\{\langle\langle$battre,*Sioux, Cavalerie du Général Custer*; 1$\rangle\rangle$, $\langle\langle$position,*Little Big Horn,1876*; 1$\rangle\rangle\}$.

Dans l'exemple précédent, à travers la lecture de l'information fournie, nous pouvons déduire que la situation indexant le document supporte également l'infon $\langle\langle$*battre*, Cavalerie du Général Custer, Sioux; 0$\rangle\rangle$. Cependant, elle ne supporte pas l'infon $\langle\langle$*battre*,Sioux,Apaches; 1$\rangle\rangle$, puisque ce dernier ne peut être perçu dans la situation D. Le fait que cet infon soit vraie ou pas dans la réalité n'a pas d'importance: il n'est pas supporté par la situation D.

Les Paramètres

Le concept des paramètres en théorie des situations[9] est très utile du point de vue de la recherche d'informations. Par exemple, si nous voulons spécifier que Tom est un CHAT ou que la relation *Poursuit* lie les types CHAT et SOURIS. Les variables $\dot{a}, \dot{b}$, etc. sont utilisées pour ce fait.

8. En anglais, *The infon ϕ holds in situation S*
9. Pour des plus amples informations, une description complète se trouve dans [Dev90], page 50.

Lorsqu'un objet utilisé dans un infon est un paramètre, alors nous parlons d'*infon paramétré*. Ainsi, pour spécifier que Tom est un CHAT, nous pouvons écrire ce qui suit:

$$s \models \langle\langle \textit{named}, \text{Tom}, \dot{a}; 1\rangle\rangle \text{ et } s \models \langle\langle \text{CHAT}, \dot{a}; 1\rangle\rangle$$

D'une façon générale, un paramètre dans l'infon exprime qu'une référence devrait être liée à un objet arbitraire. Par exemple, la relation de support $S \models \langle\langle \textit{Poursuit}, \dot{a}, \text{Souris}; 1\rangle\rangle$ ne fournit pas d'informations quant à l'"objet" qui poursuit la Souris, à moins qu'il existe une référence de $\dot{a}$ vers un individu particulier (par exemple, "Tom" ou "Minou"). Dans le cas contraire, l'information véhiculée par la situation S est vague.

Exemple 8 *La situation* $\{\langle\langle \text{Poursuit}, \dot{a}, \textit{Jerry}; 1\rangle\rangle, \langle\langle \text{nommé}, \textit{Tom}, \dot{a}; 1\rangle\rangle\}$ *explicite le fait que l'objet* $\dot{a}$ *qui poursuit "Jerry" est nommé "Tom" (puisque les deux infons sont étiquetés par le même paramètre* $\dot{a}$*).*

En théorie des situations, il existe également une généralisation des infons vers des types de situations. Considérons, par exemple, les infons $\langle\langle \textit{Poursuit}, \text{Tom}, \text{Jerry}; 1\rangle\rangle$, $\langle\langle \textit{Poursuit}, \text{Tom}, ; 1\rangle\rangle$, et $\langle\langle \textit{Poursuit}, \text{Tom}, \text{Alice}; 1\rangle\rangle$. Ces infons décrivent essentiellement la même situation dans laquelle "Tom poursuit quelqu'un". La différence est juste de savoir qui est poursuivi par Tom. Pour ces deux infons, il existe un type unifié:

$$\nu = [S \mid S \models \langle\langle \textit{Poursuit}, \text{Tom}, \dot{a}; 1\rangle\rangle]$$

Une situation est de type ν si, et seulement si, dans cette situation "Tom poursuit quelqu'un", c'est-à-dire, uniquement quand il existe $S \models \langle\langle \textit{Poursuit}, \text{Tom}, \dot{a}; 1\rangle\rangle$, et $\dot{a}$ peut être instancié à un individu donné.

La Relation d'Endiguement

Nous avons vu au début de ce chapitre qu'il est préférable d'adopter un principe de *monde ouvert*, c'est-à-dire un monde où tout ce qui n'est pas explicitement représenté par le système est considéré comme inconnu. Cependant, certaines informations sont souvent implicites. Il s'en suit qu'un mécanisme permettant leur déduction est nécessaire. En recherche d'informations, ces informations implicites peuvent favoriser la prise de certaines décisions de pertinence. Il s'agit des cas où la requête se réfère justement à ces informations implicites. La déduction de ces informations dans les systèmes de recherche d'informations classiques se base essentiellement sur la notion de la *relation d'endiguement*. Par exemple, dans le modèle Booléen, de l'infon $\langle\langle \wedge, \text{Maison}, \text{Commerce}; 1\rangle\rangle$, le mot-clé Maison peut être inféré. En effet, ce dernier est implicitement "contenu" (endigué) dans l'infon. Cela rappelle l'hypothèse de Fregean [Eps90], mentionnée au début du chapitre: si une expression est inhérente à un document, et qu'elle met en relation plusieurs sous-expressions, alors ces dernières sont toutes inhérentes au document.

Selon Landman [Lan86] et Barwise & Etchemendy [BE90], il existe une relation de préordre sur les infons donnée par la *relation d'endiguement*[10]. Cette dernière notée "$\rightarrow$", est ainsi réflexive ($\phi \rightarrow \phi$), antisymétrique (si $\phi \neq \psi$, alors au moins $\phi \not\rightarrow \psi$ ou $\psi \not\rightarrow \phi$) et transitive (si $\phi \rightarrow \psi$ et $\psi \rightarrow \gamma$

10. En anglais: *the information containment Relation*

alors $\psi \rightarrow \gamma$). La propriété de transitivité est aussi connue sous le nom du *Principe de Xerox*, dû à Dretske [Dre81].

Exemple 9 *La relation d'endiguement ci-dessous introduite est valide. Elle stipule le fait que si* "Tom est entrain de poursuivre Jerry dans un parc" *alors nécessairement* "Tom est entrain de poursuivre Jerry".

$\langle\langle$Poursuit, *Tom, Jerry, dans un Parc; 1*$\rangle\rangle \rightarrow \langle\langle$Poursuit, *Tom, Jerry; 1*$\rangle\rangle$

La relation d'endiguement "$\rightarrow$" peut aussi être donnée *a priori*. Elle exprime ainsi la *subsomption* entre infons, c'est-à-dire une sorte de relation de spécialisation/généralisation. Les systèmes de recherche d'informations munis d'un thésaurus s'appuient sur de tels relations d'endiguement pour améliorer les réponses du système. Ainsi, l'exemple 9 peut être complété par la relation d'endiguement suivante:

$\langle\langle$*Poursuit*,Tom,Jerry; 1$\rangle\rangle \rightarrow \langle\langle$*Poursuit*,Tom, *Animal*; 1$\rangle\rangle$.

En termes de situations, la définition suivante peut être énoncée:

Définition 3.1.4 (Relation d'Endiguement $(i \rightarrow j)$**)** *un infon j est dit endigué dans i (dénoté par $i \rightarrow j$) si, et seulement si, quelque soit la situations s si $s \models i$ alors $s \models j$*

D'une façon générale, en recherche d'informations la nature et les propriétés de la relation d'endiguement dépendent du modèle de recherche sous-jacent. Par exemple, dans le modèle Booléen, $\phi \rightarrow \psi$ est définie par $\phi \vdash \psi$. Cependant dans tous les cas, elle sert à représenter les connaissances du domaine, qu'elles soient de type syntaxique (e.g. relation entre expression et sous-expression) ou sémantique (e.g. thésaurus). Pour notre démarche, la base de connaissances devra inclure en plus les propriétés des relations, leur sémantique et leurs comportements. Dans ce chapitre, la relation d'endiguement sera ainsi étendue de façon à ce que la construction d'un modèle d'indexation relationnelle soit possible.

Composition d'informations

Une caractéristique de l'information est qu'elle est manipulable, donnant résultat ainsi à des formes plus complexes d'informations. Par exemple, deux pièces d'informations peuvent être composées pour donner lieu à une nouvelle pièce d'informations. Parfois, il est utile de combiner ensemble les infons pour donner lieu à une situation plus précise. Une *union informationnelle*, $\odot$, peut être employée. Par exemple, la composition de l'infon $\langle\langle$*Chat*,Tom; 1$\rangle\rangle$ avec l'infon $\langle\langle$*Souris*,Jerry; 1$\rangle\rangle$ donnera la situation $\{\langle\langle$*Chat*,Tom; 1$\rangle\rangle$, $\langle\langle$*Souris*,Jerry; 1$\rangle\rangle\}$:

$\langle\langle$*Souris*,Jerry; 1$\rangle\rangle \odot \langle\langle$*Souris*,Jerry; 1$\rangle\rangle = \{\langle\langle$*Chat*,Tom; 1$\rangle\rangle, \langle\langle$*Souris*,Jerry; 1$\rangle\rangle\}$

Une telle union informationnelle peut s'exprimer comme suit: si dans un monde réel, une personne perçoit le fait que Tom est un *Chat*, et le fait que Jerry est une *Souris*, alors cette personne peut considérer qu'il est en présence d'une situation dans laquelle un *Chat* nommé Tom et une *Souris* nommée Jerry sont observés.

À un degré de granularité supérieur, les situations abstraites, étant des ensembles d'infons, peuvent être combinées par les opérateurs ensemblistes $\cup$ et $\cap$. Une *union de situations*, prend tous les infons de la première situation avec ceux de la deuxième, puis ceux de la troisième, etc. Par exemple, l'union des situations S et T, notée $S \cup T$, représente tous les infons des deux ensembles. Si la situation $S \cup T$ supporte un infon, alors soit la situation S, soit la situation T supporte cet infon. L'opérateur $\cup$ est utilisé lorsqu'on veut combiner des situations non reliées. Par exemple, une bibliothèque peut contenir plusieurs rayons de bouquins regroupés par thèmes. Une bibliothèque peut ainsi être représentée par une situation S_B consistant en un ensemble de situations S_i, chacune ayant trait à un rayon particulier de la bibliothèque. Dans ce contexte précis, la situation $S_{roman-pol}$ est vue comme une description du rayon des romans policiers. La situation $S_{tourisme}$ décrit le rayon des livres touristiques, etc. La situation S_B est l'union de $S_{roman-pol}$ et $S_{tourisme}$ et de toutes les autres situations décrivant un rayon particulier de la bibliothèque.

Cependant, il est parfois nécessaire de relier ensemble des situations. Pour illustrer ce cas, prenons deux situations données. La première indiquant que " la Cavalerie du Général Custer était battue". La deuxième stipule, quant à elle, le fait que "les Sioux ont battu quelqu'un". Il est alors possible de créer une nouvelle information utilisant ces deux situations pour asserter que "les Sioux ont battu la Cavalerie du Général Custer". Cette déduction est basée sur l'hypothèse qu'il existe une certaine relation, ici que la cavalerie du général est la personne battue dans la seconde situation, entre les deux situations. Une opération de *fusion de situations* (notée par $\oplus$), peut être vue comme la création de nouvelles situations à partir de situations déjà existantes, ceci en les reliant ensemble.

Il existe plusieurs façons pour fusionner deux situations [HOC95, HOC96, Hui96]. On peut par exemple composer chaque infon de la première situation avec tous les infons de la seconde situation, par le biais de l'opérateur d' *union informationelle* $\odot$. Une autre manière consiste à combiner ensemble certains infons des deux situations, comme cela est illustré dans l'exemple suivant:

Exemple 10 *Soit par exemple les deux situations* $S = \{\langle\langle \text{battre}, \dot{p}, \dot{q}\ ; 1\rangle\rangle,\ \langle\langle \text{nommé}, \textit{Sioux},\ \dot{p}\ ; 1\rangle\rangle\}$ *et* $T = \{\langle\langle \text{battre}, \dot{r}, \dot{s}; 1\rangle\rangle,\ \langle\langle \text{nommé}, \textit{Cavalerie de Custer}\ ,\ \dot{s}\ ; 1\rangle\rangle\}$. *Afin de spécifier que "Les Sioux ont battu la Cavalerie de Custer", nous avons à préciser dans l'union des situation* S *et* T *que les paramètres* $\dot{p}$ *et* $\dot{r}$ *(respectivement* $\dot{q}$ *and* $\dot{s}$*) sont identiques. Le résultat est l'ensemble* $S \oplus T$ *contenant* $\{\langle\langle \text{battre}, \dot{r}, \dot{s}; 1\rangle\rangle,\ \langle\langle \text{nommé}, \textit{Sioux},\ \dot{r}\ ; 1\rangle\rangle,\ \langle\langle \text{nommé}, \textit{Cavalry de Custer}\ ,\ \dot{s}\ ; 1\rangle\rangle\}$.

Le même résultat de l'exemple précédent aurait pu être obtenu en utilisant l'opérateur d'union sur les situations et en faisant un choix correct des paramètres qui existent dans les ensembles qui leurs sont associés. Le processus de fusion est ainsi basé sur des informations sémantiques, qui sont difficiles à modéliser comme le signale Huibers dans [Hui96]. Dans ce qui suit nous nous contentons d'utiliser uniquement des unions de situations, et des compositions d'infons, puisqu'elles sont suffisantes pour nos traitements.

Enfin signalons que la fusion de deux situations identiques admet pour résultat la même situation (la fusion vérifie la propriété de l'idempotence). À ce stade, il est utile de préciser quand deux situations peuvent être considérées comme identiques. Du fait qu'une situation est présentée comme un ensemble d'infons, la sémantique de la situation ne dépend pas de l'ordre dans lequel se présentent ses infons. L'égalité de situations peut ainsi être définie comme une instance de l'équivalence ensembliste, notée $\equiv$. Pour deux situations S et T, l'énoncé $S \equiv T$ signifie intuitivement le fait que l'information véhiculée par S est égale à l'information véhiculée par T et vice-versa. Quand les deux situations contiennent des infons paramétrés, alors nous dirons qu'elles sont équivalentes si, et seulement si, les deux situations peuvent devenir totalement identiques en renommant les

paramètres. Un tel renommage de paramètres est utilisé dans les lambda calculs et est appelé α-conversion.

Des Profons vers les Relons

Les infons peuvent être long, complexes, et détaillés ou peuvent être courts et simples. La représentation des mots-clés donne lieu ainsi à des infons très simples. Huibers dans [HB94, Hui96] préfère alors parler de *profons*. Un profon est un infon dans lequel la relation R se limite à une relation unaire non spécifiée. Cette relation notée I reflète le fait qu'aucune connaissance sur les relations auxquelles prend part le mot-clé en question n'est donnée. Cela rejoint notre constatation du chapitre 2, à savoir que dans une telle démarche d'indexation, le processus de description des documents nous a dépourvu de toutes les relations présentes dans le document. Si un document d est indexé avec le mot-clé "Tom", alors nous pouvons considérer que l'information véhiculée par "Tom" est partie de, ou existe dans d. La notation $\langle\langle \mathrm{I}, \mathrm{Tom}; 1\rangle\rangle$ est utilisée dans [HB94, Hui96] pour dénoter le profon correspondant.

Pour les besoins d'une recherche orientée vers la précision des réponses, nous ne considérons pour l'indexation que des infons avec des relations spécifiées, porteuses d'une certaine sémantique et jouant un rôle défini. En d'autres termes, les profons ne correspondent pas à une notion de structure sémantique, et ne se prêtent pas ainsi à une indexation relationnelle. Seuls des infons contenant des relations n-aires ($n > 1$) seront retenus. Nous préférons alors introduire un type particulier d'infons que nous appelons *relons*. Comme son nom l'indique, ce type d'infon met l'accent sur les relations et constitue le support d'information élémentaire pour l'indexation des documents:

Définition 3.1.5 (Les Relons) *Un relon est un infon particulier de type* $\langle\langle \mathrm{R}, a_1, \ldots, a_n; i\rangle\rangle$*, où* $n > 1$.

Dans les approches traditionnelle à base de mots-clés, un terme d'indexation t présent dans un index donné peut être représenté par un relon particulier de la forme suivante:

$$\langle\langle \dot{Lien}, t, \dot{\mathrm{p}}; 1\rangle\rangle$$

où $\dot{Lien}$ peut être instancié par n'importe quelle relation du domaine d'application et $\dot{\mathrm{p}}$ peut dénoter n'importe quel élément du domaine. Ainsi, le contexte d'apparition du terme t étant perdu après la phase d'indexation du document, le relon ci-dessus introduit stipule que dans le document ce terme apparaît, et qu'il est possible qu'il soit en relation quelconque avec n'importe quel élément du domaine de l'application. L'approche classique par mots-clés est donc un cas particulier de l'indexation relationnelle. Elle engendre des index constitués de relons paramétrés, confirmant ainsi le fait qu'il s'agit de descriptions partielles et vagues.

D'une façon plus formelle, nous considérons tout au long de ce chapitre, un ensemble d'objets $\mathcal{O}$ dénotant les entités élémentaires du domaine de discours, un ensemble de paramètres *Param* et un ensemble de relations $\mathcal{R}$. L'ensemble $\mathcal{O}$ contient notamment des individus et des types d'individus (classes). L'ensemble des relations $\mathcal{R}$ contient les relations sémantiques du domaine et la relation spéciale $\dot{Lien}$ ci-dessus introduite. Le langage des relons peut alors être introduit de la manière suivante:

Définition 3.1.6 (Le langage des Relons) *Soit $\mathcal{O}$ un ensemble d'objets du domaine, Param un ensemble de paramètres, et $\mathcal{R}$ un ensemble de relations sémantiques muni de l'élément* $\dot{\text{Lien}}$. *Le langage $\Re(\mathcal{O}, Param, \mathcal{R})$ des relons est défini par:*

$$\Re(\mathcal{O}, Param, \mathcal{R}) =^{def} \{\langle\langle R, a_1, a_2, \ldots, a_n; i\rangle\rangle \mid R \in \mathcal{R}; a_i \in \mathcal{O} \cup Param; i \in \{0, 1\}\}$$

où $n > 1$. Quand $\mathcal{O}$, Param et $\mathcal{R}$ sont connus dans le contexte de l'application, nous noterons le langage des relons par $\Re$.

Par exemple, les termes $\langle\langle$*Poursuit*, Tom, Jerry; $1\rangle\rangle$, ou $\langle\langle$*Poursuit*, Tom, $\dot{\text{p}}$; $0\rangle\rangle$, ou encore $\langle\langle$*$\dot{\text{Lien}}$*, Tom, $\dot{\text{p}}$; $1\rangle\rangle$constituent des éléments du langage $\Re$. Les relons constituent les éléments élémentaires autour desquels les termes d'indexation de notre modèle seront construits.

3.2 Un Modèle Relationnel pour la RI

Notre modèle relationnel suit l'approche logique de van Rijsbergen [vR86]. Dans cette approche la correspondance entre une requête et une description d'un document est fondée sur la notion de l'inférence logique. La conception du modèle revient ainsi à la construction d'un système logique. Selon Gabbay [Gab94], la première étape pour la conception d'un système logique consiste à spécifier un langage de représentation. Ce dernier sera ensuite utilisé pour la définition de la notion de *formule bien formée*[11]. Il sera après possible de construire les axiomes et les règles d'inférence de ce système logique. En d'autres termes, il n'est pas suffisant de connaître le comportement de ce système logique, puisque nous devons également connaître comment il se présente. Cette dernière affirmation due à Gabbay montre l'importance du langage de représentation utilisé. Par exemple, nous montrons dans le chapitre suivant que le modèle des graphes conceptuels [Che92] et le modèle Booléen [vR79, SM83] ont des comportements presque identiques en termes de décisions de pertinence [HOC96, Hui96]. Cependant, la richesse d'expressivité du formalisme des graphes conceptuels par rapport à la logique propositionnelle (utilisée comme langage sous-jacent au modèle Booléen), explique pourquoi ce formalisme est mieux adéquat pour la conception des systèmes orientés vers la précision des réponses. Ainsi, les performances d'un système logique dépendent aussi bien de l'expressivité du langage utilisé que de la puissance d'inférence.

Dans ce chapitre, nous sommes intéressés par un système logique pour la recherche d'informations. Nous savons d'après notre étude du chapitre 2 que ce système devra avoir un comportement inférentiel permettant le raisonnement sur les relations. Cependant, encore faut-il que son langage de représentation soit spécifié. Ainsi si nous considérons la déduction comme étant l'opération de recherche, la relation de pertinence entre les documents et les requêtes peut être établie en termes des axiomes et des règles d'inférence inhérents au système logique. Par conséquent, le langage que nous utilisons pour la description des documents et des requêtes sera celui qui spécifiera la syntaxe d'une formule bien formée dans un système logique pour la recherche d'informations.

Toujours selon Gabbay [Gab93], pour qu'un système ait un comportement vraiment logique, il devra vérifier certains axiomes et règles d'inférence bien spécifiques: il s'agit de l'axiome de *réflexivité* et des règles du *Cut* et de *(non)-monotonicité*[12]. Nous présenterons en détail ces axiomes

11. En anglais, on parle de *well formed formula*, notée *wff* [Eps90, Eps94]
12. Selon qu'il s'agisse d'un système monotone ou non-monotone.

et règles dans la section 3.4.1. Outre ces axiomes et ces règles prédéfinies qui feront en sorte que notre système soit réellement un système logique, nous établirons un certain nombre de règles d'inférence permettant la prise en compte des propriétés des relations et les objets qu'elles lient. À chaque propriété de relation ou d'objet, qu'elle soit purement syntaxique (réécriture) ou sémantique (utilisant des connaissances sur le domaine d'application), correspondra ainsi une règle de dérivation dans notre modèle logique.

Enfin, un système logique devra avoir une sémantique claire permettant d'expliquer le processus de dérivation (dans notre cas le processus de recherche), d'asserter que toutes les décisions de dérivations élaborées par le système sont correctes (propriété de correction), et que les règles de raisonnements formalisées par le système suffisent pour démontrer toutes les relations de pertinence exprimables dans son langage (complétude).

Dans la section suivante, nous spécifions un langage $\mathcal{L}$ permettant la description des documents et des requêtes. Ce langage devra être assez expressif pour pouvoir représenter des structures sémantiques complexes, incluant un ensemble d'objets et de relations et de ce fait conforme à notre hypothèse 2 du chapitre 2 (page 22). Une fois ce langage spécifié, nous montrons en quoi consiste la prise en compte des propriétés des relations et des objets dans un processus de recherche. Cela nous amènera enfin à proposer un système logique de dérivation de pertinence, permettant la prise en compte des relations, des objets ainsi que leurs propriétés. Ce système constituera ainsi notre modèle d'indexation relationnel.

3.2.1 Le langage d'Indexation

Pour les raisons que nous avons mentionné dans la section 3.1.2 a)[page 72] , nous adoptons un langage d'indexation basé sur la théorie des situations. Ce langage devra nous permettre de représenter les éléments jugés a priori les plus porteurs des informations véhiculées par les documents. Dans notre cas, conformément à l'idée de Farradane [Far80a, Far80b] (Voir chapitre 2), nous considérons que les relations entre objets[13] traduisent le mieux le contenu sémantique des documents, en permettent une recherche orientée vers la précision des réponses. Ces relations entre objets peuvent être décrites par la notion du *relon* introduite dans la section précédente. Notre langage d'indexation se base donc sur ce type particulier d'infons:

Définition 3.2.1 (Langage d'Indexation Relationnelle) *Soit $\Re(\mathcal{O}, Param, \mathcal{R})$ le langage des relons de la définition 3.1.6. Le langage d'indexation $\mathcal{L}(\Re)$ est défini comme étant un sur-ensemble de $\Re(\mathcal{O}, Param, \mathcal{R})$ tel que:*

- *si $r_i \in \Re(\mathcal{O}, Param, \mathcal{R})$ alors $r_i \in \mathcal{L}(\Re)$*
- *si $a_i \in \mathcal{L}(\Re) \cup \mathcal{O}$ et $\mathrm{Rel} \in \mathcal{R}$ une relation n-aire donnée, alors $\langle\langle \mathrm{Rel},\ a_1\ ,\ \ldots\ ,\ a_n\ ;\ i\rangle\rangle \in \mathcal{L}(\Re)$ avec $i \in \{0, 1\}$ et $n > 1$.*

Quand il n'existe pas d'ambiguïté possible sur l'ensemble des relons $\Re$, nous désignerons le langage d'indexation simplement par $\mathcal{L}$.

13. La notion d'objet est à prendre ici au sens large, c'est-à-dire qu'elle couvre les individus, les classes d'individus ou des expressions quelconques du langage utilisé.

La première partie de cette définition fixe le *degré d'expressivité minimal* que doit présenter un langage pour une indexation efficace du contenu des documents. Ainsi, le langage d'indexation relationnelle ne peut être moins expressif que le langage des relons. La deuxième partie de la définition autorise l'imbrication des relons (voir figure 3.1). Cela permet par exemple de représenter des termes d'indexation complexes comme ceux du système RIME (Cf. Chapitre 2):

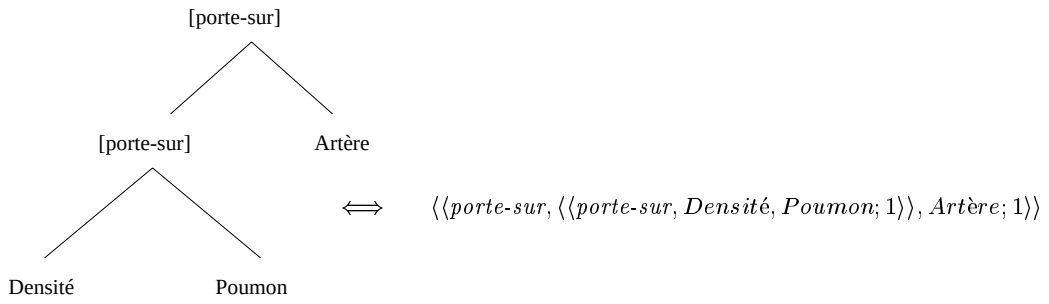


Figure 3.1. Exemple d'imbrication de Relons

Selon la terminologie de la théorie des situations, nos termes d'indexation correspondent à des infons. Il s'agit d'un ensemble de relons complexes formés à partir de $\Re$. À un niveau de granularité supérieur, nous retrouvons les situations, ou en termes de recherche d'informations, les représentations des documents et des requêtes.

3.2.2 La Fonction de Description χ

Par rapport au langage d'indexation $\mathcal{L}$ de la définition 3.2.1, le processus d'indexation normalise et sélectionne les termes d'indexation retenus pour chaque document.

Définition 3.2.2 (La Fonction χ) *Soit $\mathcal{D}$ l'ensemble des documents du corpus. La fonction d'indexation χ associe à chaque document $d \in \mathcal{D}$, un ensemble de termes dans le langage d'indexation $\mathcal{L}$:*

$$\chi : \mathcal{D} \rightarrow 2^{\mathcal{L}}$$

où $2^{\mathcal{L}}$ désigne l'ensemble des parties de $\mathcal{L}$.

Ainsi la formalisation utilisée débute avec les individus et les classes d'individus pour la modélisation des informations primitives du domaine, et finit avec les situations pour une modélisation complète du contenu des documents et des requêtes. Formellement le langage des situations est défini comme suit:

Définition 3.2.3 (Le Langage des Situations) *Soit $\mathcal{L}$ le langage d'indexation relationnelle de la définition 3.2.1. Le langage des situations $\mathcal{S}_{\mathcal{L}}$ est défini comme l'ensemble des parties de $\mathcal{L}$:*

$$\mathcal{S}_{\mathcal{L}} = 2^{\mathcal{L}}$$

Quand il n'y a pas d'ambiguïté possible sur le langage $\mathcal{L}$, nous écrirons $\mathcal{S}$ plutôt que $\mathcal{S}_{\mathcal{L}}$

L'index $\chi(d)$ est donc une situation. Dans notre système logique, un document d sera dit logiquement pertinent pour une requête q s'il existe une chaîne de dérivation dont le premier élément est la situation $\chi(d)$ et le dernier élément est la requête de l'utilisateur, également exprimée en une situation $\chi(q)$. Les éléments intermédiaires sont directement dérivables en vertu des décisions de pertinence du système. Nous adoptons le mot "à-propos"(*aboutness*) (Cf. [Mar77, Hui96]) pour exprimer la notion de pertinence logique. On dira alors qu'un document d est "*à-propos*" de la requête q, s'il est logiquement pertinent pour cette dernière. Nous noterons par $\Box\!\!\rightsquigarrow$ cette notion de pertinence logique.

Par rapport à la relation d'endiguement $\rightarrow$, la relation de pertinence $\Box\!\!\rightsquigarrow$ s'applique à des situations plutôt qu'à des infons ou des relons. Seuls les cas où les situations comportent un seul relon font que $\rightarrow$ et $\Box\!\!\rightsquigarrow$ coïncident. La relation de pertinence $\Box\!\!\rightsquigarrow$ constitue ainsi une extension de la relation d'endiguement des infons aux situations. Les relons du langage $\mathcal{L}$ contenus dans une situation $\chi(d)$ donnée forment une structure algébrique de *monoïde*. Cette dernière, munie de l'opération d'union informationnelle $\odot$, satisfait les propriétés suivantes:

- La propriété d'implication:

$$\begin{array}{lll} \forall e_i, e_j \in \chi(d) & e_i \odot e_j & \Box\!\!\rightsquigarrow \quad e_i \\ & e_i \odot e_j & \Box\!\!\rightsquigarrow \quad e_j \end{array}$$

- $\odot$ est *commutative*. Cette propriété exprime que la sémantique de l'index du document est indépendante de l'ordre dans lequel nous considérons ses termes d'indexation.

$$\forall e_i, e_j \in \chi(d), e_i \odot e_j \Box\!\!\rightsquigarrow e_j \odot e_i$$

- $\odot$ est *idempotent*. Comme le signale Gabbay [Gab93], la réflexivité est universellement satisfaite par tout raisonnement basé sur une notion de conséquence ou de dérivation logique (i.e., $\forall e$ e dérive e)[14]. Cette propriété est supportée par la relation de pertinence dans plusieurs modèles de recherche d'informations [Hui96]: i.e., $\forall e \in \mathcal{L}, e \Box\!\!\rightsquigarrow e$. En utilisant cette propriété, nous pouvons inférer le fait que $\odot$ est *idempotent*.

$$\forall e \in \mathcal{L}, e \odot e \Box\!\!\rightsquigarrow e$$

- Le terme d'indexation vide ε est un élément neutre pour l'opérateur $\odot$:

$$\forall e \in \mathcal{L}, e \odot \varepsilon \Box\!\!\rightsquigarrow e$$

Nous dénotons par $\mathcal{M}$ le modèle logique basé sur $\mathcal{D}$, $\mathcal{L}$, χ et $\Box\!\!\rightsquigarrow$, désignant respectivement l'ensemble des documents du corpus, le langage d'indexation relationnelle, la fonction de description et la notion de pertinence. Si aucune propriété sur les relations ou sur les objets n'est utilisée, ce modèle pourra être formalisé de façon très simple en associant la notion de pertinence logique $\Box\!\!\rightsquigarrow$ à une notion d'inclusion ensembliste:

Définition 3.2.4 (Relation de pertinence $\Box\!\!\rightsquigarrow$) *Une situation S est à-propos de la situation T ($S \Box\!\!\rightsquigarrow T$), si tous les infons (les termes d'indexation) de T sont inclus dans S.*

14. Les relations de conséquence logique qui ne vérifient pas cette relation expriment probablement une certaine notion de changement de théorie [KLM90].

Illustrons cette définition par un exemple:

Exemple 11 *La situation* $S = \{\langle\langle \text{Poursuit}, \textit{Tom}, \textit{Jerry}\ ; 1\rangle\rangle, \{\langle\langle \text{Localité}, \textit{Aventure}, \textit{Parc}; 1\rangle\rangle\}$, *est à-propos de la situation* $T = \{\langle\langle \text{Poursuit}, \textit{Tom}, \textit{Jerry}\ ; 1\rangle\rangle\}$. *Ce cas se traduit par le fait que* $T \cup U \ \Box\!\!\rightsquigarrow T$.

La définition 3.2.4 informelle ci-dessus introduite rappelle la notion de subsomption dans les formalismes de représentation de connaissances (comme par exemple les logiques terminologiques). Ainsi, si l'on définit une fonction d'interprétation $\mathcal{I}$ sur les situations, nous pouvons appliquer une sémantique dénotationnelle pour la théorie des situations.

Définition 3.2.5 (Une dénotation pour les Situations) *Soit une situation* $s \in \mathcal{S}$. *L'interprétation de* s, *notée* $[[s]]^{\mathcal{I}}$, *est constituée de l'ensemble des infons qu'elle supportent:*

$$\textit{si}\ s \models \phi\ \textit{alors}\ \phi \in [[s]]^{\mathcal{I}}$$

Nous pouvons alors définir une relation de subsomption sur les situations:

Définition 3.2.6 (Subsomption de Situations) *Soient* s *et* t *deux situations de* $\mathcal{S}$. *On dit que* s *subsume* t *et on écrit* $t \sqsubseteq s$ *si, et seulement si, pour toute interprétation* $\mathcal{I}$, $[[t]]^{\mathcal{I}} \subseteq [[s]]^{\mathcal{I}}$. *Autrement dit:*

$$t \sqsubseteq s \quad \textit{ssi} \quad \forall\phi, \quad t \models \phi \quad \textit{implique} \quad s \models \phi$$

Ainsi, nous dirons que $S \ \Box\!\!\rightsquigarrow T$ est valide dans notre modèle logique élémentaire $\mathcal{M}$ si, et seulement si, $T \sqsubseteq S$. Afin de différentier la notation utilisée pour la relation de support de celle de la conséquence logique, nous dénotons cette dernière par $\models\!\!\equiv$. Par conséquent, un modèle simple $\mathcal{M}$ peut être introduit de la façon suivante:

Définition 3.2.7 (Modèle Logique de Base) *Un modèle logique de base est un quadruplet* $\mathcal{M} = \langle \mathcal{D}, \mathcal{L}, \chi, \ \Box\!\!\rightsquigarrow \rangle$. *La relation de pertinence logique* $\Box\!\!\rightsquigarrow$ *entre deux situations* S *et* T *étant définie comme suit:*

$$\models\!\!\equiv_{\mathcal{M}} S \ \Box\!\!\rightsquigarrow T \ \textit{ssi}\ T \sqsubseteq S$$

Toutefois, les fonctions d'indexation ne sont généralement pas fermées sur les connaissances du domaine. Autrement dit, elles n'incluent pas dans l'index toutes les informations implicites déduites des connaissances de l'application. Par conséquent, une partie du corpus ne sera pas renvoyée en réponse à des requêtes adressant ce type d'information. Dès lors, il est légitime de considérer une notion de *préférence* sur les index lors du processus de correspondance. Autant un index est complet vis-à-vis des connaissances du système, autant il est préférable de le garder comme descripteur du document associé.

Il est cependant rare qu'un index soit exhaustif, que ce soit pour des raisons évidentes de stockage ou à cause des changements possibles de la base de connaissances pouvant altérer les index déjà construits. Il sera ainsi plus adéquat de considérer qu'un index devra se contenter de représenter le minimum d'informations stables permettant de décrire le contenu du document. La prise en compte

des connaissances du domaine se produit le plus souvent à l'étape d'interrogation. Une approche classique en recherche d'informations consiste, par exemple, à élargir la requête par tous les termes pouvant être déduits à partir des connaissances du système. Cela revient à une approche duale de celle consistant à ajouter les connaissances implicites aux documents. Nous préférons, pour notre part, effectuer les raisonnements au niveau de la fonction de correspondance. Ainsi, pour une requête et un document particuliers, nous essayons de faire passer l'index du document de son état initial vers un état regroupant toutes ses informations implicites, en tenant compte des connaissances du domaine. Ce dernier sera *préféré* à l'index originel lors de l'évaluation de la requête.

Cette notion de préférence rappelle les modèles cumulatifs de Kraus & al [KLM90], ainsi que la sémantique des mondes possibles [Kri63]. L'idée de notre modèle relationnel est assez simple: nous considérons que l'index d'un document est une situation abstraite comprenant un ensemble de relons construits selon la syntaxe du langage d'indexation $\mathcal{L}$ introduit dans la définition 3.2.1.

L'utilisation des propriétés des relations et les objets qu'elles lient permet le passage d'une situation abstraite à une autre. Ce passage se réalise par l'élargissement de l'index originel du document par les connaissances implicitement connues du système. Les connaissances ont trait aux propriétés sémantiques et syntaxiques des relations, ainsi qu'à la façon dont elles se comportent quand elles sont combinées ensemble. Les nouvelles situations ainsi élargies seront préférées lors du processus de correspondance à l'index originel.

Nous introduisons alors une relation de préférence $\prec_R \subset \chi(d) \times \chi(d)$ sur les index des documents.

Définition 3.2.8 (La relation de Préférence $\prec_R$) *L'expression $\chi'(d) \prec_R \chi(d)$ signifie, qu'étant données les propriétés des relations et des objets connues par le système, l'index élargi $\chi'(d)$ est préféré à l'index $\chi(d)$ dans le processus de correspondance.*

Illustrons cette notion de préférence sur un exemple:

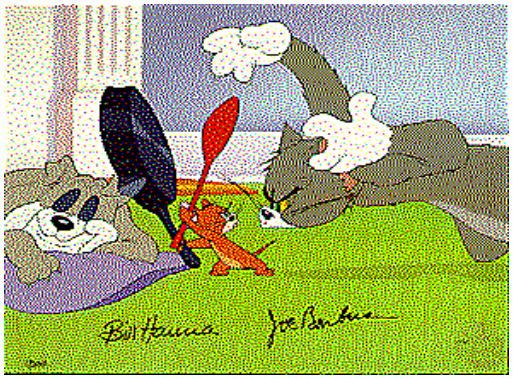
Exemple 12 *Considérons l'image de la figure 3.2, décrivant une aventure de Tom & Jerry. Soit $\chi(d)$ l'index originel de cette image, c'est-à-dire celui qui est stocké par le système. Il contient les informations jugées nécessaires par un indexeur pour la description du contenu de l'image. L'index $\chi'(d)$ sera toutefois préféré à $\chi(d)$ pour la comparaison avec la requête. En effet, il est obtenu par le biais de l'application d'une règle de dérivation exploitant le fait que la relation* "a-droite" *est l'inverse de la relation* "a-gauche". *Nous avons alors $\chi'(d) \prec_R \chi(d)$.*

Notre relation de préférence est liée à la fonction de correspondance plutôt qu'aux index eux-mêmes. En d'autres termes, il n'est pas question de dire que l'index $\chi'(d)$ est plus approprié que l'index $\chi(d)$. Une telle affirmation serait même fausse puisque dans ce cas $\chi'(d)$ est redondant.

Définition 3.2.9 (Correspondance et Préférence sur les Index) *Soit $\mathcal{D}$ un ensemble de documents. Nous notons par $P_{\chi(d)}(d)$ l'ensemble des index élargis par rapport aux connaissances K du domaine, et préférables à $\chi(d)$ lors de l'élaboration de la correspondance:*

$$\mathcal{P}_{\chi(d)}(d) = \{\chi'(d) | \chi'(d) \prec_R \chi(d)\}$$

Pour la formalisation de cette notion de préférence, nous nous basons sur les modèles cumulatifs définis par Kraus, Lehman, et Magidor dans [KLM90].



$\chi(d)$

$\{\langle\langle$*a-droite*, Tom, Jerry; 1$\rangle\rangle$.

$\{\langle\langle$*Etat*, Tom, colère; 1$\rangle\rangle\}$

$\chi'(d)$

$\{\langle\langle$*a-droite*, Tom, Jerry; 1$\rangle\rangle$.

$\{\langle\langle$*a-gauche*, Jerry, Tom; 1$\rangle\rangle$.

$\{\langle\langle$*Etat*, Tom, colère; 1$\rangle\rangle\}$

Connaissances:

$a\text{-}droite = \mathbf{Inverse}(a\text{-}gauche)$

Figure 3.2. Exemple de relation de préférence: $\chi'(d) \prec_R \chi(d)$

3.2.3 Un Modèle Préférentiel pour la RI

Notre modèle d'indexation relationnel se base théoriquement sur les logiques cumulatives de Kraus & al. Ces auteurs ont défini un cadre général pour l'étude des logiques non-monotones. Ce cadre permet d'associer à la relation d'inférence logique, une notion de préférence propre au domaine de l'application traité. En recherche d'informations, ce modèle a été appliqué dans [Won96, WBHvdW96] pour modéliser les jugements de préférence des utilisateurs concernant les réponses fournies par le système. Nous adoptons ici, pour notre part, une notion de préférence sur les index lors de la phase de correspondance.

a) Les Modèles Préférentiels

Intuitivement, un modèle préférentiel [KLM90] consiste essentiellement en un ensemble d'états, représentant des situations données dans le monde réel, et une relation binaire sur ces états. La relation représente les préférences d'un individu sur ces différents états: cet individu peut par exemple préférer les états dans lesquels il regarde un film à la télévision à des états dans lesquels il fait la vaisselle. D'une façon plus réaliste, une personne préférerai les états dans lesquels "Tweety" est un oiseau qui vole à ceux où "Tweety" est un oiseau malade qui n'arrive pas à voler. Les préférences de cet individu peuvent être décrits par un modèle, où l'individu accepte une *conséquence logique* $\alpha \rightarrow \beta$ si, et seulement si, tous les états *les plus préférables* parmi l'ensemble des états satisfaisant α, satisfont β. La notion de conséquence logique dans ce modèle est ainsi associée à la relation de préférence. Le système établira la conclusion β à partir de α, si tous les états les plus préférables satisfaisant α satisfont aussi β.

Selon les propriétés de la relation de préférence, le modèle préférentiel admet des caractéristiques propres. Dans notre cas, nous considérons que les états correspondent à des situations abstraites. La relation de préférence $\prec_R$ est définie sur ces situations. Au départ, un index de document

correspond à une situation particulière. Le document sera jugé pertinent pour une requête si les situations les plus préférables associées à l'élargissement de son index, par les propriétés des objets et des relations, sont à-propos de la requête.

Dans la section suivante, nous proposons un *modèle relationnel préférentiel*, permettant d'associer une sémantique aux dérivations logiques que nous voulons établir dans le cadre de notre modèle d'indexation relationnel. Ainsi, pour une requête donnée, l'évaluation de la pertinence d'un document se fait à la base de l'élargissement de l'index de ce dernier par toutes les informations pouvant être dérivées en utilisant les connaissances du système. Ces connaissances ont trait aux propriétés et aux caractéristiques des relations et des objets contenus dans les index des documents.

b) Application aux Index des Documents

Avant de proposer le modèle relationnel préférentiel que nous allons adopter tout au long de ce chapitre, il nous faudra d'abord spécifier une notion de conséquence logique adéquate à notre problématique en recherche d'informations. Ainsi, en utilisant la définition 3.1.1 (page 69), nous pouvons déduire une notion de conséquence logique appliquée aux situations:

Définition 3.2.10 (Conséquence Sémantique) *L'index $\chi(d)$ d'un document d correspond à une situation comprenant un ensemble de relons. Une requête est également représentée par une situation $\chi(q)$ constituée d'un ensemble de relons. Nous dirons que $\chi(q)$ est une conséquence sémantique de $\chi(d)$, notée par "$\chi(d) \models \chi(q)$", si dans toutes les interprétations dans lesquelles tous les relons de $\chi(d)$ sont évalués à vrai, il en est de même pour les relons de la requête $\chi(q)$.*

Outre la définition d'une notion de conséquence logique attribuant une sémantique aux inférences logiques établies par le système, la définition d'un modèle relationnel préférentiel nécessite aussi la précision des caractéristiques de la relation de préférence relationnelle $\prec_R$:

Propriété 1 *La relation de préférence $\prec_R$ sur les index est irréflexive, anti-symétrique et transitive. Elle constitue ainsi, une relation d'ordre partiel strict sur les situations abstraites.*

La propriété d'irréflexivité empêchera le système de dérivation de passer d'une situation S à une situation élargie qui lui soit identique[15]. Cette propriété est intéressante, car elle évitera la boucle infinie sur certaines propriétés des relations.

Si nous faisons l'analogie avec la sémantique des mondes possibles, le passage d'une situation vers une autre ne sera possible que si les propriétés de la relation de préférence $\prec_R$ sont vérifiées. La situation obtenue alors par application de la règle de dérivation associée sera préférée à la situation de départ.

À ce stade, il est intéressant de s'interroger sur les éléments minimaux de l'ordre ainsi établi sur les situations. Nous considérons alors la définition suivante:

Définition 3.2.11 (Minimal Vs. Minimum) *Soit $\prec_R$, la relation de préférence sur les index. Soit $\mathcal{S}$ le langage des situations. Soit $\mathcal{T} \subseteq \mathcal{S}$. Nous dirons que $t \in \mathcal{T}$ est* minimal *dans $\mathcal{S}$ si, et seulement si, $\forall s \in \mathcal{S}$, $s \not\prec_R t$. Nous dirons que $t \in \mathcal{T}$ est un minimum de $\mathcal{T}$ si, et seulement si, $\forall s \in \mathcal{S}$ tel que $s \neq t$, $t \prec_R s$.*

15. Voir page 78 pour l'explication de cette notion.

Dans [KLM90], le lemme suivant est énoncé:

Lemme 1 *Soit $\mathcal{E}$ un ensemble, et $\prec$ une relation binaire anti-symétrique sur $\mathcal{E}$. Si $\mathcal{E}$ a un minimum alors il est unique et il est un élément minimal dans $\mathcal{E}$*

Notre relation de préférence $\prec_R$ sur l'ensemble des situations $\mathcal{S}$ vérifie la condition de ce lemme. Il reste alors à identifier cet élément dans le cas où il existe. Dans la suite de cette section, nous tâchons de cerner cet élément.

Ainsi, le coeur d'un modèle préférentiel tourne autour de la définition des états les plus préférables. Ces derniers associées à la relation de conséquence logique permettent la construction du modèle préférentiel. Dans notre cas la relation de préférence est liée aux index des documents. Les états les plus préférables sont ainsi, les index élargis les plus préférés:

Définition 3.2.12 (Les Index les Plus Préférés) *Notons par $[\chi(d)]^{\mathcal{P}}$ l'ensemble des éléments de $\mathcal{P}_{\chi(d)}(d)$. On notera par $\mathcal{MP}(\chi(d))$ l'ensemble des index élargis les plus préférés d'un document d lors du processus de correspondance:*

$$\mathcal{MP}(\chi(d)) = \{\chi^i(d) \in [\chi(d)]^{\mathcal{P}} \mid \forall \chi^j(d) \in [\chi(d)]^{\mathcal{P}} : \chi^j(d) \not\prec_R \chi^i(d)\}$$

Dans le cas où l'ensemble $\mathcal{P}_{\chi(d)}(d)$ est vide, c'est-à-dire lorsqu'il n'est pas possible d'élargir l'index originel d'un document en utilisant toutes les connaissances du système, nous posons l'hypothèse suivante:

Hypothèse 4 *Si $\mathcal{P}_{\chi(d)}(d) = \emptyset$, alors $\mathcal{MP}(\chi(d)) = \chi(d)$*

Nous définissons alors une notion de conséquence préférentielle, notée $\mid\!\sim_R$, où $\alpha \mid\!\sim_R \beta$ correspond au fait que β est une conséquence sémantique de α étant données les propriétés des relations et des objets:

Définition 3.2.13 (Validité dans le Modèle Préférentiel) *La relation de conséquence préférentielle est dénotée par $\mid\!\sim_R$ et est définie comme suit:*

$$\chi(d) \mid\!\sim_R \chi(q) \; ssi \; \forall s \in \mathcal{MP}(\chi(d)), s \models \chi(q)$$

Par rapport aux propriétés de notre relation de préférence $\prec_R$, nous savons d'après le lemme 1 que s'il existe une situation (un index élargi) qui soit la plus préférable à l'index originel alors cette situation est unique. Dans le cas où $\chi(d)$ ne peut être élargi ($\mathcal{P}_{\chi(d)}(d) = \emptyset$), l'hypothèse 4, permet de déclarer que ce dernier est la situation unique la plus préférable lors de la phase de correspondance. Nous énonçons de plus le théorème suivant:

Théorème 1 (Unicité de la Situation la plus préférée) *Si le langage d'indexation relationnelle $\mathcal{L}$ de la définition 3.2.1 est fini, c'est-à-dire que l'ensemble des situations $\mathcal{S}$ est fini, alors il existe un élément minimal unique $\chi(d)_{Elargi}$ où:*

$$\mathcal{MP}(\chi(d)) = \{\chi(d)_{Elargi}\}$$

Preuve 1 *La relation de préférence $\prec_R$ étant irréflexive, il n'est pas possible de boucler infiniment sur la même situation. De plus, cette relation étant un ordre partiel strict, nous ne pouvons pas avoir de cycle au niveau d'une chaîne de préférence. Ainsi un cycle $\chi^1(d) \prec_R \ldots \prec_R \chi^n(d) \prec_R \chi^1(d)$ reviendrait à considérer que tous les index étendus $\chi^i(d)$ sont égaux. Ceci n'est pas possible puisque de part l'irréflexivité de la relation $\prec_R$, chaque extension d'un index devra ajouter de l'information, en un sens qu'elle ne peut faire aboutir à une situation identique à celle du départ. Il reste donc à montrer qu'il n'existe pas de chaîne descendante infinie par $\prec_R$ pour tout index de document. Ainsi, avoir une chaîne de préférence infinie $\chi^\infty(d) \prec_R \ldots \chi^i(d) \ldots \prec_R \chi(d)$ revient à dire qu'il est toujours possible d'ajouter de l'information aux mondes possibles de l'index originel. Cet ajout d'informations est assimilé comme nous l'avons vu dans la section 3.1.1 (page 69) à la précision de certaines valeurs de vérité des termes d'indexation du langage. Autrement dit à un passage de la valeur "inconnu" à la valeur "vraie" ou "faux". Au cas extrême, vu qu'il n y a pas de bouclage sur la même situation, toutes les valeurs de vérité des termes d'indexation du langage $\mathcal{L}$ seront ainsi précisées pour le document en question. Il n'est donc plus possible d'ajouter quoi que ce soit au risque de boucler sur la même situation. Il n'existe donc pas de chaîne infinie.*

Dans le cas du système RIME, l'imbrication des arborescences sémantiques peut atteindre 3 ou 4 niveaux. Nous avons préféré que le langage d'indexation relationnelle de la définition 3.2.1 soit infini pour être plus général. L'imbrication des relons est présentée comme possiblement infinie. Nous verrons dans la section 3.4.4 que dans le cas contraire, le système de dérivation proposé est décidable.

On définit alors notre modèle relationnel préférentiel comme suit:

Définition 3.2.14 (Modèle Relationnel Préférentiel) *Un modèle relationnel préférentiel $\mathcal{M}_{RP}$ est une paire $\langle\ \mathcal{S}, \prec_R\ \rangle$, où $\mathcal{S}$ est l'ensemble des situations introduit dans la définition 3.2.3 et $\prec_R$ la relation de préférence non-réflexive, anti-symétrique et transitive introduite dans la définition 3.2.8. La fonction χ du modèle logique de base $\mathcal{M}$ (Cf. définition 3.2.7) associant à chaque document une situation (ou un état) de $\mathcal{S}$.*

Le passage d'une situation à une autre est le résultat de l'application d'une règle de dérivation. Les règles de dérivation exploitent les connaissances du système. Par exemple, l'utilisation de la connaissance $\langle\langle$*Poursuit*, CHAT, SOURIS ; 1$\rangle\rangle \rightarrow \langle\langle$*Poursuit*, ANIMAL, SOURIS; 1$\rangle\rangle$permet de faire passer une situation $S = \{\langle\langle$*Poursuit*, CHAT, SOURIS ; 1$\rangle\rangle$, $\langle\langle$*État*, CHAT, COLÈRE; 1$\rangle\rangle\}$ vers une situation $S' = \{\langle\langle$*Poursuit*, ANIMAL, SOURIS ; 1$\rangle\rangle$, $\langle\langle$*État*, CHAT, COLÈRE; 1$\rangle\rangle\}$. Pour effectuer ce passage, la règle qui suit peut être utilisée:

Si $\phi_k \rightarrow \psi$ et $S = \{\phi_1, \phi_2, \ldots, \phi_k, \ldots, \phi_n\}$ Alors dériver $S' = \{\phi_1, \phi_2, \ldots, \psi, \ldots, \phi_n\}$.

Dans la section suivante, nous nous proposons d'expliciter les types de connaissances que devrait avoir un système de recherche d'informations. En particulier, nous montrons que la représentation des *propriétés des relations*, tout en étant nécessaire, permet de couvrir l'ensemble des connaissances habituellement considéré dans un système de recherche d'informations. Cela revient à étendre la relation d'endiguement classique sur les infons "$\rightarrow$", à une relation d'endiguement sur le relons "$\rightarrow_r$". Le contenu des connaissances du système spécifié, nous présentons les règles de dérivation permettant de prendre en compte ces relations.

3.3 Utilisation des Propriétés des Relations

À partir d'exemples tirés de cas réels de recherche, nous montrons dans cette section l'importance de considérer les propriétés des relations dans la base de connaissances. Généralement, il est plus facile de formaliser les propriétés des relations binaires. Cependant, même s'il est toujours théoriquement possible d'éclater une relation n-aire en un ensemble de relations binaires [Sow84], nous préférons dans ce chapitre rester à un stade général et manipuler ainsi toute sorte de relation utile en recherche d'informations. En effet, nous avons évoqué dans le chapitre précédent qu'il arrive que l'on soit amené pour des considérations pratiques à représenter des relations d'arité n. Nous avons besoin par exemple d'un certain nombre de relations tertiaires pour la représentation et la recherche des documents structurés et complexes [OC96b]. Pour représenter ces relations nous avons besoin d'au moins deux relations binaires par ailleurs difficile à gérer pour deux raisons:

- il faudra vérifier que toutes les propriétés des relations n-aires sont fidèlement véhiculées par leurs équivalents binaires.
- le nombre des relations augmente, rendant leurs traitements plus complexes et la gestion de leur cohérence plus délicate.

L'utilisation des relations n-aires dans un système opérationnel permettra de limiter le nombre des relations, de faciliter la gestion de la cohérence de la base de connaissances et enfin de limiter les opérations sur les données. Un meilleur comportement calculatoire est ainsi à prévoir.

Dans notre approche, seules les relations n-aires ayant un intérêt en recherche d'informations seront incluses dans le système. Comme les caractéristiques de ces relations sont connues et leur nombre est normalement restreint, il n'est pas difficile de les introduire dans la base de connaissances.

3.3.1 Dérivation de Connaissances Implicites

Nous énumérons ici les types des propriétés de relations dont l'utilisation permet de favoriser certaines prises de décisions de pertinence. Les termes $e_1, \ldots, e_n$ désignent des éléments quelconques de *Param* ou de $\mathcal{L}$.

Les Propriétés Mathématiques

Les propriétés mathématiques peuvent être utilisées pour augmenter l'index par des relations implicitement dérivées des connaissances du système. La prise en compte de ces relations peut améliorer le taux de rappel du système lors du processus de correspondance entre les documents et les requêtes.

Par exemple, considérons une image contenant trois objets o_1, o_2 et o_3. Supposons que l'index de cet image est donné par la situation $S = \{\langle\langle couvre,\ o_1,\ o_2\ ;1\rangle\rangle,\ \langle\langle couvre,\ o_2,\ o_3\ ;1\rangle\rangle\}$. Un nouveau relon $\langle\langle couvre,\ o_1,\ o_3\ ;1\rangle\rangle$peut être ajouté à la situation S de part la transitivité de la relation "*couvre*".

En effet, dans l'espace euclidien, un objet o_1 *couvre* un autre objet o_2 si, et seulement si, tous les points de l'objet o_1 sont inclus dans les points de l'objet o_2, i.e. point$(o_1) \subset$ point(o_2). Cette

relation spatiale est transitive comme cela a été mentionné dans [Mec95c].

De façon plus générale, une première propriété pouvant être associée à certaines relations serait la *Séquentialité*:

Séquentialité

$$\frac{\langle\langle R,e_1,e_2,\ldots,e_k;\, i\rangle\rangle \odot \langle\langle R,e_k,e_{k+1}\ldots,e_{k+m};\, i\rangle\rangle}{\langle\langle R,\ldots e^{n-1}\ldots,e_{k+m};\, i\rangle\rangle}$$

Si R est une relation n-aire, e^{n-1} dénote une combinaison de $n \Leftrightarrow 1$ termes choisis de l'ensemble $\{e_1, \ldots, e_{k+m-1}\}$. Nous pouvons ainsi avoir $n \Leftrightarrow 1$ dérivations possibles, selon la combinaison de termes considérée.

Exemple 13 *Dans le cas d'une relation tertiaire R satisfaisant la propriété de* Séquentialité, *les deux dérivations suivantes sont valides:*

Cas des Relations Tertiaires

$$\frac{\langle\langle R,e_1,e_1,e_3;\, i\rangle\rangle \odot \langle\langle R,e_3,e_2,e_4;\, i\rangle\rangle}{\langle\langle R,e_2,e_1,e_4;\, i\rangle\rangle}$$

ou

$$\frac{\langle\langle R,e_2,e_1,e_3;\, i\rangle\rangle \odot \langle\langle R,e_3,e_2,e_4;\, i\rangle\rangle}{\langle\langle R,e_3,e_1,e_4;\, i\rangle\rangle}$$

Dans un espace 2D, la relation spatiale "entre" *satisfait les deux dérivations ci-dessus. Ainsi, si dans une image il est vrai qu'un objet o_2 est* entre *les objets o_1 et o_3 et que l'objet o_3 est* entre *les objets o_2 et o_4, alors cette image est pertinente pour une requête recherchant toutes les images dans lesquelles apparaît un objet o_2 entouré des deux objets o_1 et o_4.*

La propriété de *transitivité* est un cas particulier de la *séquentialité* dans le cas des relations binaires:

Transitivité

$$\frac{\langle\langle R,e_1,e_2,;\, 1\rangle\rangle \odot \langle\langle R,e_2,e_3;\, 1\rangle\rangle}{\langle\langle R,e_1,e_3;\, 1\rangle\rangle}$$

En recherche d'informations, plusieurs relations transitives sont utilisées pour l'indexation des documents. Par exemple, les relations d'agrégation permettent de décrire comment sont composés des éléments complexes en fonction d'autres éléments plus simples: la section est une partie du chapitre, qui est lui-même une partie du document, ... Par ailleurs les relations de succession (la section s_2 *suit* la section s_1) permettent de spécifier l'ordre dans lequel il faut lire un document. Cet ordre est utile pour naviguer dans des documents hypermédia. Ces relations sont toutes transitives, et il est utile que la fonction de correspondance tienne compte de cette caractéristique.

Une deuxième propriété de relation, pouvant influencer les décisions de pertinence dans un processus de recherche est la suivante:

Permutation

$$\frac{\langle\langle R, e_1, e_2, \ldots, e_i, \ldots, e_j, \ldots, e_k; i\rangle\rangle}{\langle\langle R, e_1, e_2, \ldots, e_j, \ldots, e_i, \ldots, e_k; i\rangle\rangle}$$

La *permutation* exprime le fait que l'ordre dans lequel sont reliés certains objets dans la relation n'a pas d'influence sur la sémantique véhiculée par cette dernière.

Dans le cas des relations binaires, cette propriété correspond à la notion mathématique classique de *Symétrie* ($R(x, y) \Leftrightarrow R(y, x)$). Par exemple, la relation *marié-à* est symétrique.

Les relations de proximité (a est *loin* de b ou a est *près* de b) utiles dans l'indexation des images sont symétriques. Si une requête s'intéresse à des images contenant un objet b éloigné d'un objet a, i.e.,$\langle\langle loin, b, a; 1\rangle\rangle$, alors le document image indexé par $\langle\langle loin, a, b; 1\rangle\rangle$ est pertinent pour cette requête.

Les Agencements entre Relations

Des relations peuvent être combinées suivant certaines règles de composition pour dériver des informations implicitement contenues dans le document.

Soit, par exemple, le cas de la figure 3.3 suivante. Le document est indexé par une arborescence sémantique du type de celle utilisée dans le système RIME. Supposons qu'un utilisateur est intéressé par tous les documents à-propos des *maladies touchant le lobe*.

Comme les termes *lobe* et *lobe droit* sont plus liés par une relation de composition *partie-de* que par une relation de spécialisation, le document de cette figure ne risque pas d'être retrouvé par la requête. En effet, si à partir de la description du document, il est possible de déduire que l'*opacité pulmonaire* touche le *lobe droit*, rien ne permet de conclure qu'elle touche également le *lobe*.

Pourtant, le fait qu'un élément a est *localisé* dans un élément b *faisant partie* d'un élément c, implique nécessairement que a est également *localisé* dans c. Cette information implicite ne peut être déduite par le système que s'il a la connaissance que la relation a-pour-loc est transitive à travers la relation "*partie-de*" [16]. Dans la figure 3.3, on voit un arc en pointillé montrant ce fait.

Peu de formalismes permettent la prise en compte de telles combinaisons de relations. Le système terminologique GRAIL [GBS+94], entièrement dédié au domaine médical constitue une exception notable à ce niveau. L'absence de ce type de traitements entraînerait un taux de rappel faible pour certaines requêtes, c'est-à-dire que plusieurs documents pertinents ne seront pas renvoyés à l'utilisateur.

Plusieurs travaux [PL94] ont ainsi mentionné l'importance de ce type d'agencement entre relations. Les applications traitant des objets physiquement composés, à l'image de l'anatomie humaine, nécessitent particulièrement la représentation et la gestion de ce type de propriétés.

Certaines relations, combinées ensembles, véhiculent une sémantique implicite pouvant favoriser la prise d'une décision de pertinence. La base de connaissances du système doit donc spécifier quelles

16. En anglais, "the relation *a-pour-loc* is transitive across the *partie-de* relation [Hor97a]"

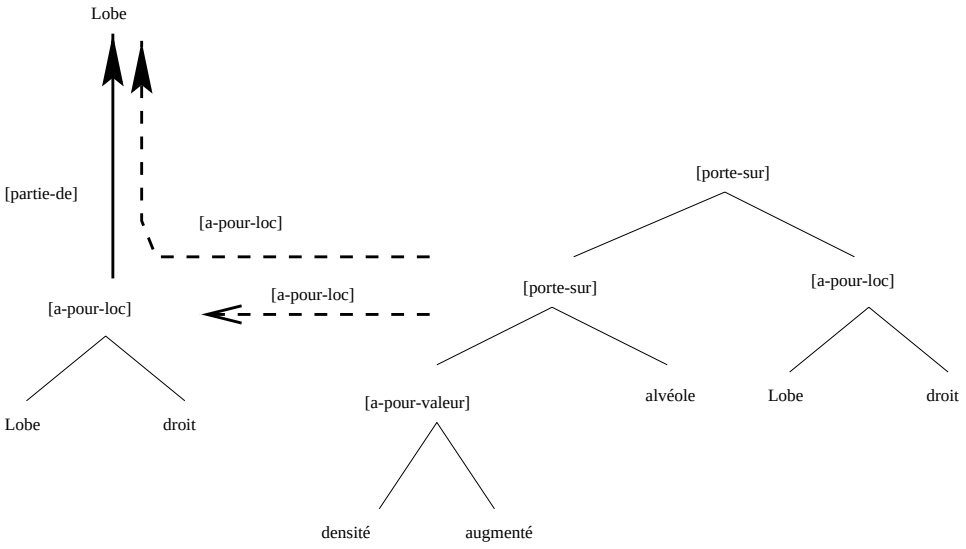


Figure 3.3. Un document à-propos d'une "*opacité alvéolaire au niveau du lobe droit*" implique que cette maladie touche également le "*lobe*"

relations répondent à cette caractéristique et de quelle façon ces dernières peuvent être combinées. La propriété de *juxtaposition* exprime plusieurs manières de le faire:

Juxtaposition

$$\frac{\langle\langle R_1,e_i,\ldots,e_k;i\rangle\rangle \odot \langle\langle R_2,e_k,\ldots e_s;i\rangle\rangle \odot \ldots \langle\langle R_n,e_s,\ldots,e_n;i\rangle\rangle}{\langle\langle R_p,e_i,\ldots,e_k,\ldots,e_s,\ldots,e_n;i\rangle\rangle}$$

Exemple 14 *La recherche des documents textuels structurés [Par96], nécessite l'introduction d'une relation de séquence transitive* "séq-trans"*, permettant de spécifier lequel de deux éléments donnés (mots, paragraphes, ...) apparaît avant l'autre dans le document. De même, ce type de recherche nécessite l'introduction d'une relation de composition transitive* "part-trans"*, spécifiant qu'une information de contenu peut être formée à partir d'autres informations. Par exemple, "recherche d'informations" est composée de «recherche», de «d'» et de «informations». Les deux relations* "séq-trans" *et* "part-trans" *vérifient la propriété de juxtaposition suivante:*

Ordonnancement Logique

$$\frac{\langle\langle \text{part-trans},e_1,e_2;1\rangle\rangle \odot \langle\langle \text{séq-trans},e_1,e_3;1\rangle\rangle \odot \langle\langle \text{part-trans},e_2,e_3;0\rangle\rangle}{\langle\langle \text{séq-trans},e_1,e_3;1\rangle\rangle}$$

Cette règle de juxtaposition exprime le fait que toutes les informations qui sont en séquence avec leurs parents, le sont aussi avec leurs fils,

Dans le cas des relations binaires, la règle de juxtaposition revient à la relation de composition

classique:

Composition

$$\frac{\langle\langle R1,x,y;1\rangle\rangle \odot \langle\langle R2,y,z;1\rangle\rangle}{\langle\langle R3,x,z;1\rangle\rangle}$$

Exemple 15 *Considérons un document à-propos* "des effets de l'intégrisme sur la société". *Supposons que son index* $\chi(d)$ *se résume à la situation* $S = \{\langle\langle$incite-à,INTÉGRISME,HAINE$;1\rangle\rangle$, $\langle\langle$cause, HAINE, VIOLENCE $;1\rangle\rangle\}$. *Ce document devra être renvoyé à un utilisateur intéressé par les* "facteurs de la violence", *i.e.,* $\langle\langle$est-un-facteur-de,p,VIOLENCE$;1\rangle\rangle$. *La relation de pertinence est établie à la base de la possibilité d'agencement entre les relations* "incite-à" *et* "cause", *puis du fait que* p *peut désigner n'importe quel objet (relation de spécialisation/généralisation).*

Example

$$\frac{\langle\langle \text{Incite-à,INTÉGRISME,HAINE};1\rangle\rangle \odot \langle\langle \text{cause,HAINE,VIOLENCE};1\rangle\rangle}{\langle\langle \text{est-un-facteur-de,INTÉGRISME,VIOLENCE};1\rangle\rangle}$$

Certaines relations comme nous l'avons vu dans l'exemple de la figure 3.3 vérifient plutôt la propriété de composition stricte. Nous distinguons les deux cas suivants:

Composition Stricte à Droite

$$\frac{\langle\langle R1,x,y;1\rangle\rangle \odot \langle\langle R2,y,z;1\rangle\rangle}{\langle\langle R2,x,z;1\rangle\rangle}$$

Composition Stricte à Gauche

$$\frac{\langle\langle R1,x,y;1\rangle\rangle \odot \langle\langle R2,y,z;1\rangle\rangle}{\langle\langle R1,x,z;1\rangle\rangle}$$

Ces deux propriétés constituent des cas particuliers de la règle de *Composition*. En termes d'Intelligence Artificielle, nous disons que dans le second cas, la relation R_1 est transitive à travers la relation R_2.

Exemple 16 *Dans le cas de la figure 3.3, la règle suivante est une partie intégrante de la base de connaissances:*

Composition Sémantique

$$\frac{\langle\langle \text{a-pour-loc},_{o_1,o_2};1\rangle\rangle \odot \langle\langle \text{partie-de},_{o_2,o_3};1\rangle\rangle}{\langle\langle \text{a-pour-loc},_{o_1,o_3};1\rangle\rangle}$$

De même, la recherche de documents hypermédia nécessite la prise en compte des liens sémantiques entre documents [Khe95]. La propriété ci-dessous donnée exprime le fait qu'un terme d'indexation t *décrivant le contenu sémantique d'un document* d_1 *décrit également le contenu sémantique de tous les documents* d_i *qui lui sont sémantiquement proches. Une requête* q *à-propos du terme* t, *retrouvera ainsi, non seulement* d_1, *mais aussi tous les documents qui lui sont sémantiquement liés:*

Composition Logique

$$\frac{\langle\langle \text{sémantiqument-proche-de},_{d_i,d_1};1\rangle\rangle \odot \langle\langle \text{contenu-sémantique},_{d_1,t};1\rangle\rangle}{\langle\langle \text{contenu-sémantique},_{d_i,t};1\rangle\rangle}$$

Liens Sémantiques entre Relations

De façon similaire aux raisonnements sur les mots-clés, les relations peuvent être classées et liées entre elles. Comme pour les mots-clés, nous distinguons deux grands types de liens: des liens hiérarchiques, et des liens non-hiérarchiques [SDV95]. Les liens hiérarchiques incluent les diverses relations de spécialisation, permettant la définition des taxinomies. Les liens non-hiérarchiques comprennent essentiellement les synonymes et les antonymes, les inversions, etc.

Commençons tout d'abord avec les liens non-hiérarchiques. La première propriété qui peut lier deux relations données dans ce type de classification est la notion d'*Alias*. Elle exprime le fait que ces deux relations ont la même extension dans le domaine et qu'elles sont ainsi logiquement équivalentes.

Alias

$$\frac{\langle\langle R1,e_1,\ldots e_n;\, i\rangle\rangle}{\langle\langle R2,e_1,\ldots,e_n;\, i\rangle\rangle}$$

En recherche d'informations, comme pour les mots-clés, cette propriété est utile pour éviter un taux élevé de silence quand la requête est exprimée par des relations différentes de celles utilisées dans l'indexation, mais qui lui sont néanmoins équivalentes.

Exemple 17 *Une requête cherchant des images dans lesquelles apparaît* "un homme habillé d'un costume blanc"*, devra retrouver une image décrite comme étant à-propos d'*"un homme vêtu de blanc"*. En effet, la relation* "vêtu-de" *est un synonyme de la relation* "habillé-de"*:*

Synonymie

$$\frac{\langle\langle \text{vêtu-de},\textsc{Homme},\textsc{Costume};\, 1\rangle\rangle \odot \langle\langle \text{couleur},\textsc{Costume},\textsc{Blanc};\, 1\rangle\rangle}{\langle\langle \text{habillé-de},\textsc{Homme},\textsc{Costume};\, 1\rangle\rangle \odot \langle\langle \text{couleur},\textsc{Costume},\textsc{Blanc};\, 1\rangle\rangle}$$

Un cas assez intéressant de lien non-hiérarchique entre relations est présenté dans la propriété d'*inversion* suivante:

Inversion

$$\frac{\langle\langle R1,e_1,e_2,\ldots,e_n;\, i\rangle\rangle}{\langle\langle R2,e_n,\ldots,e_2,e_1;\, i\rangle\rangle}$$

Cette règle exprime le fait que les deux relations R_1 et R_2 sont égales quand leurs arguments sont échangés dans leur ordre inverse d'apparition. Ce type de lien entre relations est souvent présent dans les langages de représentation, surtout dans le cas des relations binaires. Par exemple, certaines logiques terminologiques assez expressives comme LOOM [HKNP94] introduisent un opérateur spécial d'inversion de relation, noté **inv**. Cet opérateur appliqué à un rôle r donné (une relation binaire) produit une inversion de ses arguments et dénote ainsi exactement son inverse r^{-1}. Par contre, le formalisme des graphes conceptuels n'offre pas, du moins dans le modèle de base de Sowa [Sow84], une telle possibilité. L'existence dans la réalité d'une multitude de relations inverses empêche d'ignorer cette caractéristique et impose sa prise en compte par le système.

En Recherche d'informations, ce lien entre relations est très utile que ce soit pour la fonction de correspondance ou pour la représentation des connaissances du domaine tout court.

Exemple 18 *Supposons par exemple, qu'une image contienne le terme d'indexation* $\langle\langle$a-droite, MAISON, ARBRE; 1$\rangle\rangle$. *Cette image est pertinente pour une requête recherchant des* arbres à gauche d'une maison*:*

Une Inversion Possible

$$\frac{\langle\langle \text{a-droite}, \textsc{Maison}, \textsc{Arbre}; 1\rangle\rangle}{\langle\langle \text{a-gauche}, \textsc{Arbre}, \textsc{Maison}; 1\rangle\rangle}$$

Un autre type de lien non-hiérarchique est exprimé par la propriété de *Simultanéité*. Soient par exemple, n et m les arités respectives de deux relations R_1 et R_2. Soient $i_1, i_2, \ldots, i_{n-1}$ et $j_1, j_2, \ldots, j_{m-1}$ un ensemble de termes d'indexation. La propriété de *Simultanéité* expose le fait que chaque expression e prenant part à la relation R_1 par $R_1(e, i_1, i_2, \ldots, i_{n-1})$, prendra nécessairement part à la relation R_2 par $R_2(e, j_1, j_2, \ldots, j_{m-1})$ et vice-versa:

Simultanéité

$$\frac{\langle\langle_{R1,e,i_1,\ldots,i_{n-1}}; i\rangle\rangle}{\langle\langle_{R2,e,j_1,\ldots,j_{m-1}}; i\rangle\rangle}$$

Exemple 19 *Si le sens véhiculé par le relon* $\langle\langle$parents, *x, y, z* ; 1$\rangle\rangle$*consiste en* "les parents de x et y sont z, y étant la mère et z étant le père"*, alors un document à-propos de* $\langle\langle$parents, *Jean, Cathy, Joe* ; 1$\rangle\rangle$*est également à-propos de* $\langle\langle$a-pour-père, *Jean, Joe* ; 1$\rangle\rangle$.

Exemple de Simultanéité

$$\frac{\langle\langle \text{parents}_{,Jean,Cathy,Joe}; 1\rangle\rangle}{\langle\langle \text{a-pour-père}_{,Jean,Joe}; 1\rangle\rangle}$$

Comme pour les mots-clés, il arrive que deux relations s'excluent mutuellement. Le lien de préclusion exprime que les dénotations de deux relations R_1 et R_2, ayant la même arité n ont une intersection vide. Il n'existe pas ainsi, un ensemble de termes $e_1, e_2, \ldots, e_n$ qui est à la fois un tuple de R_1 et de R_2. En recherche d'informations, ce type de lien est intéressant surtout pour exprimer la notion de non-pertinence [Hui96, pp54].

Préclusion

$$\frac{\langle\langle_{R1,e_1,e_2,\ldots,e_n}; 1\rangle\rangle}{\langle\langle_{R2,e_1,e_2,\ldots,e_n}; 0\rangle\rangle}$$

Exemple 20 *Un document à-propos des horaires des trains avec réservation supporte le relon* $\langle\langle$*avec,* TRAIN, RÉSERVATION; 1$\rangle\rangle$. *On en déduit que ce document n'est pas pertinent pour une requête utilisateur cherchant* les horaires des trains sans réservation obligatoire, *i.e.,* $\langle\langle$*sans,* TRAIN, RÉSERVATION; 1$\rangle\rangle$. *Dans ce cas, les relations* "avec" *et* "sans" *s'excluent mutuellement:*

Exemple de Préclusion

$$\frac{\langle\langle \textit{avec}, \textsc{Train}, \textsc{Réservation}; 1\rangle\rangle}{\langle\langle \textit{sans}, \textsc{Train}, \textsc{Réservation}; 0\rangle\rangle}$$

Les relations peuvent être liées par des liens hiérarchiques. Ces derniers incluent en premier lieu la relation de spécialisation, permettant de définition des taxinomies. Dans notre modèle, le lien de spécialisation ne relie pas nécessairement des relations ayant la même arité n. Notre choix est motivé par le fait que plusieurs relations peuvent être hiérarchiquement liées sans pour autant qu'elles soient de même arité. Ainsi la relation tertiaire spatiale "*entre*", est une spécialisation des relations binaires "a-droite" et "a-gauche", dans un espace euclidien.

Intuitivement une condition préalable à ce qu'une relation m-aire soit une spécialisation d'une relation n-aire est que l'égalité $m \geq n$ soit vérifiée. En effet, une relation d'arité supérieure indique la présence d'une information supplémentaire. La spécialisation pourrait être ainsi vue comme une inclusion entre une projection des tuples de la relation m-aire et ceux de la relation n-aire.

Spécialisation

$$\frac{\langle\langle R1, e_1, e_2, \ldots, e_n; 1\rangle\rangle}{\langle\langle R2, e_1, e_2, \ldots e_m; 1\rangle\rangle} \qquad \frac{\langle\langle R2, e_1, e_2, \ldots e_m; 0\rangle\rangle}{\langle\langle R1, e_1, e_2, \ldots, e_n; 0\rangle\rangle}$$

Exemple 21 *Si un document est à-propos d'une situation dans laquelle* "John est un enfant de Mary*", alors ce document est également à-propos d'une requête cherchant toutes les situations dans lesquelles* "John et Mary sont parents*":*

Exemple de Spécialisation

$$\frac{\langle\langle \text{enfant}, \textit{John}, \textit{Mary}; 1\rangle\rangle}{\langle\langle \text{parent}, \textit{John}, \textit{Mary}; 1\rangle\rangle}$$

Si nous supposons que le relon $\langle\langle$entre, *X, Y, Z* ; 1$\rangle\rangle$*stipule que* "X est entre Y et Z*", un exemple de lien de spécialisation entre relations d'arités différentes pourrait être le suivant: une image supportant le relon* $\langle\langle$*entre,* FONTAINE, BÂTIMENT, PORTE-ENTREE; 1$\rangle\rangle$*, supportera le relon* $\langle\langle$*a-droite,* FONTAINE, BÂTIMENT; 1$\rangle\rangle$*mais aussi le relon* $\langle\langle$*a-gauche,* FONTAINE, PORTE-ENTREE; 1$\rangle\rangle$*.*

Arités Différentes

$$\frac{\langle\langle \textit{entre}, \text{FONTAINE}, \text{BÂTIMENT}, \text{PORTE-ENTREE}; 1\rangle\rangle}{\langle\langle \textit{a-droite}, \text{FONTAINE}, \text{BÂTIMENT}; 1\rangle\rangle} \quad \frac{\langle\langle \textit{entre}, \text{FONTAINE}, \text{BÂTIMENT}, \text{PORTE-ENTREE}; 1\rangle\rangle}{\langle\langle \textit{a-gauche}, \text{FONTAINE}, \text{PORTE-ENTREE}; 1\rangle\rangle}$$

Propriétés Basées sur les Objets

A l'inverse, des propriétés présentées dans les 3 sections précédentes et qui dépendent des relations, nous présentons ici des caractéristiques liées aux objets reliés par ces relations.

Le fait de créer une relation entre des objets implique nécessairement l'existence de ces objets. En recherche d'informations, où la notion d'inférence est généralement basée sur une relation de spécialisation, il est possible d'inférer le concept "PERSONNE" du relon $\langle\langle$*et*, PERSONNE, CHAT; 1$\rangle\rangle$. En effet, le premier est *endigué* dans le second. Dans un système comme RIME, ceci est exprimé par la règle suivante:

$$[\textit{Relation}](A, B) \Rightarrow A \quad [\textit{Relation}](A, B) \Rightarrow B$$

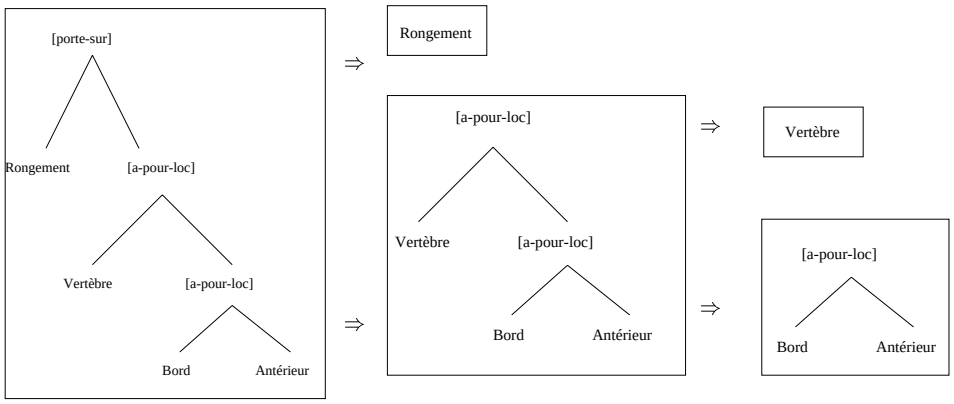


Figure 3.4. Exemple de relation d'endiguement dans RIME. (tiré de [Nie90, page 155])

En termes d'infons, cette dernière règle revient à l'énoncé de la propriété suivante de la relation d'endiguement $\rightarrow$:

$$\forall i \in [1..n],\ \langle\langle relation, \phi_1, \phi_2, \ldots, \phi_i, \ldots, \phi_n; 1\rangle\rangle \rightarrow \phi_i$$

Ainsi, la propriété suivante est valide dans notre modèle d'indexation relationnelle, $e_1, e_2, \ldots, e_n$ désignent n'importe quel terme d'indexation du langage de la définition 3.2.1:

Endiguement (ED)

$$\frac{\langle\langle relation, e_1, e_2, \ldots, e_i, \ldots, e_n; 1\rangle\rangle}{e_i}$$

Ainsi, même si notre modèle d'indexation ne permet pas d'indexer les documents par des mots-clés ou des concepts isolés, il autorise cependant leur recherche.

Exemple 22 *Le système de recherche d'informations médicales* RIME *se base sur la relation d'endiguement*[17] *pour la prise de certaines décisions de pertinence. Dans la figure 3.4, la présence d'un* "rongement au bord antérieur de la vertèbre"*, sous-tend la présence dans le document de* "rongement" *et du* "bord antérieur de la vertèbre"*. Une requête à-propos de l'un de ces deux termes devra retrouver le document en question. Dans notre modèle, la chaîne de dérivation de la figure 3.5 est valide, grâce à l'utilisation successive de la relation d'endiguement (ED).*

Une propriété sur les relations assez répandue dans les formalismes de représentation des connaissances est celle de la *restriction*. Elle stipule qu'il est possible d'effectuer une dérivation logique sur un infon en remplaçant un de ses termes par son générique.

Restriction

$$\frac{\langle\langle relation, e_1, e_2, \ldots, e_i, \ldots, e_n; 1\rangle\rangle \quad e_i \rightarrow e_j}{\langle\langle relation, e_1, e_2, \ldots, e_j, \ldots, e_n; 1\rangle\rangle}$$

17. Dans [Nie90], la relation d'endiguement est appelée *relation de dérivation*.

$$\dfrac{\dfrac{\dfrac{\langle\langle \textit{porte-sur}_{,Rongement},\langle\langle \textit{a-pour-loc}_{,Vertbre},\langle\langle \textit{a-pour-loc}_{,Bord,Antrieur};1\rangle\rangle;1\rangle\rangle;1\rangle\rangle}{\langle\langle \textit{a-pour-loc}_{,Vertbre},\langle\langle \textit{a-pour-loc}_{,Bord,Antrieur};1\rangle\rangle;1\rangle\rangle \quad Rongement}\,ED}{\langle\langle \textit{a-pour-loc}_{,Bord,Antrieur};1\rangle\rangle \quad Vertbre}\,ED}{Bord \quad Antrieur}\,ED$$

Figure 3.5. Utilisation de la relation d'endiguement pour la dérivation des connaissances implicites

Exemple 23 *Une image représentant* une femme assise *sera jugée pertinente pour une requête recherchant toutes les images dans lesquelles* des personnes assises *apparaissent.*

Exemple de Restriction

$$\dfrac{\langle\langle \text{position}_{,\textsc{Femme},\textsc{Assis}};1\rangle\rangle \quad \textsc{Femme}\rightarrow\textsc{Personne}}{\langle\langle \text{position}_{,\textsc{Personne},\textsc{Assis}};1\rangle\rangle}$$

Dans ce cas de figure, le document est une restriction *de la requête. Dans un formalisme comme les graphes conceptuels, cette dérivation est correcte. On dit que le document reprend les thèmes de la requête à une restriction près (en l'occurrence, il s'agit ici de la restriction de* Personne *à* Femme*).*

Cette façon de dériver les connaissances implicites est pratiquement présente dans tous les formalismes de représentation. Elle est basée sur un raisonnement sur les objets (concepts) plutôt que celui sur les relations. Par exemple, le système Rime la présente comme *une substitution d'arborescences sémantique*:

$$[\textit{relation}](A,B) \Rightarrow [\textit{relation}](A',B) \quad \text{si} A \Longrightarrow A'$$

Rôle fonctionnel des Relations

Les propriétés que nous avons énuméré dans les sections précédentes sont nécessaires pour assurer un bon comportement de la fonction de correspondance lors du processus de recherche. Elles forment les connaissances du domaine que cela implique les informations classiques sur les hiérarchies des objets et des relations ou encore la façon dont ces connaissances peuvent être combinées syntaxiquement et sémantiquement pour inclure d'autres informations implicites. Cependant en recherche d'informations, nous avons parfois tendance à spécifier des caractéristiques de relations au niveau de l'indexation. Cette tendance s'est accélérée récemment avec l'avènement des nouvelles applications telles que la recherchede documents structurés ou encore la recherche de documents hypermédias.

Par exemple, il arrive que les relations aient un comportement de fonctions mathématiques. Dans le travail de Chiaramella et al. [CFM96], il a été montré qu'une base formelle pour la représentation et la recherche de documents structurés nécessite l'introduction de certaines fonctions spécifiques. Cette représentation n'est toutefois pas envisageable par le biais des formalismes de représentation des connaissances. Ainsi, il n'est pas possible d'inclure la notion d'attribut calculé. Cette dernière

permettra l'intégration des divers calculs proposés sur les données, et est utile dans le cadre de la gestion des données. Pour remédier à ce problème, ces fonctions sont représentées par des relations liant les opérandes à leur résultat. Ces relations, associées à des fonctions, doivent parfois satisfaire des propriétés comme l'injectivité, la surjectivité, la bijection, etc. D'une façon plus formelle, les contraintes logiques suivantes peuvent être vérifiées par certaines relations:

- $\nexists i_1, \ldots, i_n, j_1, \ldots, j_n, e$ tel que $R(i_1, \ldots, i_n, e) \wedge R(j_1, \ldots, j_n, e)$
- $\nexists i_1, \ldots, i_n, e_1, e_2$ tel que $R(i_1, \ldots, i_n, e_1) \wedge R(i_1, \ldots, i_n, e_2)$
- $\nexists e, i_1, \ldots, i_n, j_1, \ldots, j_m$ tel que $R(e, i_1, \ldots, i_n) \wedge R(e, j_1, \ldots, j_m)$

La première contrainte peut être exprimée par la propriété suivante:

Correspondance Unique

$$\frac{\langle\langle_{R,i_1,\ldots,i_n,e}; 1\rangle\rangle}{\langle\langle_{R,j_1,\ldots,j_n,e}; 0\rangle\rangle}$$

La propriété de *Correspondance Unique* reflète la notion classique d'une fonction injective:

$$f \text{ est injective ssi } \forall x_1, x_2, x_3 (f(x_2, x_1) \wedge f(x_3, x_1)) \rightarrow x_2 = x_3$$

Exemple 24 *Pour les besoins de la recherche de documents textuels structurés, une relation de composition injective* "part" *a été introduite dans [Par96]. Elle permet de grouper des éléments textuels en différenciant principalement les descendants directs de ceux qui sont plus bas dans la hiérarchie des composants (la relation* "part" *est non transitive).*

La phrase (p) contenant l'énoncé «recherche d'informations» est composée de «recherche», de «d'» et de «informations» : part(p,«recherche»), part(p,«d'») et part(p, «informations»). La condition d'injectivité exprime le fait qu'il n'est pas possible qu'un élément soit un descendent direct de plus d'un composant, ou de façon plus simple qu'aucun élément n'a plus d'un père dans la hiérarchie de composition textuelle:

La Relation "*part*"

$$\frac{\langle\langle_{\text{part},p_1,e_1}; 1\rangle\rangle}{\langle\langle_{\text{part},p_2,e_1}; 0\rangle\rangle}$$

La seconde contrainte correspond à la notion de fonction surjective, elle peut être exprimée par le postulat suivant:

Exclusivité à Gauche

$$\frac{\langle\langle_{R,i_1,\ldots,i_n,e_1}; 1\rangle\rangle}{\langle\langle_{R,i_1,\ldots,i_n,e_2}; 0\rangle\rangle}$$

Par exemple, dans une hiérarchie de types, l'opération de calcul de la borne inférieure d'un ensemble de sous-types vérifie la contrainte d'*Exclusivité à Gauche*.

Enfin, la dernière contrainte logique peut être résumée par la propriété d'*Exclusivité à Droite* suivante:

Exclusivité à Droite

$$\frac{\langle\langle_{R,e,i_1,\ldots,i_n};1\rangle\rangle}{\langle\langle_{R,e,j_1,\ldots,j_m};0\rangle\rangle}$$

Les relations vérifiant ce type de contrainte simulent des fonctions gérant la cohérence au niveau de la base de connaissances.

Exemple 25 *Supposons qu'un document multimédia structuré est composé d'un ensemble d'objets structurels. Il peut par exemple contenir du texte, des images, des graphiques et de la vidéo. Supposons que* $Desc_{str} : \mathcal{D} \rightarrow 2^{OS}$ *est une fonction qui, pour un document structurel d donné, retourne l'ensemble de tous ses composants, où* OS *est ensemble de tous les objets structurels du corpus. Nous avons alors le postulat suivant:*

Objets Composés

$$\frac{\langle\langle_{Desc_{str},d,i_1,\ldots,i_n};1\rangle\rangle}{\langle\langle_{Desc_{str},d,j_1,\ldots,j_m};0\rangle\rangle}$$

Les Méta-Propriétés

Les propriétés des relations présentent elles-mêmes des caractéristiques qu'il faudra considérer lors de la gestion des connaissances de l'application. Ainsi, la propriété d'*Alias* est par exemple symétrique: Si a est un alias de b, alors b est également un alias de a.

Nous proposons alors de représenter ces méta-propriétés par des faits de la forme suivante: $\langle\langle$*Méta-Prop*, NOM-PROPRIÉTÉ, CARACTÉRISTIQUE; $i\rangle\rangle$

où NOM-PROPRIÉTÉ désigne le nom d'une propriété, et CARACTÉRISTIQUE dénote une caractéristique que peut présenter cette relation.

Exemple 26 *La propriété* Alias *est réflexive, symétrique, et transitive. On écrit:*

$\langle\langle$Méta-Prop, Alias, Réflexive; 1$\rangle\rangle$

$\langle\langle$Méta-Prop, Alias, Symétrique; 1$\rangle\rangle$

$\langle\langle$Méta-Prop, Alias, Transitive; 1$\rangle\rangle$

Par contre, la propriété Inversion *est symétrique mais n'est ni réflexive, ni transitive. On écrit:*

$\langle\langle$Méta-Prop, Inversion, Symétrique; 1$\rangle\rangle$

$\langle\langle$Méta-Prop, Inversion, Réflexive; 0$\rangle\rangle$

$\langle\langle$Méta-Prop, Inversion, Transitive; 0$\rangle\rangle$

L'introduction des Méta-propriétés permet d'éviter une redondance dans la représentation des connaissances. Nous désignons par $\mathcal{R}_{\mathcal{P}}$ l'ensemble des méta-propriétés connues par le système.

3.3.2 Comment Gérer les Relations et leurs Propriétés en RI

Dans les sections précédentes nous avons montré des types de propriétés de relations pouvant présenter un intérêt en recherche d'informations. Ces propriétés sont basées sur une sélection de règles de dérivation possédant des caractéristiques intéressantes, telles que nous les avons rencontré dans les systèmes de recherche d'informations d'aujourd'hui [OC96a].

La définition des relations primitives utiles en recherche d'informations mérite une investigation plus détaillée afin d'aboutir à un système opérationnel. Ces relations sont intéressantes parce qu'elles sont porteuses de connaissances pouvant améliorer le résultat de l'opération de correspondance. Ainsi, seules les propriétés d'un ensemble spécifique de relations mérite d'être caractérisées et utilisées. Par exemple, Palmer dans [Pal90], mentionne que les verbes ne sont généralement pas porteurs de propriétés intéressantes, surtout pour l'indexation des documents. D'autre part, nous savons déjà que nous avons besoin d'un ensemble de relations d'agrégation pour l'indexation et la recherche des documents structurés [Par96, CFM96]. Une autre classe de relations utiles en recherche d'informations est exposée par Mechkour dans [Mec95c]. Douze relations spatiales furent alors proposées comme par exemple, *loin, près, touche, couvre, dans*, etc.

Classification des Relations

Il est nécessaire de définir les classes spécifiques des relations utiles en recherche d'informations. Ces classes serviront de base pour exprimer les propriétés des relations, leurs liens sémantiques et leurs comportements lorsqu'elles sont agencées avec d'autres relations. En effet, outre le fait qu'il est plus facile de déterminer les propriétés des relations au niveau d'une même classe, il s'avère que souvent seules les relations d'une même classe peuvent être liées et agencées ensemble. La définition des classes de relations facilitera donc la création de la base de connaissance de l'application.

Un parcours de la littérature en recherche d'informations montre qu'essentiellement ce sont les prépositions qui furent les plus utilisées comme relations dans les termes d'indexation [Lar94]. Par exemple, dans les travaux de Palmer, un terme d'indexation consiste en un groupe nominal, reflétant la structure syntaxique et les dépendances linguistiques des termes dans le texte. Les relations syntaxiques sont matérialisées essentiellement par des prépositions. Bruza dans [Bru93] avait également adopté la même approche à savoir que les expressions d'indexation sont construites à base de termes et de relations syntaxiques (ou connecteurs) entre ces termes. Il s'agit surtout des prépositions de la langue anglaise. Une classification de ce type de relations particulières peut avoir un intérêt pour notre approche.

Nous avons ainsi présenté dans [OH97], une organisation possible des prépositions de la langue anglaise (cf. table 3.1). Un travail semblable peut être fait sur les prépositions de langue française. Cette table a été construite à partir de plusieurs livres anglais à-propos des prépositions. Seules les prépositions que nous avons jugé utiles en recherche d'informations ont été considérées dans cette table. Notre jugement a été surtout motivé par leur haute fréquence d'apparition dans les collections tests de recherche d'informations comme par exemple CACM ou Cranfield. Selon leur sens en Anglais et la façon dont elles sont employées dans les phrases, nous avons distingué quatre classes de prépositions: *Temps, Place, Direction, Conjonctions.* Chacune de ces classes peut avoir des sous-classes. Par exemple la préposition *in* appartient à la classe *Position*, laquelle est une sous-classe de *Place*.

La classification ainsi élaborée, il est possible de préciser les propriétés des relations en se basant

sur la classe à laquelle elles appartiennent. Les agencements des relations sont alors à chercher au niveau d'une même classe. Selon l'information contenue dans le corpus et la nature des documents (textuel, multimédia, structurel, etc.), d'autres classes de relations peuvent être cernées et analysées. Par exemple, la classe des relations d'agrégation pour les documents structurés, ou encore la classe des relations spatiales pour les documents images.

Préposition	Temps	Place	Direction	Conjonctions
About	Point-Période			
Above		Place		
Across			Direction	
After	Point-Période			
A long way from		Place		
Along			Direction	
Among				
And				Similitude
As	Temps			Relation
At	Point	Exact		
Back to			Direction	
Before	Point-Période			
Behind		Place		
Below		Place		
Beside		Place		
Between		Position		
But				Opposition
By	Point-Point	Position		
Down			Direction	
For	Période			
From	Point-Période			
In	Période	Position		
In front of		Place		
In the middle of		Position		
Into			Direction	
Near		Area		
Next to		Position		
Not far from		Place		
Of		Place	Direction	
Off			Direction	
Opposite		Position		
On	Point	Position		
Onto			Direction	
Or				Alternatives
Out of			Direction	
Over			Direction	
Past			Direction	
Round			Direction	
So				Relation
Since	Point-Période			
Through			Direction	
To			Direction	
When	Question			
Where		Question		
While	Question			
With				Relation
Under		Place	Direction	
Until	Point-Période			
Up			Direction	

Table 3.1. Classification des Prépositions de la Langue Anglaise

L'intérêt de la classification est que nous pouvons nous baser sur les nombreux travaux effectués sur certains types particuliers de relations. C'est le cas par exemple des SGBD Géographique où les relations spatiales et leurs propriétés sont étudiées.

Au niveau de la recherche d'informations, les ambiguïtés des représentations, doivent être évitées, surtout dans le cadre d'une approche orientée vers la précision des réponses. Ainsi comme nous l'avons signalé dans le chapitre précédent, des relations peuvent avoir plusieurs sens possibles. Cela apparaît clairement dans la table 3.1, où certaines prépositions peuvent appartenir à des classes différentes. Le dictionnaire énumère ainsi 18 sens possibles à la préposition *at* [Swa94], même si probablement peu d'entre-eux sont intéressants en recherche d'informations. Afin de différentier ces sens, nous devons spécifier exactement quelle type de relation *at* nous utilisons[18]. Nous introduisons pour ce faire, la notion de *signature*.

Les Signatures des Relations

La signature d'une relation n-aire R, notée $\Sigma_R = (R, n, O_1, O_2, \ldots, O_n)$, spécifie les types d'objets qui peuvent être liés par cette relation, ou en d'autres termes sa sémantique. Dans ce cas, une hiérarchie sur les objets est nécessaire. Si un objet o_1 est une spécialisation (ou une sous-classe) d'un autre objet o_2 alors nous écrivons que $o_1 \leq o_2$. Pour être valide, chaque argument (objet) de l'infon doit être une instance ou une spécialisation du type correspondant dans la signature. Par exemple, la relation *Loc*[19] liant un objet à une place peut avoir $\Sigma_{Loc} = (Loc, 2, \top, \text{PLACE})$ comme signature, où $\top$ désigne la classe universelle de tous les individus du domaine. Ainsi la phrase " *Les Véhicules sont* **à** *la Station*" peut être représentée par $\langle\langle Loc, \text{VÉHICULE}, \text{STATION}; 1\rangle\rangle$, où VÉHICULE $\leq \top$ et STATION $\leq$ PLACE.

Chaque relation du domaine peut être spécifiée par sa sémantique, la classe de relations à laquelle elle appartient, ses propriétés et ses liens avec d'autres relations. On spécifiera notamment:

- Le nom de la relation, comme par exemple *Loc* pour la relation de *Localité*.
- La classe de la relation, comme *Exact* pour la relation *Loc*, cette dernière est considérée comme un sens possible à la préposition *at*.
- La signature de la relation (sa sémantique), par exemple $\Sigma_{Loc} = (Loc, 2, \top, \text{PLACE})$ pour la relation *Loc*.
- Un commentaire informel à-propos de l'utilisation de cette relation. Par exemple, "Une localité est une relation liant un Objet à une Place".
- Les propriétés mathématiques de la relation, par exemple la relation *Loc* peut satisfaire la *Transitivité*.
- Ses liens sémantiques avec les autres relations de sa classe, par exemple la relation *Loc* est une spécialisation de la relation *Place* (voir table 3.1).
- Ses agencements possibles avec d'autres relations (son comportement). Ces dernières ne faisant pas nécessairement partie de sa classe.

18. Cela revient à utiliser 18 noms de relations différents pour la préposition *at*, si tous ses sens s'avèrent utiles en recherche d'informations.
19. Il s'agit d'un sens particulier de la préposition *at*.

3.3.3 Extension de la Relation d'Endiguement

Les propriétés de relations présentées dans les sections précédentes étendent la notion d'endiguement des infons "$\rightarrow$" à une notion plus large incluant des relons "$\rightarrow_r$". En particulier nous avons la propriété suivante:

Propriété 2 *Si* $\phi \rightarrow \phi$ *Alors* $\phi \rightarrow_r \phi$

La nouvelle relation "$\rightarrow_r$" tient compte, aussi bien des propriétés des objets liés que celles des relations. Ces diverses propriétés constituent les connaissances nécessaires qui doivent être représentées par le système en vue d'une recherche efficace. En d'autres termes, la base de connaissances d'un système de recherche d'informations devra inclure en plus du thésaurus classique des termes, toutes les informations connues sur les relations et leurs propriétés.

Ces propriétés n'ont toutefois pas la même sémantique, et donc les mêmes effets, lorsqu'elles sont appliquées à des relons d'une situation donnée. Nous distinguons trois types de relations d'endiguement sur les relons:

Endiguement Équivalent: Quand le résultat de l'application d'une propriété est un relon logiquement équivalent au relon de départ, la situation obtenue en remplaçant ce relon par celui qui lui est équivalent produit une nouvelle situation logiquement équivalente à celle de départ. On parle alors d'*endiguement équivalent*.

Définition 3.3.1 (Équivalence de Relons) *Si un relon* ϕ *est obtenu à partir d'un autre relon* ψ *par l'application d'une des propriétés de* Permutation, *d'*Alias, *d'*Inversion, *et de* Préclusion *alors* ϕ *et* ψ *sont porteurs de la même information. On écrira* $\phi \equiv_r \psi$.

Ceci est la conséquence du fait que les règles de *Permutation*, d'*Alias*, d'*Inversion* et de *préclusion* sont symétriques et sont gérées en tant que telles par les méta-propriétés $\mathcal{R}_{\mathcal{P}}$:

$$\text{Si} \quad \psi \overset{p}{\Leftrightarrow}_r \phi \quad \Leftrightarrow \quad \phi \overset{p}{\Leftrightarrow}_r \psi$$

où $p \in \{\text{Permutation}, \text{Alias}, \text{Inversion}, \text{Préclusion}\}$.

Exemple 27 *Soit une situation* S *à-propos d'*une personne assise à gauche d'une table*:* $S = \{\langle\langle$a-gauche, PERSONNE, TABLE; 1$\rangle\rangle$, $\langle\langle$position, PERSONNE, ASSIS; 1$\rangle\rangle\}$. *Étant donné que* $\langle\langle$a-gauche, PERSONNE, TABLE ; 1$\rangle\rangle \equiv_r \langle\langle$a-droite, TABLE, PERSONNE; 1$\rangle\rangle$, *la situation* $T = \{\langle\langle$a-droite, TABLE, PERSONNE; 1$\rangle\rangle$, $\langle\langle$position, PERSONNE, ASSIS; 1$\rangle\rangle\}$, *à-propos d'*une table à droite d'une personne assise *est équivalente à* T *:* $S \equiv T$.

D'une manière plus générale:

Équivalence

$$\frac{\phi \equiv_r \psi}{S \cup \{\phi\} \equiv S \cup \{\psi\}}$$

Endiguement Faible: Par endiguement faible, nous entendons que dans une situation donnée S, un relon ψ est remplacé par un relon ϕ quand $\psi \rightarrow_r \phi$. Le remplacement s'opère dans des cas précis de dérivation, comme cela est introduit dans la définition suivante:

Définition 3.3.2 (Endiguement Faible) *Quand un relon ϕ est obtenu à partir d'un autre relon ψ par le biais des propriétés de* Spécialisation, *d'*Endiguement *ou de* Restriction, *alors ϕ est dit* faiblement endigué *dans ψ. Ce fait est traduit par l'écriture $\psi \rightarrow_r^f \phi$.*

L'information dérivée des propriétés des relations n'est pas ajoutée à la situation de départ, mais remplace celle à partir de laquelle elle a été inférée. Les propriétés associées à ce type d'endiguement constituent des règles de *dégradation* d'informations, c'est-à-dire que la situation S_d obtenue n'est pas logiquement équivalente à celle de départ. Cette situation est plus générale que celle du départ et n'est donc que partiellement contenue dans cette dernière. L'adjectif "*faible*" associé à ce type d'endiguement exprime ce fait: il y a un faible chevauchement sémantique[20] entre la situation de départ et celle obtenue après l'utilisation d'une propriété de type de celles mentionnées dans la définition ci-dessus. La notion d'endiguement faible peut être exprimée par la relation de subsomption suivante:

$$\text{Si} \quad \psi \quad \rightarrow_r^f \quad \phi \quad \text{alors} \quad \underbrace{S' \cup \{\psi\}}_{S} \sqsubseteq \underbrace{S' \cup \{\phi\}}_{S_d}$$

Exemple 28 *Soit l'exemple de la figure 3.4, et la situation S à-propos du* rongement du bord antérieur de la vertèbre d'une personne malade. *Dans ce cas $S = \{\langle\langle$*porte-sur, *Rongement,* $\langle\langle$a-pour-loc, *Vertèbre,* $\langle\langle$a-pour-loc, *Bord, Antérieur*$;1\rangle\rangle;1\rangle\rangle\}$. *L'application de la règle d'*endiguement *permet de déduire le singleton* $\{\langle\langle$a-pour-loc, *Vertèbre,* $\langle\langle$a-pour-loc, *Bord, Antérieur*$;1\rangle\rangle;1\rangle\rangle\}$ *qui est plus général que la situation de départ, étant donné qu'il subsume S.*

Endiguement avec Retenue: Dans ce cas, à l'inverse de l'endiguement faible, l'information dérivée d'un ensemble de relons est ajoutée à ce dernier. Le terme *retenue* exprime ce fait en précisant que les informations à partir desquelles la déduction s'est faite sont gardées dans la nouvelle situation obtenue. Cet ajout explicite des informations dérivées s'opère en conformité avec la définition suivante:

Définition 3.3.3 (Endiguement avec Retenue) *L'application des propriétés de* Juxtaposition *ou de* Simultanéité *implique l'ajout explicite des informations dérivées dans la situation de départ. Ainsi, si l'endiguement $\psi \rightarrow_r \phi$ est obtenu par application de l'une de ces deux propriétés, alors nous parlons d'*endiguement avec retenue. *On écrit $\psi \rightarrow_r^{\oplus} \phi$.*

Il s'agit plutôt de propriétés de *réécriture*, c'est-à-dire que la situation dérivée n'est autre qu'une mise en forme syntaxique différente de la situation de départ. Cette réécriture n'engendre aucun changement sémantique de l'information véhiculée par la situation originelle. Ainsi, à l'opposé de l'endiguement faible, la situation résultat S_d est logiquement équivalente à la situation de départ:

$$\text{Si} \quad \psi \quad \rightarrow_r^{\oplus} \quad \phi \quad \text{alors} \quad \underbrace{\underbrace{S' \cup \{\psi\}}_{S} \cup \{\phi\}}_{S_d} \equiv \underbrace{S' \cup \{\psi\}}_{S}$$

20. Par chevauchement sémantique nous entendons l'intersection des dénotations respectives des deux situations.

ou encore (pour la Juxtaposition)

$$\text{Si} \quad \psi \quad \odot\, \varphi \quad \rightarrow_r^{\oplus} \quad \phi \quad \text{alors} \quad \underbrace{\underbrace{S' \cup \{\psi\} \cup \{\varphi\}}_{S} \cup \{\phi\}}_{S_d} \equiv \underbrace{S' \cup \{\psi\} \cup \{\varphi\}}_{S}$$

Le chevauchement entre la situation de départ et celle dérivée par ce type d'endiguement est total. Plus exactement, les deux situations sont porteuses de la même sémantique.

Exemple 29 *La situation* $S = \{\langle\langle$a-droite, PERSONNE, TABLE; $1\rangle\rangle$, $\langle\langle$a-droite, TABLE, LIT; $1\rangle\rangle\}$ *est équivalente à la situation* T *dérivée par l'utilisation d'une juxtaposition, spécifiant que la relation "*a-droite*" est transitive:* $T = \{\langle\langle$a-droite, PERSONNE, TABLE; $1\rangle\rangle$, $\langle\langle$a-droite, TABLE, LIT; $1\rangle\rangle$, $\langle\langle$a-droite, PERSONNE, LIT; $1\rangle\rangle\}$. *Le relon* $\langle\langle$a-droite, PERSONNE, LIT; $1\rangle\rangle$ *ajouté à* S *ne change pas la sémantique de cette dernière, ni par une restriction quelconque, ni par une généralisation donnée.*

Le lecteur aura remarqué que l'endiguement équivalent est un type d'endiguement avec retenue et vice-versa. Cependant, si nous avons préféré les différencier dans notre classification, c'est que même s'ils engendrent tous les deux des situations équivalentes, il n'en reste pas moins qu'ils n'ont pas la même fonctionnalité. Ainsi, contrairement aux propriétés d'endiguement avec retenue constituant des règles de réécriture syntaxiques, les propriétés de l'endiguement équivalent sont elles des règles purement sémantiques. Dans notre modèle de dérivation de pertinence, nous leur associons chacune une règle de dérivation particulière (voir la section 3.4 suivante).

3.3.4 Les Relations de Préclusion

L'expression $i \perp_r j$ exprime le fait que le relon i contredit le relon j à la lumière de la relation r. Essentiellement, elle exprime le fait qu'il n'est pas possible de combiner ces deux relons, puisqu'ils s'excluent mutuellement. Il est ainsi naturel de considérer qu'un relon avec une polarité égale à 1 exclue le même relon avec une polarité 0. Ainsi, Landman [Lan86] considère que la préclusion est une notion fondamentale de la théorie de l'information.

En recherche d'information cette notion est très utile, puisqu'elle permet d'influencer la prise d'une décision de pertinence, surtout quand les caractéristiques des relations sont connues par le système. Par exemple, supposons qu'un document est caractérisé par l'infon $\langle\langle$*Battre*, Juventus, Ajax ; $1\rangle\rangle$, où il est question de la victoire de la "Juventus" face à l'"Ajax" en Football. En supposant que la relation *Battre* est anti-symétrique, nous pouvons dériver (par la règle de *permutation* par exemple) que le document contient également l'information $\langle\langle$*Battre*, Ajax, Juventus ; $0\rangle\rangle$. En d'autres termes, le relon du document exclue le relon $\langle\langle$*Battre*, Ajax, Juventus ; $1\rangle\rangle$. Ce document sera donc retrouvé par une requête cherchant des documents représentant des situations où "*L'Ajax n'a pas battu la Juventus*".

Certaines propriétés de relations que nous avons énuméré dans les sections précédentes peuvent parfois générer des relons en contradiction avec ceux déjà existants dans le document. Dans ce cas, soit le document a été mal indexé au départ, soit une propriété de relation de la base de connaissance n'est pas toujours valide. Selon le contexte de raisonnement adopté, nous distinguons une solution particulière à ce problème. Dans le cas d'un raisonnement *non-monotone*, nous retenons

la dernière information ajoutée tout en retirant les infons qui contredisent cette dernière dans l'index du document. Par contre, dans le cas d'un raisonnement *monotone*, seules les informations en cohérence avec celles déjà existantes dans l'index peuvent être ajoutées. Dans le cas d'un conflit, la nouvelle information n'est pas ajoutée mais tout simplement rejetée. Par exemple, comme la relation $\langle\langle \textit{Père-De}, a, b; 1\rangle\rangle$ exclue la relation $\langle\langle \textit{Enfant-De}, a, b; 1\rangle\rangle$, il sera impossible dans le cas d'une approche de raisonnement monotone d'ajouter une information comme $\langle\langle \textit{Enfant-De}, \text{John}, \text{Jack}; 1\rangle\rangle$, si $\langle\langle \textit{Père-De}, \text{John}, \text{Jack}; 1\rangle\rangle$ existe déjà dans l'index du document.

Nous pensons toutefois qu'un raisonnement *non-monotone* est plus adéquat pour notre problématique de recherche. D'autant plus que les connaissances à-propos des propriétés des relations et leurs agencements possibles sont valides et choisies avec le plus grand soin. S'il existe un conflit, cela est due essentiellement à une mauvaise indexation des documents. Le problème de la non-monotonicité sera détaillé dans la section suivante.

3.4 Un Modèle Relationnel de Dérivation de Pertinence

L'étude de l'impact des relations sur le processus de correspondance peut se faire par le biais d'un système logique de dérivation dont la sémantique est basée sur la notion de préférence présentée dans la section 3.2. Afin de mettre l'emphase sur le fait que les décisions de pertinence sont basées sur les relations et leurs propriétés, nous préférons introduire une notion de *pertinence relationnelle*, notée $\Box\!\!\!\!\rightsquigarrow_R$, au lieu de la relation de pertinence logique $\Box\!\!\!\!\rightsquigarrow$.

Définition 3.4.1 (Le Langage de Dérivation) *Étant donné, le langage d'indexation relationnelle $\mathcal{L}$, le langage de dérivation de pertinence $\mathcal{P}(\mathcal{L})$ est défini comme le plus petit ensemble tels que:*

1. *si $\varphi, \psi \in \mathcal{L}$ alors $\varphi \rightarrow_r \psi$, $\varphi \rightarrow_r^f \psi$, $\varphi \equiv_r \psi$, $\varphi \perp_r \psi$, $\varphi \not\perp_r \psi$, $\in \mathcal{P}(\mathcal{L})$*
2. *si $\phi_1, \phi_2, \ldots, \phi_n, \psi \in \mathcal{L}$ alors $\phi_1 \odot \phi_2 \odot \ldots \phi_n \rightarrow_r^{\oplus} \psi \in \mathcal{P}(\mathcal{L})$*
3. *si $S, T \in \mathcal{S}_{\mathcal{L}}$ alors $S \cup T$, $S \cap T$, $S \Box\!\!\!\!\rightsquigarrow_R T$, $S \equiv T \in \mathcal{P}(\mathcal{L})$*

Quand il n'y a pas d'ambiguïté sur le langage $\mathcal{L}$, nous écrirons simplement $\mathcal{P}$, au lieu de $\mathcal{P}(\mathcal{L})$.

Définition 3.4.2 (Le Système de Dérivation) *Soit le langage $\mathcal{P}$ défini dans 3.4.1. Un système de dérivation relationnelle $\mathcal{S}_{\Re}$ est un triplet $\langle \mathcal{P}, Ax, \textit{Règle}\rangle$ avec:*

- *Ax est un ensemble d'axiomes*
- *Règle est un ensemble de règles $\{R_1, \ldots, R_n\}$ de la forme $R(T_1, \ldots, T_k, T_{k+1})$, avec $\forall T_i, T_i \in \mathcal{P}$. Ici, $T_1, \ldots, T_k$ sont les prémisses de la règle et T_{k+1} sa conclusion.*

La représentation des règles suivantes est celle classiquement utilisée dans les systèmes logiques, i.e, $\boxed{\frac{A}{B}}$ signifie que si A est valide dans notre modèle relationnel, alors B est également valide.

3.4.1 Les règles et Axiomes du Système

Le premier axiome, la *réflexivité*, exprime le fait que dans notre modèle un document est à-propos de lui-même. Ceci est très courant dans les systèmes de recherche d'informations [Hui96], mais aussi dans les systèmes logiques non-monotones [Gab85].

Réflexivité

$$S \boxrightarrow_R S$$

La différence avec un système logique monotone apparaît ici en ces termes: en logique monotone la réflexivité $\Delta \mid\!\sim A$ est valide quand il existe un $\Delta_A \subseteq \Delta$ reprenant des formules de A ($\Delta \cap A \neq \emptyset$), alors que dans les logiques non-monotones, l'ensemble des formules tout entier Δ est utilisé pour dériver A. Ainsi, si Δ est élargi à Δ', il n'y aura pas de changement dans le cas d'un raisonnement monotone, alors qu'il y a des fortes chances que la dérivation n'est plus valide dans le cas d'un raisonnement non-monotone.

Les conséquences logiques d'une formule doivent dépendre de sa sémantique et non pas de sa forme. Cette nécessité est exprimée dans notre modèle par la règle suivante:

Équivalence Logique à Gauche

$$\frac{\phi \equiv_r \psi \quad S \cup \{\phi\} \boxrightarrow_R T}{S \cup \{\psi\} \boxrightarrow_R T}$$

L'*Équivalence Logique à Gauche* exprime le fait que des situations comprenant des relons équivalents ont exactement les mêmes conséquences logiques, c'est-à-dire qu'elles engendrent les mêmes décisions de pertinence. Un document à-propos d'une situation contenant un relon ϕ donné, sera également à-propos de la situation obtenue en remplaçant ϕ par un relon qui lui est équivalent.

Les trois règles suivantes expriment toutes une instanciation d'une règle très classique dans les logiques non-monotones, à savoir un affaiblissement à droite[21]. Il s'agit du fait que le système accepte comme conséquence logique valide tout ce qui est logiquement impliqué par ce qui était déjà dérivé. En d'autres termes, l'affaiblissement à droite exprime, que les conséquences logiques sont fermées sur les implications logiques. Similairement, notre modèle relationnel exprime que tous les index (les situations) sont fermés sur les propriétés des relations, exprimées alors en une relation d'endiguement. Cette dernière joue le rôle d'une implication informationnelle logique. Selon le type de l'endiguement, nous spécifions le détail de la règle associée.

La première des règles précise que toutes les règles valides comme relations d'endiguement le sont aussi comme décisions de pertinence.

Endiguement Élémentaire

$$\frac{\phi \rightarrow_r \psi}{\{\phi\} \boxrightarrow_R \{\psi\}}$$

21. L'affaiblissement à droite ou *Right Weakening* en anglais [KLM90] est défini comme suit:

Affaiblissement à Droite

$$\frac{\models \alpha \rightarrow \beta \quad \gamma \mid\!\sim \alpha}{\gamma \mid\!\sim \beta}$$

La deuxième règle est une adaptation de l'*affaiblissement à droite* à l'endiguement faible. Elle exprime le fait que si une situation S est à-propos d'une situation T', avec $\phi \in T'$ et s'il est possible de déduire un relon ψ de ϕ par une règle d'endiguement faible, alors nous pouvons remplacer le relon ϕ par ψ dans T' en préservant la relation de pertinence avec S.

Endiguement Faible

$$\frac{\phi \rightarrow_r^f \psi \quad S \boxrightarrow_R T \cup \{\phi\}}{S \boxrightarrow_R T \cup \{\psi\}}$$

Les deux règles suivantes transforment l'*Endiguement Faible* de façon à ce que les relons dérivés s'ajoutent à ceux à partir desquels ils ont été endigués dans la situation originelle. Elles expriment qu'il n'est possible d'ajouter des relons que s'il s'agit de l'endiguement avec retenue "$\rightarrow_r^{\oplus}$". La règle à gauche a trait à une dérivation d'un relon vers un relon, alors que la seconde concerne la dérivation d'un relon à partir d'un ensemble de relons.

Endiguement avec Retenue

$$\frac{\phi \rightarrow_r^{\oplus} \psi \quad S \boxrightarrow_R T \cup \{\phi\}}{S \boxrightarrow_R T \cup \{\psi\} \cup \{\phi\}} \qquad \frac{\phi_1 \odot \phi_2 \odot \ldots \odot \phi_n \rightarrow_r^{\oplus} \varphi \quad S \boxrightarrow_R T \cup \{\phi_1, \phi_2, \ldots, \phi_n\}}{S \boxrightarrow_R T \cup \{\phi_1, \phi_2, \ldots, \phi_n\} \cup \{\varphi\}}$$

La règle du "*Cut*" [22] est très commune aux systèmes logiques qu'ils soient monotones ou non-monotones. Elle exprime le fait que pour prouver qu'un document est à-propos d'une requête, il suffit d'élargir la description de ce dernier par un ensemble de relons qu'il lui est possible de dériver. Pour décider de la pertinence du document, il suffit alors de montrer que l'ensemble élargi est à-propos de la requête.

Cut

$$\frac{S \cup T \boxrightarrow_R U \quad S \boxrightarrow_R T}{S \boxrightarrow_R U}$$

La règle du *Cut* est aussi appelée *règle de transitivité* [Gab93]. Elle n'exprime pas autre chose que la génération de lemmes dans un système logique de preuve, à savoir si $\Delta \mid\!\sim A$, alors la formule A peut être utilisée comme un lemme pour obtenir B de Δ ($\Delta \mid\!\sim A$; $\Delta, A \mid\!\sim B$ implique $\Delta \mid\!\sim B$).

22. Il existe plusieurs versions de la règle du Cut. Selon Gabbay [Gab93], elles sont toutes équivalentes dans le cas des logiques classiques ou intuitionnistes. Une autre version possible peut être:

Cut

$$\frac{\alpha \mid\!\sim \beta \quad \gamma, \beta \mid\!\sim \sigma}{\gamma, \alpha \mid\!\sim \sigma}$$

Cette dernière version ajoutée à la réflexivité implique nécessairement la propriété de monotonicité. À l'inverse, la version ici introduite tirée de [KLM90] ne conduit pas à ce résultat.

Une relation de conséquence logique devrait au moins vérifier la *réflexivité*, la règle du *cut* et une forme restreinte de la monotonicité [KLM90, Gab85, Gab93][23]:

Monotonicité Restreinte

$$\frac{S \boxrightarrow_R T \quad \text{Condition}}{S \cup U \boxrightarrow_R T}$$

Si nous avons choisi ici de donner un comportement non-monotone à notre système c'est surtout parce que nous privilégions la conception des systèmes orientés vers la précision des réponses. L'addition d'information par la règle de monotonicité a pour conséquence l'augmentation du nombre des déductions logiques, sans toutefois jamais les réduire. En d'autres termes, il est possible avec la propriété de monotonicité de renvoyer plus de documents pertinents à l'utilisateur mais jamais moins. Cette propriété est intéressante puisqu'elle améliore le taux du rappel dans le système. Cependant, vu qu'il y a souvent une antinomie entre le taux de rappel et de précision, il est fort probable que cela engendre du bruit, c'est-à-dire que le système retournera à l'utilisateur des documents non pertinents. Par exemple, prenons le cas où la requête utilisateur est le terme anglais «dog». Par application de l'axiome de réflexivité, nous pouvons déduire que «dog» est à-propos de «dog». L'ajout de l'information «hot» par le biais de la règle de monotonicité permettra de déclarer le document «hot dog» à-propos de «dog». Ainsi, alors que l'utilisateur est intéressé par des documents sur les chiens, le système retourne des documents ayant trait aux sandwichs. Dans [BH95, Hui96], la non-monotonicité de la relation de pertinence est liée au fait que l'acceptabilité d'une déduction dépend de l'utilisateur et de ce que sa requête véhicule comme sens sémantique. L'information ajoutée pourrait contredire le thème de recherche de l'utilisateur, et générer ainsi du bruit. Pour éviter ce problème nous adoptons la règle de *Monotonicité Circonspecte* suivante:

Monotonicité Circonspecte

$$\frac{S \boxrightarrow_R T \quad S \boxrightarrow_R U}{S \cup U \boxrightarrow_R T}$$

Cette règle est proposée dans [Gab93, KLM90]. Elle exprime le fait que la connaissance d'un nouveau fait, la validité duquel peut être conclue ne remet pas en cause les conclusions précédentes. Dans notre cas, la relation de pertinence n'est pas violée par l'ajout dans S de toutes les informations dont S est à-propos. La règle de la *Monotonicité Circonspecte* admet ainsi un intérêt pragmatique.

Cette règle semble parfois assez contraignante dans la réalité. En effet, si elle évite l'introduction du bruit, en favorisant la précision des réponses, elle peut constituer un écueil à la dérivation de certains documents pertinents. Par exemple, dans un formalisme comme les graphes conceptuels [Sow84], une forme moins forte de la *Monotonicité Circonspecte* peut être proposée:

Monotonicité Prudente

$$\frac{S \boxrightarrow_R T \quad S \cap U \neq \emptyset}{S \cup U \boxrightarrow_R T}$$

23. Un système monotone devra vérifier par contre, outre la réflexivité et la règle du cut, la propriété de monotonicité:

Monotonicité

$$\frac{\Delta \mid\!\sim A}{\Delta, \Delta' \mid\!\sim A}$$

Il suffit dans ce cas que l'information U ajoutée partage des relons avec la situation originelle S. La situation U reprend donc une partie de la sémantique véhiculée par S. L'union des deux situations reviendra à une fusion basée sur les relons en commun, et constitue ainsi une spécialisation cohérente commune. Cela est traduit en termes de graphes, par l'existence d'une jointure possible entre l'information ajoutée et le graphe de départ.

Une autre règle qui découle de la *Monotonicité Restreinte*, exprime le fait que l'ajout de relons ne peut être valide que s'ils ne sont pas en préclusion avec ceux existants dans la situation originelle:

Monotonicité Rationnelle

$$\frac{S \boxrightarrow_R T \quad S \equiv \{\varphi_1, \ldots, \varphi_n\} \quad \varphi_1 \not\downarrow_r \phi \quad \ldots \quad \varphi_n \not\downarrow_r \phi}{S \cup \{\phi\} \boxrightarrow_R T}$$

Cette règle a été originellement introduite par Huibers [Hui96] pour formaliser le comportement par défauts des utilisateurs. Certaines relations de préclusion sont alors données a priori comme des données par défaut sur les utilisateurs et leurs préférences.

La règle de *Disjonction* suivante ne remet pas en cause le principe de non-monotonicité de notre modèle relationnel, même si elle est présente dans les raisonnements monotones classiques. Elle exprime le fait que si une formule est séparément une conséquence logique de deux formules différentes, alors elle est une conséquence logique de leur disjonction. En recherche d'informations, il est facile de constater que l'élargissement d'un index avec des informations connues pour être à-propos de la requête ne viole pas les conclusions de pertinence déjà élaborées.

Disjonction

$$\frac{S \boxrightarrow_R T \quad U \boxrightarrow_R T}{S \cup U \boxrightarrow_R T}$$

Enfin, la règle du *Bouclage*[24] exprime le fait que si des situations peuvent être arrangées en un cycle, dans un sens que chacune de ces situations est une conséquence logique de l'autre, alors chacune de ces situations est une conséquence logique de n'importe quelle situation de cette boucle. En d'autres termes, elles sont toutes équivalentes par la relation $\equiv$. Dans notre modèle relationnel, cela revient à dire que si d'un index de document, on obtient le même index après application de plusieurs règles, alors toutes les situations générées entre ces deux états sont équivalentes à cause du fait que la relation de préférence $\prec_R$ sur les index est transitive et anti-symétrique.

Bouclage

$$\frac{S_0 \boxrightarrow_R S_1 \quad S_1 \boxrightarrow_R S_2 \quad \ldots \quad S_{k-1} \boxrightarrow_R S_k \quad S_k \boxrightarrow_R S_0}{S_0 \boxrightarrow_R S_k}$$

3.4.2 Autres Règles de Dérivation

Les règles de dérivation présentées dans la section précédente permettent la déduction d'autres dérivations logiques. Ainsi, les règles de *Monotonicité Circonspecte* et du *Cut* dérivent la propriété

24. Dans [KLM90], elle est appelée Loop. Cette règle est valide dans tous les modèles dans lesquels la relation de préférence $\prec_R$ constitue un ordre partiel strict. Cela explique sa présence dans notre système de dérivation.

de *Conjonction* suivante:

Conjonction

$$\frac{S \boxrightarrow_R T \quad S \boxrightarrow_R U}{S \boxrightarrow_R T \cup U}$$

La règle d'*Équivalence* est obtenue par utilisation de la *Monotonicité Circonspecte* et de la règle d'*Équivalence Logique*:

Équivalence

$$\frac{S \boxrightarrow_R T \quad T \boxrightarrow_R S \quad S \boxrightarrow_R U}{T \boxrightarrow_R U}$$

L'application de cette dernière règle, permet de dériver la *Disjonction Transitive*:

Disjonction Transitive

$$\frac{S \cup T \boxrightarrow_R S \quad S \boxrightarrow_R U}{S \cup T \boxrightarrow_R U}$$

Une conséquence directe des règles de *Bouclage* et d'*Équivalence* est la règle suivante:

Cycle

$$\frac{S_0 \boxrightarrow_R S_1 \quad S_1 \boxrightarrow_R S_2 \quad \ldots \quad S_{k-1} \boxrightarrow_R S_k S_k \boxrightarrow_R S_0}{S_i \boxrightarrow_R S_j}$$

Une règle très intuitive et utile en recherche d'informations est la règle d'*Union Contextuelle* suivante:

Union Contextuelle

$$\frac{S \boxrightarrow_R T \quad U \boxrightarrow_R V}{S \cup U \boxrightarrow_R T \cup V}$$

Cette règle découle directement de l'application des règles d'*Équivalence Logique à Gauche*, de l'*Endiguement Faible* et de la *Disjonction*.

Nous mentionnons à la fin la règle d'*Endiguement Inverse*, une règle directement dérivée à partir de l'*Équivalence Logique à Gauche*:

Endiguement Inverse

$$\frac{\psi \rightarrow_r \phi \quad S \cup \{\phi\} \boxrightarrow_R T}{S \cup \{\psi\} \boxrightarrow_R T}$$

Cette règle fut ainsi présentée dans [Hui96] [25] comme un postulat intéressant en recherche d'informations. Il exprime le fait que si une situation S' est à-propos d'une situation T, avec $\phi \in S'$ et si $\psi \rightarrow_r \phi$ est valide, alors nous pouvons remplacer l'infon ϕ par ψ dans S' sans perdre le fait que cette dernière soit à-propos de la situation T. Les preuves de la validité de ces règles de dérivation à partir de celles de la section précédente sont données dans [KLM90].

25. Elle fut présentée par Huibers sous le nom de *The Union Containment*.

3.4.3 Complétude et Correction

Notre modèle de dérivation de pertinence devra être complet et correct vis-à-vis de la conséquence préférentielle $|\!\sim_R$ introduite dans la définition 3.2.13 (page 88). Dans ce cas, la correspondance se fera bien entre la requête et les représentations les plus élargies des index des documents. Dans [KLM90], une notion de relation de conséquence cumulative[26] et préférentielle est ainsi définie:

Définition 3.4.3 (Relation de Conséquence Préférentielle) *Une relation de conséquence* $|\!\sim$ *est dite* préférentielle ordonnée[27] *si, et seulement si, elle contient toutes les instances de l'axiome de* réflexivité*, et est fermée sur les règles de l'*Équivalence Logique à Gauche*, de l'*Affaiblissement à Droite*, du* Cut*, de la* Monotonicité Circonspecte*, de la* Disjonction *et du* Bouclage*.*

Notre système de dérivation de pertinence de la section 3.4 est donc défini comme suit:

Définition 3.4.4 (Dérivation de Pertinence Relationnelle) *Le système relationnel de dérivation de pertinence* $\mathcal{S}_{\Re}$ *est défini comme étant le triplet* $\langle\ \mathcal{P}$, {*Réflexivité*}, {*Équivalence Logique à Gauche, Endiguement Élémentaire, Endiguement Faible, Endiguement avec Retenue, Cut, Monotonicité Circonspecte, Disjonction, Bouclage*}$\rangle$.

Les règles d'endiguement étant la contre-part logique de l'*Affaiblissement à Droite*, il en découle de la définition 3.4.3 que la relation de pertinence Relationnelle $\Box\!\!\rightsquigarrow_R$ est une relation préférentielle ordonnée:

Lemme 2 *La pertinence relationnelle* $\Box\!\!\rightsquigarrow_R$ *est une relation préférentielle ordonnée.*

Le théorème de correction suivant est ainsi énoncé. La preuve se trouve dans [KLM90]:

Théorème 2 (Correction) *la relation de conséquence* $|\!\sim_R$ *définie par le modèle relationnel préférentiel* $\mathcal{M}_{RP}$ *de la définition 3.2.14 (page 89)est une relation préférentielle ordonnée, i.e, si* $\models_{\mathcal{M}_{RP}} \chi(d) \mid\!\sim_R \chi(q)$ *alors* $\vdash_{\mathcal{S}_{\Re}} \chi(d) \Box\!\!\rightsquigarrow_R \chi(q)$

De plus il est montré dans [KLM90] qu'étant donnée une relation préférentielle $|\!\sim$, il est possible de construire un modèle préférentiel dont la relation de conséquence est exactement $|\!\sim$:

Théorème 3 (Complétude) *Le système de dérivation de pertinence* $\mathcal{S}_{\Re}$ *est complet vis-à-vis de la relation de conséquence préférentielle* $|\!\sim_R$*, i.e, si* $\vdash_{\mathcal{S}_{\Re}} \chi(d) \Box\!\!\rightsquigarrow_R \chi(q)$ *alors* $\models_{\mathcal{M}_{RP}} \chi(d) \mid\!\sim_R \chi(q)$.

3.4.4 Décidabilité

Supposons que nous posons une question à une machine, et que cette dernière est munie d'un certain algorithme qu'elle utilise pour produire une réponse correcte. Si la machine garantie qu'à un moment donné, elle s'arrêtera pour nous donner une réponse (positive ou négative), alors le problème est dit *décidable*. Si la machine s'arrête uniquement quand elle est en possession d'une réponse positive, et qu'elle pourrait infiniment tourner quand la réponse est négative, alors le

26. Le terme "*Cumulative*" est introduit afin de mentionner le fait que tous les axiomes et règles qui ne contredisent pas la non-monotonicité sont ajoutées au système de dérivation.

27. En Anglais *Loop-Preferential*.

problème est dit *semi-décidable*. Les raisonnements non-monotones conduisent généralement à des approches qui ne sont même pas *semi-décidables* [Isr80, Rei80]. Par conséquent, n'importe quel système qui utilise une approche non-monotone fera dans les meilleurs des cas des erreurs de déductions de temps à autres, ou dans les pires des cas, il pourra boucler infiniment [Gin87].

Dans notre approche, ceci serait le cas si le langage d'indexation relationnelle est infini. En effet, il se pourrait alors que l'ajout d'information (ou en d'autres termes la spécification de nouvelles valeurs de vérité pour les termes d'indexation) soit infini. Par conséquent, le système de dérivation bouclera infiniment et ne sera pas en mesure de fournir une réponse à la question posée. Il s'agit du cas où l'on n'est pas certain d'avoir un élément minimal dans la chaîne de préférence.

Théorème 4 (Décidabilité) *Si le langage d'indexation relationnelle $\mathcal{L}$ est fini, le système de dérivation $\mathcal{S}_{\Re}$ est décidable.*

Preuve 2 *D'après le théorème 1 de la page 88, si le langage d'indexation relationnelle est fini, alors il existe un élément minimal unique constituant l'index le plus élargi du document par rapport aux propriétés des relations et des objets. Dans ce cas, le système de dérivation relationnel $\mathcal{S}_{\Re}$ étant complet et correct vis-à-vis du modèle préférentiel, l'évaluation de la relation de pertinence $\Box\!\!\rightsquigarrow_R$ s'effectuera à la base de cet élément qui existe et qui est unique. L'opération qui consiste à savoir si un ensemble fini est inclus dans un autre ensemble fini (ou que la situation associée à la requête est incluse dans celle associée à l'index le plus élargi) étant décidable, la décision de pertinence du système est un problème décidable.*

Ce résultat est assez important. En effet, vu que nous choisirons un langage d'indexation basé sur les graphes conceptuels, et que ce dernier est fini une fois l'ontologie du domaine d'application spécifiée, notre système de dérivation à base de graphes conceptuels sera décidable (voir chapitre 4, page 198).

3.4.5 Exemple de Dérivation de Pertinence

Nous reprenons ici comme exemple, l'image de la page 86, décrivant une aventure de Tom & Jerry. Supposons que cette image d est maintenant munie de l'index suivant:

$\chi(d) = \{\langle\langle$*a-droite*, Tom, Jerry; 1$\rangle\rangle$, $\langle\langle$*État*, Tom, colère; 1$\rangle\rangle$, $\langle\langle$*Tenant*, Jerry, Cuillère; 1$\rangle\rangle\}$

Supposons qu'une requête est à la recherche de toutes les images où *Jerry tenant un objet apparaît à gauche de Tom dans le plan de la photo*. Cette requête est exprimée de la façon suivante en théorie des situations:

$\chi(q) = \{\langle\langle$*a-gauche*, Jerry, Tom; 1$\rangle\rangle$, $\langle\langle$*Tenant*, Jerry, Objet; 1$\rangle\rangle\}$

Considérons que la base de connaissances du système est partiellement formée des relations d'endiguement suivantes:

Connaissances
$\langle\langle$*a-droite*, e_1, e_2; 1$\rangle\rangle \equiv_r \langle\langle$*a-gauche*, e_2, e_1; 1$\rangle\rangle$
$\langle\langle I$,Cuillère ; 1$\rangle\rangle \rightarrow \langle\langle I$,Objet; 1$\rangle\rangle$

Nous montrons alors que le document d est pertinent à la requête q à travers l'utilisation des connaissances du domaine:

$$\dfrac{\dfrac{\dfrac{\langle\langle I,\text{Cuillère};1\rangle\rangle \rightarrow_r \langle\langle I,\text{Objet};1\rangle\rangle \quad \{\langle\langle \textit{a-gauche},\text{Jerry},\text{Tom};1\rangle\rangle,\langle\langle \textit{Tenant},\text{Jerry},\text{Objet};1\rangle\rangle\} \boxminus\!\!\rightarrow_{R\chi}(q)}{\{\langle\langle \textit{a-gauche},\text{Jerry},\text{Tom};1\rangle\rangle,\langle\langle \textit{Tenant},\text{Jerry},\text{Cuillère};1\rangle\rangle\} \boxminus\!\!\rightarrow_{R\chi}(q)}\,(EI) \quad \langle\langle \textit{a-droite},e_1,e_2;1\rangle\rangle \equiv_r \langle\langle \textit{a-gauche},e_1,e_2;1\rangle\rangle}{\{\langle\langle \textit{a-droite},\text{Tom},\text{Jerry};1\rangle\rangle,\langle\langle \textit{Tenant},\text{Jerry},\text{Cuillère};1\rangle\rangle\} \boxminus\!\!\rightarrow_{R\chi}(q)}\,(ELG) \quad \langle\langle \textit{a-droite},\dot{\text{p}},\dot{\text{q}};1\rangle\rangle \not\equiv_r \langle\langle \textit{État},\dot{\text{s}},\dot{\text{t}};1\rangle\rangle \quad \langle\langle \textit{Tenant},\dot{\text{p}},\dot{\text{q}};1\rangle\rangle \not\equiv_r \langle\langle \textit{État},\dot{\text{s}},\dot{\text{t}};1\rangle\rangle}{\{\langle\langle \textit{a-droite},\text{Tom},\text{Jerry};1\rangle\rangle,\langle\langle \textit{Tenant},\text{Jerry},\text{Cuillère};1\rangle\rangle,\textit{État},\text{Tom},\text{colère};1\rangle\rangle\} \boxminus\!\!\rightarrow_{R\chi}(q)}\,(MR)$$

Il s'en suit que par application successive des règles d'*Endiguement Inverse* (EI), d'*Équivalence Logique à Gauche* (ELG) et de *Monotonicité Rationnelle* (MR), il est possible de déduire que le document d est à-propos de la requête. Les pas de dérivation se basent sur les propriétés des relations incluant, entre autres, les propriétés sur leurs arguments (il s'agit ici de la relation de spécialisation/généralisation entre Cuillère et Objet).

Il est à noter, que comme nous l'avons expliqué précédemment, nous préférons dans la pratique l'utilisation de la *Monotonicité Rationnelle*, par rapport à la *Monotonicité Circonspecte*. En effet, la règle de *Monotonicité Rationnelle* permet de favoriser l'établissement de certaines dérivations comme pour l'exemple ci-dessus. Comparée à la trop contraignante règle de *Monotonicité Circonspecte*, elle favorise le taux de rappel tout en respectant la contrainte de précision des réponses, puisqu'elle vérifie qu'il n'y a pas de contradictions entre les informations ajoutées et celles existantes dans une situation donnée.

3.5 Extensions Possibles du Modèle

Notre système relationnel de dérivation de pertinence $\mathcal{S}_{\Re}$ utilise uniquement des relations d'endiguement certaines, c'est-à-dire des connaissances valides dans tous les contextes d'application. Les règles de $\mathcal{S}_{\Re}$ permettent ainsi la dérivation des connaissances implicites véhiculées par les relations et les objets qu'elles lient. Chaque passage d'une situation à une autre se faisait ainsi avec une certitude totale.

En recherche d'informations, il est parfois intéressant de considérer des dérivations de connaissances incertaines. Ces dérivations peuvent se baser sur des relations d'endiguement plausibles mais pas toujours certaines. Ainsi, la valuation du fait que le relon $\langle\langle$*parle-à*, John, Mary,; $1\rangle\rangle$ est à-propos de $\langle\langle$*écoute*, Mary, John, ; $1\rangle\rangle$revient très souvent à vraie, même si ce n'est pas toujours le cas. Cela dépend du contexte, par exemple si Mary et John sont dans une classe ou dans une conférence, ou si Mary est déjà entrain d'écouter une autre personne. S'il s'avère que dans le contexte où l'on opère que Mary est *attentive*, cette valuation serait évaluée à vraie. Ceci peut se traduire par la relation d'endiguement plausible suivante:

Exemple de Contexte

$$\dfrac{\langle\langle \textit{parle-à},\text{John},\text{Mary},;1\rangle\rangle \quad : \quad \langle\langle \textit{état},\text{Mary},\text{Attentive},;1\rangle\rangle}{\langle\langle \textit{écoute},\text{Mary},\text{John},;1\rangle\rangle}\,[\alpha]$$

où α est un réel de $]o..1]$, désignant le degré de plausibilité d'une telle relation d'endiguement. De cet exemple apparaît clairement comment le contexte peut influencer la dérivation de certaines connaissances implicites. Si l'index du document s'intègre à ce contexte, il sera possible de lui

ajouter la conséquence de la relation d'endiguement en question et favoriser ainsi certaines prises de décisions de pertinence.

La base de connaissances du système pourra donc être enrichie par des relations d'endiguement incertaines. Ces dernières viendront s'ajouter aux relations d'endiguement faible, d'endiguement équivalent ou d'endiguement avec retenue déjà présentes dans le système. L'idée est ainsi d'ajouter aux règles du système relationnel de dérivation de pertinence $\mathcal{S}_{\Re}$ (voir Définition 3.4.4), des règles d'endiguement incertaines. Ces dernières seront basées sur les relations d'endiguement incertaines. À la différence des autres règles de dérivation de $\mathcal{S}_{\Re}$, elles introduisent de l'incertitude dans le processus de dérivation logique. En d'autres termes, elles ne conduisent pas nécessairement à des situations strictement équivalentes ou plus génériques que celle assignée à l'index originel. La plausibilité du résultat fourni par ce type de règles est égal au degré de plausibilité α assigné à la relation d'endiguement incertaine utilisée.

L'ajout de telles règles de dérivation incertaines permettra de favoriser le taux de rappel dans certains cas de figure. La connaissance contextuelle ci-dessus mentionnée rappelle ainsi les logiques par défaut et peut se lire de la manière suivante: "*Si* $\langle\langle$parle-à, *John, Mary,* ; 1$\rangle\rangle$ *est valide, et si l'infon* $\langle\langle$état, *Mary, Attentive,* ; 0$\rangle\rangle$ *n'est pas présent dans le contexte de l'application (l'index du document), alors il est possible d'inférer* $\langle\langle$écoute, *Mary, John,* ; 1$\rangle\rangle$.

Il existe d'autres types de relations d'endiguement incertaines. Par exemple, supposons que le document est toujours à-propos du relon $\langle\langle$*parle-à*, John, Mary, ; 1$\rangle\rangle$. Ce document sera faiblement pertinent pour une requête représentée par le relon $\langle\langle$*Loc*, $\langle\langle$*parle-à*, John, Mary, ; 1$\rangle\rangle$, classe; 1$\rangle\rangle$. Cela s'exprime par la relation d'endiguement suivante:

Exemple d'Endiguement Incertain

$$\frac{\langle\langle \textit{parle-à}, \text{John}, \text{Mary}, ; 1\rangle\rangle}{\langle\langle \textit{Loc}, \langle\langle \textit{parle-à}, \text{John}, \text{Mary}, ; 1\rangle\rangle, \text{classe}; 1\rangle\rangle}\,[\alpha]$$

La prise en compte de telles relations d'endiguement passe également par l'introduction d'un certain nombre de règles incertaines dans le système de dérivation $\mathcal{S}_{\Re}$. Dans la suite nous nous proposons d'étudier l'impact de l'introduction des règles incertaines dans le processus de recherche. Nous appliquerons ainsi deux types d'extension au système relationnel de dérivation $\mathcal{S}_{\Re}$. Dans une première étape, nous introduisons des règles par défaut permettant de gérer les connaissances contextuelles. Le système $\mathcal{S}_{\Re}$ sera ensuite étendu par des règles incertaines duales à celles exploitant les relations d'endiguement certaines. Le résultat de ces extensions est un système relationnel étendu $\mathcal{S}_{\Re}^{Ext}$ de dérivation de pertinence, permettant d'exploiter aussi bien les connaissances certaines que les connaissances plausibles. Nous introduisons ensuite une technique permettant de justifier les résultats du système ainsi étendu. Cette technique, basée sur une assignation de labels aux règles de dérivation incertaines, permet également la prise en compte des jugements de pertinence des utilisateurs lors de la reformulation des requêtes, à l'image de ce qui se fait dans les procédés de bouclage de pertinence (*relevance feedback*). Enfin, l'introduction de la notion de *facette* dans le processus d'indexation relationnelle permet une description fidèle et complète des documents multimédias.

3.5.1 Introduction de l'Incertitude

Supposons qu'après l'élargissement de l'index d'un document par toutes les informations implicites qu'il contient, le système de dérivation ne puisse toujours pas le déclarer comme à-propos de la représentation de la requête. Au lieu d'arrêter le processus à ce stade, il est parfois souhaitable de faire dériver des informations incertaines pouvant rendre possible une décision de pertinence positive. Dans ce cas, le degré de pertinence du document en question sera inférieur à celui des documents retrouvés par la seule application des règles d'inférence certaines.

Plusieurs transformations incertaines sont possibles à partir de l'index élargi d'un document. Seulement, à l'opposé des règles basées sur des connaissances certaines, elles ne conduisent pas nécessairement à la même situation finale, quand elles sont appliquées dans un ordre différent. En effet, selon l'ordre d'application de ces règles, des informations incertaines peuvent être ajoutées ou pas, modulant ainsi le déclenchement d'autres règles. Par exemple, il se peut que l'application d'une règle utilisant une connaissance par défaut soit possible dans un premier temps, mais ne peut l'être si elle est précédée par des règles dérivant des connaissances en exclusion mutuelle avec le contexte de la dite connaissance.

En termes de la sémantique des mondes possibles, à partir de l'index le plus élargi d'un document (s'il existe[28]), l'application d'une règle plausible correspond à un passage vers une situation probable. Ce passage se fait avec un degré de plausibilité α fourni par la connaissance incertaine utilisée par cette règle. Le processus de dérivation incertain prend fin quand une situation probable $\chi(d)^n_{Prob}$ permet de satisfaire la requête ou quand il n'est plus possible d'appliquer d'autres règles incertaines.

Pour un index élargi donné, il peut donc exister plusieurs *transformations possibles*. En traçant toutes ces transformations, nous obtenons une arborescence dont les noeuds représentent des situations probables. Les transformations effectuées d'une situation initiale vers une situation probable possible sont couramment appelées *chemins de transformation* [Nie90]. Selon le principe d'incertitude de Rijsbergen [vR86] (voir page 67), parmi les divers chemins de transformation possibles résultant en une situation probable satisfaisant la requête, le calcul du degré de pertinence devra se baser sur celui qui aura le moindre coût en termes de transformations, i.e, celui qui inclura le maximum de certitude (plausibilité) possible. Dans notre cas, les situations correspondent aux diverses transformations possibles de l'index de document déjà élargi par les informations implicites qu'il contient ($\chi(d)_{Elargi} = \chi(d)^0_{Prob}$), vers d'autres index élargis probables $\chi(d)^i_{Prob}$ (voir figure 3.6).

La plausibilité de pertinence d'un document pour une requête est alors obtenue en mesurant la plausibilité de chaque transformation élémentaire d'une situation probable $\chi(d)^i_{Prob}$ vers une autre $\chi(d)^{i+1}_{Prob}$, dénotée par $\sigma(\chi(d)^i_{Prob}, \chi(d)^{i+1}_{Prob})$. Une fois toutes ces plausibilités élémentaires calculées, nous pouvons établir la mesure de plausibilité de chaque chemin de transformation $p = \{\chi(d)_{Elargi}, \ldots, \chi(d)^n_{Prob}\}$. Ceci est donné à travers une fonction $\bigotimes$, qui peut par exemple être le produit des réels:

$$\sigma(p) = \bigotimes_{i=0, i=n-1} \sigma(\chi(d)^i_{Prob}, \chi(d)^{i+1}_{Prob}) = \prod_{i=1}^{n-1} \sigma(\chi(d)^i_{Prob}, \chi(d)^{i+1}_{Prob})$$

28. Nous rappelons que si le langage d'indexation relationnelle est fini, cet élément existe et est unique. S'il n'est pas possible d'appliquer aucune règle stricte, l'index originel du document sera pris comme l'index le plus élargi (Cf Hypothèse 4). Dans le cas ou le langage n'est pas fini, le processus n'est pas décidable.

Dans notre cas, le degré de plausibilité élémentaire $\sigma(\chi(d)^i_{Prob}, \chi(d)^{i+1}_{Prob})$ est donné par le degré de plausibilité α associé à une règle d'inférence plausible: $\sigma(\chi(d)^i_{Prob}, \chi(d)^{i+1}_{Prob}) = \alpha$. Nous énonçons alors la définition du degré de pertinence (ou valeur de pertinence):

Définition 3.5.1 (Degré de Pertinence) *Soit $\mathcal{S}^{ext}_{\Re}$ notre système relationnel de dérivation étendu avec les règles incertaines. Le degré de pertinence d'un document d pour une requête utilisateur q, noté $D_{Rel}(d, q)$, est déterminé de la façon suivante:*

- *Si l'index élargi $\chi(d)_{Elargi}$ du document d est pertinent pour $\chi(q)$ ($\chi(d)_{Elargi} \ \Box\!\!\rightsquigarrow_R \ \chi(q)$) alors:*

$$D_{Rel}(d, q) = 1.$$

- *Si l'index élargi $\chi(d)_{Elargi}$ du document d n'est pas jugé pertinent pour $\chi(q)$ par le système ($\chi(d)_{Elargi} \ \not\Box\!\!\rightsquigarrow_R \ \chi(q)$), alors:*

$$D_{Rel}(d, q) = \sigma(\chi(d)_{Elargi}, \chi(q)) = Max_{p \in P(\chi(d), \chi(d)^n_{Prob}}[\sigma(p)],$$

où $P(\chi(d), \chi(d)^n_{Prob})$ est l'ensemble de tous les chemins de transformation ayant pour racine $\chi(d)_{Elargi}$ et pour feuille $\chi(d)^n_{Prob}$ tel que $\vdash_{\mathcal{S}^{ext}_{\Re}} \chi(d)^n_{Prob} \ \Box\!\!\rightsquigarrow_R \ \chi(q)$.

Dans le premier cas, nous dirons que le document d est strictement pertinent *pour la requête, alors que dans le second cas, nous dirons qu'il est* plausiblement pertinent. *Dans ce cas, nous choisirons le chemin ayant la mesure de plausibilité maximale. Nous appelons par $Path_{Max}$ un chemin de transformations à plausibilité maximale.*

Définition 3.5.2 (Connaissances Incertaines) *D'une façon générale, les connaissances incertaines du système sont représentées à travers des relations d'endiguement ayant la forme suivante:*

Connaissance Incertaine

$$(label)\frac{Prémisse}{Conclusion}\ [\alpha]$$

où:

(label): *désigne une étiquette* unique *associée à la connaissance incertaine. Cette étiquette comme on le verra ensuite sera par exemple utile pour la prise en compte des jugements de pertinence de l'utilisateur lors d'un bouclage de pertinence.*

[α]: *désigne le degré de plausibilité de la relation d'endiguement.*

À chaque fois que le système relationnel de dérivation étendu $\mathcal{S}^{Ext}_{\Re}$ applique une règle ayant trait à une connaissance incertaine, il laisse une trace de son label et tient compte de sa plausibilité pour le calcul du degré de pertinence.

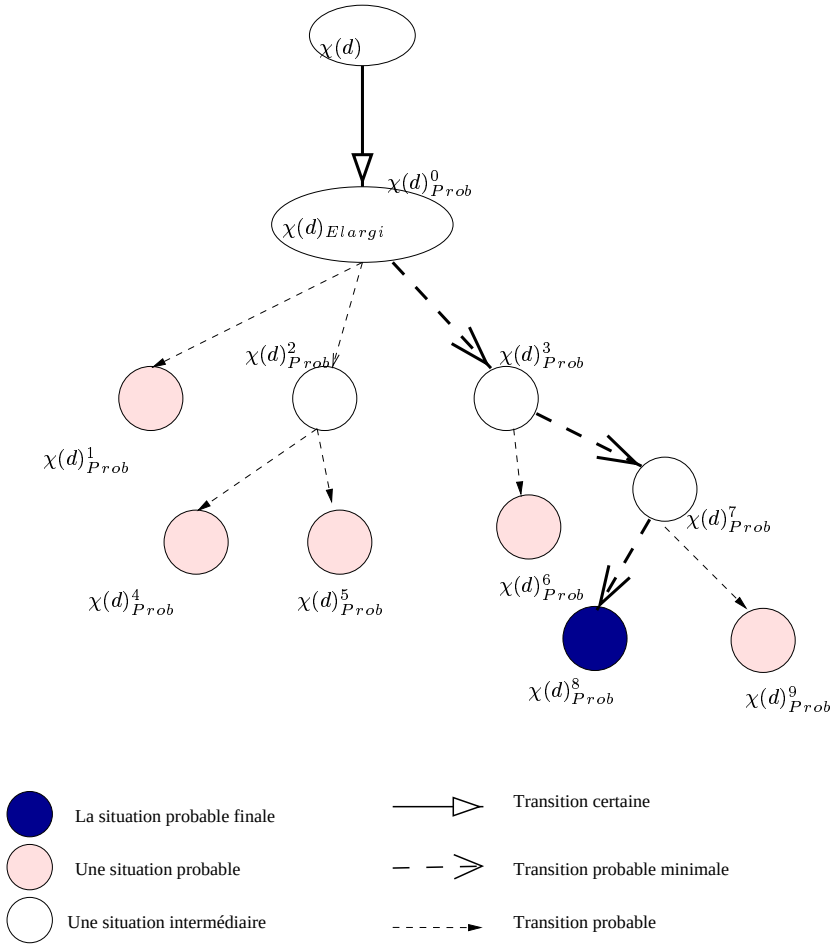


Figure 3.6. Chemins de transformation d'un index de document

Afin de tenir compte du principe d'incertitude de Rijsbergen, il est nécessaire, dans le cas des dérivations incertaines, d'associer la relation de préférence au chemin de transformation à plausibilité maximale. Nous définissons ainsi une relation de préférence étendue:

Définition 3.5.3 (Relation de Préférence Étendue $\prec_R^{Ext}$)

$$\chi(d)^{i+1}_{Prob} \prec_R^{Ext} \chi(d)^{i}_{Prob} \ \ ssi \ \ \chi(d)^{i+1}_{Prob} \prec_R \chi(d)^{i}_{Prob} \wedge \chi(d)^{i+1}_{Prob} \in Path_{Max}$$

En d'autres termes, au niveau de la fonction de correspondance un index sera considéré au détriment d'un autre, non seulement quand il peut être obtenu à partir de ce dernier par le biais de l'utilisation des connaissances du système, mais aussi uniquement dans le cas où il appartient à un chemin de transformation $Path_{Max}$ à plausibilité maximale. La relation de préférence étendue étant basée sur $\prec_R$, elle vérifie les mêmes propriétés que cette dernière:

Propriété 3 *La relation de préférence étendue $\prec_R^{Ext}$ sur les index élargis probables est irréflexive, anti-symétrique et transitive. Autrement dit, elle constitue un ordre partiel strict sur les situations*

abstraites probables.

Il est à noter que l'index $\chi(d)_{Elargi} = \chi(d)^0_{Prob}$ appartient nécessairement à tout chemin de type $Path_{Max}$. Par exemple, dans la figure 3.6 précédente nous avons la relation de préférence suivante:

$$\chi(d)^8_{Prob} \prec^{Ext}_R \chi(d)^7_{Prob}$$

Par contre même si $\chi(d)^9_{Prob} \prec_R \chi(d)^7_{Prob}$ est vrai étant donné que $\chi(d)^9_{Prob}$ est obtenu de $\chi(d)^7_{Prob}$ par utilisation d'une connaissance du système, nous avons:

$$\chi(d)^9_{Prob} \not\prec^{Ext}_R \chi(d)^7_{Prob}$$

En effet, dans cette figure, $\chi(d)^9_{Prob}$ n'appartient pas à un chemin de transformation à plausibilité maximale. Dans cette même figure, si on part de l'index élargi $\chi(d)_{Elargi}$, on en déduit que la situation $\chi(d)^8_{Prob}$ constitue l'index probable élargi le plus préférable au niveau de la fonction de correspondance. Le calcul du degré de pertinence du document se fera ainsi par rapport à cet index élargi, et donc en complète harmonie avec le principe d'incertitude de Rijsbergen.

Il peut exister toutefois plusieurs chemins de transformations $Path_{Max}$ à certitude maximale. Ainsi, contrairement au cas de l'élargissement de l'index originel par le biais des connaissances certaines du système, il peut exister un ensemble de situations pouvant être considérées comme les plus préférables lors de l'élaboration de la fonction de correspondance. Le résultat du calcul du degré de pertinence peut en effet être le même pour plusieurs situations probablement pertinentes à la requête. La figure 3.7 suivante montre deux chemins de transformation à plausibilité maximale. Dans cette figure, nous avons indiqué les connaissances utilisées par le système pour faire passer un index d'un état à un autre. Supposons que $\chi(d)^5_{Prob}$ et $\chi(d)^7_{Prob}$ sont pertinents pour la requête utilisateur. Par R^α_l, nous mentionnons le fait que la connaissance portant le label l et ayant α pour degré de plausibilité a été utilisée par une règle incertaine du système pour faire évoluer une situation abstraite d'un état à un autre. Pour les deux chemins à plausibilité maximale en question, nous avons les degré de pertinence suivants:

$$\sigma(\{\chi(d)_{Elargi}, \chi(d)^2_{Prob}, \chi(d)^5_{Prob}) = 0.7 \times 0.5 = 0.35$$

et

$$\sigma(\{\chi(d)_{Elargi}, \chi(d)^3_{Prob}, \chi(d)^7_{Prob}) = 0.5 \times 0.7 = 0.35$$

Les deux situations probables $\chi(d)^5_{Prob}$ et $\chi(d)^7_{Prob}$ ont donc le même degré de pertinence. Nous énonçons alors la propriété suivante:

Propriété 4 (Multiplicité des Situations Probables les Plus Préférables) *Pour un index élargi $\chi(d)_{Elargi}$, il peut exister plusieurs chemins de transformation $Path_{Max}$ à plausibilité maximale. Au bout de chaque chemin se trouve un index élargi faisant partie de l'ensemble des situations élargies les plus préférées d'un document d lors du processus de correspondance.*

Le problème est de savoir si pour un document donné, il est possible de trouver tous ses chemins de transformation $Path_{Max}$ à certitude maximale. Pour ce faire, il faudra déjà calculer la valeur de certitude de tous ses chemins de transformation aboutissant à des situations pertinentes pour la requête. Cette dernière tâche ne sera possible à réaliser que si ces chemins de transformations sont finis, au sens qu'ils n'engendrent pas de chaînes de transformation infinies. Cela revient au problème de la décidabilité de l'approche. Dans la suite, nous prouvons que l'approche est décidable, uniquement dans le cas où le langage d'indexation relationnel utilisé est fini.

Commençons tout d'abord par noter que même s'il existe plusieurs chemins de transformation possibles pour les raisons que nous avons évoqué au début de cette section (voir page 118), chaque chemin de transformation est fini quand le langage d'indexation relationnelle est fini. Nous avons en effet le théorème suivant:

Théorème 5 (Unicité de la Situation Finale d'un Chemin de Transformation) *Si le langage d'indexation relationnelle $\mathcal{L}$ de la définition 3.2.1 (page 81) est fini, c'est-à-dire que l'ensemble des situations $\mathcal{S}$ est fini, alors il existe nécessairement pour chaque chemin de transformation p un élément minimal $\chi(d)^p_{Prob}$ unique par rapport à $\prec_R$:*

$$\forall \chi(d)^i_{Prob} \in p, \quad \chi(d)^p_{Prob} \prec_R \chi(d)^i_{Prob}$$

Preuve 3 *La démonstration ressemble à celle donnée pour le théorème 1 de la page 88. En effet, les connaissances incertaines n'ont pas de répercussions à vrai dire sur la façon dont les informations s'ajoutent. Il s'agit toujours de préciser certaines valeurs de vérité des termes d'indexation du langage, sauf que dans ce cas la plausibilité de cette précision est tenue en compte lors de l'évaluation de pertinence. L'utilisation d'une connaissance incertaine a ainsi le même effet que celui produit par l'utilisation d'une connaissance certaine. Il s'agit toujours de faire passer la valeur de vérité d'un terme d'indexation de la valeur "inconnu" à la valeur "vrai" ou "faux". Si le langage est fini, la précision de ces valeurs de vérité se répète jusqu'au cas extrême où toutes les valeurs de vérité des termes d'indexation du langage sont précisées. À partir de ce moment, étant donné que $\prec_R$ est irréflexive, il n'est plus possible d'ajouter de l'information, car l'index contient déjà tous les termes du langage. Pour les mêmes raisons que celles citées pour le cadre du théorème 1, il n'est pas également possible d'avoir des cycles dans un chemin de transformation p ($\prec_R$ étant un ordre partiel strict). Il ne peut donc exister de chemin de transformation infinie.*

Il s'en suit que dans le cas d'un langage d'indexation fini, il est toujours possible de décider si un chemin de transformation engendre une situation pertinente pour la requête, et si oui d'évaluer le degré de pertinence du document correspondant. Cependant, pour que l'approche soit décidable dans sa globalité, il faudra connaître toutes les valeurs de plausibilité des autres chemins de transformation avant de décider si un chemin donné est à plausibilité maximale ou pas. Il est au fait facile de constater le théorème suivant:

Théorème 6 (Finitude des Chemins de Transformation) *À partir d'une situation abstraite, il n'existe qu'un nombre fini de chemins de transformation "p" possibles.*

Preuve 4 *Les chemins de transformation sont construits à partir de l'ensemble des connaissances du système. Ce dernier étant fini pour un domaine d'application particulier, à partir d'une situation abstraite donnée, il existe un maximum de possibilités de dérivation égal à la cardinalité de cet ensemble. Le nombre des chemins de transformation pour une situation quelconque est ainsi fini.*

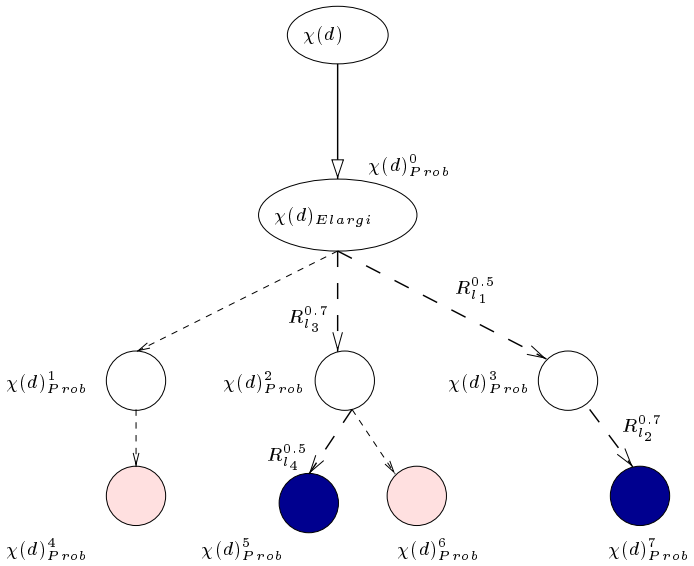


Figure 3.7. Multiplicité des chemins à certitude maximale

Nous pouvons alors énoncer le théorème suivant:

Théorème 7 (Décidabilité de l'Extension au Règles incertaines) *Si le langage d'indexation relationnelle $\mathcal{L}$ est fini, le système de dérivation relationnelle $\mathcal{S}^{Ext}_{\Re}$ étendu par des règles incertaines est décidable.*

Preuve 5 *La preuve découle directement des deux théorèmes précédents. En effet, les chemins de transformations étant finis, et leur nombre limité, il est possible pour chaque document d'identifier tous ses chemins de transformations aboutissant à des situations probables satisfaisants la requête, de cerner ensuite parmi eux ceux à certitude maximale, enfin de spécifier le degré de pertinence du document à la base de ces chemins à certitude maximale. Ainsi, pour chaque document, on est certain que la processus de recherche s'arrêtera pour nous donner une réponse qu'elle soit positive ou négative.*

Comme nous le verrons dans la section 3.5.2 (page 128), si le système est tenu de justifier à l'utilisateur les raisons de ses jugements de pertinence, il est utile de s'interroger sur la démarche à suivre quand il existe plusieurs chemins de transformation à certitude maximale. Le système pourra par exemple, choisir comme justification le premier chemin $Path_{Max}$ qu'il retrouvera dans l'arborescence des transformations, ou encore choisir le chemin ayant le plus petit nombre de pas d'inférence (le chemin le plus court). Il pourra enfin donner toutes les justifications possibles, c'est-à-dire retourner à l'utilisateur la trace d'exécution de tous les chemins $Path_{Max}$. De la nature de la justification souhaitée par l'utilisateur, dépendra le choix de la stratégie d'explication adoptée par le système.

a) Utilisation des Règles par Défaut

Dans la section précédente, nous avons montré un exemple où les connaissances peuvent prendre la forme d'un défaut. Les défauts sur les relations sont représentés comme dans les logiques par défaut de la manière suivante:

Les Connaissances par Défaut

$$(\mathrm{l})\ \frac{\beta \quad : \quad \rho}{\gamma}\,[\alpha]$$

où $\beta, \rho, \gamma \in \mathcal{L}$, $\alpha \in]0..1]$ et l désigne le label de la connaissance. En termes de la théorie des situations, les connaissances de ce type peuvent être représentées comme des relations d'endiguement particulières:

$$\beta : \rho \overset{l}{\rightarrow}{}^{\alpha}_{r} \quad \gamma$$

Définition 3.5.4 (Langage de Dérivation par Défaut) *Étant donné le langage d'indexation relationnelle $\mathcal{L}$, le langage de dérivation de pertinence $\mathcal{P}^{Ext}_{df}(\mathcal{L})$, étendu à la prise en compte des connaissances par défaut du domaine est défini comme étant le plus petit sur-ensemble de $\mathcal{P}(\mathcal{L})$ tel que:*

1. *si $\varphi, \rho, \psi \in \mathcal{L}$ alors $\varphi : \rho \overset{l}{\rightarrow}{}^{\alpha}_{r}\ \psi \in \mathcal{P}^{Ext}_{df}(\mathcal{L})$*
2. *si $S, T \in \mathcal{S}_{\mathcal{L}}$ alors $S \boxed{\sim}\!\!\rightarrow^{\alpha}_{R} T$, $S \not\boxed{\sim}\!\!\rightarrow^{\alpha}_{R} T \in \mathcal{P}^{Ext}_{df}(\mathcal{L})$*

où $\mathcal{L}$ et $\mathcal{S}_{\mathcal{L}}$ sont respectivement le langage d'indexation relationnelle et le langage des situations.

Le système de dérivation $\mathcal{S}_{\Re}$ est quant à lui étendu par les trois règles de dérivation incertaines ci-dessous introduites. De plus, nous avons la propriété suivante:

Propriété 5 *Si $S \boxed{\sim}\!\!\rightarrow_{R} T$ alors $S \boxed{\sim}\!\!\rightarrow^{\alpha}_{R} T$ avec $\alpha = 1$*

$S \boxed{\sim}\!\!\rightarrow^{\alpha}_{R} T$ signifie que la situation S est à-propos de la situation T avec un degré de plausibilité α. Ce degré est déduit du degré de plausibilité de la connaissance incertaine utilisée.

La première règle incertaine ajoutée à $\mathcal{S}_{\Re}$ exprime le fait que tous les défauts constituent des décisions de pertinence plausibles ayant comme degré de pertinence α.

Défaut Élémentaire

$$\frac{\phi : \varphi \overset{l}{\rightarrow}{}^{\alpha}_{r}\ \psi}{\{\phi\} \boxed{\sim}\!\!\rightarrow^{\alpha}_{R} \{\psi\}}$$

La deuxième règle exprime le fait que si une situation S contient un relon φ_1 qui fait partie d'un *pré-requit* d'un défaut, et que tous les autres relons de S ne sont pas en contradiction avec la *justification* ρ de ce défaut, alors il est possible d'ajouter la conséquence du défaut à la situation

S, et faire dériver un index élargi avec une plausibilité α. La relation de pertinence étant préservée après cet ajout.

Défaut Positif

$$\frac{\varphi_1{:}\rho \xrightarrow{l}{}^{\alpha}_{r} \psi \quad S \boxminus\!\!\!\rightarrow^{\alpha}_{R} T \quad S \equiv \{\varphi_1,\ldots,\varphi_n\} \quad \varphi_2 \not\perp^{p}_{r} \rho \quad \ldots \quad \varphi_n \not\perp^{p}_{r} \rho}{S \cup \{\psi\} \boxminus\!\!\!\rightarrow^{\alpha}_{R} T}$$

La dernière règle montre qu'il suffit qu'un seul relon de la situation S soit en contradiction avec le pré-requit d'un défaut pour que l'ajout de la conséquence de ce dernier à une situation T permet d'affirmer avec un degré de plausibilité α que les deux situations S et T (ainsi étendue) ne sont plus liées avec une relation de pertinence.

Défaut Négatif

$$\frac{\varphi_1{:}\rho \xrightarrow{l}{}^{\alpha}_{r} \psi \quad S \boxminus\!\!\!\rightarrow^{\alpha}_{R} T \quad S \equiv \{\varphi_1,\ldots,\varphi_n\} \quad \varphi_i \perp^{p}_{r} \rho}{S \not\boxminus\!\!\!\rightarrow^{\alpha}_{R} T \cup \{\psi\}}$$

b) Utilisation des Règles Incertaines

Un exemple de transformation incertaine a été introduit au début de la section 3.5.1. Deux types de connaissances incertaines sont généralement considérées dans la littérature [Bru93, Che94, OC96b, Nie90]:

Élargissement

$$(\text{élargir}_i)\frac{\langle\langle \mathit{relation}, e_1, e_2, \ldots, e_j, \ldots, e_n; 1\rangle\rangle \quad e_i \rightarrow e_j}{\langle\langle \mathit{relation}, e_1, e_2, \ldots, e_i, \ldots, e_n; 1\rangle\rangle}\,[\alpha]$$

Contenir

$$(\text{englober}_i)\frac{\langle\langle \mathit{relation}, e_1, e_2, \ldots, e_i, \ldots, e_n; 1\rangle\rangle}{\langle\langle \mathit{Nouvelle\text{-}Rel}, \langle\langle \mathit{relation}, e_1, e_2, \ldots, e_i, \ldots, e_n; 1\rangle\rangle, e; 1\rangle\rangle}\,[\alpha]$$

La présence de l'indice i s'explique par le fait qu'à chacune des deux propriétés citées ci-dessus correspond plusieurs relations d'endiguement incertaines, selon les connaissances mises en jeu. L'élargissement est l'inverse de la propriété de restriction. Ainsi, si nous reprenons l'exemple de la page 99, nous pouvons avoir la dérivation incertaine suivante:

Exemple de Restriction

$$(\text{élargir}_{85})\frac{\langle\langle \mathit{position}, \textsc{Personne}, \textsc{Assis}; 1\rangle\rangle \quad \textsc{Femme} \rightarrow \textsc{Personne}}{\langle\langle \mathit{position}, \textsc{Femme}, \textsc{Assis}; 1\rangle\rangle}\,[0.5]$$

Dans cet exemple, un utilisateur cherchant des images dans lesquelles apparaît une femme assise, serait possiblement intéressé par une image où l'on voit une personne assise (pouvant d'ailleurs être une femme).

La deuxième propriété est la duale de la relation d'endiguement (ED) définie dans la page 98. On considère ainsi qu'un document contenant un relon particulier pourra intéresser un utilisateur cherchant une situation dans laquelle ce relon participe à une relation sémantique donnée. Par exemple, un document traitant des effets de la pollution en général, pourra vraisemblablement être utile pour une requête cherchant les effets de la pollution engendrés par les métaux:

Exemple

$$(\text{englober}_8)\frac{\langle\langle \textit{Provenant-de}, \textsc{Effets}, \textsc{Pollution}; 1\rangle\rangle}{\langle\langle \textit{Engendré-par}, \langle\langle \textit{Provenant-de}, \textsc{Effets}, \textsc{Pollution}; 1\rangle\rangle, \textsc{Métaux}; 1\rangle\rangle}\,[0.4]$$

Les deux propriétés ci-dessus citées concernent plutôt un raisonnement plausible sur les arguments des relations. Elles adressent donc les propriétés des objets, plus que celles des relations. Cela explique le fait qu'elles soient très présentes dans les systèmes de recherche d'informations actuels [Bru93, Che94]. Dans le cadre de notre approche relationnelle, à ces deux types de relations d'endiguement s'ajoutent les connaissances incertaines sur les propriétés des relations, qu'elles soient mathématiques, sémantiques, ou exprimant des agencements plausibles entre relations. Par exemple, il est souvent remarqué que le verbe *aimer* n'est pas généralement symétrique. Il n'empêche que si un document décrit une situation dans laquelle *Jean aime Marie*, alors il y a des fortes chances que ce document soit pertinent pour une requête cherchant les situations dans lesquelles *Marie aime Jean*:

Symétrie Plausible

$$(\text{Aimer})\frac{\langle\langle \textit{aime}, \text{Jean}, \text{Marie}; 1\rangle\rangle}{\langle\langle \textit{aime}, \text{Marie}, \text{Jean}; 1\rangle\rangle}\,[0.8]$$

D'une façon plus générale, toutes les propriétés que nous avons précédemment défini pour l'extension de la relation d'endiguement aux propriétés mathématiques des relations, à leurs liens sémantiques et leurs comportements peuvent avoir un dual plausible. Par la suite nous adoptons la notation suivante pour une relation d'endiguement incertaine:

$$\phi \overset{l}{\rightarrow}{}^{\alpha}_{r}\ \psi$$

ce qui signifie que ϕ peut dériver ψ avec une certitude de valeur α. Cette dérivation a le label unique l. La valeur de certitude est définie dans l'intervalle]o,1]. En particulier $\phi \rightarrow_r \psi$ signifie que $\phi \rightarrow^1_r \psi$. Il s'agit du cas que nous avons jusqu'ici traité, à savoir celui où la relation d'endiguement est toujours valide. Les règles relatives à des relations d'endiguement certaines ne sont pas munies de labels. La raison est que, comme nous le verrons dans la section suivante, les labels sont introduits pour détecter les règles d'inférence incertaines ayant des effets nuisibles sur la pertinence du point de vue de l'utilisateur. Les règles certaines étant valides dans tous les contextes de l'application, elles n'engendrent que des connaissances implicitement vraies. Ces dernières ne peuvent avoir qu'une influence positive sur les décisions de pertinence.

Afin de tenir compte des règles de dérivation probables avec un degré de plausibilité α, le langage de dérivation $\mathcal{P}(\mathcal{L})$ est étendu de la façon suivante:

Définition 3.5.5 (Langage de Dérivation Incertain) *Étant donné le langage d'indexation relationnelle $\mathcal{L}$, le langage de dérivation de pertinence $\mathcal{P}^{Ext}_{prob}(\mathcal{L})$, étendu à la prise en compte des*

connaissances probables du domaine est défini comme étant le plus petit sur-ensemble de $\mathcal{P}(\mathcal{L})$ tel que:

1. *si $\varphi, \psi \in \mathcal{L}$ alors $\varphi \overset{l}{\rightarrow}_r^{\alpha} \psi \in \mathcal{P}_{prob}^{Ext}(\mathcal{L})$*
2. *si $S, T \in \mathcal{S}_{\mathcal{L}}$ alors $S \boxed{\sim}\!\!\rightarrow_R^{\alpha} T \in \mathcal{P}_{prob}^{Ext}(\mathcal{L})$*

avec $\mathcal{L}$ et $\mathcal{S}_{\mathcal{L}}$ les langages précédemment définis.

Comme pour le cas de l'ajout des règles par défaut, $S \boxed{\sim}\!\!\rightarrow_R^{\alpha} T$ signifie que la situation S est à-propos de la situation T avec un degré de plausibilité α. Ce dernier est déterminé par la connaissance incertaine utilisée.

Les trois règles suivantes remplacent ainsi les règles d'*Endiguement Élémentaire*, d'*Endiguement Faible* et d'*Endiguement avec Retenue* du système de dérivation de pertinence $\mathcal{S}_{\Re}$:

Incertitude Élémentaire

$$\frac{\phi \overset{l}{\rightarrow}_r^{\alpha} \psi}{\{\phi\} \boxed{\sim}\!\!\rightarrow_R^{\alpha} \{\psi\}}$$

Cette règle est une extension de l'*Endiguement Élémentaire* à la prise en compte des connaissances incertaines. Dans le cas où la connaissance est valide, la plausibilité de la dérivation est égale à 1.

La règle suivante est une extension de la règle d'*Endiguement Faible* aux cas où les propriétés des relations ne sont pas forcément valides dans le domaine de l'application:

Endiguement Faible Incertain

$$\frac{\phi \overset{l}{\rightarrow}_r^{\alpha,f} \psi \quad S \boxed{\sim}\!\!\rightarrow_R^{\alpha} T \cup \{\phi\}}{S \boxed{\sim}\!\!\rightarrow_R^{\alpha} T \cup \{\psi\}}$$

Dans ce cas, $\phi \overset{l}{\rightarrow}_r^{\alpha,f} \psi$ exprime le fait que le relon ψ est obtenu de ϕ par une règle d'endiguement faible incertaine.

Enfin, certaines compositions de relations sont plausibles, mais favorisent parfois l'obtention de plusieurs documents pouvant intéresser l'utilisateur:

Endiguement Incertain avec Retenue

$$\frac{\phi \overset{l}{\rightarrow}_r^{\alpha,\oplus} \psi \quad S \boxed{\sim}\!\!\rightarrow_R^{\alpha} T \cup \{\phi\}}{S \boxed{\sim}\!\!\rightarrow_R^{\alpha} T \cup \{\psi\} \cup \{\phi\}} \qquad \frac{\phi_1 \odot \phi_2 \odot \ldots \odot \phi_n \overset{l}{\rightarrow}_r^{\alpha,\oplus} \varphi \quad S \boxed{\sim}\!\!\rightarrow_R^{\alpha} T \cup \{\phi_1, \phi_2, \ldots, \phi_n\}}{S \boxed{\sim}\!\!\rightarrow_R^{\alpha} T \cup \{\phi_1, \phi_2, \ldots, \phi_n\} \cup \{\varphi\}}$$

Ici, $\phi \overset{l}{\rightarrow}_r^{\alpha,\oplus} \psi$ stipule que le relon ψ est ajouté à la situation S contenant le relon ϕ quand il peut être déduit de ce dernier avec une plausibilité α. Il s'agit d'étendre la règle d'*Endiguement avec Retenue* définie pour les connaissances certaines.

3.5.2 Utilisation des Labels de Règles

Les labels peuvent s'avérer utiles pour expliquer le processus de recherche ou pour prendre en compte les préférences des utilisateurs lors d'une session d'interrogation. Grâce aux labels, il est possible de pouvoir identifier, pour un document donné, les règles que le système a appliqué pour établir sa décision de pertinence. Nous rappelons en effet, qu'à chaque fois que le système de dérivation de pertinence utilise une connaissance incertaine, il garde une trace de son label et tient compte de sa plausibilité pour le calcul du degré de pertinence (voir page 119). Le chemin de transformation $Path_{Max}$ incluant le plus de certitude possible étant choisi pour le calcul du degré de pertinence, nous aurons à la fin une liste des labels correspondant aux règles que le système a appliqué lors de l'élaboration de ce chemin. Appelons, $trace_{d_i}$, la liste des labels correspondant au chemin $Path_{Max}$ d'un document d_i. Pour chaque document jugé pertinent par le système, il est ainsi possible de cerner les connaissances que ce dernier a utilisé pour *justifier* sa décision de pertinence.

Cette caractéristique de notre modèle à expliquer ses décisions de pertinence rappelle les avantages des systèmes experts [Bla86]. Généralement, ces systèmes ne se contentent pas uniquement de fournir un diagnostic, mais ils sont souvent capables de le justifier. Dans le domaine de la logique, l'utilisation des labels (ou étiquettes) a été largement discutée par Gabbay [Gab94]. Nous partageons ainsi avec d'autres [Con94], le fait qu'un système de recherche d'informations devrait fournir un minimum d'explications à l'utilisateur, surtout pour ce qui est des décisions de pertinence plausibles. L'utilisateur pourra ainsi détecter les connaissances incertaines qui lui paraissent inadéquates. Par conséquent, il pourra ensuite demander explicitement au système d'ignorer ces connaissances lors de l'évaluation de ses prochaines requêtes.

Par ailleurs, une forme de justification dans un système de recherche d'informations est également utile pour ses concepteurs. Ainsi, par rapport aux résultats prévus sur une collection test, il est possible de tirer des conclusions, quant aux problèmes que poseraient certaines connaissances incertaines. Comme dans les systèmes experts, les développeurs pourraient se servir des traces d'exécution du système pour attribuer des degrés de confiance aux diverses connaissances utilisées.

Nous pouvons également imaginer une autre forme d'utilisation des labels en recherche d'informations. Prenons par exemple, le cas de la figure 3.8 suivante. Les degrés de pertinence des documents d_1, d_2 et d_3 sont calculés par rapport à leur chemin de transformation $Path_{Max}$. Considérons les traces $trace_{d_i}$ associées à ces chemins. Il s'agit des traces suivantes:

Document	**trace**$_{d_i}$
d_1	$[l_1, l_6, l_3, l_4]$
d_2	$[l_2, l_6, l_9]$
d_3	$[l_{10}, l_7, l_2]$

Supposons que pour une requête utilisateur donnée, ces trois documents sont jugés pertinents par le système. Il sont par conséquent affichés à l'utilisateur selon leur ordre de pertinence. De plus, imaginons que l'interface du système est munie d'un procédé de bouclage de pertinence (*relevance feedback*) [Jr.71], où le système reformule la requête en fonction du "feedback" fourni par l'utilisateur sur les documents retrouvés. Il existe plusieurs "feedback" possibles. La forme la plus classique étant de demander à l'utilisateur de désigner les documents pertinents et non pertinents de la réponse

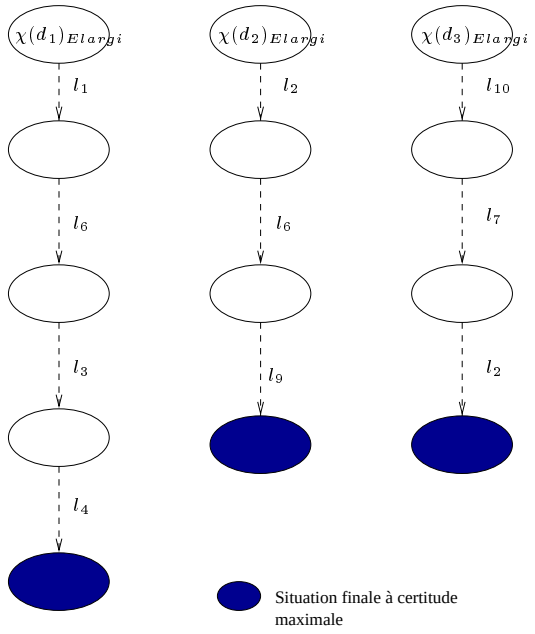


Figure 3.8. Exemple de transformation d'index

(jugements de pertinence binaires). Le système demande ainsi à l'utilisateur de choisir parmi les documents affichés ceux qu'il juge réellement intéressants pour son besoin en informations.

Dans notre cas, nous supposons que le seul document jugé pertinent par l'utilisateur est d_1. Dans ce cas, nous devons analyser pour quelles raisons l'utilisateur semble rejeter deux documents, jugés pourtant comme pertinents par le système. Les labels peuvent alors être utilisés pour donner une explication possible à ce cas de figure.

Ainsi, pour les deux documents rejetés par l'utilisateur, nous remarquons l'utilisation de la connaissance portant le label l_2. Cette dernière n'apparaît pas dans la trace associée au document d_1. Nous pouvons ainsi déduire qu'il est fort probable que l'utilisateur dans son contexte de recherche ne semble pas apprécier l'utilisation de cette connaissance. Le système peut alors utiliser ce constat lors de la reformulation de la requête, en excluant la connaissance portant le label l_2 de sa base de connaissances, durant tout le reste de la session de recherche. Cette approche pourra avoir deux conséquences possibles sur la suite des évènements:

(1): Le retrait de la connaissance portant le label l_2 ne permettra plus au système de retrouver les deux documents en question (d_2 et d_3). Dans ce cas, les deux documents ne seront plus affichés à l'utilisateur.

(2): Le chemin $Path_{Max}$ n'étant qu'une possibilité parmi autres engendrant une situation pertinente à la requête, il se peut que le retrait de la connaissance ayant le label l_2 n'aura pour conséquence que la mise en relief d'un autre chemin à certitude plus petite. En d'autres termes, il peut exister un autre chemin de transformation pour les documents en question, n'utilisant pas la connaissance ayant le label l_2 et aboutissant, quand même, à une décision de pertinence positive.

Dans les deux cas, l'approche est bénéfique. Si pour le premier cas, son intérêt est évident, le cas (2) mérite une explication plus poussée. Ainsi, supposons que, même après le retrait de la connaissance ayant le label l_2, les documents d_2 et d_3 sont jugés comme plausiblement pertinents par le système, à travers l'utilisation d'autres chemins de transformation. Le degré de pertinence de ces deux documents sera nécessairement moindre par rapport à leur degré de pertinence avant le retrait de la connaissance l_2 en question. En effet, $Path_{Max}$ est par définition le chemin où les transformations du document associé sont les plus plausibles. Les documents d_2 et d_3 auront ainsi des degrés de pertinence plus petits si d'autres chemins sont utilisés. Après la reformulation de la requête, les documents d_2 et d_3 auront un degré de pertinence plus faible. Ils seront ainsi repoussés plus bas dans l'ordre des documents pertinents, ou simplement ils peuvent être supprimés si leur degré de pertinence ne dépasse pas un seuil prédéfini par le système[29]. Dans ce cas, le système affichera en premier lieu les documents susceptibles de mieux intéresser l'utilisateur.

Le processus est ensuite répété, l'utilisateur désignera toujours les documents d_2 et d_3 comme indésirables mais, cette fois-ci, d'autres documents peuvent s'afficher à l'utilisateur, car la suppression de la connaissance l_2 pourrait les promouvoir à un degré de plausibilité supérieur au seuil. Il est ainsi probable que l'utilisation de la connaissance l_2 et de son degré de plausibilité font qu'en premier lieu, ces documents ne soient pas affichés. Grâce à un autre retour de l'utilisateur, le système pourra détecter d'autres connaissances indésirables par l'utilisateur et les supprimer ensuite de la session de travail[30].

Le lecteur se demandera certainement pourquoi nous continuons à afficher des documents, jugés pourtant indésirables par l'utilisateur. Nous voulons par ce fait, détecter toutes les connaissances jugées inadéquates par l'utilisateur. Ainsi, si un document continue à s'afficher alors que la connaissance l_2 est supprimée, c'est qu'il existe potentiellement une autre connaissance impertinente qui fait en sorte que le document soit toujours affiché à l'écran. En effet, la connaissance portant le label l_2 peut faire partie de tout un ensemble de connaissances qui soient en contradiction avec les préférences d'un utilisateur. Par le procédé que nous avons décrit au paragraphe précédent, cet ensemble a toutes les chances d'être détecté, en analysant les résultats dérivés par le système après la suppression de la connaissance l_2 et en demandant encore une fois son avis à l'utilisateur.

Plus formellement, la détection des connaissances non conformes aux préférences de l'utilisateur peut être réalisée de la manière suivante:

Définition 3.5.6 (Élimination de Connaissances) *Supposons que nous avons n (resp. m) réponses jugées pertinentes (resp. non pertinentes) par l'utilisateur. Les labels Lab_{Ret} des connaissances à retirer de la base sont déterminés comme suit:*

$$Lab_{Ret} = \bigcap_{i \in m} trace_{d_i} \Leftrightarrow \bigcup_{j \in n} trace_{d_j}$$

Pour tous les m documents jugés non pertinents par l'utilisateur, le système tente de retrouver les labels communs aux chemins de transformation $Path_{Max}$ qui leurs sont associés (c'est une intersection de listes de labels). Les connaissances relatives à ces labels sont considérées comme potentiellement non conformes aux préférences de l'utilisateur. Elles sont en effet utilisées dans tous

29. Dans la pratique, un seuil est généralement fixé. Si le degré de pertinence du document est au-dessous de de ce seuil, le document correspondant n'est pas jugé pertinent.

30. Une session de travail pourra comporter plusieurs requêtes de la part du même utilisateur. Il s'agit de la période couvrant tout ce que fait l'utilisateur depuis l'instant où il se connecte au système jusqu'à l'instant où il le quitte.

les chemins de transformation des documents rejetés par ce dernier. Les connaissances réellement éliminées sont celles qui n'ont jamais été utilisées dans les chemins de transformation des documents jugés utiles par l'utilisateur (c'est une union de labels). En effet, si une connaissance a été utilisée pour déduire un "bon" document selon l'utilisateur, et qu'elle apparaît en même temps dans la liste des connaissances potentiellement non conformes à ce dernier, alors il n'y a pas de raisons particulières pour la considérer comme étant à l'origine des mauvaises réponses obtenues.

Dans notre exemple, le label l_2 fait partie des traces des documents rejetés par l'utilisateur (d_1 et d_2). Comme l_2 ne fait pas partie de la trace de d_1, la connaissance qui lui est associée est exclue de la base selon la définition ci-dessus introduite. Il est à remarquer toutefois, que notre contrainte de faire supprimer une connaissance de la base, uniquement dans le cas où le label de celle-ci n'apparaît dans aucune trace des n documents pertinents, est assez forte. Nous pouvons alléger la contrainte de la façon suivante:

$$Lab_{Ret} = \bigcap_{i \in m} trace_{d_i} \Leftrightarrow \bigcap_{j \in n} trace_{d_j}$$

Avec cette formule, une connaissance douteuse avec un label l sera supprimée de la base, sauf si l fait partie de tous les chemins de transformations des documents déclarés pertinents par l'utilisateur. Ainsi, même si la connaissance a été utilisée pour dériver certains "bons" documents, elle sera retirée de la base. Il est difficile de savoir quelle approche est la plus adéquate pour le retrait des connaissances douteuses. Il revient en effet à l'expérimentation de déterminer laquelle des deux formules ci-dessus introduites est plus à même à tenir compte des préférences de l'utilisateur. Dans cette section, nous avons juste mis l'accent sur l'utilité des labels à favoriser une recherche interactive avec l'utilisateur.

3.5.3 Un Système Multifacette

Tout au long de ce chapitre, nous avons considéré que l'index $\chi(d)$ d'un document correspond à une situation unique formée d'un ensemble de relons. L'application des règles d'inférence strictes ou plausibles permet de faire passer cette situation vers d'autres situations abstraites possibles pouvant correspondre à la représentation de la requête. Cette vue fut originellement adoptée par Nie [Nie90], puis suivie par d'autres comme Sembok & van Rijsbergen dans [SvR93], ou par Amati & Kerpedjiev dans [AK92]. Cependant, nous pensons que cette approche est inadaptée pour des documents complexes comme les images ou la vidéo. Par exemple, la description d'une image par un ensemble de relons, implique souvent une perte d'informations, car il s'agit de remplacer une image riche en informations (couleurs, textures, etc.) par des descripteurs sémantiquement incomplets, adressant le plus souvent un aspect particulier de l'image, à savoir son contenu sémantique. Il existe en effet plusieurs autres interprétations possibles d'une image, rendant très subjective sa description. Cela dépend principalement de la perception d'un sujet dans un contexte d'application donné, ses connaissances générales et ses besoins particuliers de recherche. Les interprétations possibles sont nombreuses et chaque sujet peut donner une description de l'image différente des autres. Une personne peut s'intéresser par exemple à la forme des objets dans l'image, une autre appréciera les couleurs et leurs agencement, enfin une autre pourrait trouver un contenu émotionnel dans l'image considérée. Il existe ainsi plusieurs *points de vue* à partir desquels un utilisateur peut estimer la valeur de pertinence fournie par le système. Ces points de vue sont reliés au document lui-même et correspondent à des différentes vues de ce dernier. Dans ce qui suit nous les désignerons par le terme *facette* [OC96b].

a) Les Facettes

Une *facette* permet surtout de représenter et de décrire le plus fidèlement possible les documents complexes (images, vidéos, documents structurés, ...). Chaque facette correspond alors à une vue du document selon laquelle il pourra être décrit par le biais d'un ensemble homogène de termes. En effet, on pourra remarquer que les termes d'indexation n'ont pas tous la même nature: il y a ceux relatifs aux contenu sémantique du document, ceux correspondant à la texture, aux attributs externes, etc. Les utilisateurs peuvent faire référence à ces facettes lors de la formulation de leur besoin.

Il existe une façon simple pour détecter la notion de facette dans un document. Supposons qu'étant donné l'ordre de pertinence fourni par le système, un utilisateur est capable de décider si un document est plus pertinent ou plus utile qu'un autre document. L'hypothèse suivante peut être introduite:

Hypothèse 5 (Facette) *Si un utilisateur établie qu'un document d_1 est plus pertinent qu'un document d_2, et en même temps que d_2 est plus pertinent que d_1, alors nous pouvons expliquer cette situation par le fait qu'il existe deux facettes différentes f_1 et f_2, dont l'utilisateur s'est servi pour ordonner différemment les deux documents.*

La notion d'interprétation introduite dans dans la définition 3.1.1 de la page 69, est fortement liée à la notion de facette. En effet, les facettes sont définies comme des points de vue différents sur les documents: une interprétation des expressions d'indexation constitue ainsi une vue particulière du document. Nous partageons donc les opinions exprimées par Meghini et al. dans [MSST93] ou par Mechkour dans [Mec95b, Mec95c] sur la nécessité d'inclure la notion de facette dans les descriptions des documents. Afin d'aboutir à la plus complète des descriptions, nous considérons un ensemble de facettes, chacune correspondant à des descriptions des divers éléments de ce document. Notre modèle d'indexation relationnel peut ainsi être étendu comme nous le verrons dans la section suivante.

b) Un Système Relationnel Multifacette

Avant d'introduire formellement ce modèle, illustrons à travers un exemple comment le système devra fonctionner avec les facettes.

Exemple 30 *Reprenons l'exemple 5 de la page 71. Supposons que ce document peut avoir trois facettes. La première décrit l'histoire véhiculée par le document (le contenu sémantique tel qu'il peut être compris par une personne consultant le document), la seconde décrit le contexte général de cette histoire, enfin la troisième facette précise le lieu où se produit l'histoire.*

$\chi(d)$	*Facette* 1	*Facette* 2	*Facette* 3
	{*Aventures de Tom et Jerry, Tom poursuit Jerry dans un Parc, Échappée de Jerry* }	{*Enfant jouant avec Ballon, Oiseaux sur Arbres, Personnes sur Chaises*}	{*Action dans un Jardin, Aventure à Londres* }

Un document sera déclaré pertinent pour une requête, si l'une de ses facettes est pertinente pour la requête en question. Par exemple, une requête à-propos d'"actions se déroulant dans un jardin" *retrouvera le document d de cet exemple, étant donné que sa facette 3 est pertinente pour la requête.*

Définition 3.5.7 (Un Système Relationnel Multifacette) *Soient $\mathcal{L}$ le langage d'indexation relationnelle de la définition 3.2.1, et $\mathcal{S}_{\Re}^{fct}$ le système relationnel de dérivation étendu avec la notion de facette. Soit $\mathcal{D}$ l'ensemble des documents du corpus, avec $d \in \mathcal{D}$. Nous proposons le système relationnel multifacette suivant:*

- *L'index $\chi(d)$ du document d est associé à un ensemble de facettes.*
- *Une facette correspond à un ensemble de relons dans $\mathcal{L}$, i.e, $\chi(d) \in 2^{\mathcal{L}} \times 2^{\mathcal{L}} \times \ldots \times 2^{\mathcal{L}}$.*
- *Une requête q est représentée comme une situation dans $\mathcal{L}$ et peut exprimer des informations ayant trait à différentes facettes du document.*
- *Un document d est dit pertinent par rapport à la représentation $\chi(q)$ de la requête si, et seulement si, il existe au moins une facette f de $\chi(d)$ telle que f est pertinente pour la requête. Nous dénotons par $Rel(d,q)$ le fait que le document d est pertinent pour la requête q. Nous avons alors:*

$$Rel(d,q) \text{ ssi } \exists f \in \chi(d) \mid f \vdash_{\mathcal{S}_{\Re}^{fct}} \Box\!\!\rightsquigarrow_R \chi(q).$$

Cette définition peut être étendue pour incorporer des requêtes complexes, c'est-à-dire des requêtes contenant plusieurs facettes mentionnées par l'utilisateur. Nous pouvons alors s'attendre à des améliorations dans les performances du système de recherche, surtout si les facettes du document sont proches de celles de l'utilisateur.

En termes de la sémantique des mondes possibles, nous adoptons ici une approche plus affinée que celle généralement proposée dans la littérature: au lieu de considérer que l'index du document correspond à un état (ou à un monde), nous considérons plutôt les facettes $\mathcal{F}$ comme des mondes possibles. Autrement dit, à un index de document est associé un ensemble de mondes possibles ou d'états. Le processus de transformation qu'il soit stricte ou plausible reste globalement le même que celui décrit dans les sections précédentes (un ajout d'informations certain ou plausible). Ce qui change réellement c'est plutôt la mesure de pertinence. Ainsi, nous pensons que dans un index de document, certaines facettes peuvent avoir plus d'importance que d'autres, ceci selon la nature du document. Par exemple, le contenu sémantique sera préféré à la forme de présentation [31] dans le cas des documents textuels. Un poids w sera donc associé à chaque facette f du document, représentant la signification de l'information qu'elle décrit.

Définition 3.5.8 (Facettes Pondérées) *Soit $\mathcal{D}$ l'ensemble des documents, avec $d \in \mathcal{D}$. Soit $\mathcal{F}$ l'ensemble des mondes possibles, i.e. l'ensemble de toutes les facettes f. L'index $\chi(d)$ est associé à un ensemble particulier de facettes décrivant le document d. La fonction δ: $\chi(d), f \rightarrow [0..1]$ définie pour chaque facette f de $\chi(d)$ son poids tel que: $\sum_{f \in \chi(d)} \delta(f) = 1$*

Quand une facette f_i ne peut pas directement faire dériver la requête $\chi(q)$, elle peut être transformée en un ensemble de *facettes abstraites* f_{i+1} en appliquant les connaissances certaines et plausibles du système de dérivation. Chaque facette dérivée f_{i+1} est dite être un monde possible pour f_i. Étant donné un index de document $\chi(d) = d_0$, une transformation est appliquée à toutes ses facettes non pertinentes, afin d'obtenir un document abstrait élargi d_1 contenant les facettes dérivées. S'il existe encore dans d_1, des facettes dérivées non pertinentes, nous répétons le processus

31. En anglais on parle de *layout*.

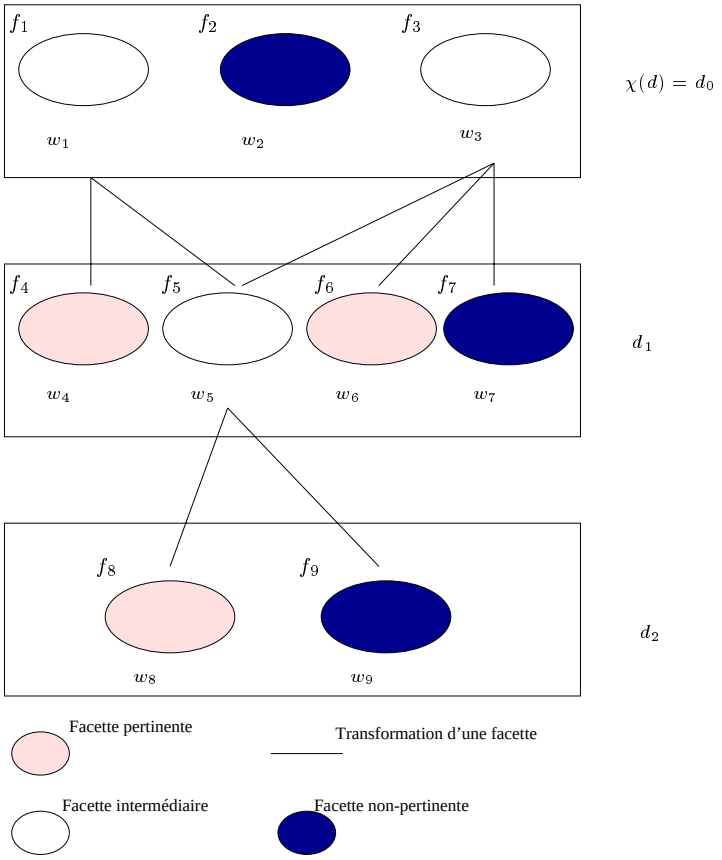


Figure 3.9. Exemple de transformation de facettes

et obtenons un autre document élargi d_2. Le processus de transformation d'une facette prendra fin quand elle devient pertinente pour la requête ou quand il n'est plus possible d'appliquer d'autres transformations (voir figure 3.9). Chaque facette dérivée aura un poids dont l'évaluation peut être inspirée par la théorie d'évidence de Dempster-Shafer comme cela a été présenté par Lalmas et al. dans [LvR93]:

$$\delta(f_{i+1}) = \sum_{f \in w_i^{-1}(f_{i+1})} \sigma(f_i, f_{i+1}) \times \delta(f_i)$$

ici w_i^{-1} est l'ensemble de toutes les facettes dans d_i qui ont été transformés en f_{i+1}, et $\sigma(f_i, f_{i+1})$ est définie comme dans la section précédente.

Le degré de pertinence d'un document d par rapport à une requête $\chi(q)$ est défini de la façon suivante:

Définition 3.5.9 (Degré de Pertinence Multifacette) *Soit $\mathcal{S}_{\Re}^{fct}$ le système relationnel de dérivation étendu aux facettes. Soit d un document et $\chi(d)$ son index correspondant à un ensemble de facettes. De plus, soit $\chi(q)$ la représentation d'une requête unifiée pouvant référencer plusieurs*

facettes du document. Le degré de pertinence de d par rapport à q, dénoté par $D_{Rel}(d,q)$, est donné comme suit:

$$D_{Rel}(d,q) = D_{Rel}(\chi(d),\chi(q)) = D_{Rel}(d_0,\chi(q)) = \bigoplus_{f \in \{f \in d_i \mid f \vdash_{\mathcal{S}_{\Re}^{fct}} \chi(q)\}} \delta(f),$$

où d_i correspond à un index élargi du document et où $\bigoplus$ est un opérateur binaire, qui peut par exemple être l'addition des réels.

Le degré de pertinence d'un document est ainsi calculé en faisant la somme des poids affectés aux facettes dérivées de l'index originel $\chi(d)$ et considérées comme pertinentes pour la requête. Plus, le nombre de ces facettes est grand, et leur poids est important, plus le document sera considéré comme pertinent.

Cette section a présenté une première approche pour la prise en compte des diverses interprétations possibles que peut avoir un document. Chaque interprétation correspond à une description partielle du contenu de ce document. Une telle approche est utile dès lors que les documents adressent des notions subjectives, pouvant être perçues de manières différentes d'un individu à un autre. C'est le cas par exemple des images où de la vidéo. En décrivant le document selon toutes ses facettes, on privilégie une indexation précise et complète, limitant ainsi les cas de silence. Plusieurs systèmes de recherche d'images incluent déjà cette notion de facette, en permettant aux utilisateurs d'adresser plusieurs caractéristiques d'une image à la fois. C'est le cas par exemple des CBIR (Content-Based Image Retrieval systems) [GR95, JV96, SB91] où des facettes comme celles relatives aux couleurs, aux formes des objets dans l'image, à la texture, ainsi qu'au contenu sémantique sont utilisées. L'utilisateur n'a pas à se rappeler exactement du contenu des images, il peut faire appel à son imagination et à la façon dont il perçoit les choses pour spécifier son besoin. Ceci en adressant différentes facettes de l'image dans sa requête. Par exemple, il peut spécifier la couleur *Jaune* et le mot-clé sous-marin pour voir s'afficher à l'écran le dernier clip musical des Beatles.

3.6 Conclusion et Résumé

Dans ce chapitre, nous avons défini un modèle d'indexation relationnel pour la recherche d'informations. Ce modèle est basé sur la théorie des situations comme langage de représentation des requêtes et des documents. Il précise les connaissances et les règles d'inférence qui doivent être supportées par un système orienté vers la précision des réponses. En particulier, ce modèle explicite les données et les types d'inférence que doit présenter un système de recherche en vue de gérer convenablement les relations sémantiques apparaissant dans les index des documents. Ainsi, en plus des règles d'inférence classiques modélisant le comportement logique d'un système, l'exploitation des relations constitue le principal facteur motivant la prise des décisions de pertinence dans un système conceptuel. En formalisant les propriétés des relations et leur comportement sémantique, puis en les considérant dans le processus de recherche, certains cas de silence peuvent être évités et une amélioration des performances qualitatives du système est attendue.

Notre problématique dans ce chapitre peut ainsi être vue comme la définition d'une logique "idéale" pour la recherche d'informations, c'est-à-dire une formalisation d'un langage de représentation suffisant pour la description des documents complexes en recherche d'informations, puis

la spécification des connaissances et des règles d'inférence nécessaires pour la modélisation d'une notion de pertinence adéquate aux nouvelles applications de recherche. En effet, ce modèle relationnel s'intègre harmonieusement aux intentions suggérées par le modèle logique de Rijsbergen. Il constitue une logique non classique adaptée aux nouvelles exigences du domaine de la recherche d'informations. De plus, les dérivations logiques se font toujours par rapport à une transformation qui est matérialisée par le passage d'un monde à un autre, à travers l'utilisation d'un ensemble de connaissances sur le domaine. Nous avons identifié cet ensemble de connaissances, en mettant l'accent sur l'importance des propriétés des relations et la sémantique qu'elles véhiculent. Nous avons alors montré qu'un thésaurus de termes ne constitue qu'une très faible portion de ce que doit contenir la base de connaissances d'un système.

En plus de ce que doit représenter une logique en recherche d'informations, il est important de s'interroger sur ses caractéristiques et de justifier ainsi ses résultats de recherche. Nous nous sommes donc intéressé à des critères comme la complétude et la correction de cette logique. Pour ce faire, nous avons proposé une sémantique adéquate à notre problématique en recherche d'informations. Par rapport à cette sémantique nous avons prouvé que ces deux critères sont satisfaits par notre logique. Enfin, nous nous sommes interrogé sur l'efficacité de cette logique. Nous avons alors montré qu'elle peut être décidable dans le cas où le langage utilisé pour la représentation des situations est fini.

Une caractéristique importante du modèle proposé est qu'il peut être à la fois considéré comme une logique pour la recherche d'informations, mais aussi comme un cadre formel par rapport auquel nous pouvons détecter les limites et les insuffisances des formalismes opérationnels existants, un peu à la manière de Huibers [Hui96]. Ces insuffisances sont soit au niveau de la représentation des connaissances, principalement la prise en compte des propriétés des relations, soit au niveau de la fonction de correspondance elle-même, qui peut ainsi présenter moins de décisions de pertinence qu'il en faut pour une recherche efficace et complète[32].

Au fait, le modèle relationnel ici proposé est conçu plutôt comme une méta-logique devant être implantée par un formalisme opérationnel existant. Ce dernier devra vérifier les exigences du modèle et se soustraire à un certain nombre de contraintes pratiques. Par exemple, il est souhaitable que le formalisme à choisir devra supporter une recherche rapide, c'est-à-dire qu'il doit retourner les documents pertinents en un laps de temps assez réduit. Ainsi, si l'implantation d'un système à base de situations reste envisageable, et donc que la logique véhiculée par le modèle relationnel soit directement implantée, il est fort à parier que les temps de réponses du système engendré par cette logique seront très en deçà de ce qui peut être acceptable en recherche d'informations.

Ainsi, quelques essais ont été réalisés pour construire un cadre opérationnel basé sur la théorie des situations [Bla93, NSHP88, TA93, TV94]. Ces essais ont abouti à trois langages de programmation orientés situations [TV95], à savoir PROSIT [NSHP88], ASTL [Bla93] et BABY-SIT [TA93]. Une revue détaillée de ces trois langages se trouve dans [TV95]. La plupart de ces travaux offrent des cadres opérationnels de programmation basés sur des langages semblables à Prolog ou à Lisp. Ces derniers n'étant pas connus pour donner lieu à des algorithmes rapides, il s'en suit qu'une implantation à base de théorie des situations ne sera pas efficace en termes de complexité des traitements.

Dans le chapitre suivant, nous nous intéressons au formalisme des graphes conceptuels. Nous

32. Par une recherche complète, nous désignons un taux de silence nul. Ainsi, s'il est possible en utilisant les connaissances du système de faire dériver la requête à partir de la représentation d'un document, alors ce dernier devra être retourné à l'utilisateur.

nous proposons ainsi d'analyser les insuffisances de ce formalisme par rapport au modèle relationnel proposé. Ces insuffisances détectées, nous nous déployons à lui appliquer les extensions théoriques nécessaires, de façon à aboutir à un système logique ayant des capacités de représentation et de dérivation proches de celles fournies par le modèle relationnel ici proposée. Le problème de la complexité des traitements est résolu par l'adoption d'une technique d'interprétation issue de la logique de premier ordre.

Chapitre 4

Une Vision Relationnelle des Graphes Conceptuels

4.1 Introduction

Le modèle d'indexation relationnel que nous avons adopté dans le chapitre précédent peut être appliqué à de nombreux formalismes de représentation, moyennant une extension du modèle qu'ils utilisent. Il s'agit d'un modèle théorique spécifiant les caractéristiques et les hypothèses dont l'intégration au processus de recherche pourrait améliorer l'efficacité du système. Différents formalismes peuvent être proposés pour une approche d'indexation relationnelle. Par exemple, la logique de premier ordre, les logiques terminologiques, les graphes conceptuels, Datalog, etc. Chaque formalisme a des inconvénients et des avantages, que ceci concerne la définition formelle des éléments de base qu'il adopte, la complexité des algorithmes de traitements qu'il propose ou la richesse de son expressivité.

Dans la suite nous retenons le formalisme des graphes conceptuels de Sowa [Sow84]. Les raisons de ce choix trouvent leurs origines aussi bien dans des considérations théoriques que dans un but pragmatique. Elles seront présentées et approfondies tout au long de ce chapitre.

Dans une première partie, nous discutons les éléments intéressants de ce formalisme qui justifient son choix en recherche d'informations en général, et comme une implantation de notre approche en particulier.

Dans la partie suivante, nous présentons les notions de base de ce formalisme telles qu'elles sont présentées dans [Sow84].

Nous apportons ensuite les adaptations nécessaires pour l'utilisation de ce formalisme comme un langage d'indexation relationnelle. Nous étendons ainsi le modèle des graphes conceptuels par des notions qui nous seront utiles pour la suite de la thèse.

Enfin, nous proposons une nouvelle interprétation logique pour les graphes conceptuels. Cette interprétation a l'avantage, outre l'élimination des ambiguïtés sémantiques du formalisme, d'offrir un cadre formel permettant un processus de recherche d'informations efficace. En effet, elle conduit à rendre possible une approche purement algébrique pour les graphes conceptuels. Les opérations sur les graphes conceptuels correspondent alors à des opérateurs ensemblistes. Cette approche s'appuie sur une technique classique en recherche d'informations, à savoir les fichiers inverses, pour proposer

un algorithme de recherche rapide. Cet algorithme, correct et complet vis-à-vis de la nouvelle interprétation logique, présente la particularité d'être polynomial et de travailler uniquement sur la requête.

4.1.1 Pourquoi les Graphes Conceptuels?

Les démonstrateurs de théorèmes sont réputés être difficiles à implanter et complexes à mettre en oeuvre. Ce fait s'aggrave quand la base de faits est de taille importante. Les raisonnements en logiques terminologiques sont ainsi par essence complexes, que ce soit pour un langage puissant comme BACK [Neb88] ou un langage aussi réduit que $\mathcal{FL}$ [Neb90b]. La subsomption, à la base de tout mécanisme de classification dans ces logiques, repose ainsi sur un ensemble de règles d'inférence dont la complexité engendre un algorithme de calcul co-NP-hard [Neb90a] même pour un langage aussi dénué que $\mathcal{FL}$. Les recherches tendant à apporter une solution à ce problème de complexité n'ont pas débouché sur le résultat espéré. Patel-Schneider & al. résument ce constat d'échec dans [PSOKK+90](page 21):

> "There was a consensus among the discussants that [these logics] should be efficient and principled ... However, interpreting the two terms efficient and principled as worst-case tractability and soundness and completeness with respect to standard model-theoric semantics lead us to conclude that we should not try to implement [terminological logics] at all ... "

Pourtant les logiques terminologiques peuvent facilement être étendues avec notre modèle d'indexation relationnel. Les termes générés par ces logiques sont assez expressifs pour permettre l'implantation d'une approche d'indexation relationnelle. De plus, les relations (les rôles) occupent une place importante dans ces formalismes et certaines logiques terminologiques sont même munies de plusieurs opérateurs sur les rôles. Cependant, cette extension pourrait avoir des conséquences importantes sur la complexité des traitements dans ces logiques, puisqu'il existe une opposition originelle entre leur puissance d'expression et leur efficacité de traitement [LB85]. Autrement dit, il devient rapidement évident que l'ajout d'autres règles d'inférence sur les rôles ferait basculer irrémédiablement l'algorithme de subsomption vers une complexité exponentielle. La fonction de correspondance dans les logiques terminologiques étant fondée sur cette notion de subsomption, il est clair que nous ne pouvons pas les utiliser dans le cadre d'un système de recherche d'informations opérationnel où le coût du calcul doit être raisonnable.

Ce même constat s'applique à toutes les logiques et à tous les formalismes basés sur des démonstrateurs de preuves. Ainsi, que ce soit la logique de premier ordre ou même un langage comme Prolog, les temps de calcul sont trop élevés pour être convenables en recherche d'informations. Il faudrait ainsi choisir un formalisme assurant en même temps une richesse d'expression suffisante et des traitements efficaces et maniables (d'une complexité polynomiale). À ce titre, les graphes conceptuels de Sowa présentent bien des atouts.

Dans son livre sur les structures conceptuelles [Sow84], Sowa introduit, à travers un formalisme logique convivial, un nouveau modèle de représentation de connaissances. Ce formalisme, appelé *graphes conceptuels*, fut présenté comme un compromis entre les langages formels et les langues naturelles. Les fondements logiques des graphes conceptuels reposent sur la définition de l'opérateur ϕ, qui associe à tout graphe conceptuel une formule logique de premier ordre, et la définition d'un certain nombre de règles de réécriture. L'aspect graphique du formalisme procure une facilité

de lecture fort appréciable en représentation de connaissances, permettant ainsi la construction d'interfaces homme-machine utiles dans tout processus interactif.

Un des atouts de ce formalisme est justement le fait qu'il constitue une représentation graphique de la logique, s'inspirant ainsi de l'idée défendue par C.S. Peirce [Sow93], à savoir qu'un graphe est plus facilement déchiffré qu'une formule logique. À travers les opérateurs algébriques de graphes qu'il offre, le formalisme des graphes conceptuel simule un raisonnement logique. L'intérêt d'une telle approche est que ces opérateurs algébriques peuvent bénéficier des multiples algorithmes efficaces développés en théorie des graphes, autorisant ainsi des implantations rapides. Les graphes conceptuels offrent donc un bon compromis entre les approches algébriques et les démonstrateurs de preuves: plutôt que de considérer les graphes conceptuels comme une logique, et réaliser les calculs à l'aide d'un démonstrateur logique, l'approche adoptée consiste à effectuer exclusivement des opérations algébriques sur les graphes, et à leur associer une interprétation logique par le biais de l'opérateur ϕ.

Comme cela a été mentionné par Gaines [Gai93], les graphes conceptuels peuvent donc servir de base à un système opérationnel de représentation de connaissances, pouvant manier aussi bien la langue naturelle, la logique, les langages formelles ainsi que les raisonnements qui leurs sont associés. Récemment, plusieurs travaux se sont employés à adopter ce formalisme en recherche d'informations [Che92, Mya92, ML93, BZ92, MKL94, Sch92, Khe95, MA96]. Les raisons de ce choix peuvent se résumer ainsi:

- Le formalisme des graphes conceptuels est assez expressif pour représenter et raisonner sur des connaissances complexes et structurées. Il va ainsi au delà des représentation plates des mots-clés.
- Les graphes conceptuels offrent des grandes possibilités pour exprimer la sémantique de l'information, ce qui permet une indexation précise des documents [HOC96].
- Les graphes conceptuels semblent mieux modéliser les nouvelles applications de la recherche d'informations tels que les systèmes hypermédia [MA96, Khe95], et la recherche de documents multimédia [Sch94, Mec95c].
- Le formalisme des graphes conceptuels peut être couplé avec le modèle logique de recherche d'informations [Che92]. Les propriétés du formalisme lors d'une implantation opérationnelle peuvent ainsi être contrôlées.

En plus de la dernière propriété, une des raisons qui nous a poussé à choisir ce formalisme comme implantation de notre modèle relationnel est que les graphes conceptuels se prêtent bien aux extensions. En effet, si le modèle de base des graphes conceptuels avait été défini de façon précise par Sowa, ce dernier présentait des idées d'extensions du formalisme d'une manière parfois informelle. Plusieurs de ces extensions restent assez floues et il appartient donc à chaque application de les préciser. C'est le cas par exemple des relations. Sowa définit bien un treillis de types de relations mais son avantage n'a jamais été explicité, et aucun des opérateurs de graphes n'en tienne compte. Ceci confirme notre constatation que peu de formalismes de représentation de connaissances gèrent convenablement les relations. Notre choix des graphes conceptuels s'appuie aussi sur les constatations suivantes:

- Les graphes conceptuels peuvent être utilisés dans une approche d'indexation relationnelle. En effet un graphe conceptuel peut être vu comme un ensemble de relations reliées entre elles

par des concepts en commun.

- L'extension des graphes conceptuels par un raisonnement sur les relations est facile à intégrer, et ne nécessite aucun recours à un autre formalisme de représentation.
- Il est possible d'associer au modèle des graphes conceptuels ainsi étendu, une sémantique logique complète et correcte.
- Une fonction de correspondance polynomiale intégrant aussi bien un raisonnement sur les concepts que sur les relations peut être proposée. Elle constitue une extension de l'opérateur de projection par une approche relationnelle, et est basée sur une nouvelle interprétation logique associée aux graphes conceptuels. Cette interprétation a l'avantage de garder intactes les notions de base de Sowa tout en autorisant des traitements plus rapides.

Dans la section suivante, nous rappelons les éléments de base du modèle tels qu'ils sont introduits par Sowa.

4.1.2 Notions de Base

Les graphes conceptuels sont présentés comme un modèle général d'écriture de réseaux pour la représentation des connaissances. L'accent a été mis sur le fait que toutes les autres formes de représentation pourraient être exprimées sous la forme de graphes conceptuels. Cet aspect de généralité est au coeur des 5 avantages du formalisme que la communauté des graphes conceptuels [1] se plaît à rappeler: généralité, existence de standards, expressivité, rapidité de traitements et enfin les potentialités d'interaction homme-machine. La représentation des connaissances se fait à travers la description des concepts et des relations entre ces concepts.

Définition 4.1.1 (Graphe Conceptuel) *Un graphe conceptuel est un graphe fini, orienté, connexe et biparti. On y trouve ainsi deux types de noeuds: les noeuds concepts et les noeuds relations.*

Les concepts sont représentés graphiquement par des rectangles et correspondent à des objets du domaine de l'application. Par définition, ils ont chacun un type noté en majuscules et parfois un référent noté en minuscules. On parle alors de concept individuel dans le cas de la présence du référent, et de concept générique dans le cas contraire. Par exemple, $\boxed{PERSONNE : *}$ représente un concept générique dénotant l'ensemble des personnes du domaine de l'application. Ce concept est également noté $\boxed{PERSONNE}$. Par contre, le concept $\boxed{PERSONNE : \#}$ représente une personne particulière non identifiée. Enfin le concept $\boxed{PERSONNE : \#John}$ dénote l'individu de nom John. Les référents différents de $*$, sont appelés *marqueurs individuels*. Le marqueur $\star$ est un marqueur générique. L'ensemble des marqueurs est dénoté par $\mathcal{M}$.

Le type représente la classe à laquelle se rattache le concept exprimé, tandis que le référent est vu comme une instanciation du concept. Ces deux éléments sont reliés par une relation de conformité $Conf$, assertant que le référent est un élément de la classe du type de concept. L'ensemble des référents conformes à un type de concept forme l'extension de ce dernier, ou sa *dénotation*:

Définition 4.1.2 (Dénotation de Type) *Soit $\mathcal{T}_c$ l'ensemble de tous les types de concepts. La dénotation d'un type de concept $t_c \in \mathcal{T}_c$ est l'ensemble des marqueurs individuels de $\mathcal{M}$ conformes à t_c, c'est-à-dire qui sont des référents possibles de ce type de concept.*

1. Les discussions dans cette communauté sont menées sur la mailing-list "cg@cs.uah.edu"

Les types de concepts $\mathcal{T}_c$ sont organisés selon une hiérarchie des types qui constitue un treillis fini. Ce treillis est muni d'une relation d'ordre partiel $\leq$, définie de la façon suivante: si la dénotation d'un type de concept c_1 est incluse dans la dénotation du type de concept c_2 alors $c_1 \leq c_2$ dans $\mathcal{T}_c$. De plus $\leq$ vérifie les propriétés de réflexivité et de transitivité. Par exemple, HOMME $\leq$ PERSONNE. Nous disons que HOMME est une restriction (sous-type) de PERSONNE, et que PERSONNE est une généralisation de HOMME. Les relations conceptuelles sont représentées graphiquement par des ovales étiquetés par le type de la relation. Une relation a au moins un arc, et chaque arc d'une relation doit être lié à un concept. Les types de relations, dénotés par l'ensemble $\mathcal{T}_r$, sont aussi classés dans une structure de treillis. Ces éléments de base étant posés, Sowa introduit la notion de graphe simple:

Définition 4.1.3 (Graphe Simple) *Un graphe simple est formé d'un ensemble de concepts et un ensemble de relations tels que chaque concept est relié à au moins une relation (connexité) et chaque relation est reliée à un nombre de concepts égal à son arité. Un concept seul est aussi considéré comme un graphe simple.*

Un exemple de graphe simple est donné dans la figure 4.1 suivante:

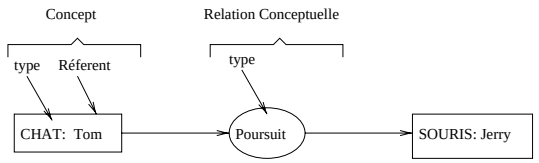


Figure 4.1. Un graphe simple pour "Tom poursuit Jerry"

Les rectangles et les ovales forment la *notation graphique* du formalisme de Sowa. Le graphe de la figure 4.1 peut être également mis sous une forme de *notation linéaire*:

[CHAT:Tom]→(Poursuit)→[SOURIS:Jerry].

La définition 4.1.3 permet de construire des graphes qui n'ont pas de sens [2] Sowa définit ainsi un ensemble de graphes de départ noté $\mathcal{B}_{can}$, dit *base canonique*, exprimant des conditions à toujours respecter, comme par exemple le fait que l'*agent d'une action doit être une personne*:

[ACTION]→(Agt)→[PERSONNE].

L'avantage de cette notion est de permettre l'expression de contraintes sémantiques sans recours à un autre formalisme. Un graphe de la base canonique $\mathcal{B}_{can}$ peut être très précis, puisqu'il peut s'agir d'un graphe assez complexe contenant plusieurs relations et concepts. Cependant, la base canonique ne permet pas d'exprimer des contraintes négatives (*l'autruche ne vole pas*), ni de spécifier des contraintes sur le nombre des liens d'un concept ou d'une relation (*une action n'a qu'un agent*).

2. Dans [Sow84, pp90-91] Sowa écrit "A conceptual graph is a combination of concept nodes and relation nodes [...]. But not all such combination make sense [...]. To distinguish the meaningful graphs that represent real or possible situations in the external world, certain graphs are declared to be *canonical*".

Elle peut toutefois être considérée comme un bon compromis entre la simplicité de sa manipulation et sa puissance d'expression.

Afin que ces contraintes de départ soient respectées par les graphes conceptuels générés, Sowa fait munir son formalisme des opérateurs algébriques de formation de graphes suivants:

Copie: Si u est un graphe conceptuel alors la copie v de u est aussi un graphe conceptuel.

Restriction: Le type d'un concept peut être remplacé par un type de niveau inférieur dans le treillis $\mathcal{T}_c$. Le référent doit alors se conformer au nouveau type. Par exemple, $[PERSONNE : \#John]$ peut être restreint à $[HOMME : \#John]$ si le type HOMME et le référent $John$ sont en relation de conformité. La restriction peut aussi porter sur le marqueur générique $\star$ en le réduisant à un marqueur individuel i donné. Par exemple, $[HOMME]$ peut être restreint à $[HOMME : \#John]$.

Simplification: Quand deux concepts sont reliés par deux relations identiques, alors une des deux relations peut être supprimée. Par exemple, $[A] \rightarrow (r) \rightarrow [B]$ est une simplification de $[A] \rightarrow (r) \rightarrow [B] \leftarrow (r) \leftarrow [A]$.

Jointure: Si un concept $[c]$ dans un graphe u est identique à un concept $[d]$ dans un graphe v alors u et v peuvent être joints pour former un graphe conceptuel w, en supprimant $[d]$ et en liant à $[c]$ toutes les relations qui étaient liées à $[d]$. Par exemple, le graphe $[A] \rightarrow (r) \rightarrow [B]$ peut être joint avec le graphe $[C] \rightarrow (r') \rightarrow [B]$ sur le concept $[B]$. Le graphe résultat étant $[A] \rightarrow (r) \rightarrow [B] \leftarrow (r') \leftarrow [C]$. On notera qu'il peut exister plusieurs façons de joindre deux graphes conceptuels. Ainsi, le graphe $[A] \rightarrow (r) \rightarrow [A]$ peut être joint au graphe $[A] \rightarrow (r') \rightarrow [B]$ de deux manières différentes en produisant soit le graphe $[A] \rightarrow (r) \rightarrow [A] \rightarrow (r') \rightarrow [B]$ soit le graphe $[B] \leftarrow (r') \leftarrow [A] \rightarrow (r) \rightarrow [A]$.

Seuls les graphes construits à partir de la base canonique par ces quatre opérateurs sont des expressions du langage des graphes conceptuels. Sowa introduit alors un ordre partiel $\leq$ sur les graphes (relation à ne pas confondre avec la relation d'ordre sur les types). Cet ordre définit une relation d'inférence sur les graphes permettant de déterminer si les informations représentées par un graphe induisent les informations représentées par un autre graphe. En d'autres termes, Sowa étend la relation d'ordre partiel sur les types de concepts aux graphes conceptuels. Cette dernière est définie en termes d'une séquence d'application des règles de formation précédentes:

Définition 4.1.4 (Dérivation de Graphes Conceptuels) *Si u est le résultat de l'application d'une séquence des quatre opérateurs algébriques à un graphe de départ v, alors on dit que u est dérivable de v et on note $u \leq v$.*

On dit également que u est une spécialisation de v ou que v est une généralisation de u. Cette relation peut aussi être définie par l'opérateur de projection, considéré comme un morphisme de graphes bipartis (les sommets concepts sur les sommets concepts, les sommets relations sur les sommets relations):

Définition 4.1.5 (Projection) *Il existe une projection d'un graphe h sur un graphe g si, et seulement si, il existe une application π de h vers g satisfaisant les conditions suivantes:*

- *pour chaque concept c de h, $\pi(c)$ est soit une spécialisation de c, soit identique à c.*

- *pour chaque relation conceptuelle r de h, $\pi(r)$ est identique à r.*
- *si les concepts $[c_1]$ et $[c_2]$ sont reliés par une relation r dans h, alors $\pi(c_1)$ et $\pi(c_2)$ sont également reliés par $\pi(r)$.*

Le sous-graphe de g identifié par π, est appelé la projection h sur g.

La projection π telle qu'elle a été définie par Sowa n'est pas nécessairement injective: si x_1 et x_2 sont deux concepts ou deux relations conceptuelles où $x_1 \neq x_2$, il est possible d'avoir $\pi(x_1) = \pi(x_2)$. Le résultat de la projection n'est pas toujours unique: le graphe g peut avoir plusieurs sous-graphes différents vérifiant la condition de la projection. La figure 4.2 montre un exemple de projection.

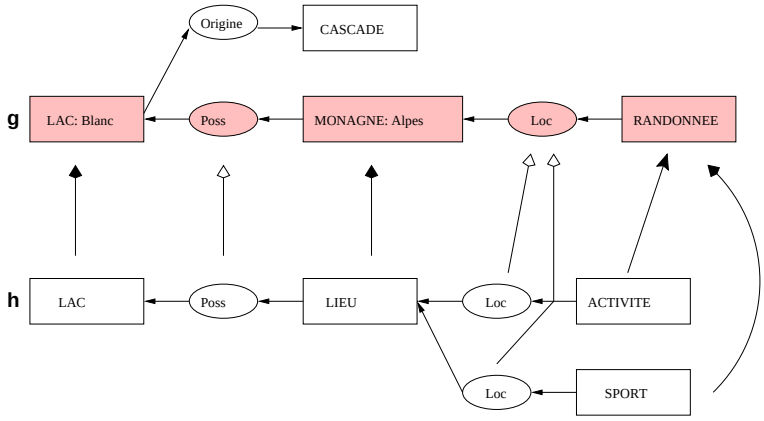


Figure 4.2. En grisé est représenté la projection de h sur g. h peut être interprété comme suit: "il existe une activité et un sport localisés sur le même lieu possédant un lac".

Sowa prouve que si un graphe g est une spécialisation d'un graphe h, alors il existe une projection de h sur g. De plus dans [CM92], il est démontré que s'il existe une projection de h sur g, alors $g \leq h$.

Théorème 8 *Soient g et h deux graphes conceptuels. On dit que g est une spécialisation de h (noté $g \leq h$) si, et seulement si, il existe une projection π de h dans g.*

Une des caractéristiques importantes du modèle de Sowa est que le raisonnement précédent sur les graphes peut être fait tout en conservant un lien avec la logique de premier ordre. Sowa définit un opérateur d'interprétation logique, noté ϕ, qui transforme chaque élément du modèle des graphes conceptuels en un élément de la logique du premier ordre. L'opérateur ϕ associe à tout type de concept un prédicat unaire et à tout type de relation, un prédicat de même arité que le type. À un marqueur individuel est associé une constante.

Définition 4.1.6 (L'opérateur ϕ) *L'association d'une formule logique $\phi(u)$ à un graphe conceptuel u est réalisée de la façon suivante:*

1. *Pour chaque concept $[c]$, on associe un prédicat $t_c(p)$ dont le nom est égal au type de concept t_c de $[c]$, et dont le paramètre p est soit une constante dans le cas où $[c]$ est muni d'un marqueur*

individuel, soit une variable indépendante et quantifiée si $[c]$ *est muni du marqueur générique* $\star$*. Par exemple au concept* $[T : \#r]$ *est associé le prédicat* $T(r)$ *alors qu'au concept générique* $[T : *]$ *est associé la formule logique* $\exists x T(x)$*.*

2. *À chaque relation n-aire* (r) *est associée un prédicat n-aire dont le nom est égal au type de la relation* t_r*, et dont les paramètres sont égaux aux paramètres des prédicats correspondant aux concepts liés par cette relation. L'ordre des paramètres est convenu pour chaque type de relation.*

3. *Si u est un graphe conceptuel,* $\Phi(u)$ *est une formule logique de premier ordre obtenue par conjonction des prédicats associés aux composants de u dans 1) et 2).*

Par exemple, au graphe

$$[CHAT : \#Tom] \rightarrow (Poursuit) \rightarrow [SOURIS]$$

est associé la formule logique suivante:

$$\exists x\, CHAT(Tom) \wedge Poursuit(Tom, x) \wedge SOURIS(x)$$

Une propriété importante de l'opérateur ϕ est qu'il vérifie le théorème suivant:

Théorème 9 *Pour tout couple de graphes conceptuels u et v, si* $u \leq v$*, alors* $\phi(u) \supset \phi(v)$

En d'autre termes, si u est dérivable de v par les quatre opérateurs de formation de graphes (copie, restriction, simplification, jointure) alors $\phi(u) \supset \phi(v)$. Ainsi, la relation d'ordre partiel, qui est à la base du raisonnement dans le modèle des graphes conceptuels, peut être définie par l'existence d'une projection d'un graphe sur un graphe spécialisé. On en déduit des deux théorèmes précédents et de la définition de la projection, la propriété suivante:

Théorème 10 *S'il existe une projection d'un graphe h sur un graphe g, alors* $g \leq h$ *et* $\phi(g) \supset \phi(h)$

Ainsi, la projection est un opérateur algébrique qui simule une inférence logique. Elle peut être vue comme la combinaison de plusieurs opérateurs de formation de graphes. La projection résume plusieurs pas de dérivation de graphes en une seule opération: si la projection d'un graphe h sur un autre graphe g est possible, alors g peut être dérivé de h. Comme on le verra dans la section 4.2, cet opérateur est très important dans un processus de recherche fondé sur les graphes conceptuels.

4.1.3 Les Problèmes du Formalisme

Les graphes conceptuels apparaissent comme un modèle de représentation général offrant un niveau de représentation intermédiaire entre le langage et la logique formelle. Cependant, ce modèle se voulant général, de nombreux points restent à éclaircir si l'on veut le mettre réellement en oeuvre. Ainsi, le manque de détails de certains opérateurs de graphes, en particulier la correspondance entre graphes conceptuels et formules logiques, et l'ambiguïté de certaines notions de base constituent des points faibles qu'il convient de régler avant toute utilisation efficace.

Dans [Che92, CM93, Mug93, Mug95, CM95], la relation de spécialisation $\leq$ ne constitue qu'un préordre partiel sur les graphes. Si elle est bien réflexive et transitive, en fait elle n'est pas anti-symétrique. Cela est dû à la définition de l'opérateur de jointure. En effet, telle qu'elle a été présentée par Sowa, cette définition n'indique aucune condition sur les graphes utilisés. Ainsi, selon que les graphes à joindre sont distincts ou pas, nous distinguons l'opération de jointure externe de l'opération de jointure interne [Che92, CM95].

Par exemple considérons la séquence d'opérations suivantes sur le graphe (1):

$$(1)[PERSONNE] \rightarrow (Parle) \rightarrow [FILLE]$$

L'application d'une restriction sur *PERSONNE*, permet de faire dériver le graphe suivant:

$$(2)[GARCON] \rightarrow (Parle) \rightarrow [FILLE]$$

Une jointure externe appliquée aux graphes (1) and (2) produit:

$$(3)[PERSONNE] \rightarrow (Parle) \rightarrow [FILLE] \leftarrow (Parle) \leftarrow [GARCON]$$

Le concept $[PERSONNE]$ peut encore être restreint pour obtenir le graphe:

$$(4)[GARCON] \rightarrow (Parle) \rightarrow [FILLE] \leftarrow (Parle) \leftarrow [GARCON]$$

En appliquant tour à tour une jointure externe sur $[GARCON]$ puis une simplification de la relation redondante, nous obtenons finalement le graphe:

$$(5)[GARCON] \rightarrow (Parle) \rightarrow [FILLE]$$

Les graphes (2) et (5) sont égaux. Si la définition de l'ordre partiel sur les graphes est appliquée, nous devons alors considérer la situation paradoxale suivante: $1 \geq 2 \geq 3 \geq 4 \geq 2$. Les graphes (2), (3) et (4) doivent alors être considérés comme égaux. La notion de relation d'ordre $\leq$ pose ainsi le problème de l'égalité entre graphes, et plus particulièrement le problème d'égalité entre concepts en vue de les joindre. Afin de ramener la relation $\leq$ vers une relation d'ordre partiel, il convient d'éviter l'utilisation des jointures internes. Seulement, une telle démarche induit des changements importants dans le modèle. Par exemple, l'opérateur de simplification ne sera jamais utilisé puisqu'une redondance de relations ne peut apparaître que lorsqu'on effectue des jointures internes. De plus cela induit que la projection est une application injective, ce qui restreint la définition de base de Sowa. Ainsi, Chevallet dans [Che92] préfère plutôt parler de *graphes normalisés*, pour désigner les représentants des graphes équivalents par la relation d'ordre. Un graphe normalisé, est un graphe sur lequel aucune séquence d'opérateurs de formation de graphes ne peut être appliquée. Par conséquent, la relation de spécialisation $\leq$ sur ces graphes normalisés est un ordre partiel. Nous retrouvons la même approche dans [CM92], où il est montré que chaque classe d'équivalence ε définie par $\leq$ possède un seul graphe qui n'admet aucune projection de lui-même dans un de ses sous-graphes stricts. Ce graphe appelé *graphe conceptuel irredondant* de ε est le plus petit au nombre de noeuds. Il est facile de montrer qu'il correspond à la notion de graphe normalisé définie par Chevallet. La relation $\leq$ sur les graphes irredondants est ainsi une relation d'ordre partiel. La propriété suivante est énoncée:

Propriété 6 (Graphes Équivalents) *La relation $\leq$ est réflexive, transitive mais n'est pas anti-symétrique. Deux graphes g et h sont dits équivalents si $g \leq h$ et $h \leq g$. On écrit alors que $g \equiv h$.*

Par exemple, la figure 4.3 montre deux graphes équivalents. Notons qu'il est aussi possible de limiter la relation de spécialisation $\leq$ à l'existence d'une *projection injective*[3] afin d'aboutir à un ordre partiel. Ainsi dans cette même figure 4.3, seule la relation de spécialisation $g \leq h$ sera vérifiée, et le problème d'anti-symétrie ne se pose pas.

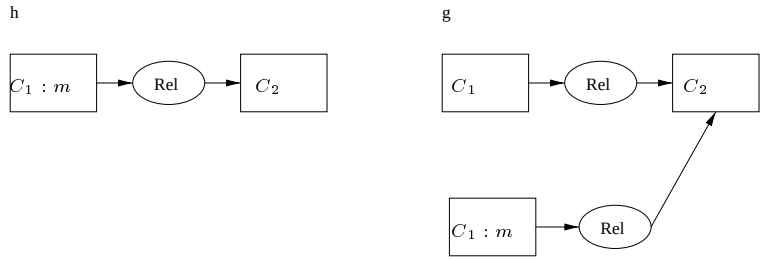


Figure 4.3. Les graphes g et h vérifient $g \leq h$ et $h \leq g$, mais $h \neq g$

Cependant, de toutes les imprécisions du modèle, c'est la définition de l'opérateur de translation ϕ qui semble être l'écueil de ce formalisme. Il est en effet facile de prouver que cet opérateur n'est pas complet vis-à-vis de la relation de spécialisation. Ainsi deux graphes différents peuvent avoir la même formule logique, comme le montre l'exemple de la figure 4.4 suivante.

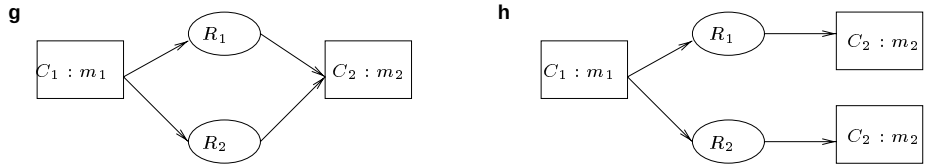


Figure 4.4. Ces deux graphes sont logiquement équivalent ($\phi(g) = \phi(h)$), cependant ils ne sont pas équivalents par la relation de spécialisation. De plus $g \neq h$.

Ce problème fut discuté dans la littérature [CM95, Gui96a] mais ne semble pas occuper une place importante dans les recherches jusqu'ici menées au sein de la communauté des graphes conceptuels. La plupart des travaux se sont en effet concentrés sur la théorie des graphes [MMS93, TDS94, ELRS95, EEM96] et la recherche d'algorithme efficace permettant l'implantation des opérateurs de graphes dans des applications réelles. Dans [CM95], Chein et al. se sont ainsi interrogés sur la nécessité de lier l'étude des graphes conceptuels à la logique[4]. Seul, Michel Wirmelinger [Wer95] s'est intéressé formellement au problème de la complétude de ϕ, en proposant un autre opérateur de translation.

Comme pour la relation d'ordre partiel, des classes de graphes conceptuels pour lesquelles la fonction ϕ de Sowa est complète furent évoquées. Ainsi, dans [CM95, Car96, Gui96a], la notion de

3. Nous rappelons que la projection dans la définition originelle de Sowa n'est pas forcément injective.

4. Dans [CM95], on trouve la remarque suivante: "*.. the ease of doing important reasoning- for instance plausible reasoning by using some maximal join- on CGs, without, at least for the moment, logical semantics, strengthens our belief that CGs must be studied and developed "independently" from logic*".

graphe sous forme normale est introduite. La forme normale d'un graphe est obtenue en fusionnant les noeuds concepts ayant le même marqueur individuel. Une telle démarche nous oblige à travailler sur des graphes d'un type particulier, ce qui constitue une limitation pour un formalisme se voulant général et assez expressif. La duplication d'un noeud dans un graphe peut être une façon de mentionner son importance et peut être utile en recherche d'informations, où un poids plus élevé sera assigné à ce concept.

La complétude n'est malheureusement pas le seul inconvénient de la fonction ϕ. Ainsi, l'interprétation logique associée aux concepts munis de marqueurs n'est pas toujours claire à saisir. Cela est dû au fait que la définition de ϕ dans [Sow84] n'est pas très détaillée. Il en résulte que les concepts $[A : \#]$ et $[A : \star]$ se voient attribuer la même formule logique par ϕ, à savoir $\exists x A(x)$. C'est très étrange puisque qu'il s'agit de confondre le concept désignant *tous les individus* de A avec celui dénotant un *élément particulier* de la dénotation de A. Le marqueur générique $\star$ est interprété existentiellement tout comme le marqueur #. Si cette interprétation semble être correcte pour #, assimilé ainsi au quantificateur existentiel $\exists$, la question se pose quant à la validité de l'interprétation du marqueur $\star$. En effet, il est légitime de se demander si ce marqueur ne coïncide pas plutôt avec le quantificateur universel. Nous discutons de ce point précis dans la section 4.5.

Une question souvent soulevée par ceux qui découvrent le formalisme pour la première fois, concerne le sens des dérivations logiques de ϕ par rapport à celui de la dérivation des graphes. Cela n'est pas très intuitif à premier abord. Ainsi, quand on restreint un concept, ou quand on applique une jointure sur les noeuds d'un graphe donné, l'interprétation logique associée au nouveau graphe dérivé, implique l'expression logique associée au graphe initial.

Si g dérive g' alors$\phi(g') \supset \phi(g)$.

La dérivation du graphe g' dans la figure 4.5 pourrait paraître absurde à sa première lecture. Cependant, elle est valide relativement à la logique véhiculée par les graphes. La dérivation devra se lire dans ce sens: si nous observons un crocodile rouge alors nous pouvons inférer qu'il existe un objet qui a une couleur. Ceci qui est évidemment correct en termes d'inférence logique, mais pas très intuitif par rapport à la notion de base canonique[5].

Enfin des ambiguïtés subsistent quand à l'importance des relations conceptuelles dans le formalisme. Ainsi, les hypothèses 3.2.7 et 3.6.13 dans le livre de Sowa, mentionnent une hiérarchie de types de relations dont l'élément le plus grand est la relation binaire "LIEN". Cependant, dans tout le reste du livre, il n'a jamais été question de l'utiliser. De plus, la signification exacte d'une relation d'ordre comme ENTRE $\leq$ LIEN, où ENTRE est une relation tertiaire et LIEN une relation binaire reste ambiguë dans le formalisme. À notre avis, ceci explique pourquoi personne ne s'est intéressé à définir une projection incluant cet hiérarchie ainsi qu'un opérateur de restriction sur les relations.

Les problèmes mentionnés dans cette section expliquent l'existence, à ce jour, d'une multitude de points de vue sur les graphes conceptuels. Selon les besoins des différentes applications conçues pour des domaines très divers comme le traitement de la langue naturelle, la médecine ou la recherche d'informations, des choix ont été fait au niveau du modèle de base. Le danger étant évidemment que chaque application redéfinisse son propre formalisme et que, ce faisant, certaines propriétés du

5. Comme cela a été indiqué dans la section précédente, la base canonique n'est qu'un compromis entre précision et facilité de gestion. Si nous voulons exprimer le fait que les crocodiles sont toujours de couleur verte, alors il faudra remplacer la contrainte initiale par le graphe [CROCODILE]→(A-Coul)→[COULEUR:vert].

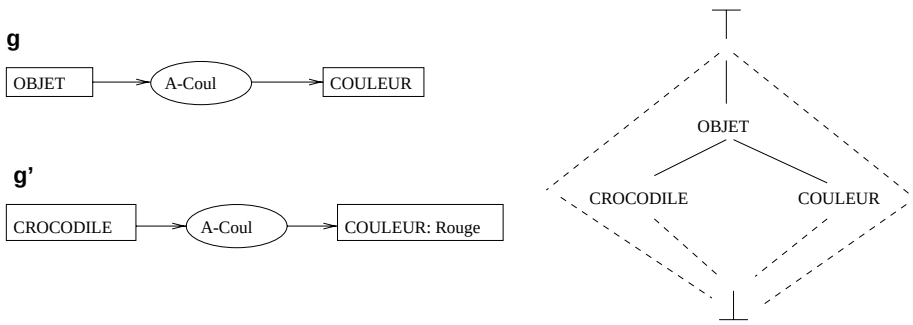


Figure 4.5. L'opérateur ϕ n'est pas intuitif. Les dérivations logiques sont opposées aux dérivations sur les graphes. Nous avons $g \vdash_{Copie+Restriction} g'$ alors que $\phi(g') \supset \phi(g)$

modèle initial soient perdue. Nous présenterons tout au long de ce chapitre une formalisation des graphes conceptuels pour l'implantation d'une approche relationnelle en recherche d'informations. Nous tâcherons de garder intactes les notions de base du formalisme tout en l'adaptant à notre problématique. La généralité du modèle sera préservée, et parfois même étendue. Cette adptation tend à résoudre les problèmes ci-dessus évoquées tout en gardant le pouvoir expressif du formalisme. Afin de comprendre les fondements de l'utilisation de ce formalisme dans la recherche d'informations, nous présentons dans la section suivante un aperçu sur les principales caractéristiques des décisions de pertinence élaborées par ce modèle. Ces caractéristiques seront détectées à l'aide de notre modèle relationnel.

4.2 Le Modèle des Graphes Conceptuels et la RI

Il existe aujourd'hui de nombreux systèmes fondés sur les graphes conceptuels intégrant un module de recherche. Citons par exemple, le système KALIPSOS d'IBM [BDFL88, Far89] destiné à la compréhension des textes écrit en français, le système MENELAS [MEN94, Za93, BZ92] dédié à la recherche médicale et le système de recherche de Myaeng [Mya92, ML93, MKL94]. Cependant, la plupart de ces systèmes ne se situent pas dans le cadre particulier de la recherche d'informations, mais sont plutôt destinés à la représentation et à la gestion des connaissances. Autrement dit, ils ne suivent pas un modèle théorique de recherche d'informations dans leur conception. Le système ELEN [Che92] déroge à cette règle puisqu'il propose une adaptation du modèle des graphes conceptuels au domaine de la recherche d'informations. Il existe ainsi un lien entre le modèle logique de van Rijsbergen [vR86] et le formalisme des graphes conceptuels. Ce lien est discuté dans [Che92, Oun94, Oun95, HOC96].

Si l'on représente les documents et les requêtes avec des graphes conceptuels, la relation de spécialisation $\leq$ définie sur les graphes peut servir à modéliser la notion de pertinence. De plus, l'opérateur de projection fournit la méthode opérationnelle de déduction. Ainsi, un document d indexé par un graphe conceptuel $\chi(d)$ est pertinent pour une requête q représentée par un graphe conceptuel $\chi(q)$ si, et seulement si, $\chi(d)$ est une spécialisation de $\chi(q)$, c'est-à-dire $\chi(d) \leq \chi(q)$.

Cette formalisation de la notion de pertinence repose sur un constat classique en recherche d'informations, à savoir qu'on cherche en général à vérifier que les éléments décrits dans la requête figurent dans les documents pertinents sous une forme identique ou plus précise (plus spécifique).

Définition 4.2.1 (Graphes Conceptuels et Pertinence) *Soit $\mathcal{D}$ l'ensemble des documents du corpus, avec $d \in \mathcal{D}$. Soit $\mathcal{G}$ un ensemble de graphes conceptuels, avec $\chi(q)$ and $\chi(d) \in \mathcal{G}$, où $\chi(d)$ et $\chi(q)$ sont respectivement la représentation du document et de la requête. Nous avons alors,*

d est pertinent pour q si et seulement si $\chi(d) \leq \chi(q)$

L'opérateur de projection permet la mise en oeuvre d'un processus de recherche basé sur cette notion de pertinence. Nous avons en effet montré dans la section précédente que la relation de spécialisation peut être définie par l'existence d'une projection d'un graphe sur sa spécialisation. Ainsi, de part le théorème 10, s'il existe une projection de $\chi(q)$ sur $\chi(d)$, la relation d'ordre $\chi(d) \leq \chi(q)$ est établie[6]. Il en découle d'après les propriétés de l'opérateur ϕ que $\phi(\chi(d)) \supset \phi(\chi(q))$. En d'autres termes la requête est une conséquence logique du document. Il existe ainsi une connexion explicite avec le modèle logique. La projection se comporte comme le principal opérateur de recherche: retrouver les documents impliquant logiquement la requête de l'utilisateur revient à retrouver les documents qui contiennent une projection de cette requête. Elle remplace ainsi un démonstrateur de preuves.

Seuls des graphes conceptuels normalisés sont acceptables comme index dans le système ELEN (Cf. section 4.1.3), afin d'éviter que deux documents différents ne soient indexés par des graphes équivalents. Le système ELEN n'utilise ainsi que des jointures maximales sur les graphes. Une jointure maximale consiste à appliquer l'opération de fusion de deux graphes, non pas sur un seul noeud, mais sur un sous-graphe commun. Cette opération s'exprime par une composition de jointures et de simplifications et ne construit que des graphes normalisés. De plus les relations des graphes sont toutes binaires.

Nous avons mentionné à maintes reprises dans les chapitres 2 et 3 que le formalismes des graphes conceptuels ne permet pas de raisonner sur les relations. D'une part à cause du fait que les propriétés des relations ne sont pas incluses dans la base de connaissances, et d'autre part du fait que la projection ne tient compte que des spécialisations (restriction) de concepts. Par rapport au modèle relationnel que nous avons proposé, il est intéressant à ce stade d'expliciter clairement les propriétés des décisions de pertinence du système ELEN. Cela permet de montrer les potentialités de ce formalisme, mais aussi de détecter ses limites afin d'y remédier.

En collaboration avec Huibers, nous avons dégagé les décisions de pertinence du système ELEN vis-à-vis du cadre logique neutre proposé par ce dernier pour la comparaison des systèmes de recherche d'informations existants. Les résultats obtenus sont présentés dans [HOC95, HOC96]. Ainsi, dans sa modélisation de la notion de pertinence en recherche d'informations, Huibers [HB94, Hui96] élabore un cadre logique formel permettant d'évaluer puis de comparer des systèmes de recherche d'informations sans passer nécessairement par des expérimentations. L'idée est la suivante: expliciter des axiomes et des règles qui sous-tendent la notion de pertinence puis traduire le comportement de chaque système particulier de recherche d'informations en leurs termes. Ainsi, il a été montré, par exemple, que les modèles classiques de recherche d'informations (Booléen, vectoriel,

6. Cela correspond bien à ce que l'on entend en général par correspondance en recherche d'informations, si l'on prend le point de vue offert par les formalismes de représentation de connaissances fondés sur la notion de hiérarchie générique/spécifique.

...) peuvent être spécifiés par une sélection d'un sous-ensemble de ces axiomes et de ces règles. La théorie des situations a été choisie pour modéliser la notion de pertinence, vu la grande liberté et la marge de manoeuvre importante qu'elle présente. On parle alors d'un cadre général pour la recherche d'informations basé sur la théorie des situations. Ce cadre permet de comparer des systèmes de recherche d'informations relativement aux axiomes et règles qu'ils supportent.

La démarche d'évaluation théorique d'un système de recherche d'informations consiste à le transformer vers ce cadre formel. Des fonctions de transformations complètes doivent être proposées à cet effet. La relation de pertinence étant alors définie comme une relation entre situations, elle peut être établie par un système de dérivation de pertinence composé d'un ensemble de postulats. Cet ensemble de postulats est présenté comme une série d'axiomes et de règles établissant les propriétés de relation de pertinence entre situations. Les systèmes de recherche d'informations peuvent ainsi être comparés par rapport aux postulats qu'ils satisfont.

Nous nous sommes basé sur le travail d'évaluation du système ELEN dans le cadre ci-dessus présenté pour spécifier les caractéristiques des graphes conceptuels en termes de notre modèle relationnel. En effet, notre langage de représentation dans ce modèle étant la théorie des situations augmentée d'un ensemble de relations d'endiguement spécifiques à notre démarche, il suffisait de rétablir par rapport à ce dernier la translation effectuée dans [HOC96]. L'analyse de ELEN par rapport à notre modèle relationnel sera présentée dans la section suivante.

4.2.1 Lien avec la Théorie des Situations

Dans ELEN, un graphe conceptuel véhicule des informations. Il peut donc être vu comme une situation dont il convient de spécifier les relons. Il suffit pour cela de comprendre comment est construit un graphe conceptuel. En effet, un graphe n'est autre qu'une collection de relations connectées entre elles par des concepts. Les concepts peuvent être génériques ou individuels. Toutes les composantes d'un graphe ont un rôle spécifique. Par exemple, les concepts décrivent les classes d'objets, alors que les marqueurs dénotent des éléments particuliers. L'idée proposée est donc de transformer chaque composante du graphe (concepts, marqueur, relation) en un relon spécifique. La seule contrainte qu'il faut respecter est de conserver les informations véhiculées par le graphe, en respectant essentiellement les connexions entre types de concepts et marqueurs d'une part, et entre les relations d'autre part.

La translation des graphes conceptuels en situations passe donc par la définition de ces relons, associés à chaque composante du graphe. Par conséquent, le langage des relons $\Re(\mathcal{O}, Param, \mathcal{R})$ du chapitre 3 (section 3.1.6, page 79) doit être instancié:

Définition 4.2.2 (Le Langage de Relons des Graphes Conceptuels) *Soient $\mathcal{T}_c$ un ensemble de types de concepts et $\mathcal{M}$ un ensemble de marqueurs. De plus, soient $\mathcal{T}_r$ un ensemble de types de relations et Param un ensemble de paramètres. Nous dénotons par $\mathcal{C}$ l'union de $\mathcal{T}_c$ et de $\mathcal{M}$. Le langage de relons des graphes conceptuels $\Re_{GCs}(\mathcal{C}, Param, \mathcal{T}_r)$ est défini comme étant le plus petit ensemble tel que:*

- *si $r \in \mathcal{T}_r$, et $\dot{p}$, $\dot{q} \in Param$ alors le relon $\langle\langle r, \dot{p}, \dot{q}; 1\rangle\rangle \in \Re_{GCs}(\mathcal{C}, Param, \mathcal{T}_r)$*
- *si $\text{TYPE}_c \in \mathcal{T}_c$, et $\dot{p} \in Param$ alors le relon $\langle\langle \text{Type}, \text{TYPE}_c, \dot{p}; 1\rangle\rangle \in \Re_{GCs}(\mathcal{C}, Param, \mathcal{T}_r)$*
- *si $m \in \mathcal{M}$, et $\dot{p} \in Param$ alors le relon $\langle\langle \text{Marqueur}, m, \dot{p}; 1\rangle\rangle \in \Re_{GCs}(\mathcal{C}, Param, \mathcal{T}_r)$*

Quand $\mathcal{C}$, Param et $\mathcal{T}_r$ sont connus, nous dénotons le langage $\Re_{GCs}(\mathcal{C}, Param, \mathcal{T}_r)$ simplement par $\Re_{GCs}$.

Dans ELEN, seuls des graphes simples sont utilisés, c'est-à-dire que les noeuds concepts ne peuvent pas avoir pour marqueurs des graphes conceptuels. De ce fait, il n'est pas possible que les arguments des relons soient eux-mêmes des relons. Le langage d'indexation relationnelle $\mathcal{L}(\Re)$ du chapitre 3 (Définition 3.2.1 de la page 81) est ainsi confondu avec le langage $\Re_{GCs}$. Nous notons $\mathcal{L}_{GCs}(\Re_{GCs}) = \Re_{GCs}$ le langage d'indexation relationnelle instancié aux graphes conceptuels. L'ensemble des situations est défini pour être l'ensemble $\mathcal{S}(\Re_{GCs})$ (ou simplement $\mathcal{S}_{GCs}$).

Chaque graphe conceptuel devra être ainsi transformé en une situation de $\mathcal{S}_{GCs}$. Pour ce faire, une fonction de translation *trans* sur les graphes conceptuels est introduite:

Définition 4.2.3 (La Fonction de Translation *trans***)** *Étant donné deux graphes conceptuels g et $h \in \mathcal{G}$. Soit $\mathcal{S}_{GCs}$ l'ensemble des situations, la fonction de translation trans : $\mathcal{G} \rightarrow \mathcal{S}_{GCs}$ des graphes conceptuels vers les situations est définie comme suit:*

- *Pour chaque noeud de concept générique $[u]$ avec un type de concept U, la fonction trans$([u])$ a comme résultat $\{\langle\langle \text{Type}, U, \dot{p}; 1\rangle\rangle\}$, avec $\dot{p}$ un paramètre unique. Un relon de cette forme sera appelé* relon concept.

- *Pour chaque noeud de concept individuel u avec un type de concept U, et un marqueur individuel m, la fonction trans$([U : m])$ a comme résultat $\{\langle\langle \text{Type}, U, \dot{p}; 1\rangle\rangle, \langle\langle \text{Marqueur}, m, \dot{p}; 1\rangle\rangle\}$, avec $\dot{p}$ un paramètre unique. Le relon $\langle\langle \text{Marqueur}, m, \dot{p}; 1\rangle\rangle$ sera appelé* relon marqueur.

- *Pour chaque relation binaire* R *entre deux sous-graphes g et h (dans cet ordre particulier),* R *lie nécessairement deux concepts*[7] *C et D de g et h. Soient $\langle\langle \text{Type}, C, \dot{p}; 1\rangle\rangle \in$ trans(g) et $\langle\langle \text{Type}, D, \dot{q}; 1\rangle\rangle \in$ trans(h), alors trans$(gRh) = \{\langle\langle \text{R}, \dot{p}, \dot{q}; 1\rangle\rangle\} \cup$ trans$(g) \cup$ trans(h). Un relon de type $\langle\langle \text{R}, \dot{p}, \dot{q}; 1\rangle\rangle$ sera appelé* relon relation.

Illustrons cette translation par un exemple:

Exemple 31 $[PERSONNE : Jean] \rightarrow (Parle) \rightarrow [FILLE]$

La translation du graphe ci-dessus aura pour résultat la situation $\{\langle\langle \text{Marqueur}, Jean, \dot{s}; 1\rangle\rangle, \langle\langle \text{Type}, \text{PERSONNE}, \dot{s}; 1\rangle\rangle, \langle\langle \text{Parle}, \dot{s}, \dot{q}; 1\rangle\rangle, \langle\langle \text{Type}, \text{FILLE}, \dot{q}; 1\rangle\rangle\}$.

Deux relons partageant le même paramètre seront dit *relons liés*. Par exemple, les relons $\langle\langle$ *Type*,PERSONNE,$\dot{\text{p}}$; $1\rangle\rangle$ et $\langle\langle$ *Marqueur*,Jean,$\dot{\text{p}}$; $1\rangle\rangle$ sont liés.

Avec l'utilisation des paramètres, il est utile d'établir une notion d'équivalence sur les situations.

Définition 4.2.4 (Équivalence de Situations) *Étant données deux situations S, T de $\mathcal{S}_{GCs}$, la relation d'équivalence $\equiv$ entre situations est définie comme suit:*

$$S \equiv T =^{def} ((\varphi \in (S \cup T)^{(\dot{p_1},\dot{q_1})\ldots(\dot{p_n},\dot{q_n})}) \Leftrightarrow (\varphi \in T)) \text{ et } ((\varphi \in (T \cup S)^{(\dot{r_1},\dot{s_1})\ldots(\dot{r_n},\dot{s_n})}) \Leftrightarrow (\varphi \in S)).$$

7. Nous rappelons que dans le cas du système ELEN , les relations utilisées sont binaires.

où la notation $S^{(\dot{x},\dot{y})}$ représente le remplacement du paramètre $\dot{x}$ dans l'ensemble S par le paramètre $\dot{y}$.[8]

Par exemple, les situations $S = \{\langle\langle Marqueur,\text{John},\dot{p};1\rangle\rangle\}$ et $T = \{\langle\langle Marqueur,\text{John},\dot{q};1\rangle\rangle\}$ sont équivalentes, alors que $\{\langle\langle Parle,\dot{p},\dot{p};1\rangle\rangle\} \not\equiv \{\langle\langle Marche,\dot{r},\dot{r};1\rangle\rangle\}$.

Une situation résultant d'un graphe conceptuel simple par l'application de la fonction de translation *trans* contient au plus trois types de relons: les relons relations, les relons concepts et les relons marqueurs:

Propriété 7 *Selon la nature du graphe conceptuel g en entré, la situation qui lui correspond contient les types de relons suivants:*

- *un relon concept si le graphe g n'est autre qu'un concept générique;*
- *un relon concept et un relon marqueur si g est un concept individuel;*
- *des relons relations, des relons concepts et des relons marqueurs si le graphe g ne se résume pas à un seul concept. Si aucun concept individuel n'apparaît dans g, alors sa situation n'inclura pas de relons marqueurs.*

Ainsi, les situations résultants de la translation des graphes conceptuels de ELEN constituent un sous-ensemble de $\mathcal{S}_{GCs}$. En d'autres termes, il existe des situations de $\mathcal{S}_{GCs}$ ne correspondant pas à des graphes conceptuels. La définition suivante permet d'identifier les situations de $\mathcal{S}_{GCs}$ pouvant correspondre à des graphes conceptuels:

Définition 4.2.5 (Graphe Situation) *Dans le cas du système* ELEN, *une situation* $S_g \in \mathcal{S}_{Gcs}$ *est appelée* graphe situation *si, et seulement si, elle satisfait les conditions suivantes:*

1. *Pour chaque paramètre utilisé dans les relons relation de S_g, il existe un relon concept qui lui soit lié dans S_g.*
2. *Pour chaque relon concept dans S_g il existe au plus un relon marqueur qui lui soit lié dans S_g.*
3. *S'il existe plus d'un relon concept dans la situation S_g, alors pour chaque relon concept il existe un relon relation qui lui soit liée dans S_g.*
4. *Pour chaque relon marqueur dans S_g, il existe un concept relon qui lui soit lié dans S_g.*
5. *Chaque relon relation a exactement deux paramètres.*
6. *Chaque paire de relons relation r_i et r_j a un paramètre en commun, ou il existe une liste de paires $(r_i, r_k) \ldots (r_l, r_j)$ telle que $r_n \in S_g$ et chaque paire a un paramètre en commun.*

La première condition établie le fait que chaque noeud relation doit être connecté avec des concepts. La seconde condition exprime le fait qu'un type de concept a au plus un marqueur qui lui soit associé, La troisième condition exige que dans le cas ou le graphe conceptuel est formé de deux noeuds concepts ou plus, il est nécessaire que ces derniers soient reliés à des relations[9]. La quatrième

8. Comme pour les substitutions en logique, les remplacements de paramètres vérifient la propriété suivante: $S^{(\dot{w},\dot{x})(\dot{y},\dot{z})} =^{def} (S^{(\dot{w},\dot{x})})^{(\dot{y},\dot{z})}$

9. Nous rappelons qu'un graphe conceptuel peut être formé d'un seul noeud concept.

condition stipule que chaque marqueur du graphe doit nécessairement être associé à un type de concept. La cinquième condition tient compte du fait que les relations dans le système ELEN sont toutes binaires. Cette condition peut être facilement supprimée sans que la démarche ici présentée ne subsisse un changement quelconque[10]. Enfin la sixième condition vérifie le fait que les relations sont bien connectées et que le graphe est ainsi connexe.

Dans le cas des graphes normalisés, la fonction *trans* est injective:

Lemme 3 (Graphes Normalisés) *Si* $trans(g) \equiv trans(h)$ *alors* $g = h$

Preuve 6 *Les graphes conceptuels étant normalisés, il n'est pas possible d'avoir des relations redondantes. Les situations* $trans(g)$ *et* $trans(h)$ *vérifient bien entendu les conditions de la définition 4.2.5. Ces deux situations contiennent au plus trois types de relons, à savoir des relons concepts, des relons relations et des relons marqueurs. La structure des deux graphes peut être directement déduite de ces relons. Chaque relon concept peut être directement transformé en un noeud concept, chaque relon relation peut être transformé en un noeud relation dont les arcs et leur orientation sont déterminés par l'ordre de ces paramètres dans le relon relation. Ces même paramètres indiquent de façon unique quels concepts sont reliés à la dite relation. Enfin, un relon marqueur peut être transformé en un marqueur d'un type de concept, selon le paramètre qu'il véhicule. Ainsi, chaque graphe situation correspond à un unique graphe. Quand deux situations sont équivalentes, elles peuvent être mises sous forme identiques par renommage de variables (définition 4.2.4). Elles donneront ainsi lieu à un même graphe conceptuel.*

Par exemple, les deux graphes g et h de la figure 4.6 ont la même translation en termes de situations. Seulement ils ne sont pas normalisés. ils ne vérifient donc pas le lemme précédent.

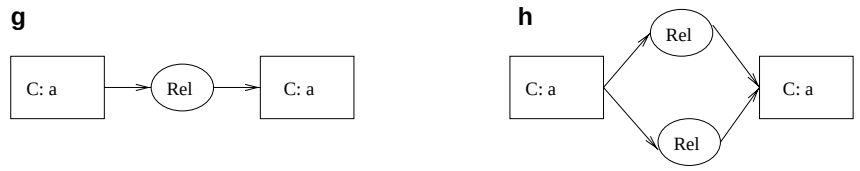


Figure 4.6. Ici $g \equiv h$ mais $g \neq h$. Ces deux graphes ont cependant la même translation en termes de situations

Si les graphes conceptuels utilisés par ELEN n'étaient pas normalisés, le lemme précédent serait le suivant:

Lemme 4 (Graphes Quelconques) *Si* $trans(g) \equiv trans(h)$ *alors* $g \equiv h$

où la relation $\equiv$ désigne dans la première apparition la notion d'équivalence entre situations (définition 3.3.2) et dans la seconde apparition la notion d'équivalence entre graphes conceptuels définie par la propriété 6 (page 147). En effet, dans le cas d'un graphe conceptuel avec des relations redondantes, la translation du graphe obtenu après la simplification de ces relations redondantes

10. Nous avons vu dans les chapitres 2 et 3 qu'il est parfois souhaitable d'avoir des relations n-aire. Un système de recherche d'informations fondé sur les graphes conceptuels devra offrir une telle caractéristique.

aura le même résultat que celle obtenue pour le graphe du départ. Cela est dû aux propriétés ensemblistes: $\mathcal{S} \cup \{\langle\langle R, \dot{p}, \dot{q}; 1\rangle\rangle\} = \mathcal{S} \cup \{\langle\langle R, \dot{p}, \dot{q}; 1\rangle\rangle\} \cup \{\langle\langle R, \dot{p}, \dot{q}; 1\rangle\rangle\}$.

L'inverse du lemme précédent n'est évidemment pas vrai. Les paramètres étant affectés de façon unique aux concepts, deux noeuds non joints auront nécessairement des paramètres différents. Dans la figure 4.7 qui suit, nous avons les translations suivantes:

$trans(g) = \langle\langle \mathit{Marqueur}, \mathrm{a}, \dot{p}; 1\rangle\rangle, \langle\langle \mathit{Type}, \mathrm{C}, \dot{p}; 1\rangle\rangle, \langle\langle \mathit{Rel}, \dot{p}, \dot{q}; 1\rangle\rangle, \langle\langle \mathit{Type}, \mathrm{C}, \dot{q}; 1\rangle\rangle, \langle\langle \mathit{Marqueur}, \mathrm{a}, \dot{q}; 1\rangle\rangle\}$

alors que

$trans(h) = \langle\langle \mathit{Marqueur}, \mathrm{a}, \dot{p}; 1\rangle\rangle, \langle\langle \mathit{Type}, \mathrm{C}, \dot{p}; 1\rangle\rangle, \langle\langle \mathit{Rel}, \dot{p}, \dot{q}; 1\rangle\rangle, \langle\langle \mathit{Type}, \mathrm{C}, \dot{q}; 1\rangle\rangle, \mathit{Marqueur}, \mathrm{a}, \dot{q}; 1\rangle\rangle, \langle\langle \mathit{Rel}, \dot{p}, \dot{s}; 1\rangle\rangle, \langle\langle \mathit{Type}, \mathrm{C}, \dot{s}; 1\rangle\rangle, \langle\langle \mathit{Marqueur}, \mathrm{a}, \dot{s}; 1\rangle\rangle\}$

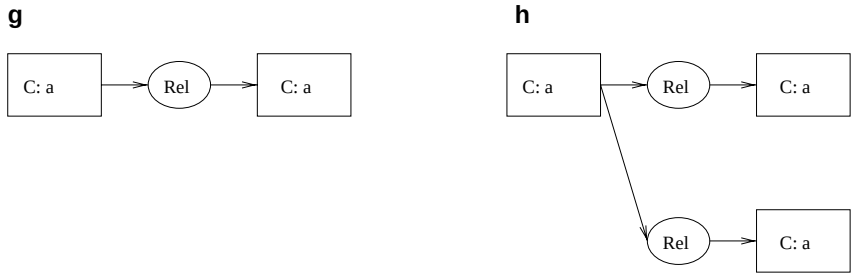


Figure 4.7. Les deux graphes g et h sont équivalents. Ils n'ont cependant pas la même translation en termes de situations.

Dans ELEN, les raisonnements sont uniquement basés sur les propriétés des concepts. Pour ce faire, le treillis des types de concepts $\mathcal{T}_c$ est utilisé. Ce treillis ne contient que des relations de spécialisation/généralisation. En termes de représentation de connaissances du domaine, il est beaucoup plus pauvre que les thésaurus généralement utilisés par les systèmes de recherche d'informations classiques. En particulier, par rapport aux diverses types de relations d'endiguements que nous avons présenté dans le chapitre 3, seule la propriété de restriction est utilisée. Ainsi, la hiérarchie des types de concepts et les propriétés qui en découlent sont simulés à l'aide d'une relation d'endiguement faible $\rightarrow_r^f$ restreinte à la seule propriété de restriction. Par exemple, si HOMME $\leq$ PERSONNE est définie dans la hiérarchie, alors $\langle\langle \mathit{Type}, \mathrm{HOMME}, \dot{p}; 1\rangle\rangle \rightarrow_r^f \langle\langle \mathit{Type}, \mathrm{PERSONNE}, \dot{p}; 1\rangle\rangle$ est valide pour n'importe quel paramètre $\dot{p}$.

À partir de cette constatation, il ne reste plus qu'à spécifier les règles de dérivation que présente un système logique à base de graphes conceptuels. Le langage de dérivation étant $\mathcal{P}(\Re_{GCs})$ construit comme dans la définition 3.4.1 (page 108).

4.2.2 Les Propriétés du Modèle

Partant de la translation des graphes conceptuels décrite dans la section précédente et du langage de dérivation qui en découle, nous pouvons modéliser les décisions de pertinence véhiculées par le système ELEN . Rappelons qu'une décision de pertinence au sens des graphes conceptuels est fondée sur une implication logique entre les documents et les requêtes. Cette implication est liée à la relation d'ordre $\leq$ sur les graphes, laquelle est matérialisée par l'opérateur de projection

(théorème 8 de la page 145).

Ce fondement logique du système ELEN explique la présence de l'axiome de *Réflexivité* et de la règle du *Cut*. Nous avons en effet mentionné dans le chapitre précédent que, selon Gabbay [Gab93], un système logique doit vérifier un certain nombre d'axiomes et de règles. Ainsi la relation de pertinence dans ELEN est réflexive. Pour s'en rendre compte, il suffit de constater que tout graphe se projette sur lui-même.

Réflexivité

$$S \boxrightarrow_R S$$

La règle du *cut* existe dans pratiquement tous les système logiques. Les graphes conceptuels ne dérogent pas à cette règle. Il s'agit d'une propriété simple de la relation d'ordre $\leq$

Cut

$$\frac{S \cup T \boxrightarrow_R U \quad S \boxrightarrow_R T}{S \boxrightarrow_R U}$$

Nous avons également vu dans le chapitre précédent que selon la nature monotone du système logique considéré, il devra présenter une règle de monotonicité ou de non-monotonicité. Nous avons alors mentionné que les graphes conceptuels présentent une forme de monotonicité circonspecte, à savoir la règle suivante:

Monotonicité Prudente

$$\frac{S \boxrightarrow_R T \quad S \cap U \neq \emptyset}{S \cup U \boxrightarrow_R T}$$

Le terme "Monotonicité" provient du fait que la pertinence est préservée lors de l'ajout d'informations. Le terme "Prudente" indique lui que cet ajout n'est pas acceptable dans tous les cas. En effet dans le domaine des graphes conceptuels, l'ajout d'informations signifie un ajout d'un graphe conceptuel à un graphe de départ. Comme les graphes sont connexes, cet ajout correspondra nécessairement à une jointure entre le graphe de départ et l'information ajoutée. La jointure n'étant envisageable que si ces deux graphes présentent un sous-graphe commun, il est clair qu'il faudra vérifier que les situations qui leurs correspondent partagent un certain nombre de relons. Par exemple étant donné, que l'ensemble $\{\langle\langle \mathit{Type}, \text{COUREUR}, \dot{p}; 1\rangle\rangle, \langle\langle \mathit{Gagne}, \dot{p}, \dot{q}; 1\rangle\rangle, \langle\langle \mathit{Type}, \text{COURSE}, \dot{p}; 1\rangle\rangle\}$ est pertinent pour $\{\langle\langle \mathit{Type}, \text{COUREUR}, \dot{p}; 1\rangle\rangle, \langle\langle \mathit{Gagne}, \dot{p}, \dot{q}; 1\rangle\rangle, \langle\langle \mathit{Type}, \text{COURSE}, \dot{p}; 1\rangle\rangle\}$ (en utilisant la réflexivité), une nouvelle situation est formée par l'ajout (au sens union ensembliste) de $\{\langle\langle \mathit{Type}, \text{COUREUR}, \dot{p}; 1\rangle\rangle, \langle\langle \mathit{Etat}, \dot{p}, \dot{q}; 1\rangle\rangle, \langle\langle \mathit{Type}, \text{CONTENT}, \dot{p}; 1\rangle\rangle\}$ au premier ensemble. Dans ce cas, la condition de "*Prudence*" est vérifiée puisque le relon $\langle\langle \mathit{Type}, \text{COUREUR}, \dot{p}; 1\rangle\rangle$est présent aussi bien dans la situation de départ que dans l'information ajoutée[11].

La règle de *Monotonicité Prudente* nous permet de conclure que le nouvelle situation est toujours pertinente pour $\{\langle\langle \mathit{Type}, \text{COUREUR}, \dot{p}; 1\rangle\rangle, \langle\langle \mathit{Gagne}, \dot{p}, \dot{q}; 1\rangle\rangle, \langle\langle \mathit{Type}, \text{COURSE}, \dot{p}; 1\rangle\rangle\}$. En termes de graphes conceptuels, cela revient à dire, tout simplement, que si un graphe $\chi(q)$ se projette sur

11. Le lecteur aura remarqué qu'un renommage de variables est implicite pour pouvoir appliquer une opération de jointure, et donc une intersection, entre la situation de départ et celle ajoutée. Nous verrons plus loin dans la section 4.5 qu'une jointure n'est autre qu'un procédé de renommage de variables.

un graphe $\chi(d)$ alors il se projettera nécessairement sur la jointure de $\chi(d)$ avec un autre graphe. De plus l'affectation d'un référent à un concept générique est également supportée par cette règle. Il s'agit d'ajouter à la situation de départ un graphe situation avec un relon concept et un relon marqueur, de façon à ce que l'intersection entre les deux situations en jeu soit non vide[12]. En résumé, la règle de *Monotonicité Prudente* ne viole pas l'opérateur de projection, mais applique simplement une spécialisation au graphe de départ.

La règle qui suit simule l'opérateur de restriction des types de concepts sur les graphes. Comme nous l'avons vu à la fin de la section précédente, cet opérateur correspond à un endiguement faible de type restriction. Un graphe conceptuel est pertinent pour sa généralisation. Il s'agit de la définition même de pertinence dans le système ELEN. Le graphe résultat est le même que celui de départ, sauf que un des types de concepts de ce dernier a été spécialisé. En particulier, pour un graphe g constitué d'un unique concept, l'application de cette règle revient à affirmer qu'il existe un relon concept i associé à $trans(g)$, tel que i est en relation d'endiguement faible de type restriction avec un relon concept j ($i \rightarrow_r^f j$) dans le graphe situation résultant.

Endiguement Faible

$$\frac{\phi \xrightarrow{Restr.}{}^{f}_{r} \psi \qquad S \boxrightarrow_R T \cup \{\phi\}}{S \boxrightarrow_R T \cup \{\psi\}}$$

La propriété d'endiguement (ED) (voir chapitre précédent) n'est pas applicable dans le contexte d'un système à base de graphes conceptuels simples. En effet, dans ce cas, le langage d'indexation relationnelle est confondu avec le langage des relons puisqu'il n'est pas possible d'avoir des concepts qui soient eux-mêmes des graphes conceptuels (voir page 153).

Enfin, les deux règles d'équivalence suivantes expriment le fait que deux graphes équivalents ont exactement les mêmes dérivations de pertinence. Si un graphe g se projette sur un graphe h alors g se projettera également sur un graphe qui lui soit équivalent. Cela découle du fait que la projection n'est pas nécessairement injective. Les deux règles d'équivalence mentionnent également le fait qu'il importe peu la façon dont les graphes ont été construits, c'est-à-dire que l'ordre d'application des opérateurs de formation de graphes n'a pas une influence sur les décisions de pertinence. Par exemple avec ces deux règles, nous pouvons à partir de $S \cup T \boxrightarrow S$ inférer $T \cup S \boxrightarrow S$.

Équivalence

$$\frac{S \boxrightarrow_R T \qquad S \equiv U}{U \boxrightarrow_R T} \qquad\qquad \frac{S \boxrightarrow_R T \qquad T \equiv U}{S \boxrightarrow_R U}$$

Les postulats de pertinence ci-dessus présentés constituent l'ensemble minimal des décisions de pertinence de ELEN, à partir desquelles toutes les autres décisions du système peuvent être déduites. Par exemple, la transitivité est valide dans ELEN, puisque qu'il s'agit d'une propriété intrinsèque de la relation de spécialisation $\leq$.

Transitivité

$$\frac{S \boxrightarrow_R T \qquad T \boxrightarrow_R U}{S \boxrightarrow_R U}$$

12. Nous rappelons ainsi que les situations ajoutées doivent être conformes à celles spécifiées dans la définition 4.2.5(page 154).

La transitivité est une règle de dérivation déduite de tout système logique contenant les règles de monotonicité, d'équivalence et du Cut.

De plus, la disjonction découle de la règle de monotonicité prudente et de l'endiguement faible, elle constitue également une propriété évidente de la relation d'ordre $\leq$:

Disjonction

$$\frac{S \boxrightarrow_R T \quad U \boxrightarrow_R T}{S \cup U \boxrightarrow_R T}$$

En termes de graphes conceptuels, si $g_1 \leq g$ et $g_2 \leq g$ alors $g_1 \oplus_j g_2 \leq g$, où $\oplus_j$ correspond à l'opérateur de jointure entre graphes. En effet, par définition nous avons $g_1 \oplus_j g_2 \leq g_1$ et $g_1 \oplus_j g_2 \leq g_2$. Comme $g_1 \leq g$ et $g_2 \leq g$, on en déduit par transitivité la conséquence $g_1 \oplus_j g_2 \leq g$.

En raisonnant maintenant sur les situations, si g_1 est une spécialisation de g c'est qu'il reprend le graphe g tout en l'enrichissant soit par des restrictions, soit par des ajouts d'informations (jointures avec d'autres graphes). Autrement dit, g est inclus (au sens sous-graphe) à des restrictions de concepts près dans g_1. De la même façon g est également inclus dans g_2 à des restrictions de concept près. Il est alors facile de constater que g_1 et g_2 partagent un sous-graphe commun à des restrictions de concepts près. La règle d'endiguement faible permet de faire passer une des deux translations associée au graphe g_1 ou au graphe g_2 vers une autre plus spécifique (correspondant à une spécialisation du graphe de départ) de façon à ce qu'elle partage un même sous-ensemble (sous-graphe) avec la translation de l'autre graphe. Cet ensemble étant nécessairement non vide (g n'étant pas vide), il suffit ensuite d'appliquer la règle de monotonicité prudente.

Enfin, la règle du bouclage peut également être déduite directement à partir de la transitivité. Elle est aussi une propriété intrinsèque de la relation d'ordre $\leq$.

Bouclage

$$\frac{S_0 \boxrightarrow_R S_1 \quad S_1 \boxrightarrow_R S_2 \quad \ldots \quad S_{k-1} \boxrightarrow_R S_k \quad S_k \boxrightarrow_R S_0}{S_0 \boxrightarrow_R S_k}$$

Nous prouvons dans [HOC96], que les 5 premiers postulats précédemment présentés sont suffisant pour simuler la notion de pertinence dans ELEN :

Définition 4.2.6 (Les Décisions de Pertinence dans ELEN) *Le système de dérivation de pertinence $\mathcal{E}$ correspondant aux décisions de pertinence du système* ELEN *est défini comme l'ensemble des postulats {Réflexivité, Monotonicité Prudente, Endiguement faible, Équivalence, Cut}.*

En particulier les deux théorèmes suivants sont énoncés. Les preuves sont laissées comme exercices. Un résumé des deux démonstrations est donné comme indication. Pour une preuve détaillée, le lecteur pourra consulter [HOC96].

Théorème 11 (Correction) *Le système de dérivation de pertinence $\mathcal{E}$ est correct. C'est-à-dire, pour tous les graphes conceptuels g et $h \in \mathcal{G}$, si $\vdash_{\mathcal{E}} trans(g) \boxrightarrow_R trans(h)$ alors $g \leq h$.*

Preuve 7 *Résumé: il suffit de montrer que chaque postulat du système de dérivation de pertinence $\mathcal{E}$ est associé à un ou plusieurs opérateurs de formation de graphes. Nous avons déjà mentionné à quoi correspondent certains des è postulats précédents.*

Théorème 12 (Complétude) *Le système de dérivation de pertinence $\mathcal{E}$ est complet. C'est-à-dire, pour tous les graphes conceptuels g et $h \in \mathcal{G}$, si $g \leq h$ alors $\vdash_{\mathcal{E}} trans(g) \Box\!\!\rightsquigarrow_R trans(h)$*

Preuve 8 *Résumé: supposons que $g \leq h$, cela signifie que g est construit à partir de h par le biais des quatre opérateurs de formation de graphes. Pour chaque opérateur, nous avons à inspecter si cet opérateur peut être traduit en termes du système de dérivation de pertinence $\mathcal{E}$. Pour une preuve détaillée, le lecteur pourra consulter [HOC96].*

4.2.3 Conclusions sur le Modèle

L'utilisation des graphes conceptuels dans le système ELEN avait pour principal objectif, la création d'un système orienté vers la précision des réponses, évitant ainsi aux utilisateurs d'être submergés par la masse de documents retournés. Les décisions de pertinence de ELEN sont donc strictes, dans un sens où finalement très peu de règles de pertinence sont données. De plus, la base de connaissances du système est très pauvre puisqu'elle se limite à un treillis de types de concepts et à un treillis de types de relations. Par conséquent, les possibilités de dérivation d'une requête à partir d'un document sont très minces, essentiellement quand la requête fait allusion à des connaissances implicites sur le domaine non considérées par le système. Dans ELEN, les connaissances représentées sont non seulement pauvres, mais pour le peu qu'il représente, le système du fait qu'il se base sur l'opérateur de projection comme support de ses décisions de pertinence n'exploite pas le treillis des relations. C'est que l'opérateur de projection ne tient pas compte des relations. Il n'y a pas donc de quoi être surpris quand très peu de documents sont retrouvés par le système. Afin de délivrer plus de documents et augmenter ainsi le rappel, d'autres décisons de pertinence doivent être ajoutées au système de dérivation de pertinence.

Notre démarche dans la suite de ce chapitre est d'abord d'étendre la base de connaissances du système de façon à ce qu'elle intègre les divers types de relations d'endiguement que nous avons présenté dans le chapitre précédent. En particulier cette base devra inclure les propriétés des relations, qu'elles soient syntaxiques ou sémantiques. Une fois cette extension réalisée, nous pouvons ajouter d'autres règles de pertinence essentiellement pour tenir compte de toute la richesse de la nouvelle base de connaissances ainsi obtenue. Ces règles devraient étendre la définition de la projection. Par rapport à notre modèle relationnel du chapitre 3, nous pouvons imaginer l'ajout de deux types de règles de pertinence:

Des règles de dérivation sur les relations: Il s'agit principalement de tenir compte des relations et leurs propriétés sémantiques dans l'opérateur de projection. L'ajout de telles règles dans l'opérateur de projection est un des objectifs de notre approche. La précision des réponses n'est pas affectée puisque nous nous contentons d'exploiter des informations toujours valides et propres à la sémantique des relations. Une fois, la base de connaissances étendue par les relations d'endiguement mettant en jeu les propriétés des relations et leurs comportements, nous pouvons ajouter au système de dérivation $\mathcal{E}$, les règles qui leurs correspondent dans le modèle relationnel du chapitre 3.

Par exemple, si une relation R est un alias d'une autre relation R' alors nous déduisons que quand deux concepts C et D sont reliés par la relation R, alors ces mêmes concepts peuvent être reliés par la relation R'. Nous pouvons ainsi utiliser la règle d'*Équivalence Logique à*

Gauche définie dans le chapitre précédent:

Equivalence Logique à Gauche (instanciée)

$$\frac{\langle\langle R,\dot{c},\dot{d};1\rangle\rangle \equiv_r \langle\langle R',\dot{e},\dot{f};1\rangle\rangle \quad S\cup\{\langle\langle R,\dot{c},\dot{d};1\rangle\rangle\} \boxrightarrow_R T}{S\cup\{\langle\langle R',\dot{e},\dot{f};1\rangle\rangle\} \boxrightarrow_R T}$$

Cette règle de dérivation ne s'applique que dans le cas où la base de connaissances inclue la relation d'endiguement équivalent ci-dessus mentionnée. D'une façon plus générale, les décisions de pertinence du système ELEN doivent être étendues par les diverses règles que nous avons spécifié dans le chapitre 3 et qui manquent dans $\mathcal{E}$.

Des règles de dérivation incertaines: Les règles strictes précédentes favorisent le taux de rappel sans pour autant diminuer la précision des réponses En effet, elles reposent toutes sur des connaissances implicites toujours valides dans le domaine d'application considéré. Cependant, en recherche d'informations, une fonction de correspondance doit être capable d'établir une mesure de pertinence non binaire, entre les documents et la requête. En général une requête exprime la recherche d'un document satisfaisant le plus d'éléments de la requête. Intuitivement cela revient à exhiber également un sous-graphe de la requête $\chi(q)$ capable de se projeter sur un document puis d'évaluer son importance. Chevallet [Che92] introduit ainsi la notion de projection partielle. Afin d'avoir des classes de pertinence, nous pouvons aussi alléger les contraintes de l'opérateur de projection. Par exemple, il n'est pas possible de déduire qu'un graphe représentant l'information "*Jean aime les pâtisseries*" est pertinent pour un graphe représentant la phrase "*Jean aime les gâteaux*", car le second graphe est une spécialisation du premier. En termes de projection, c'est plutôt la déduction inverse qui est valide. Pourtant un utilisateur cherchant une situation où Jean est un amateur de gâteaux (un marqueur possible pour pâtisseries) serait intéressé par les documents mentionnant que le même Jean est un amateur de pâtisseries tout court. La règle de dérivation incertaine suivante peut être ajoutée aux système de dérivation de pertinence du système ELEN:

Union de Relation Monotone

$$\frac{S \boxrightarrow_R T}{S \boxrightarrow_R T\cup\{\langle\langle Marqueur,\mathrm{m},\dot{\mathrm{p}};1\rangle\rangle\}}$$

Dans la règle précédente, tout se passe comme si la projection ne tient pas compte des marqueurs. Une contrainte est ainsi éliminée des conditions que doit vérifier l'opérateur de projection. Cette même approche pourra être généralisée à une notion de projection partielle plus globale que celle définie par Chevallet, où des relations, des concepts et des référents peuvent être ajoutés (ou supprimés) aux documents (ou aux requêtes) de façon à ce qu'une projection de la requête sur le document soit possible. Nous pouvons également représenter des relations d'endiguement incertaines de type de celles utilisées à la fin du chapitre 3 (section 3.5.1, page 118). Sur ces connaissances nous pouvons appliquer les règles de dérivation incertaines développées dans la même section.

À l'inverse des décisions de pertinence incluant les propriétés des relations dans l'opérateur de projection, les règles de dérivation incertaines introduisent du bruit. Les documents générés par ces règles auront un degré de pertinence moindre que celui assigné aux documents obtenus par une

projection exacte ou par l'application des propriétés des relations. Selon le taux d'information non implicites ajoutées au document ou à la requête, le degré de pertinence diminue comme cela est indiqué dans le principe d'incertitude de Rijsbergen.

Dans la suite, nous proposons les adaptations nécessaires au formalisme des graphes conceptuels, afin que des décisions de pertinence basées sur les relations et leur propriétés puissent être prises.

4.3 Un Canon Adapté à une Approche Relationnelle

Nous reprenons dans cette section les éléments de base du modèle des graphes conceptuels tout en adoptant une formulation légèrement différente de celle introduite par J. Sowa [Sow84]. Il s'agit surtout d'étendre le modèle de base par des notions dont nous avons besoin pour qu'une approche relationnelle puisse être possible avec les graphes conceptuels. La notion de *canon* introduite par Sowa définit le vocabulaire de base avec lequel les connaissances relatives à un domaine sont représentées. Il constitue le niveau terminologique du formalisme des graphes conceptuels. Un graphe conceptuel n'a de sens que relativement à un canon donné. Cette notion de canon a été volontairement définie par Sowa en des termes généraux afin qu'elle puisse facilement se prêter à des multiples extensions selon le type d'application à laquelle le modèle s'attaque. Ainsi, si des notions comme le *treillis des types de concepts* ou la *relation de conformité* sont clairement définies, les notions de *marqueur*, de *référent* ou de *graphe canonique* pouvaient donner lieu à diverses interprétations.

Les personnes qui se sont intéressées ensuite au modèle des graphes conceptuels ont proposé chacune leur propre vision du modèle de façon à ce que le résultat soit conforme à leurs besoins. Ainsi en 1992, M. Chein et M.L. Mugnier [CM92] ont publié un travail de formalisation d'une partie du modèle. Leur objectif était de faire munir les graphes conceptuels d'un cadre formel facilitant tout travail d'investigation ultérieur pour la réalisation d'un système de représentation de connaissances et de raisonnements fondé sur les graphes conceptuels. Ceci en précisant une partie des notions introduites par Sowa. Un travail semblable a été aussi réalisé par M. Wermelinger et al. en 1994 dans [WL94] et par Willems en 1995 [Wil95].

Notre objectif est de spécifier un modèle théorique pour les graphes conceptuels, conforme à nos besoins en recherche d'informations. En se servant des travaux antérieurs de formalisation [CM92, Mug93, Car96], nous introduisons des nouvelles définitions permettant la conception d'un modèle opérationnel ayant pour support les graphes conceptuels et intégrant une approche d'indexation relationnelle.

Le canon que nous allons proposer ici est plus général que ceux présentés dans [Sow84, CM92]. Il tient compte principalement des propriétés des relations et prépare le terrain à une extension du formalisme par des nouvelles règles de formation de graphes. Il permet une implantation du modèle d'indexation relationnel que nous avons présenté dans le chapitre précédent.

Définition 4.3.1 (Le Canon) *Le canon des graphes conceptuels est un 7-uplet* $\langle \mathcal{T}_c, \mathcal{T}_r, \mathcal{M}, Conf, \mathcal{S}_r, \mathcal{K}_s, \mathcal{E} \rangle$ *regroupant les éléments suivants:*

- *Un ensemble* $\mathcal{T}_c$ *de types de concepts, ordonné en une structure de treillis fini, notée* $\langle \mathcal{T}_c, \leq, \wedge, \vee \rangle$
- *Un ensemble* $\mathcal{T}_r$ *de types de relations strictement disjoint de l'ensemble* $\mathcal{T}_c$*, muni d'une relation d'ordre partiel, notée* $\leq_\sigma$*. Cette relation d'ordre est associée à une fonction injective* σ

permettant de comparer deux relations d'arité différentes.

- *Un ensemble dénombrable* $\mathcal{M} = \zeta \cup W \cup \{\star, \tilde{\star}, \exists\}$ *de* marqueurs quantifiés, *permettant de dénoter les individus du domaine de discours. Les marqueurs peuvent être vus comme des substituts des référents dans le monde réel. Le marqueur "*", appelé* marqueur générique *substitue n'importe quel individu du domaine de discours, alors que le* marqueur absurde "$\tilde{\star}$" *n'est représentant d'aucun individu du domaine de discours. Le* marqueur existentiel "$\exists$" *incarne un individu particulier du domaine. Les ensembles* ζ *et* W *sont utilisés pour dénoter des individus appartenant aux dénotations des types de concepts.* ζ *est appelé l'ensemble des* marqueurs individuels *ou l'ensemble des référent.* W *est un ensemble abstrait d'individus du domaine de discours. C'est l'ensemble des* témoins *dont l'utilisation sera largement discutée tout au long de ce chapitre.*

  $\mathcal{M}$ *est ordonné en une structure de treillis.*

- *Une relation de conformité* $Conf$ *sur* $\mathcal{M}$ *et* $\mathcal{T}_c$*, qui définit des contraintes d'association entre un type de concept et un marqueur quantifié.*

- *Une signature* $\Sigma_r = (r, n, C_1, \ldots, C_n)$ *est associée à chaque type de relation* $r \in \mathcal{T}_r$*. Chaque signature* Σ_r *est représentée par un graphe conceptuel, dit* graphe signature. *Dans une signature* Σ_r*,* n *est l'arité du type de relation* r*, et* $C_1, \ldots, C_n$ *constituent les types de concepts maximaux du treillis* $\mathcal{T}_c$*, pouvant être reliés par le type de relation* r*. Nous notons par* $\mathcal{S}_r$ *l'ensemble des graphes signatures et par* $\Sigma_i(r)$ *le* i^{eme} *type de concept* C_i *dans la signature de* r*.*

- *Un ensemble* $\mathcal{K}_s$ *de* propriétés *sur les types de relations de* $\mathcal{T}_r$*. Cet ensemble est représenté sous forme de graphes conceptuels utilisant des relations d'ordre supérieur à deux.*

- *Un ensemble* $\mathcal{E}$ *de relations d'*endiguement *de type de celles spécifiées dans le chapitre 3. Cet ensemble précise par exemple ce que l'on entend par une relation* transitive *ou par le fait qu'une relation vérifie la propriété de permutation. Plus généralement, cet ensemble définie les propriétés des relations. Cet ensemble sera représenté par un ensemble de graphes conceptuels avec contextes. Les contextes permettent d'exprimer des règles de production sur les graphes.*

Les éléments de cette définition seront explicités dans les sections suivantes.

4.3.1 Le Treillis des Types de Concepts $\mathcal{T}_c$

Le treillis des types de concepts est ordonné par une relation "sorte de", notée $\leq$. Ce treillis est orienté du type le plus spécifique vers le type le plus général tel que pour tout $t_1, t_2, t_3 \in \mathcal{T}_c$:

1. si $t_1 \leq t_2$ alors t_1 est dit *sous-type de* t_2, et t_2 est dit *super-type de* t_1
2. si $t_1 \leq t_2$ et $t_1 \neq t_2$ alors t_1 est dit *sous-type propre de* t_2, et t_2 est dit *super-type propre de* t_1
3. si $t_1 \leq t_2$ et $t_1 \leq t_3$ alors t_1 est *un sous-type commun de* t_2 et de t_3
4. si $t_1 \geq t_2$ et $t_1 \geq t_3$ alors t_1 est *un super-type commun de* t_2 et de t_3

Le treillis des types de concepts $\langle \mathcal{T}_c, \leq, \wedge, \vee \rangle$ est muni de deux opérateurs binaires notés $\wedge$ et $\vee$ ayant les propriétés suivantes:

1. $\forall t_1, t_2 \in \mathcal{T}_c$, $t_1 \wedge t_2 \leq t_1$ et $t_1 \wedge t_2 \leq t_2$
2. Si $t_1, t_2, t_3 \in \mathcal{T}_c$, avec $t_3 \leq t_1$ et $t_3 \leq t_2$ alors $t_3 \leq t_1 \wedge t_2$
3. $\forall t_1, t_2 \in \mathcal{T}_c$, $t_1 \leq t_1 \vee t_2$ et $t_2 \leq t_1 \vee t_2$
4. Si $t_1, t_2, t_3 \in \mathcal{T}_c$, avec $t_1 \leq t_3$ et $t_2 \leq t_3$ alors $t_1 \vee t_2 \leq t_3$

Tout couple d'éléments (t_1, t_2) de $\mathcal{T}_c$ a donc un unique *sous-type maximal* noté $t_1 \wedge t_2$ et un unique *super-type minimal* noté $t_1 \vee t_2$, appelés respectivement *borne inférieure* et *borne supérieure* de t_1 et t_2. Le treillis des types de concepts $\langle \mathcal{T}_c, \leq, \wedge, \vee \rangle$ est borné par le type *Universel*, noté $\top$ et par le type *Absurde*, noté $\perp$, respectivement le supremum et l'infimum du treillis.

Un treillis de types $\langle \mathcal{T}, \leq, \wedge, \vee \rangle$ est construit par rapport à un domaine d'application particulier. Chaque treillis de types peut être interprété à travers une algèbre relative à ce domaine.

Définition 4.3.2 (Une C-Algèbre pour $\mathcal{T}_c$) *Une algèbre associée à un treillis de types de concepts $\langle \mathcal{T}_c, \leq, \wedge, \vee \rangle$ est une structure*

$A = \langle \mathcal{D}^A, (\mathcal{T}_c^A)_{t \in \mathcal{T}_c} \rangle$

où

- *$\mathcal{D}^A$ est un ensemble non vide, appelé le* domaine *de A (ou univers de discours)*
- *pour tout type t dans $\mathcal{T}_c$, t^A est un sous-ensemble du domaine, en particulier $\top^A = \mathcal{D}^A$ et $\perp^A = \emptyset$*
- *l'opération binaire $\wedge$ déterminant la borne inférieure de deux types donnés dans $\mathcal{T}_c$ est interprétée comme une intersection sur $\mathcal{D}^A$, c'est-à-dire $(t_1 \wedge t_2)^A = t_1^A \cap t_2^A$ pour deux types t_1 et t_2 dans $\mathcal{T}_c$*
- *l'opération binaire $\vee$ déterminant la borne supérieure de deux types donnés est interprétée comme une union sur $\mathcal{D}^A$, c'est-à-dire $(t_1 \vee t_2)^A = t_1^A \cup t_2^A$ pour deux types t_1 et t_2 dans $\mathcal{T}_c$*

L'ensemble t^A associé à un type de concept $t \in \mathcal{T}_c$ par la C-Algèbre est appelé la *dénotation* de t.

4.3.2 Le Treillis des Types de Relations $\mathcal{T}_r$

Nous avons vu au chapitre 2 (section 2.2.1[page 22] que le nombre des primitives est différent d'un système à un autre. Le fait d'avoir de nombreux types de relations facilite la précision des représentations. Par exemple, il est facile de représenter efficacement les relations d'agrégation si le système sous-jacent est muni des types de relation *section*, *portion*, *division*, *pièce*, *membre*, *segment*, *fragment* [CH88, ILE88]. Quand il devient difficile d'exprimer un sens particulier, il est possible d'introduire un nouveau type de relation. Les systèmes munis d'un grand nombre de types

de relations sont ainsi capables d'exprimer des différences de sens. Autrement dit, ils sont capables de présenter un langage de représentation très précis et assez discriminatoire.

Certains systèmes offrent un grand nombre de types de relations [MNE92], alors que d'autres ne présentent qu'un ensemble réduit de types de relations, c'est le cas par exemple du système RIME. Dans les deux cas, il est difficile de saisir certaines similarités entre ces types de relations. En effet, souvent leur sens se chevauchent, comme c'est le cas des deux types de relations *courir* et *marcher*. Ces deux types sont peut être indépendants sémantiquement, mais ils ne sont certainement pas orthogonaux.

De plus, il est souvent difficile de distinguer les types de concepts des types de relations. En intelligence artificielle, nous sommes généralement ramenés à introduire une distinction artificielle entre ces deux notions. Soit par exemple, la phrase suivante:

Le ciel est bleu. John aime sa couleur.

Dans la première partie de la phrase, nous sommes tentés d'utiliser la relation →(COULEUR)→, alors que dans sa seconde partie, l'utilisation du concept [COULEUR] paraît plus adéquate. Dans les bases de données, il est généralement admis que les relations ne peuvent pas avoir d'attributs. Dans le cas contraire, elles sont représentées comme un concept. Ainsi, dans l'exemple ci-dessus, la "couleur" sera représentée comme un concept dans les deux parties de la phrase. Cela semble également être le choix adopté pour la définition des graphes de connaissances dans [vdB93]. Bien que ce choix permette de distinguer entre les concepts et les relations, il n'en demeure pas moins qu'il est sémantiquement injustifié. Au niveau des graphes conceptuels, la décision de considérer une entité comme un concept ou comme une relation est arbitraire, revenant ainsi au concepteur.

Sowa autorise l'utilisation d'un grand nombre de types de relations. Ainsi, des types de relations primitifs tels que →(*attr*)→ (pour 'attribut') et →(*nom*)→ (pour 'a-pour-nom'), ou des types de relations plus complexes comme →(*agt*)→ (pour 'agent') et →(*enft*)→(pour 'enfant') peuvent être différemment utilisés dans le formalisme. Par contre

Nous avons montré dans le chapitre précédent que nous avons besoin d'un ordre de spécialisation sur les types de relations. Cet ordre nous permettra d'étendre la projection par des décisions de pertinence basées sur les propriétés sémantiques des relations. De plus, nous avons également mentionné dans les chapitre 2 et 3 qu'en recherche d'informations, il est parfois utile de pouvoir établir cet ordre sur des relations d'arités différentes.

Cependant, dans [Sow84], il n'est nullement indiqué comment la hiérarchie des types de relations est définie, ni comment il convient de la construire. D'autant plus que comme nous l'avons souligné dans la section 4.1.3, le sens d'une relation de type *Entre* $\leq$ *Lien* n'est pas spécifié par le formalisme de base, même si dans le livre il a été toujours indiqué que la relation *Lien* peut être restreinte à tous les autres types de relations. Ainsi dans [Gui96a, CM95], l'ensemble des types de relations $\mathcal{T}_r$ est divisé en des ensembles partiellement ordonnés de types de relations de même arité; $\mathcal{T}_r = \mathcal{T}_{r_1} \cup \mathcal{T}_{r_2} \ldots \mathcal{T}_{r_n}$, où $\mathcal{T}_{r_i}$ est l'ensemble des types de relations de même arité i, $i > 0$.

Nous préférons adopter une approche plus adéquate à la recherche d'informations. Nous aurons alors une seul treillis de types de relations, quelque soit l'arité des relations. Chaque type de relation r étant munie d'une signature (i.e., nom, arité, ordre des arguments):

Définition 4.3.3 (Signature) *Chaque type de relation r de $\mathcal{T}_r$ est munie d'une et une seule signature spécifiant son ordre, et les types de concepts maximaux qu'elle peut relier dans un graphe*

conceptuel. Ces types de concepts sont donnés selon un ordre spécifiant la sémantique de cette relation et la façon dont elle peut être lue. On note par $\Sigma_R = (R, n, C_1, \ldots, C_n)$*, la signature du type de relation* R*, d'arité* n *et pouvant relier des individus de type* $C_1, \ldots, C_n$*.*

Exemple 32 *La signature de la relation de localité "*Loc*" liant un objet quelconque à une place donnée est la suivante:* $\Sigma_{Loc} = (Loc, 2, \top, \text{PLACE})$

L'ordre des types de concepts dans la signature est important, il est régie par la fonction partielle *arg* suivante:

Définition 4.3.4 (Arguments de Type de Relation) *La fonction partielle arg:* $\mathcal{T}_r \times \mathbb{N} \rightarrow \mathcal{T}_c$ *retourne pour un type de relation* r *donné et un entier naturel* i*, le* i^{eme} *argument de la signature de* r*:*

$$arg(r, i) = a \text{ ssi } \Sigma_r = (r, n, C_1, \ldots C_i, \ldots, C_n) \wedge a = C_i$$

Pour simplifier l'écriture on notera par $\Sigma_i(r)$ *l'argument* $arg(r, i)$*. La fonction args:* $\mathcal{T}_r \rightarrow 2^{\mathcal{T}_c}$ *assigne à chaque relation ses arguments:*

$$args(r) = \bigcup_{i=1}^{n} arg(r, i)$$

où $2^{\mathcal{T}_c}$ *désigne l'ensemble des parties de* $\mathcal{T}_c$*.*

La définition de notre relation d'ordre partiel $\leq_\sigma$ peut ainsi être énoncée:

Définition 4.3.5 (La Relation D'ordre Partiel $\leq_\sigma$) *Soit* R_1 *et* R_2 *deux relations avec* $\Sigma_{R1} = (R_1, n, C_1, \ldots, C_n)$ *et* $\Sigma_{R2} = (R_2, m, C'_1, \ldots, C'_m)$*, et* $m \geq n$*. Soit* σ *une fonction injective de l'ensemble des indices* $[1 \ldots n]$ *vers celui de* $[1 \ldots m]$*. La relation* R_2 *est une spécialisation de* R_1 *si, et seulement si,* $\Sigma_{\sigma(i)}(R_2) \leq \Sigma_i(R_1)$ *pour tout* $i = 1, .., n$*, où* $\Sigma_i(r)$ *dénote le* i^{eme} *type de concept* C_i *dans la signature associée au type de relation* r *dans le canon:*

$$R_2 \leq_\sigma R_1 \quad ssi \quad \Sigma_{\sigma(i)}(R_2) \leq \Sigma_i(R_1) \text{ pour tout } i \in [1 \ldots n], \text{ et où } j = \sigma(i)$$

En d'autres termes, une relation R_2 est une spécialisation de R_1 si, et seulement si, chaque j^{eme} argument de Σ_{R_2} est inférieur ou égal dans le treillis de types de concepts $\mathcal{T}_c$ au i^{eme} argument de Σ_{R_1}, où $j = \sigma(i)$.

Exemple 33 *Soient* $\Sigma_{entre} = (\text{entre}, 3, \text{OBJET}, \text{OBJET}, \text{OBJET})$ *et* $\Sigma_{a-droite} = (\text{a-droite}, 2, \text{OBJET}, \text{OBJET})$*. Nous avons alors,* entre $\leq_\sigma$ a-droite.

À partir de l'algèbre associée au treillis des types de concepts, nous pouvons déduire une algèbre pour le treillis des types de relations $\langle \mathcal{T}_r, \leq_\sigma, \wedge, \vee \rangle$, où les opérateurs binaires $\wedge$ et $\vee$ vérifient les propriétés de la section précédente.

Définition 4.3.6 (Une R-Algèbre pour $\mathcal{T}_r$) *Une algèbre associée à un treillis de types de relation* $\langle \mathcal{T}_r, \leq_\sigma, \wedge, \vee \rangle$ *est une structure*

$A = \langle \mathcal{D}^A, (\mathcal{T}_r^A)_{r \in \mathcal{T}_r} \rangle$

où

- $\mathcal{D}^A$ *est un ensemble non vide, appelé le* domaine *de* A *(ou univers de discours)*
- *pour tout type de relation* r *dans* $\mathcal{T}_r$ *avec* $\Sigma_r = (\mathrm{r}, n, C_1, C_2, \ldots, C_n)$, r^A *est un sous-ensemble du produit cartésien* $(\mathcal{D}^A)^n$, *en particulier :*
  1. r^A *est un sous-ensemble du produit cartésien* $C_1^A \times C_2^A \times \ldots C_n^A$
  2. $\top^A = (\mathcal{D}^A)^{Min}$,*où* Min *est l'arité minimale que peut avoir une relation dans* $\mathcal{T}_r$, $Min \geq 2$.
  3. $\bot^A = \emptyset$
- *l'opération binaire* $\wedge$ *déterminant la borne inférieure de deux types de relations donnés* r_1 *et* r_2 *dans* $\mathcal{T}_r$ *est interprétée comme un* produit algébrique*:* $(r_1 \wedge r_2)^A = r_1^A \star r_2^A$
- *l'opération binaire* $\vee$ *déterminant la borne supérieure de deux types de relation donnés* r_1 *et* r_2 *est interprétée comme une* somme algébrique*:* $(r_1 \vee r_2)^A = r_1^A + r_2^A$

L'ensemble r^A associé à un type de relation $r \in \mathcal{T}_r$ par la R-Algèbre est appelé la *dénotation* de r. Les treillis des types de concepts et de relations sont strictement distincts, c'est-à-dire qu'ils n'ont de commun que le type Universel $\top$, et le type Absurde $\bot$.

Dans la suite, nous confondons type de relation et relation, chaque fois qu'il est facile de les distinguer dans le contexte du discours. Ces deux notions sont au fait très liées. Ainsi par propriétés de relations, nous désignons implicitement les propriétés des types de relations et vice-versa.

4.3.3 L'ensemble $\mathcal{M}$ des Marqueurs

Les types de concepts correspondent à des noms communs. Ils constituent des termes génériques spécifiant certains attributs des entités désignées comme par exemple, CHAT, HOMME, et COMMANDANT. À l'opposé, les marqueurs correspondent à des étiquettes particulières servant à nommer et à distinguer les individus vérifiant les attributs des entités désignées par les types de concepts. Par exemple, "Minou", "John" et "Cousteau" permettent de nommer respectivement un individu appartenant à CHAT, à HOMME, et à COMMANDANT.

Il existe un ensemble $\zeta = \{\#\text{John}, \#\text{Cousteau}, \#\text{Jaques}, \ldots\}$, dont les éléments sont appelés *marqueurs individuels* ou *référents*. Chaque élément désigne le nom d'un individu connu par le système comme vérifiant les attributs d'une des entités correspondant à un type de concept dans $\mathcal{T}_c$. Ce nom doit être unique dans le contexte de l'application.

Un marqueur #, lorsqu'il n'est pas suivi d'une chaîne de caractères, est utilisé dans le formalisme pour désigner un individu particulier. Cet individu est distingué dans le contexte de l'application comme étant un élément de la dénotation du type de concept associé, mais reste non identifié. La marqueur #, appelé *marqueur existentiel*, incarne la sémantique du quantificateur existentiel. En effet, il dénote un individu dont l'existence est assertée mais dont l'identité n'est pas connue. Nous préférons utiliser le quantificateur $\exists$ à la place de # pour une plus grande cohérence avec la sémantique véhiculée.

L'ensemble des *témoins* W est fortement lié à la sémantique du marqueur existentiel. Ainsi, quand un concept de la forme $[C : \exists]$ apparaît dans un graphe conceptuel, l'individu non identifié correspondant au marqueur $\exists$ est désigné par une constante abstraite w, appelée témoin, introduite pour asserter l'existence de cet individu en lui assignant un nom abstrait sous lequel il sera connu dans l'application. Ce nom abstrait devra alors appartenir à la dénotation du type de concept C. Les témoins doivent être uniques dans le contexte de l'application. L'ensemble W est disjoint de l'ensemble ζ.

Nous désignerons par $\mathcal{C}$ l'ensemble de tous les concepts $[c : m]$ tel que c est un type de concept et m un marqueur quantifié, et par $\mathcal{C}_\exists$ l'ensemble des concepts de $\mathcal{C}$ existentiellement marqués $(m = \#)$.

Définition 4.3.7 *Un fonction bijective $\mathcal{W}$: $\mathcal{C}_\exists \rightarrow W$ produit pour chaque marqueur existentiel d'un concept $[c : \#] \in \mathcal{C}_\exists$ le témoin unique w qui lui est associé, avec $w \in c^{\mathcal{A}}$*

Le marqueur générique $\star$ associé à un type de concept t dénote tous les éléments de $t^{\mathcal{A}}$. Il joue ainsi le rôle d'un quantificateur universel. À l'opposé, le marqueur absurde $\tilde{\star}$ ne peut être le représentant d'aucun élément du domaine de discours, il représente la notion de non-existence.

L'ensemble des marqueurs individuels ζ muni du marqueur existentiel $\exists$, et des deux marqueurs spéciaux "*" (marqueur générique) et "$\tilde{\star}$"(marqueur absurde) forme une structure de treillis:

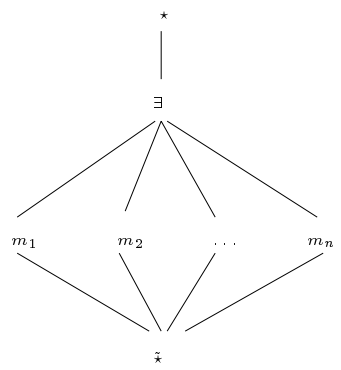


Figure 4.8. Treillis des Marqueurs

Une fonction particulière référent: $\mathcal{C} \rightarrow \mathcal{M}$ produit pour chaque concept $c \in \mathcal{C}$ le marqueur qui lui est associé:

$$\text{référent}(c) = m \text{ ssi } c = [t_c : m]$$

4.3.4 La Relation de Conformité *Conf*

La relation de conformité *Conf*, notée "::", définit des contraintes d'association entre un type de concept et un marqueur. Si $t_c \in \mathcal{T}_c$ et $m_i \in \mathcal{M}$ avec $t_c :: m_i$ est vrai alors m_i est dit conforme au type t_c. Intuitivement, autoriser une association entre un type de concept t_c et un marqueur m revient à attester que l'individu repéré par m est un t_c, ou que m est une instance de type t_c.

Hypothèse 6 (La Relation de Conformité) *La relation de conformité $Conf$ entre les types de concepts et les marqueurs est définie comme suit:*

$$\forall t_c \in \mathcal{T}_c, \quad t_c :: m \quad ssi \quad m = \star \; ou \; m = \# \; ou \; m \in t_c^{\mathcal{A}}$$

$t_c^{\mathcal{A}}$ désigne la dénotation du type de concept t_c.

En particulier, pour chaque élément $d \in \zeta$ ou $w \in W$ et pour chaque $t_c \in \mathcal{T}_c$ la relation de conformité indique si les prédicats $t_c(d)$ et $t_c(w)$ peuvent être interprétés comme vrais.

Théorème 13 $\forall t_1, t_2 \in \mathcal{T}_c \quad et \quad \forall m \in \mathcal{M}$, *si* $t_1 :: m$ *et* $t_1 \leq t_2$ *alors* $t_2 :: m$*;*

Preuve 9 *Si m est un marqueur générique ou existentiel, le théorème est facilement établie. Autrement, si $t_1 :: m$ alors $m \in t_1^{\mathcal{A}}$ d'après l'hypothèse 6. Comme $t_1 \leq t_2$, d'après les propriétés des hiérarchies des types $t_1^{\mathcal{A}} \subseteq t_2^{\mathcal{A}}$. Nous avons alors $m \in t_2^{\mathcal{A}}$, et $t_2 :: m$*

Ainsi si le marqueur "#Minou" est conforme au type de concept CHAT alors "#Minou" est aussi conforme à tous les super-types de CHAT dans le treillis des types de concepts, par exemple "#Minou" est conforme à ANIMAL, du fait de la relation *sorte de* qui existe entre CHAT et ANIMAL.

Corollaire 1 *Il est facile de vérifier que les propriétés suivantes sont valides:*

(1) $\forall m \in \mathcal{M}$, $\top :: m$ *et* $\neg(\bot :: m)$*, autrement dit tout marqueur est conforme au type Universel mais aucun ne peut être associé avec le type absurde;*

(2) $\forall t_c \in \mathcal{T}_c$, $\neg t_c :: \star$ *et* $\neg(t_c :: \tilde{\star})$*, c'est-à-dire que le marqueur générique est conforme à tous les types de concepts, alors que le marqueur absurde n'est conforme à aucun type de concept;*

(3) $\forall t_1, t_2 \in \mathcal{T}_c$ *et* $m \in \mathcal{M}$*, si* $t_1 :: m$ *et* $t_2 :: m$ *alors* $(t_1 \wedge t_2) :: m$ *et* $t_1 \wedge t_2 > \bot$

Preuve 10 *La première propriété découle directement du fait que la dénotation du type universel $\top$ contient tous les éléments du domaine de discours, en particuliers les éléments dénotés par l'ensemble des marqueurs $\mathcal{M}$. La dénotation du type absurde $\bot$ est par contre vide. Elle ne contient donc aucune instance. La deuxième propriété est une conséquence directe de l'hypothèse 6 pour ce qui est du marqueur générique. Pour le marqueur absurde, étant donné qu'il modélise la notion de la non-existence, il est clair qu'il n'appartient à aucune dénotation de type de concept. La troisième propriété est une conséquence directe de la propriété de l'opérateur algébrique $\wedge$: si $t_1 :: m$ et $t_2 :: m$ alors $m \in t_1^{\mathcal{A}}$ et $m \in t_e^{\mathcal{A}}$. Le marqueur m appartient donc à l'intersection $t_1^{\mathcal{A}} \cap t_2^{\mathcal{A}}$. Par conséquent $m \in (t_1 \wedge t_2)^{\mathcal{A}}$ et donc $t_1 \wedge t_2 :: m$.*

Nous utiliserons une relation binaire $=$ contenue dans $\zeta \times W$ entre les éléments de ζ et W. Cette relation sera utilisée lors de l'introduction de la nouvelle sémantique des graphes, et est définie comme suit: chaque fois qu'un marqueur individuel a est conforme à un type de concept donné, l'interprétation $a = w$ est ajoutée à la relation $=$, avec w une nouvelle constante dans W.

4.3.5 L'ensemble $\mathcal{S}_r$ des Graphes Signatures

Nous avons vu que chaque type de relation $r \in \mathcal{T}_r$ a exactement une et une seule signature Σ_r (Définition 4.3.3. Si un type de relation pourrait admettre deux signatures différentes ou plus, alors ce type de relation a deux sens ou plus qu'il faudra spécifier par d'autres types de relations.

Définition 4.3.8 *Soit S_Σ l'ensemble de toutes les signatures. La fonction bijective $associe_{sign}$: $\mathcal{T}_r \rightarrow S_\Sigma$, retourne pour chaque type de relation $r \in \mathcal{T}_r$, sa seule et unique signature.*

Chaque signature Σ_r est représentée par un graphe conceptuel g_{Σ_r}, dit *graphe signature.*

Définition 4.3.9 (Graphe Signature) *Étant donné S_Σ, l'ensemble de toutes les signatures de (types de) relation, la fonction bijective $construit_{sign}$: $S_\Sigma \rightarrow \mathcal{G}$, produit pour chaque signature de type de relation $\Sigma_r = (r, n, C_1, \ldots, C_n)$ un graphe conceptuel biparti connexe et orienté dénoté par g_{Σ_r} construit de la manière suivante:*

- *un noeud relation est associé au type de relation r.*
- *un noeud de concept générique $[C_i : \star]$ est associé à chacun des types de concepts C_i de la signature.*
- *La relation de type r admet e arcs entrants et s arcs sortants, avec $e + s = n$. Le i^{eme} arc lie le i^{eme} concept à la relation r. L'ordre des types de concepts dans la signature est respecté. Ainsi, le i^{eme} concept correspond au i^{eme} argument du type de relation r renvoyé par la fonction arg définie dans la section 4.3.2.*

Nous désignons par $\mathcal{S}_r$ l'ensemble des graphes signatures générés par la fonction $construit_{sign}$.

Tous les concepts des graphes signatures vérifient la relation de conformité $Conf$ (étant donné qu'ils sont tous des concepts génériques). De plus, les graphes signatures doivent présenter les propriétés suivantes.

Propriétés 1

(1) *Si la relation de type r est binaire ($n = 2$), alors le noeud relation du graphe signature de r admet exactement un arc entrant et un un arc sortant ($e = s = 1$). Par convention, on considère que le premier argument de la signature correspond à l'arc entrant;*

(2) *Si la relation de type t_r est n-aire, alors le sommet relation du graphe signature de t_r admet au moins un arc entrant et au moins un arc sortant;*

(3) *$\forall r_1, r_2 \in \mathcal{T}_r$, tel que $\Sigma_{r_1} = (r_1, n, C_1, \ldots, C_n)$ et $\Sigma_{r_2} = (r_2, m, C_1, \ldots, C_m)$ si $r_2 \leq_\sigma r_2$, alors le i^{eme} arc du graphe signature de r_1 et le $\sigma(i)^{eme}$ arc du graphe signature de r_2 ont la même direction (ils sont soit tous les deux entrants, soit tous les deux sortants).*

Exemple 34 *En utilisant la propriété (1), le graphe signature associé à $\Sigma_{Loc} = (Loc, 2, \top, \text{PLACE})$ est le suivant:*

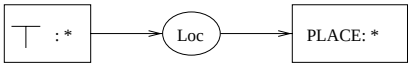


Figure 4.9. Un graphe signature pour la relation de localité *Loc*

Si l'on admet comme dans [Sow84] le fait que toutes les relations n-aires peuvent être exprimées en un ensemble de relations primitives binaires, il est facile d'établir dans ce cas la direction des arcs (entrant ou sortant) dans les graphes signatures qui leur sont associés. En effet, puisque ces relations sont construites à partir d'un ensemble de relations binaires, chacune est une spécialisation des relations binaires contenues dans ce ensemble. En appliquant la propriété (3) ci-dessus citée, la direction des arcs est automatiquement déduite. Pour illustrer ce fait, nous supposons dans l'exemple suivant que la relation tertiaire «*entre*» est construite à partir des deux relations primitives binaires «*a-droite*» et «*a-gauche*».

Exemple 35 *Supposons que* $\Sigma_{a-droite} = (a \Leftrightarrow droite, 2, A, B)$, $\Sigma_{a-gauche} = (a \Leftrightarrow gauche, 2, A, C)$ *et finalement que* $\Sigma_{entre} = (entre, 3, A, B, C)$. *Dans ce cas on a* entre $\leq_\sigma$ a-droite *et* entre $\leq_\sigma$ a-gauche. *Les graphes signatures des deux relations* a-droite *et* a-gauche *sont donnés dans la figure 4.10. Il s'en suit le graphe signature de la relation* entre.

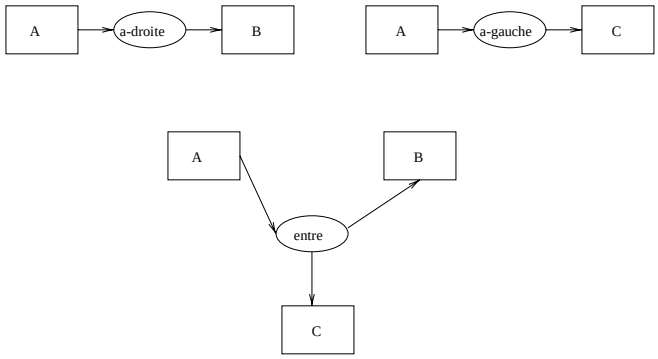


Figure 4.10. Le sens des arcs de la relation "*entre*" est conforme à la propriété (3)

Nous avons aussi le théorème suivant:

Théorème 14 (L'ensemble $\mathcal{S}_r$) *L'ensemble* $\mathcal{S}_r$ *des* graphes signatures g_{Σ_r} *vérifie les deux propriétés suivantes:*

1. *Chaque graphe de* $\mathcal{S}_r$ *a exactement une relation.*
2. *Chaque type de relation apparaît une et une seule fois dans* $\mathcal{S}_r$

Preuve 11 *Ce théorème découle directement des propriétés des fonctions* $associe_{sign}$ *et* $construit_{sign}$. *En effet,* $g_{\Sigma_r} = construit_{sign}(associe_{sign}(r))$, *d'où* $g_{\Sigma_r} = construit_{sign} \circ associe_{sign}(r)$. *Les fonctions* $associe_{sign}$ *et* $construit_{sign}$ *étant bijectives, leur composé est bijective. Les deux propriétés du théorème découlent alors directement de ce fait.*

4.3.6 L'ensemble des Connaissances $\mathcal{K}_s$

Les types de relations que nous avons utilisé jusqu'à présent sont tous de premier ordre, c'est-à-dire qu'ils lient les types de concepts primitifs de $\mathcal{T}_c$. Tel est le cas par exemple des relations →(Loc)→ ou →(agt)→. Afin de permettre un raisonnement sur les relations comme le stipule le chapitre 3, nous avons besoin d'introduire des relations d'ordre supérieur. Ainsi, tout comme pour les fonctions d'ordre supérieur, une relation d'ordre $i+1$ est une relation qui a au moins un argument constituant un type de concept d'ordre i.

Les relations d'ordre supérieur ne font pas partie des graphes signatures $\mathcal{S}_r$. Elles lient des types de concepts particuliers d'ordre 1. Ces derniers sont organisés en un treillis de types de concepts spécial, appelé $\mathcal{T}_{Cspe}$ (voir figure 4.11).

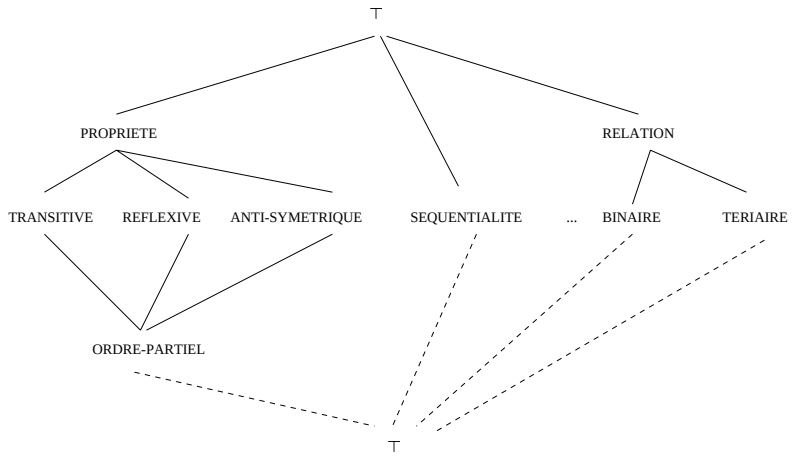


Figure 4.11. Le treillis des types de concepts $\mathcal{T}_{Cspe}$

Ces relations permettent de représenter les propriétés des (types de) relations dans le canon de l'application. Ainsi, nous avons vu dans le chapitre 3 qu'il existe essentiellement trois types de raisonnements sur les relations: un raisonnement sur leurs propriétés mathématiques (transitivité, symétrie, ...), un raisonnement sur leurs agencements possibles (juxtaposition, composition, ...) et enfin un raisonnement sur leurs liens sémantiques (inversion, exclusion, ...). Pour chacun de ces raisonnements, nous introduisons des graphes conceptuels représentant les connaissances requises pour sa mise en place. Ces graphes sont énumérés dans l'ensemble $\mathcal{K}_s$ du canon. Trois types de graphes sont ainsi identifiés. À cet effet, un treillis de types de relations d'ordre deux est introduit, nous le désignons par $\mathcal{T}_{Rspe}$ (voir figure 4.12).

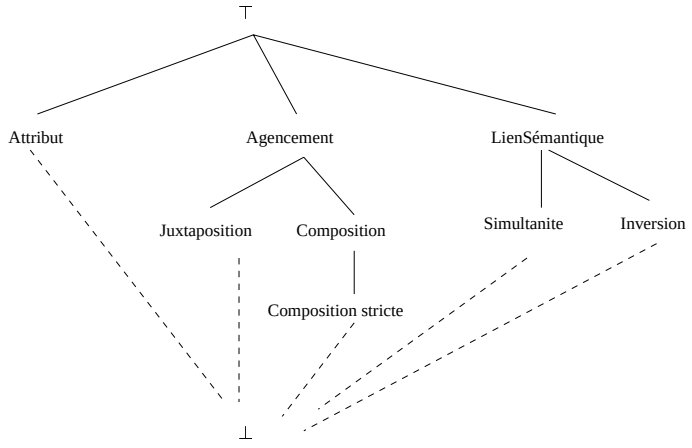


Figure 4.12. Le treillis des types de Relations $\mathcal{T}_{Rspe}$

Propriétés Mathématiques

Les propriétés mathématiques des relations sont représentées dans $\mathcal{K}_s$ par des graphes de la forme suivante:

$$[RELATION] \rightarrow (Attribut) \rightarrow [PROPRIETE]$$

Ces graphes contiennent une relation d'ordre deux $\rightarrow$(Attribut)$\rightarrow$, attachée à un type de concept particulier, appelé PROPRIETE. Ce dernier véhiculera la caractéristique mathématique d'une relation donnée. Par exemple, pour spécifier que la relation d'agrégation "*Compose*" est transitive, nous ajoutons le graphe suivant à $\mathcal{K}_s$:

$$[RELATION : Compose] \rightarrow (Attribut) \rightarrow [TRANSITIVE]$$

Agencements entre Relations

Les agencements constituent une manière possible de combiner les relations et d'obtenir ainsi des informations implicitement contenues dans les graphes. Ils sont introduits dans $\mathcal{K}_s$ par des graphes conceptuels de la forme suivante:

$$\begin{array}{c} [RELATION] \rightarrow (Agencement) \leftarrow [RELATION] \\ \downarrow \\ [RELATION] \end{array}$$

Le type de l'agencement est spécifié par la relation $\rightarrow$(Agencement)$\rightarrow$ ou par un de ses spécifiques (Juxtaposition, Composition, ...). Les relations en entrée sont spécifiées grâce aux deux arcs entrants du graphe. La nouvelle relation dérivée est indiquée, quant à elle, par l'arc sortant.

Par exemple, s'il existe un agencement de composition stricte entre les relations "*Compose*" et "*ContenuDans*", le graphe suivant est ajoutée à $\mathcal{K}_s$:

$$[RELATION : Compose] \rightarrow (CompositionStricte) \leftarrow [RELATION : ContenuDans]$$
$$\downarrow$$
$$[RELATION : ContenuDans]$$

Liens Sémantiques entre Relations

Les liens sémantiques possibles entre les relations sont ajoutées à $\mathcal{K}_s$ par des graphes de la forme suivante:

$$[RELATION] \rightarrow (LienSémantique) \rightarrow [RELATION]$$

La nature du lien est spécifiée par la relation $\rightarrow$(LienSémantique)$\rightarrow$, ou par un de ses spécifiques dans $\mathcal{T}_{Rspe}$. Par exemple, la relation *a-droite* est liée à la relation *a-gauche* par un lien d'inversion:

$$[RELATION : a \Leftrightarrow droite] \rightarrow (Inversion) \rightarrow [RELATION : a \Leftrightarrow gauche]$$

Cohérence entre les Ensembles $\mathcal{K}_s$ et S_Σ

Tout comme la définition de la relation d'ordre partiel $\leq_\sigma$ sur les relations, la définition des propriétés des relations doit être en conformité avec les signatures des relations en question. En effet, ces propriétés engendrent des interprétations sémantiques dont la cohérence avec les signatures doit être vérifiée. Ainsi, asserter qu'une relation binaire r est symétrique sous-tend le fait qu'on peut permuter les tuples de cette relation sans altérer la relation de conformité $Conf$. Autrement dit, les types de concepts maximaux de la signature $\Sigma_r = (r, 2, C_1, C_2)$ de r doivent être les mêmes au risque de violer la relation de conformité: si r est symétrique, l'énoncé $\forall x, y$ r(x, y) $\rightarrow$ r(y, x) est valide. Il s'en suit que pour tout couple (x, y) appartenant à la dénotation de r, si $x \in C_1^{\mathcal{A}}$ (resp. $y \in C_2^{\mathcal{A}}$) alors nécessairement $x \in C_2^{\mathcal{A}}$ (resp. $y \in C_1^{\mathcal{A}}$). Considérer que $C_1 = C_2$ est alors la seule façon possible de garder la relation de conformité.

La contrainte qui suit doit être vérifiée entre les éléments de $\mathcal{K}_s$ et les signatures de S_Σ:

Définition 4.3.10 (Cohérence des éléments de $\mathcal{K}_s$) *Selon le type de la relation d'endiguement introduite par le graphe conceptuel de $\mathcal{K}_s$, une des trois conditions suivantes doit être vérifiée:*

- *S'il s'agit d'une propriété mathématique d'une relation r, la signature $\Sigma_r = (r, n, C_1, C_2, \ldots, C_n)$ devra vérifier la condition $\Sigma_i(r) = \Sigma_j(r) = \ldots = \Sigma_k(r)$ dès lors que la propriété porte sur le $i, j, \ldots, k^{eme}$ arguments de la signature en question.*

- *Si le graphe conceptuel introduit a trait à un agencement entre deux relations r_1 et r_2 avec $\Sigma_{r_1} = (r_1, n, C_1, C_2, \ldots, C_n)$ et $\Sigma_{r_2} = (r_2, n, D_1, D_2, \ldots, D_n)$ alors la condition $\Sigma_i(r_1) = \Sigma_j(r_2)$ doit être vérifiée si l'agencement porte sur n'importe quel paire (i, j) se référant au i^{eme} argument de r_1 et le j^{eme} argument de r_2, respectivement.*

- *Exceptée la relation de spécialisation matérialisée par $\leq_\sigma$ et ayant ses propres contraintes (voir section 4.3.2), un lien sémantique entre deux relations r_1 et r_2, avec $\Sigma_{r_1} = (r_1, n, C_1, C_2, \ldots, C_n)$ et $\Sigma_{r_2} = (r_2, n, D_1, D_2, \ldots, D_n)$ implique que $\Sigma_i(r_1) = \Sigma_j(r_2)$ dès que la propriété touche au i^{eme} argument de r_1 et au j^{eme} argument de r_2, respectivement.*

Ainsi, si nous reprenons l'exemple des deux relations *a-droite* et *a-gauche* liées par une inversion, l'ensemble des signatures S_Σ comptera les deux signatures $\Sigma_{a-droite} = (a\text{-}droite, n, O_1, O_2)$ et $\Sigma_{a-gauche} = (a\text{-}droite, n, O_3, O_4)$, tels que $O_1 = O_4$ et $O_2 = O_3$. Lors de l'implantation de notre prototype RELIEF (voir chapitre 5), nous avons respecté ces contraintes en contrôlant la saisie des propriétés des relations par rapport aux signatures.

4.3.7 L'ensemble d'Endiguement $\mathcal{E}$

Spécifier qu'une relation est transitive ou que deux relations peuvent être agencées pour donner lieu à une autre relation ne signifie rien en soi si nous ne précisons pas ce que cela veut dire d'être une relation transitive ou d'être composée à une autre relation. Il faut spécifier par exemple que pour une relation binaire transitive R, le deuxième argument de R(x, **y**) doit être le même que le premier argument de R(**y**, z). Le résultat étant une autre apparition de la relation R avec comme arguments x et z. Les diverses relations d'endiguement que nous avons présenté dans le chapitre 3 doivent donc être spécifiées au niveau du canon. Ces relation d'endiguement expriment les conditions que doivent satisfaire les relations pour posséder telle ou telle autre propriété. Pour représenter ces relations, nous utilisons l'adaptation de Sowa des graphes existentiels de Peirce [Pei58, Rob73]:

Notation graphique	*Notation standard*
$\boxed{p\ q\ r}$	$(p \wedge q \wedge r)$
$\neg\ \boxed{p\ q\ r}$	$\sim(p \wedge q \wedge r)$
$\neg\boxed{\neg\ \boxed{p\ q\ r}}$	$\sim\sim(p \wedge q \wedge r)$
$\neg\boxed{\neg\ \boxed{p}\ \neg\ \boxed{q}\ \neg\ \boxed{r}}$	$(p \vee q \vee r)$
$\neg\boxed{p\ q\ \neg\ \boxed{r\ s}}$	$(p \wedge q) \supset (r \wedge s)$
$\neg\boxed{p\ \neg\ \boxed{q}}$	$p \supset q$

Figure 4.13. Les notions de Contexte positif et de contexte négatif

Peirce propose une représentation graphique de la logique où toutes les formules logiques peuvent

être représentées par des contextes négatifs imbriqués[13] (voir figure 4.13). Cela est dû au fait que tous les opérateurs Booléens peuvent être réduits à des conjonction et des négations. Par exemple, la disjonction $p \vee q$ peut être représentée comme $\sim (\sim p \wedge \sim q)$. La conjonction de deux termes est représentée par un contexte (graphiquement une boîte) où ces deux derniers sont mis côte à côte. Peirce représente la négation par le symbole $\neg$ placé avant un contexte, donnant ainsi lieu à un *contexte négatif*. Dans la figure précédente, les trois premières lignes du tableau montrent les termes p, q et r assertés d'abord dans un contexte positif, puis dans un contexte négatif, puis dans une double négation où un contexte négatif est imbriqué dans un autre contexte négatif. Les deux dernières lignes du tableau montrent la notation qu'adopte Peirce pour la représentation de l'implication logique. Cette dernière provient du fait que $p \supset q \Leftrightarrow \sim p \vee q \Leftrightarrow \sim (p \wedge \sim q)$, ce qui explique le graphe proposé.

À premier abord cette notation paraît difficile à appréhender. Cependant, Sowa précise qu'avec un peu de pratique, l'utilisateur peut facilement "voir" les différentes combinaisons des boîtes comme des disjonctions et des implications[14].

Cette notion de contexte fut alors reprise par Sowa en permettant aux concepts d'avoir des graphes comme référents. De tels concepts sont appelés des *contextes* et ont la forme générale suivante: [TYPE: $\{u_1, u_2, \ldots, u_n\}$], où les u_i sont des graphes conceptuels, pouvant éventuellement être eux-mêmes des contextes. Graphiquement, les contextes sont représentés par des boîtes comme dans les graphes existentiels de Peirce. Nous disons que les graphes u_i sont "assertés" par le contexte. Bien qu'à priori le type d'un contexte puisse être quelconque, c'est généralement le type PROPOSITION qui est le plus souvent utilisé dans le livre de Sowa ainsi que dans la littérature sur les graphes conceptuels en général. Ce type sert à affirmer une proposition ou à décrire une situation.

La relation unaire $(NEG)_{\rightarrow}$ appliquée à un concept de type PROPOSITION nie la conjonction des graphes conceptuels assertés dans son contexte. Le symbole $\neg$ utilisé dans le tableau précédent est une abréviation informelle de la relation $(NEG)_{\rightarrow}$. Comme les situations complexes peuvent avoir beaucoup de contextes imbriqués, il y aura plusieurs répétitions du label PROPOSITION dans les contextes. Sowa propose la notation $\neg[u\ v]$ comme abréviation de $(NEG)_{\rightarrow}$[PROPOSITION:$\{u, v\}$]. Nous retrouvons alors la notation de Peirce, sauf qu'ici les termes apparaissant dans le contexte constituent des graphes conceptuels.

Par définition [Sow84] (page 140), un graphe de type $(NEG)_{\rightarrow}$[PROPOSITION:$\{u_i, \ldots, u_n\}$] est dit *contexte négatif* ("negative context"). Un contexte négatif contenant un unique graphe conceptuel qui est lui-même un graphe négatif est appelé *une double négation*. L'hypothèse suivante est ensuite introduite:

Hypothèse 7 *Si p est une proposition assertant les graphes $u_1, \ldots, u_n$, alors la formule logique ϕp associée à cette proposition est la suivante: $(\phi u_1 \wedge \ldots \wedge \phi u_n)$. Si c est un contexte négatif constitué de la relation (*NEG*)$_{\rightarrow}$ liée à une proposition p, alors ϕc est égale à $\sim \phi p$. Tous les concepts génériques apparaissant dans p doivent être assignés d'un symbole de variable distinct par ϕ.*

Il peut être parfois nécessaire dans un contexte d'exprimer le fait que deux concepts représentent le même individu. Pour identifier ces liens anaphoriques entre concepts, les *liens de coréférence*[15]

13. *nested negative contexts.*

14. Dans [Sow84, pp139-140], l'auteur écrit notamment : "Although nested boxes may look odd to logicians with many years of experience with the linear form, Peirce's graphs are very readable: with a little practice, the reader can immediately "see" the various combinations of boxes as disjunctions and implications".

15. En anglais *coreference links.*

sont alors utilisés. Les liens de coréférence sont représentés sous une forme graphique par des traits pointillés. Dans une formule logique, ces liens sont représentés en utilisant le même référent pour les concepts ainsi liés, ceci en explicitant au besoin les variables pour les concepts génériques[16].

Exemple 36 *Le contexte négatif suivant représente le graphe conceptuel "Chaque pays a un drapeau"*

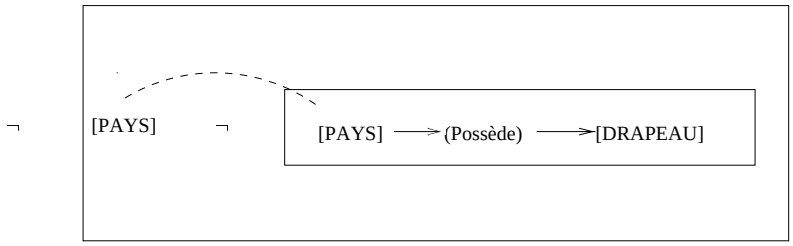


Figure 4.14. Un lien de coréférence sur le concept [PAYS]

Les déductions sur ces graphes conceptuels complexes se basent sur le système de déduction logique imaginé par Peirce pour les graphes existentiels. Il s'agit d'un système constitué d'un seul axiome et de cinq règles d'inférence. L'unique axiome correspond à l'ensemble vide des graphes et les règles d'inférence sont l'effacement (Erasure), l'insertion (Insertion), l'itération (Iteration), la déitération (Deiteration) et la double négation (Double negation). Ces règles sont essentiellement basées sur la profondeur du graphe, c'est-à-dire sur le nombre des contextes négatifs qui doivent être traversés pour arriver au contenu du graphe, ceci en partant du contexte extérieur. Selon ce nombre, le graphe est dit à contexte pair ou impair. Les contextes pairs contiennent des graphes à valeur de vérité positive, tandis que les contextes impairs contiennent, eux, des graphes à valeur de vérité négative. Dans l'exemple de la figure 4.14, le concept [PAYS] est inclus dans un contexte ayant la profondeur 1. Les deux concepts [PAYS] et [DRAPEAU] sont par contre inclus dans un contexte ayant la profondeur 2. On dit que le premier contexte *domine* le second. Si un contexte x apparaît dans un contexte y, alors x est dit dominé par y. Si x domine un autre contexte z, alors z est également dominé par y.

Définition 4.3.11 (Le système de déduction Alpha de Peirce) *Soit une proposition p. Les graphes conceptuels qui sont assertés dans cette proposition peuvent eux-mêmes être des propositions (des contextes). Nous dénotons par S le* contexte le plus en dehors de *p, c'est-à-dire l'ensemble des graphes conceptuels qui ne sont le référent d'aucun concept. Tout graphe dérivé de S par les règles d'inférence suivantes est dit* inférable *de S.*

- *L'Effacement (ou "Erasure"): tout graphe apparaissant dans un contexte pair peut être effacé.*
- *L'Insertion (ou "Insertion") : n'importe quel graphe peut être inséré dans un contexte impair.*
- *L'Itération (ou "Iteration"): la copie de n'importe quel graphe u peut être insérée dans le contexte dans lequel u apparaît ou dans n'importe quel contexte dominé par u.*

16. Pour plus de détails sur ces liens de coréférence, le lecteur pourra consulter l'article de J. Esch [Esc93].

- *La Déitération (ou "Deiteration") : n'importe quel graphe dont l'occurrence peut être le résultat de l'application de la règle d'itération peut être supprimé (i.e, si ce graphe est identique à un autre graphe dans le même contexte ou dans un contexte dominant).*
- *La Double Négation: une double négation peut être dessinée autour d'un graphe ou d'un ensemble de graphes dans n'importe quel contexte. Elle peut être également effacée de n'importe quel graphe ou ensemble de graphes dans un contexte quelconque.*

L'ensemble vide des graphes est le seul axiome logique de ce système de déduction: il est noté soit par { } ou simplement par un espace blanc. Tout graphe inférable de { } par les règles de déduction ci-dessus citées est dit un théorème.

Sowa mentionne que ce système permet la déduction de tous les axiomes des logiques classiques. De plus, la règle de *modus ponens* peut être simulée: étant donnés les énoncés p et $p \supset q$, l'énoncé q est dérivé. En termes de graphes conceptuels, nous commençons par p et $\neg[p\ \neg[q]]$. L'application de la règle de déitération permet d'effacer p: $\neg[\neg[q]$. Comme une double négation peut être effacée de n'importe quel graphe, le graphe q peut être dérivé.

Le système de déduction de Peirce, ci-dessus mentionné, fut alors généralisé par Sowa, afin de prendre en compte les liens de coréférence et les quatre opérateurs de formation de graphes. Les 5 règles précédentes ont été légèrement modifiées. Par exemple la règle d'insertion est devenue la suivante:

- Insertion: dans un contexte impair, n'importe quel graphe conceptuel peut être inséré, un lien de coréférence peut être dessiné entre deux concepts identiques, et une restriction peut être effectuée sur n'importe quel concept.

De plus, deux règles ont été ajoutées, il s'agit de la jointure de coréférent (Coreferent join) et l'individualisation (Individuals). Pour plus de détail sur le système ainsi étendu, le lecteur pourra consulter [Sow84] ([pp154–155]).

Grâce à l'utilisation du *modus ponens*, des règles de déduction de type "**Si** *Condition* **Alors** *Résultat*" peuvent être représentées et exploitées par le formalisme des graphes conceptuels. Nos relations d'endiguements du chapitre 3 peuvent donc être définies et représentées à l'aide de contextes.

Par exemple, la propriété de transitivité peut être représentée de la façon suivante:

Grâce aux liens de coréférence, ce graphe contexte détermine de façon précise ce que l'on entend par une relation binaire transitive. Les concepts des graphes conceptuels apparaissant dans le contexte de cette figure sont tous munis du type de concept universel $\top$. Le type de relation "*Relation*" est un type particulier supposé être égal à tout type de $\mathcal{T}_r$. Ainsi, lors de l'établissement d'une projection, il peut avoir pour image n'importe quel type de relation. Si l'opérateur de projection permet la prise en compte des restrictions sur les types de relations, le type *Relation* se comporterait comme une généralisation de tous les types de $\mathcal{T}_r$ (une sorte de relation universelle). Nous verrons dans la section 4.4.5, qu'il s'agit par exemple de s'assurer qu'une relation décrite comme étant transitive dans l'ensemble des connaissances $\mathcal{K}_s$ verra la prémisse de la règle représentée par le contexte ci-dessus se projeter sur son graphe signature ainsi que une copie de ce dernier.

Plus formellement, l'ensemble d'endiguement $\mathcal{E}$ regroupe des contextes véhiculant des propriétés de relations. Il s'agit des diverses propriétés que nous avons mentionné dans le chapitre précédent.

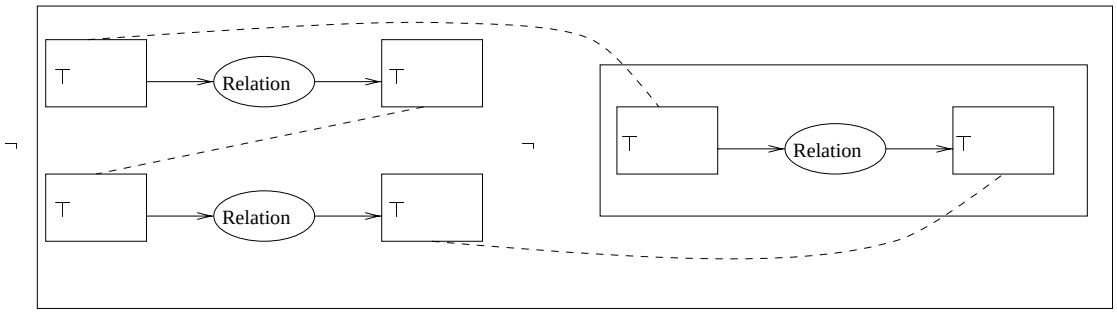


Figure 4.15. Représentation de la propriété de transitivité

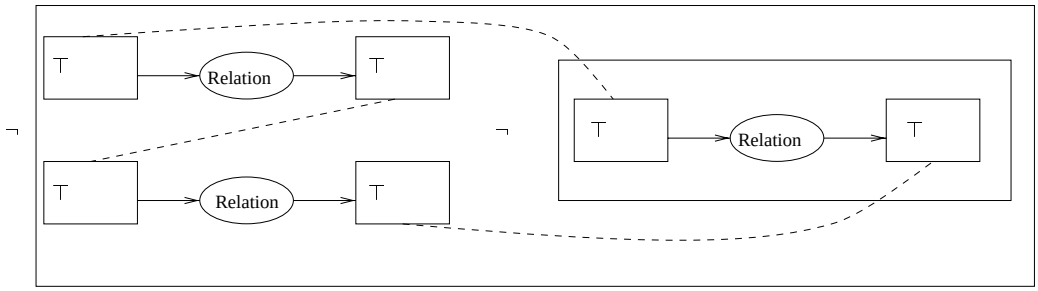


Figure 4.16. Représentation de la propriété de composition

Par exemple, nous avons vu que la composition des relations (un cas particulier de la propriété d'agencement) exprime le fait que si $R_1(x, y) \wedge R_2(x, y)$ alors $R_3(x, y)$. Ceci peut être exprimé par le contexte de la figure 4.16.

Enfin la figure 4.17 qui suit représente la symétrie qui est une instanciation de la propriété de *permutation* aux relations binaires.

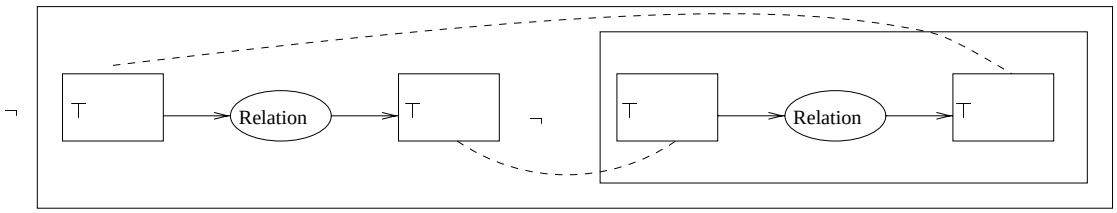


Figure 4.17. Représentation de la propriété de symétrie

Si toutes ces règles sont de la forme "**Si** *Condition* **Alors** *Résultat*", nous devons toutefois être vigilant quant aux relations qui rendent valide la condition. Ainsi, les contextes inclus dans l'ensemble $\mathcal{E}$ doivent être utilisés d'une manière duale avec les graphes de la base de connaissances $\mathcal{K}_s$. C'est-à-dire que seules les relations vérifiant la propriété véhiculée par le contexte se verront

appliquer la règle et déclencheront ainsi le résultat.

Pour illustrer comment se passent réellement les choses, reprenons la propriété de symétrie et le graphe contexte qui lui est associé. Nous avons montré dans la section précédente que les propriétés des relations sont représentées par des graphes conceptuels ajoutés à l'ensemble des connaissances $\mathcal{K}_s$. Pour appliquer la règle de production véhiculée par ce graphe contexte à une relation R, il faudra vérifier que le graphe $[RELATION\colon *] \rightarrow (Attribut) \rightarrow [SYMETRIQUE]$ se projette sur un des graphes assertant les propriétés de R dans $\mathcal{K}_s$. Dans ce cas, la relation R est bien symétrique et le résultat de la règle de production du contexte peut lui être appliqué. Ainsi, les graphes contextes que nous avons présenté dans cette section se voient ajouter un graphe conceptuel mentionnant de quel type de relation il s'agit. Ce graphe joue le rôle d'un identificateur pour le contexte. Il est une spécialisation de l'un des trois graphes définissant le type des propriétés dans $\mathcal{K}_s$. Il s'agit:

- pour les propriétés de type mathématique, du graphe:

$$[RELATION] \rightarrow (Attribut) \rightarrow [PROPRIETE]$$

- pour les propriétés de type agencement, du graphe:

$$\begin{array}{c} [RELATION] \rightarrow (Agencement) \leftarrow [RELATION] \\ \downarrow \\ [RELATION] \end{array}$$

- pour les liens sémantiques entre relations, du graphe:

$$[RELATION] \rightarrow (LienSémantique) \rightarrow [RELATION]$$

La spécialisation de ces graphes peut porter sur des types de concepts ou de relations. Les référents devant toujours être des marqueurs génériques ($\star$). Il est à noter que plusieurs relations d'endiguement peuvent être représentées par le même graphe contexte. La composition, la composition stricte à gauche et la composition stricte à droite (voir chapitre 3) peuvent être décrites par le même graphe contexte du type de la figure 4.16, muni de l'identificateur suivant:

$$\begin{array}{c} [RELATION : \star] \rightarrow (Composition) \leftarrow [RELATION : \star] \\ \downarrow \\ [RELATION : \star] \end{array}$$

L'utilisation du treillis des types de relations $\mathcal{T}_{Rspe}$ permet d'assurer la projection de cet identificateur sur des graphes conceptuels de $\mathcal{K}_s$ faisant état, par exemple, d'une relation de composition stricte.

Exemple 37 *La figure 4.18 suivante montre l'identificateur de la propriété de symétrie.*

Si dans $\mathcal{K}_s$ apparaît le graphe [RELATION: R_1]$\rightarrow$ (Attribut) $\rightarrow$ [SYMETRIQUE], alors l'identificateur du contexte de la figure 4.18 précédente se projettera sur ce graphe. Par la suite, il nous sera possible d'exploiter la sémantique de la propriété de symétrie pour la relation R_1.

Nous verrons dans la section 4.4.5(page 189) comment ces contextes identifiés peuvent être exploités pour inclure un raisonnement sur les relations et leurs propriétés dans le formalisme des graphes conceptuels.

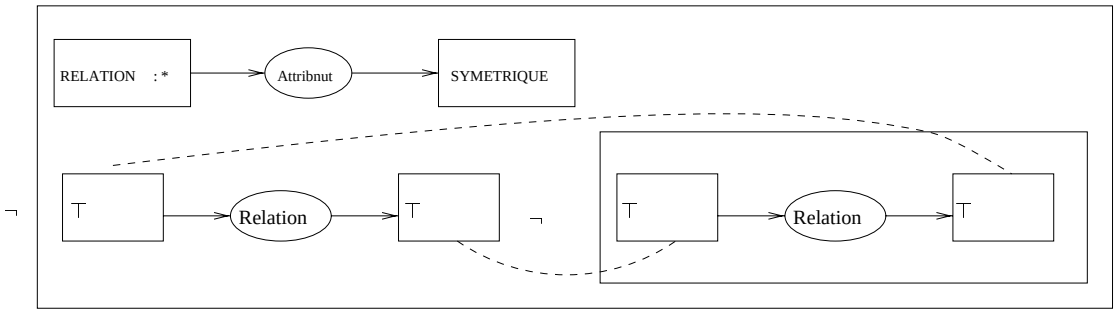


Figure 4.18. Applicabilité de la condition du contexte

4.4 Un Modèle de GCs Propre à la RI

L'ingénierie des connaissances est la science qui consiste à appliquer la logique et l'ontologie pour la construction de modèles opérationnels dédiés à satisfaire un but particulier dans un domaine donné [Sow97]. Un formalisme de représentation de connaissances s'attache donc, étant donnée l'ontologie associée au domaine d'application, à utiliser la logique pour analyser les connaissances de ce domaine et à fournir un modèle opérationnel pour l'exploiter. Typiquement, à l'inverse de l'ontologie représentant des connaissances qui ne changent pas au cours des inférences, un formalisme de représentation de connaissances gère une base de connaissances construite à partir du comportement et des interactions entre les éléments de l'ontologie grâce à un mécanisme d'inférence logique adéquat.

Les définitions que nous avons introduit dans les sections précédentes forment l'ontologie sur la base de laquelle nous construirons le formalisme des graphes conceptuels que nous utilisons pour la suite de ce chapitre et qui nous permettra d'implanter une approche relationnelle.

4.4.1 Graphe Conceptuel Bien Formé

Nous avons vu que le formalisme des graphes conceptuels met en oeuvre deux notions de base: les concepts et les relations conceptuelles. Ces deux notions sont les noeuds d'un graphe biparti, connexe, orienté et fini, appelé par définition *graphe conceptuel*.

Les concepts sont des entités discrètes, qui peuvent correspondre à des images mentales de l'être humain (concepts concrets), ou non (concepts abstraits). Par définition [Sow84], à chaque concept c est associé un *type* (ou TYPE_c), et un *marqueur* m. Un concept est représenté soit en forme linéaire par [TYPE_c: m], soit en forme graphique par une boîte: $\boxed{\text{TYPE}_c\text{: m}}$.

Définition 4.4.1 (Concept) *Un concept c est un couple (TYPE_c, m_c), où:*

- TYPE_c *est un élément de $\mathcal{T}_c$, l'ensemble des types de concepts du domaine;*
- *m est un élément de $\mathcal{M}$, l'ensemble des marqueurs quantifiés du canon. Si $m \in \zeta$, alors le concept c est dit* concept individuel, *si $m = \star$, alors le concept c est dit* concept générique, *enfin si $m = \exists$ le concept est dit* concept existentiel*;*

Les relations ont uniquement un type ($type_r$). Elles spécifient les rapports qui existent entre les concepts du graphe. Elles sont représentées soit par ($type_r$) en forme linéaire, soit par un ovale ($type_r$) en forme graphique. Les relations peuvent avoir n'importe quel nombre d'arcs (n-aires), même si l'usage montre que les relations unaires et binaires peuvent suffire pour représenter convenablement la connaissance du domaine [Sow84]. Ceci permet de limiter leur nombre à un ensemble restreint de *cas conceptuels*, indépendant du domaine. L'avantage est aussi de mieux expliciter le rôle joué par chaque concept relié à la relation. Cependant, comme nous l'avons montré dans le chapitre 2 (page 59), il est parfois utile en recherche d'informations d'introduire certaines relations n-aire. Leur sémantique étant bien précise, le rôle de chaque concept lié à la relation reste bien précis.

Définition 4.4.2 (Graphe conceptuel) *Un graphe conceptuel g est une structure $g = (C, R, A, ord, label)$, où:*

- *C et R désignent deux classes de noeuds concepts et de noeuds relations avec $C \neq \emptyset$*
- *A désigne un ensemble d'arcs. Chaque élément de A est une paire (n_i, n_j) de noeuds avec $n_i \in C$ et $n_j \in R$.*
- *les arcs adjacents à chaque relation $r \in R$ sont totalement ordonnés par la fonction ord. Ils sont numérotés de 1 à l'arité n de r. Le type de concept du i^{eme} noeud concept voisin de r dans g est dénoté par $g_i(r)$.*
- *chaque noeud concept et noeud relation du graphe conceptuel g admet un label défini par la fonction* label*, tel que:*
  - *si $r \in R$ alors $label(r) \in \mathcal{T}_r$*
  - *si $c \in C$ alors $label(c) \in \mathcal{T}_c \times \mathcal{M}$*

Le label d'un concept est ainsi une paire (TYPE_c, m_c), avec $\text{TYPE}_c \in \mathcal{T}_c$ et $m_c \in \mathcal{M}$. Tous les graphes formés à partir de la définition 4.4.2 ne vérifient pas forcément le canon du domaine. Ainsi, seuls les graphes respectant les contraintes du canon seront retenus par l'application. La notion de *graphe conceptuel bien formé* tend donc à distinguer des graphes $(C, R, A, ord, label)$ respectant les contraintes des graphes canoniques du départ. Afin d'introduire formellement cette notion de graphe conceptuel bien formé, nous utilisons les définitions suivantes:

Définition 4.4.3 (Concept bien formé) *Soit $g = (C, R, A, ord, label)$ un graphe conceptuel. Un concept $c \in C$ est dit bien formé si, et seulement si, c est muni d'un type de concept $\text{TYPE}_c \in \mathcal{T}_c$ et d'un marqueur $m \in \mathcal{M}$ avec: $\text{TYPE}_c :: m$, où "::" est la relation de conformité* Conf *définie dans le canon.*

Définition 4.4.4 (Relation bien formée) *Soit $g = (C, R, A, ord, label)$ un graphe conceptuel. Une relation $r \in R$ est dite bien formée si, et seulement si:*

- *Le nombre d'arcs adjacents à r est égal à l'arité n de sa signature Σ_r;*
- *les arcs adjacents à r sont totalement ordonnés par ord. Pour chaque $a = 1..n$, le type de concept du a^{eme} voisin de r dans g doit être, dans le treillis $\mathcal{T}_c$, inférieur ou égal au a^{eme} type de concept de la signature de r, i.e $g_a(r) \leq \Sigma_a(r)$ avec $a = 1..n$.*

- *La direction des arcs adjacents à r doit être conforme à celle de son graphe signature. Le i^{eme} arc du graphe signature de r a la même direction que le i^{eme} arc de la relation r.*

Définition 4.4.5 (graphe conceptuel bien formé) *Soit $g = (C, R, A, label)$ un graphe conceptuel. Le graphe g est dit bien formé relativement au canon du domaine d'application, si et seulement si:*

- *$\forall c \in C$, c est un concept bien formé;*
- *$\forall r \in R$, r est une relation bien formée;*
- *$\forall c_1, c_2 \in C$, si $label(c_1) = (type_{c_1}, m)$ et $label(c_2) = (type_{c_2}, m)$, où m est un marqueur individuel dans ζ, alors $type_{c_1} = type_{c_2}$.*

4.4.2 Compléments de Définitions

Afin de limiter les ambiguïtés et les imprécisions du modèle de base de Sowa, nous introduisons les compléments de définitions suivants.

Définition 4.4.6 (Égalité de Concepts) *Deux concepts sont dits égaux ou identiques si, et seulement si, ils ont le même type de concept et le même marqueur individuel:*

$$c = d \text{ ssi } label(c) = label(d) \text{ et } référent(c), référent(d) \in \zeta$$

Exemple 38 *Les graphes suivants contiennent des concepts différents:*

1. $[A : *] \rightarrow (r) \rightarrow [A : *]$*: Nous avons vu que le marqueur $\star$ correspond plutôt à un quantificateur universel. Considérer que les deux concepts de ce graphe sont égaux revient à asserter que la relation (r) est réflexive. Ceci n'est pas forcément le cas.*
2. $[A : \exists] \rightarrow (r) \rightarrow [A : \exists]$ *: Les deux marqueurs $\exists$ dénotent l'existence de deux éléments particuliers du domaine de discours, lesquels ne sont pas forcément les mêmes. Autrement dit, ils ne correspondent pas nécessairement au même témoin.*
3. $[A : *] \rightarrow (r) \rightarrow [A : \exists]$ *: Alors que le marqueur générique peut dénoter tous les éléments de la dénotation $A^{\mathcal{A}}$ de A, le marqueur existentiel dénote un élément particulier non identifié. Asserter que les deux concepts sont égaux revient à dire que le marqueur $\star$ est réduit à l'élément dénoté par $\exists$. Cela constitue une restriction de taille.*
4. $[A : *x] \rightarrow (r) \rightarrow [A : \exists x]$ *: Dans le premier concept, "*x" dénote une variable quelconque, alors que dans le second cas, elle identifie un seul élément particulier du domaine. Il ne s'agit donc pas de la même sémantique. Intuitivement, ce graphe indique que* "quelque soit x, x est relié par r à un élément particulier a du domaine de discours.

Cependant, les graphes suivants contiennent chacun deux concepts égaux:

1. $[A : *x] \rightarrow (r) \rightarrow [A : *x]$
2. $[A : \exists x] \rightarrow (r) \rightarrow [A : \exists x]$

Comme le souligne Sowa[17], la notion de "concepts identiques" n'est pas utilisée dans le modèle des graphes conceptuels. Cependant, l'opérateur de jointure introduit la notion de concepts communs:

Définition 4.4.7 (Concepts Communs) *Deux concepts $[c]$ et $[d]$ sont dits communs si, et seulement si, ils ont le même type de concept et le même référent. On écrit $c \asymp d$ pour désigner le fait que deux concepts sont communs.*

$$c \asymp d \; ssi \; label(c) = label(d)$$

La propriété suivante est ainsi facile à démontrer:

Propriété 8 *Deux concepts égaux sont forcément communs: si $c = d$ alors $c \asymp d$*

Afin d'introduire la notion d'égalité entre graphes conceptuels, nous définissons la notion de sous-graphe:

Définition 4.4.8 (Sous-graphe) *Un graphe g est un sous-graphe de h s'il existe une correspondance Π des relations et des concepts de g dans les concepts et les relations de h tels que:*

- *Pour chaque concept c de g, $\Pi(c)$ est un concept de h avec le même type de concept et le même marqueur.*
- *Pour chaque relation r de g, $\Pi(r)$ est une relation de h ayant le même type de relation*
- *Si c et r sont reliés dans g, alors $\Pi(c)$ et $\Pi(r)$ sont reliés de la même façon dans h.*

Nous avons alors la définition suivante:

Définition 4.4.9 (Égalité de Graphes) *Deux graphes g and h sont égaux si g est un sous-graphe de h, et h est un sous-graphe de g.*

4.4.3 Les Graphes Isomorphes

La notion de graphes isomorphes a déjà été utilisée par Guinaldo dans [Gui96a] ainsi que par Kheirbek [Khe95]. Nous spécifions ici cette notion dans le cadre d'une implantation d'un système à base de graphes conceptuels. Ainsi, le vocabulaire terminal du langage des graphes conceptuels est construit à partir des graphes signatures en recherchant tous leurs isomorphes. Intuitivement, un graphe est isomorphe à un autre graphe si le premier est une copie du second à des restrictions de concepts près.

Définition 4.4.10 (Graphes Isomorphes)

17. Note récente adressée par Sowa à la mailing-list des graphes conceptuels "cg@pluto.cs.uah.edu" en date du 6 Octobre 1997.

Le graphe $g = (C_1, R_1, A_1, ord_1, label_1)$ *est dit* isomorphe *au graphe* $h = (C_2, R_2, A_2, ord_2, label_2)$ *s'il existe une fonction bijective* f *de* C_1 *vers* C_2*, et une fonction bijective* g *de* R_1 *vers* R_2 *satisfaisant les contraintes suivantes:*

1. *les arcs et leur ordonnancement sont conservés: pour tout arc* r_1c_1 *de* A_1*,* $g(r_1)f(c_1)$ *est un arc de* A_2*. Si le* i^{eme} *arc de* r_1 *est lié à* c_1*, le* i^{eme} *arc de* $g(r_1)$ *est lié à* $f(c_1)$*.*
2. *Les labels des noeuds concepts de* C_1 *peuvent être diminués: pour tout* c *de* C_1*,* $label_2(f(c_1)) \leq label_1(c_1)$
3. *Les labels de noeuds relations de* C_1 *sont conservés: pour tout* r *de* R_1*,* $label_2(g(r_1)) = label_1(r_1)$

On notera par $g \cong h$ *le fait que* g *est un graphe isomorphe à* h*.*

La relation d'isomorphisme est ainsi une projection bijective entre deux graphes g et h ayant le même nombre de concepts et de relations conceptuelles. Nous introduisons alors la fonction *Ism* suivante:

Définition 4.4.11 *Soit* $\mathcal{G}$ *l'ensemble des graphes conceptuels. La fonction* $Ism : \mathcal{G} \rightarrow 2^{\mathcal{G}}$ *assigne à chaque graphe conceptuels* g*, les graphes qui lui sont isomorphes:*

$$h \in Ism(g) \text{ ssi } g \cong h$$

où $2^{\mathcal{G}}$ *désigne l'ensemble des parties de* $\mathcal{G}$*.*

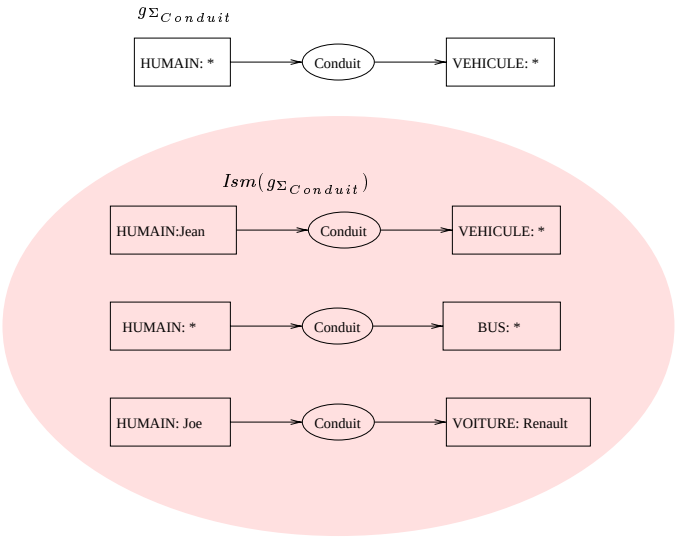


Figure 4.19. En Grisé les graphes isomorphes à $g_{\Sigma_{Conduit}}$

Il est facile de remarquer que les graphes de *Ism* ne sont pas comparables par la relation de spécialisation $\leq$. De plus, la relation d'isomorphisme entre graphes vérifie le corollaire 2:

Corollaire 2 *La relation d'isomorphisme* $\cong$ *est une relation réflexive, anti-symétrique et transitive.*

Preuve 12 *La réflexivité est facilement prouvée en prenant $f = g =$ la fonction identité. L'anti-symétrie découle directement des propriétés de la relation de spécialisation entre concepts et de la caractéristique bijective des deux fonctions f et g. Si $u \cong v$, alors il existe deux fonctions bijectives f et g vérifiant la première propriété de la définition 4.4.10. Si $v \cong u$, alors la relation d'isomorphisme est également munie de deux fonctions bijectives f' et g' vérifiant la même propriété. Les quatre fonctions étant bijectives, nous avons alors $f^{-1} = f'$ et $g^{-1} = g'$. Il en résulte du fait que la relation $\leq$ est anti-symétrique, une égalité entre les labels de noeuds concepts et les labels des noeuds relations de u et v. L'utilisation de la définition 4.4.9 permet de déduire que $u = v$. La propriété d'anti-symétrie est ainsi vérifiée. La transitivité est prouvée de la façon suivante: si $u \cong v$, alors la relation d'isomorphisme est munie de deux fonctions bijectives f et g. De plus si $v \cong w$, alors nous assignons de la même manière deux fonctions bijectives f' et g'. Du fait que les quatre fonctions sont bijectives et que les relations d'égalité $=$ et de spécialisation $\leq$ sont transitives, il existe $f'' = f' \circ f$, et $g'' = g' \circ g$, tel que f'' et g'' sont bijectives et vérifient les conditions de la définition 4.4.10.*

Exemple 39 *Soit* $\Sigma_{Conduit} = (Conduit, 2, \text{HUMAIN}, \text{VEHICULE})$ *la signature de la relation* Conduit. *La figure 4.19 montre trois graphes isomorphes à son graphe signature* $g_{\Sigma_{Conduit}}$.

D'une manière plus générale, nous définissons la relation de E-isomorphisme de la façon suivante:

Définition 4.4.12 (E-isomorphisme) *Deux graphes g et h sont dits E-isomorphes, s'ils deviennent isomorphes après une ou plusieurs applications des opérations suivantes:*

(1) *"Éclatement" d'un noeud concept dans g ou h afin d'obtenir un graphe non connexe*

(2) *Si un des deux graphes, disons g, a deux sous-graphes g et g', tel que $g = g'$, alors un de ces deux sous-graphes est supprimé*

Dans ce cas, on écrit que $g \cong_E h$.

Exemple 40 *Les deux graphes g et h de la figure 4.20 sont E-isomorphes. L'éclatement du noeud concept* [HUMAIN : *Jean*] *dans g produit deux sous graphes identiques. L'application de l'opération (2) produit un graphe isomorphe à h*

Comme nous le verrons dans la section b), le E-isomorphisme nous permet d'obtenir la forme standard (ou normale) d'un graphe conceptuel.

4.4.4 Le Langage d'Indexation Relationnelle

Le langage des graphes conceptuels, comme tout langage de représentation, est spécifié par la donnée d'un alphabet $\mathcal{A}_{GC}$ et des règles de formation syntaxiques $\mathcal{S}_{GC}$. Le langage des graphes conceptuels $\mathcal{L}_{GC}$ est donc défini par un couple $\langle \mathcal{A}_{GC}, \mathcal{S}_{GC} \rangle$. Un graphe conceptuel est construit à partir de concepts individuels ou génériques, et de relations. L'alphabet $\mathcal{A}_{GC}$ est un quadruplet $\langle \mathcal{T}_c, \mathcal{M}, \mathcal{T}_r, \mathcal{B}_{can} \rangle$, où:

- $\mathcal{T}_c$ est un ensemble de types de concepts.

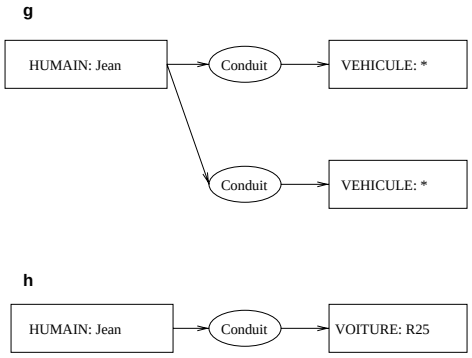


Figure 4.20. Deux graphes E-isomorphes

- $\mathcal{M}$ est un ensemble de marqueurs, correspondant à des éléments du domaine de discours.
- $\mathcal{T}_r$ dénote un ensemble de types de relations.
- $\mathcal{B}_{can}$ est un ensemble de graphes conceptuels de départ, considérés comme bien formés et modélisant les connaissances primitives d'un domaine de discours.

L'alphabet $\mathcal{A}_{GC}$ constitue le vocabulaire primitif des graphes conceptuels. Des graphes conceptuels plus complexes peuvent être formés par le biais des règles de formation $\mathcal{S}_{GC}$ appliquées à $\mathcal{B}_{can}$. Ces règles sont au nombre de quatre, il s'agit de la copie, de la restriction, de la jointure et de la simplification [Sow84].

Définition 4.4.13 (Le langage des GCs) *Le langage des graphe conceptuels $\mathcal{L}_{GC}$ est le couple $\langle \mathcal{A}_{GC}\ ,\ \mathcal{S}_{GC}\ \rangle$, où $\mathcal{A}_{GC} = \langle \mathcal{T}_c, \mathcal{M}, \mathcal{T}_r, \mathcal{B}_{can}\ \rangle$et $\mathcal{S}_{GC} = \{$copie, restriction, jointure, simplification$\}$.*

Dans une approche d'indexation relationnelle, seules des structures sémantiques incluant au moins une relation peuvent être considérées comme des bons termes d'indexation. Ainsi, un graphe conceptuel formé d'un seul concept, même s'il est correct syntaxiquement, n'est pas un élément du langage d'indexation. Il serait en effet incohérent d'indexer les documents par des termes de granularité inférieure à celle des signatures des relations. Les signatures constituant par définition l'alphabet de base d'un langage d'indexation relationnelle. Nous énonçons alors l'hypothèse suivante:

Hypothèse 8 (Graphe Index) *Un graphe conceptuel bien formé $(C, R, A, ord, label)$ peut être considéré comme un index dans une approche relationnelle s'il contient au moins une relation conceptuelle, autrement dit si $R \neq \emptyset$. Nous notons par $\mathcal{G}$ l'ensemble des graphes index.*

De plus un graphe canonique de $\mathcal{B}_{can}$, ayant deux relations ou plus, peut être construit à partir des graphes isomorphes aux graphes signatures des relations qu'il contient. Cette constatation rappelle la définition de minimalité d'une base canonique introduite par Chevallet [Che92, pp91]:

> "Une base canonique est minimale pour un ensemble de graphes donné lorsque toute base obtenue par suppression de l'un de ses éléments ne lui est pas équivalente, c'est-à-dire qu'elle engendre un ensemble de graphes différents."

Formellement, une base canonique minimale $\mathcal{B}_{min}$ peut être définie comme suit:

Définition 4.4.14 (Base canonique Minimale) *Une base canonique minimale devra vérifier les conditions suivantes:*

(1) *Pour tout m_i, $m_j \in \mathcal{B}_{min}$, m_i et m_j ne sont pas comparables par la relation d'ordre $\leq$ sur les graphes : $\neg(m_i \leq m_j)$ et $\neg(m_j \leq m_i)$*

(2) *Étant donné un graphe conceptuel bien formé g, g est un élément de la fermeture de $\mathcal{B}_{min}$ sur les opérateurs de formation de graphes: $\forall g = (C, R, A, ord, label)$, $g \in fermeture(\mathcal{B}_{min})$*

(3) *Pour tout m_i, $m_j \in \mathcal{B}_{min}$, les ensembles des graphes isomorphes à m_i et à m_j sont disjoints: $\forall m_i, m_j, Ism(m_i) \cap Ism(m_j) = \emptyset$.*

Il est alors facile de vérifier que l'ensemble $\mathcal{S}_r$ des graphes signatures est une base minimale:

Propriété 9 ($\mathcal{S}_r$ est une Base Minimale) *Par rapport à l'ensemble des graphes index $\mathcal{G}$ de l'hypothèse 8 ci-dessus, l'ensemble des graphes signatures $\mathcal{S}_r$ est une base canonique minimale.*

Preuve 13 *Le théorème 14(page 171) précise que chaque type de relation apparaît une et une seule fois dans $\mathcal{S}_r$, et que chaque graphe signature contient une seule relation. La propriété (3) de la définition 4.4.14 ci-dessus est directement vérifiée de part la notion même de graphes isomorphes. En effet, les labels des noeuds relations entre un graphe et son isomorphe étant conservés, tous les graphes isomorphes à m_i auront sa même relation. Similairement, les graphes isomorphes à m_j contiennent sa relation. Il n'est donc pas possible d'avoir une intersection entre les graphes isomorphes à m_i et ceux isomorphes à m_j, puisque la relation apparaissant dans m_i est différente de celle contenue dans m_j. Pour la même raison, il n'est pas également possible d'obtenir m_i de m_j (ou l'inverse) par une ou plusieurs applications des quatre opérateurs de formation de graphes (leur relation étant différente). La propriété (1) est ainsi vérifiée. La propriété (2) découle du fait que chaque graphe index peut être formé par l'application d'une jointure sur les graphes isomorphes aux signatures des relations qu'il contient.*

Chevallet [Che92](page 92) montre qu'il n'existe qu'une base canonique minimale pour un ensemble de graphes canoniques. La démonstration se base sur les propriétés de la relation d'ordre $\leq$ sur les graphes. Ce résultat important signifie que, connaissant un ensemble de graphes conceptuels qu'on se donne le droit d'utiliser, c'est-à-dire un ensemble de graphes canoniques, on sait qu'il n'existe qu'une seule base contenant le moins de graphes possibles qui permettent de construire tous ces graphes. Ainsi dans notre cas, l'ensemble des graphes signatures est non seulement une base canonique minimale, mais elle est la seule qui le soit.

Théorème 15 *Dans une approche d'indexation relationnelle, l'ensemble des graphes signatures $\mathcal{S}_r$ constitue la seule base canonique minimale pour l'ensemble des graphes index $\mathcal{G}$.*

Preuve 14 *La preuve est triviale. Elle se base sur l'unicité de la base canonique minimale pour un ensemble de graphes canoniques donné. Cette unicité a été démontrée par Chevallet. L'ensemble $\mathcal{S}_r$ étant une base canonique minimale, il est le seul qui le soit.*

Il est à noter que si un concept peut être utilisé comme un graphe index, on aura comme base canonique minimale $\mathcal{B}_{Can_{Min}} = \mathcal{S}_r + [\top]$. Il suffit en effet d'ajouter le concept universel, pour dériver tout les graphes concepts possibles. Ce résultat est intéressant puisque quelque soit le domaine de l'application et le type des graphes autorisé, la construction d'une base canonique minimale ne nécessite que la définition des graphes signatures des types de relations utilisée par l'application. Ce résultat n'est toutefois valide que si l'on admet qu'un type de relation ne peut avoir qu'une seule signature, c'est-à-dire un seul sens possible. Cette condition, par ailleurs fort logique, ne semble pas être respectée dans tous les systèmes à base de graphes conceptuels. C'est le cas par exemple du système de recherche d'images EMIR_2 dont la base canonique minimale comporte plusieurs triplets $[C_1]\rightarrow(\text{r})\rightarrow[C_2]$ contenant la même relation binaire r.

L'union des graphes isomorphes aux graphes signatures de $\mathcal{S}_r$ forme l'alphabet de base de notre langage d'indexation relationnelle, ou en d'autres termes les connaissances de base utilisées pour représenter le contenu des documents. Nous appelons *Alphabet du Langage d'indexation Relationnelle*, noté par $\mathcal{A}_{\mathcal{LR}}$, ce vocabulaire terminal:

$$\mathcal{A}_{\mathcal{LR}} = Ism(\mathcal{S}_r) = \bigcup_{\Sigma_r \in \mathcal{S}_r} Ism(\Sigma_r)$$

Au plus haut niveau, le langage d'indexation relationnelle est équivalent à l'ensemble des graphes canoniques pouvant être générés à partir de $\mathcal{S}_r$, et par application des opérateurs de formation de graphes. On désignera ce langage par le terme *langage d'indexation maximal.* Par ailleurs, étant donné le fait que l'opérateur de *Restriction* est implicitement pris en compte dans le vocabulaire $\mathcal{A}_{\mathcal{LR}}$, le langage d'indexation maximal est introduit de la façon suivante:

Definition 4.4.1 (Le Langage d'Indexation Relationnelle Maximal) *Le langage d'indexation relationnelle maximal $\mathcal{LR}_{Max}$, fondé sur les graphe conceptuels est le couple $\langle \mathcal{A}_{\mathcal{LR}}, \mathcal{R}_{\mathcal{LR}} \rangle$ où $\mathcal{A}_{\mathcal{LR}} = Ism(\mathcal{S}_r)$ et $\mathcal{R}_{\mathcal{LR}} = \{$Copie, Simplification, jointure$\}$.*

Au plus bas niveau, notre langage d'indexation relationnelle est égal à $Ism(\mathcal{S}_r)$. Plus généralement, il peut être défini en extension à partir de compositions (jointures) d'éléments du vocabulaire terminal $Ism(\mathcal{S}_r)$. Il est ainsi un sous-ensemble de tous les graphes canoniques pouvant être générés à partir de la base canonique minimale $\mathcal{S}_r$, soit un sous-langage de $\mathcal{LR}_{Max}$:

$$\mathcal{LR} = \langle Ism(\mathcal{S}_r), \mathcal{R}_{Stx} \rangle \subseteq \mathcal{LR}_{Max}$$

où $\mathcal{R}_{Stx}$ est un ensemble de règles syntaxiques particulières, constituant les éléments du langage à partir de $Ism(\mathcal{S}_r)$. Dans le cas, où $\mathcal{R}_{Stx}$ est vide et où le langage est défini par sa syntaxe, $\mathcal{LR}$ est réduit à $Ism(\mathcal{S}_r)$, ce qui correspond au cas minimal évoqué plus haut.

Dans la suite, et par mesure de généralité, nous supposons que le langage d'indexation relationnelle est maximal par rapport à la base de canonique. Par $\mathcal{LR}$ nous désignons ainsi $\mathcal{LR}_{Max}$. Par conséquent, les graphes index de $\mathcal{G}$ constituent des expressions de $\mathcal{LR}_{Max}$.

4.4.5 Méta-Connaissances et Règles de Dérivation

Les méta-connaissances représentent des informations sur l'ensemble des graphes $\mathcal{K}_s$, c'est-à-dire des informations qui se rattachent aux propriétés des relations en précisant leur nature ou en

spécifiant leurs caractéristiques. Il s'agit principalement de règles de dérivation de graphes. Elles sont de deux types: les règles de dérivation sur les relations d'ordre supérieur et les règles de pertinence définies dans notre modèle relationnel du chapitre 3. Dans les deux cas, ces règles sont représentées par des graphes contextes.

Les règles de dérivation sur les relations d'ordre deux

Il s'agit de spécifier quelles dérivations de connaissances on peut établir sur les relations d'ordre deux. Par exemple, que tel lien sémantique spécifique à $\rightarrow(\mathit{LienSémantique})\rightarrow$est symétrique, que les relations de type inférieur ou égal à $\rightarrow(\mathit{Agencement})\rightarrow$sont symétriques, etc.

Par exemple, la relation d'inversion étant symétrique, le graphe

$$[RELATION : a \Leftrightarrow droite] \rightarrow (Inversion) \rightarrow [RELATION : a \Leftrightarrow gauche]$$

dérive

$$[RELATION : a \Leftrightarrow droite] \leftarrow (Inversion) \leftarrow [RELATION : a \Leftrightarrow gauche]$$

Les règles de ce type seront dénotées par $\mathcal{R}_{R_{Spe}}$. L'intérêt de ces règles est de permettre la constitution d'un ensemble minimal $\mathcal{K}_{s_{Min}}$ de connaissances sur le domaine. Si ce dernier est défini, il aura la propriété de produire toutes les autres connaissances du domaine par fermeture sur les règles $\mathcal{R}_{R_{Spe}}$ (voir figure 4.21). Cet ensemble minimal permet d'une part de limiter le volume du stockage des connaissances (ainsi dans l'exemple précédent, un seul des deux graphes sera retenu dans $\mathcal{K}_{s_{Min}}$, et d'autre part de garantir que toutes les connaissances implicites pouvant être déduites des propriétés des relations d'ordre supérieur sont bien considérées par le système.

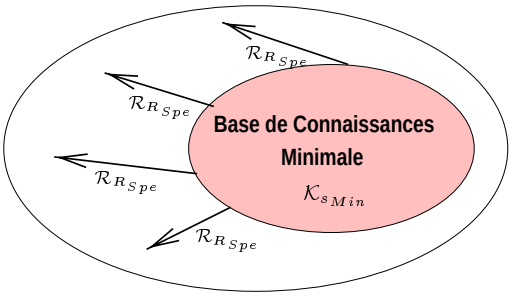


Figure 4.21. Base de Connaissances Minimale

En pratique, la constitution d'une base de connaissances minimale n'est pas une tâche facile. Elle requiert des connaissances expertes sur le domaine que l'on veut modéliser. Elle est d'autant plus difficile à définir et à maîtriser qu'elle augmente en taille. Nous devons nous intéresser à l'existence de bases de connaissances équivalentes, puis à la plus petite de ces bases équivalentes que nous noterons par $\mathcal{K}_{s_{Min}}$. Deux bases sont *équivalentes* quand les ensembles des connaissances engendrées par fermeture sur les règles de dérivation $\mathcal{R}_{R_{Spe}}$, sont égaux: soient deux bases de connaissances $\mathcal{K}_1$ et $\mathcal{K}_2$, on dit que $\mathcal{K}_1$ est équivalente à $\mathcal{K}_2$ si, et seulement si $\mathcal{F}_{\mathcal{R}_{R_{Spe}}}(\mathcal{K}_1) = \mathcal{F}_{\mathcal{R}_{R_{Spe}}}(\mathcal{K}_2)$, où $\mathcal{F}_{\mathcal{R}_{R_{Spe}}}(x)$ désigne la fermeture générée par le biais de $\mathcal{R}_{R_{Spe}}$ à partir de x.

Comme pour les bases canoniques minimales, nous pouvons établir la définition intuitive suivante d'une base de connaissances minimale:

Définition 4.4.15 (Base de Connaissances Minimale) *Une base de connaissances est minimale pour un ensemble de connaissances donné lorsque toute base obtenue par suppression de l'un de ses éléments ne lui est pas équivalente, c'est-à-dire qu'elle engendre un ensemble de connaissances différent.*

Contrairement aux bases canoniques, il peut exister plusieurs bases de connaissances minimales $\mathcal{K}_{s_{Min}}$. En effet, Considérons le cas des deux bases de connaissances $\mathcal{K}_1$ et $\mathcal{K}_2$ de la figure 4.22. Supposons que l'ensemble des règles $\mathcal{R}_{R_{Spe}}$ est constitué d'un seul graphe contexte comme cela est indiqué dans la figure. Par fermeture sur $\mathcal{R}_{R_{Spe}}$, ces deux bases de connaissances engendrent le même ensemble de connaissances $\mathcal{K}_s$.

Pourtant, les deux bases de connaissances $\mathcal{K}_1$ et $\mathcal{K}_2$ vérifient la condition de minimalité. Dans ce cas, les connaissances $\mathcal{K}_s$ du système peuvent être générées à partir de deux bases minimales équivalentes différentes. Ainsi, à l'inverse du cas des bases canoniques, pour un ensemble de connaissances donné, il n'existe pas une seule base de connaissances minimale, mais plusieurs. Du fait des propriétés de la relation d'ordre $\leq$ sur les graphes engendrée par les quatre opérateurs de formation de graphes, il a été possible dans [Che92] d'attester l'existence d'une seule base canonique minimale. Les règles de dérivation $\mathcal{R}_{R_{Spe}}$ ont manifestement d'autres propriétés et caractéristiques. Nous énonçons alors le théorème suivant:

Théorème 16 (Multiplicité des Bases de Connaissances Minimales) *Pour un ensemble de connaissances $\mathcal{K}_s$ et un ensemble de règles de dérivation $\mathcal{R}_{R_{Spe}}$, il peut exister plusieurs bases de connaissances minimales équivalentes $\mathcal{K}_{s_{Min}}$ pouvant engendrer $\mathcal{K}_s$ par fermeture sur les règles $\mathcal{R}_{R_{Spe}}$.*

Preuve 15 *Démonstration par contre-exemple. Il suffit de prouver que pour un ensemble donné de connaissances $\mathcal{K}_s$, et un ensemble de règles de dérivation, il peut exister plusieurs bases de connaissances minimales à partir desquelles il est possible de construire $\mathcal{K}_s$ par fermeture sur les règles de dérivation. L'exemple de la figure suivante illustre un tel cas.*

Cette multiplicité des bases minimales complexifie la tâche de création d'une base de connaissances dans le cas d'une démarche relationnelle. Afin d'éviter les redondances en termes de connaissances, la démarche que nous avons implémenté dans le système RELIEF [OP97b], lors de la saisie de la base de connaissances $\mathcal{K}_s$ de l'application, est de vérifier que chaque fois qu'une connaissance est ajoutée, elle ne peut être déduite par $\mathcal{R}_{R_{Spe}}$ à partir de celles déjà existantes dans la base. Dans le cas contraire, un message d'erreur est affiché à l'écran et la connaissance n'est pas introduite. Cette démarche consiste simplement à vérifier que la base de connaissances introduite est minimale.

Les règles de pertinence

Il s'agit des règles spécifiées par le modèle relationnel du chapitre 3, tendant à inclure les relations et leurs propriétés dans le processus de recherche. Intuitivement, si le sous-graphe

$$[RELATION : compose] \rightarrow (Attribut) \rightarrow [TRANSITIVE]$$

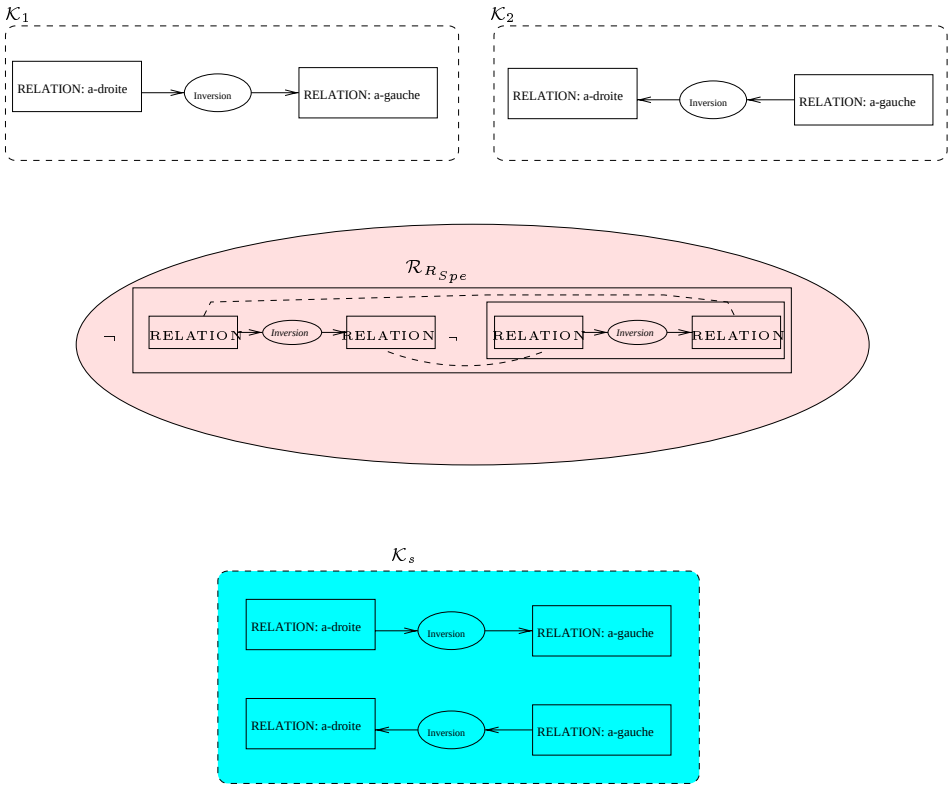
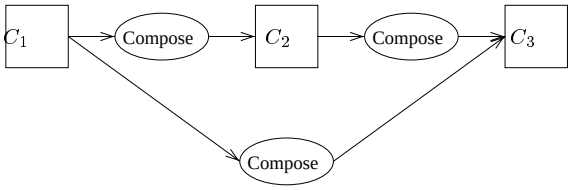


Figure 4.22. Deux bases de connaissances minimales

est présent dans $\mathcal{K}_s$, tout sous-graphe de la forme $\boxed{C_1} \rightarrow \text{Compose} \rightarrow \boxed{C_2} \rightarrow \text{Compose} \rightarrow \boxed{C_3}$, sera remplacé par le sous-graphe:



Nous noterons par $\mathcal{R}_{M_{Rel}}$ l'ensemble des règles proposées par le modèle relationnel pour la prise en compte des propriétés des relations. Ces règles ont pour objectif de combler les limites de l'opérateur de projection, en lui associant un moteur d'inférence spécialisé dans la gestion des relations.

Dans [SM96], un cadre logique permettant le raisonnement sur des règles de production, sous forme de graphes conceptuels, est présenté. Un système de chaînage avant permet alors d'enrichir un graphe de départ par des informations implicites dérivées des dites règles. L'application d'une règle sur un graphe g sera exécutée s'il existe une projection d'un sous-graphe de g sur la prémisse de la règle. Nous nous inspirons de ce cadre général pour établir un système de dérivation de graphes conceptuels incluant un raisonnement sur les relations.

D'une façon informelle, quand une propriété sur les relations apparaît sous forme d'une connaissance dans l'ensemble $\mathcal{K}_s$, les graphes signatures des relations que cette connaissance met en jeu seront utilisés pour spécifier le graphe prémisse à l'application de la règle de dérivation de la propriété de relation en question. Par exemple, quand une relation R_1 peut être composée avec une relation R_2 pour donner lieu à la relation R_3, les graphes signatures de R_1 et de R_2 vont être joints de manière à reproduire la prémisse de la propriété de composition : **Si** R_1(x, y) et R_2(y, z) **Alors** R_3(x, z). Dans ce cas, le deuxième concept du graphe signature de R_1 va être joint au premier concept du graphe signature de R_2. C'est l'ensemble d'endiguement $\mathcal{E}$ qui nous donnera cette information.

Dans l'exemple de la figure 4.23 suivante, la projection de la prémisse du contexte représentant la relation de composition sur les graphes signatures associés aux relations apparaissant dans un graphe de $\mathcal{K}_s$, fournit le graphe prémisse à l'application de la règle de dérivation de la propriété en question. Ici nous avons supposé que l'identificateur du contexte se projette correctement sur le graphe de $\mathcal{K}_s$.

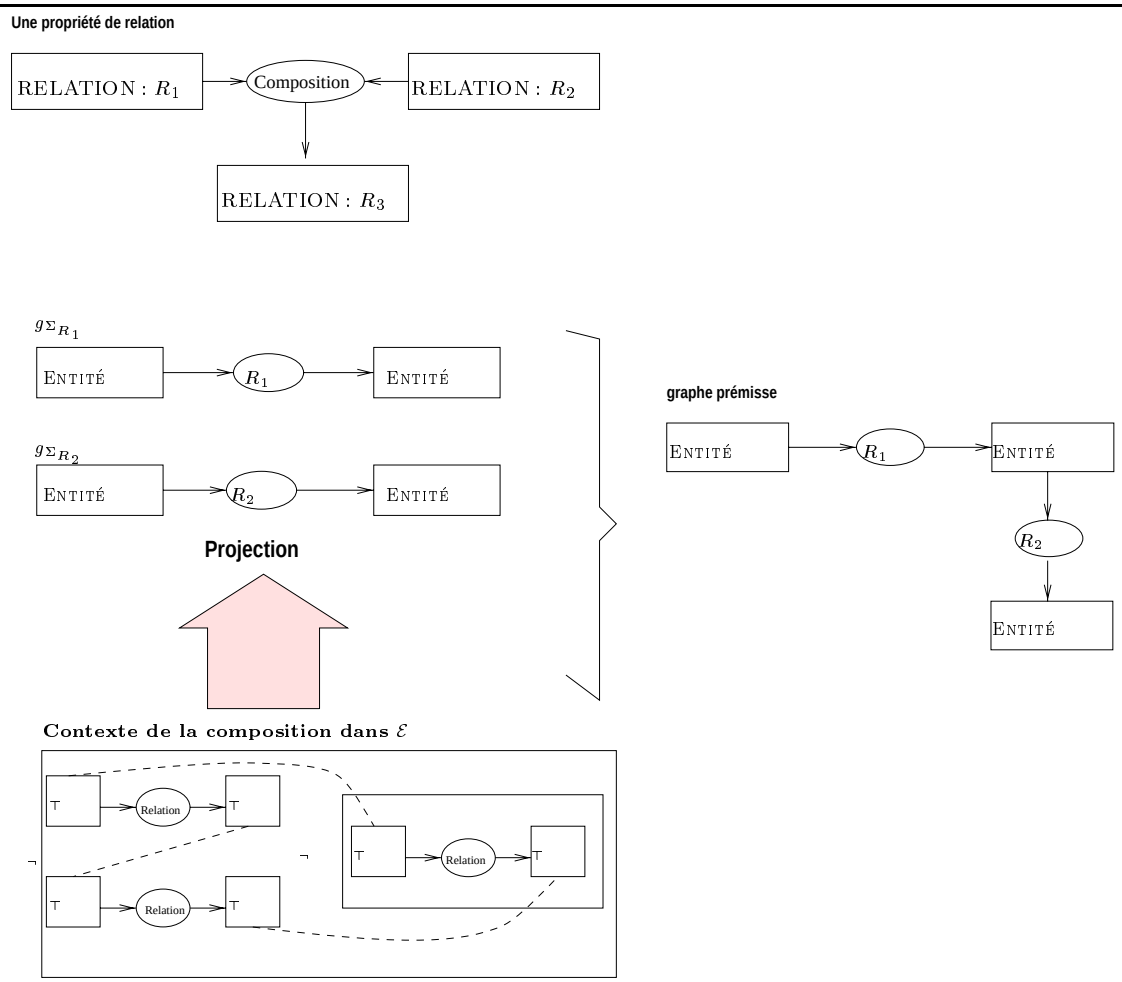


Figure 4.23. Exemple de dérivation de graphes

Les prémisses des règles sont ainsi construites à partir de tous les contextes de $\mathcal{E}$ en projetant leurs prémisses sur les graphes signatures associés aux relations pour lesquels la propriété du contexte est vérifiée. Il ne reste plus qu'à spécifier le résultat de la relation d'endiguement sous-jacente en termes de graphes conceptuels. Comme nous l'avons vu au chapitre 3, le résultat de l'application d'une règle de dérivation de $\mathcal{R}_{M_{Rel}}$ aux graphes prémisses varie selon qu'il s'agit d'un endiguement équivalent, d'un endiguement faible ou d'un endiguement avec retenue. Par exemple, supposons que $\mathcal{K}_s$ contient également le graphe suivant:

$$[RELATION : r] \rightarrow (Attribut) \rightarrow [SYMETRIQUE]$$

Considérons le cas où le graphe signature g_{Σ_r} de la relation r est défini comme suit:

$$[\text{ENTITÉ}] \rightarrow (r) \rightarrow [\text{ENTITÉ}]$$

Dans la page 179, nous avons donné le contexte de la propriété de symétrie. La projection de sa prémisse sur le graphe signature de r donne le graphe "Graphe prémisse" de la figure 4.24 suivante. Ce même contexte fournit la conséquence de la règle de production qu'il véhicule. Dans ce cas, il s'agit du graphe [ENTITÉ]$\leftarrow$ (r) $\leftarrow$ [ENTITÉ]. Cependant dans le chapitre 3, nous avons défini la propriété de symétrie (i.e, permutation) comme étant une relation d'endiguement équivalent. Dans ce cas, le graphe dérivé s'ajoute (par jointure) au graphe de départ. Le graphe de départ ici est le graphe signature de r alors que le graphe dérivé est celui fournit par le contexte de la propriété de symétrie. Le graphe "Résultat" de la figure 4.24 mentionne bien cet ajout.

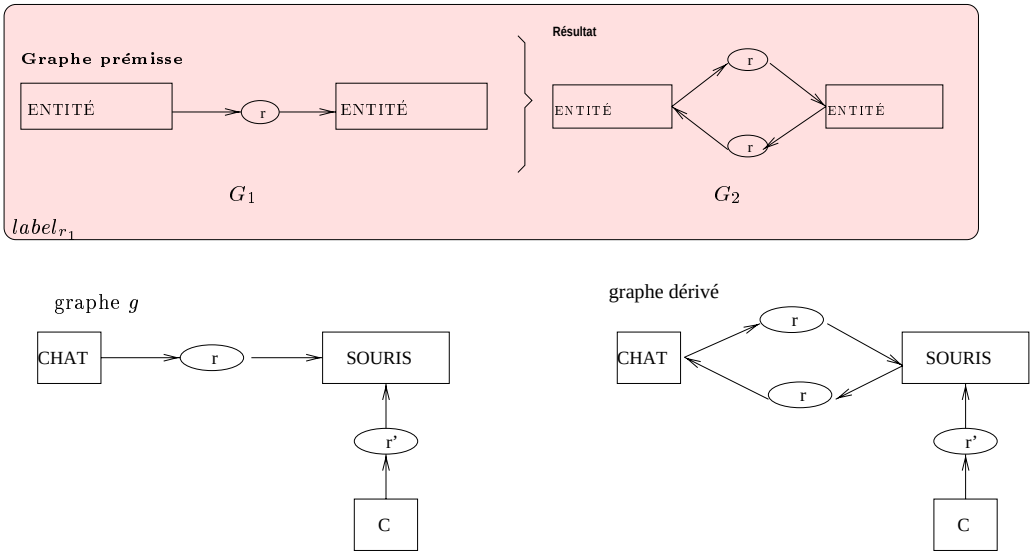


Figure 4.24. Exemple de dérivation de graphes

D'une façon plus générale, les règles $\mathcal{R}_{M_{Rel}}$ reprennent les conséquences des graphes contextes de l'ensemble d'endiguement $\mathcal{E}$ en tenant compte du type d'endiguement véhiculé. Dans le cas d'un endiguement équivalent ou d'un endiguement avec retenue, l'information dérivée vient s'ajouter au graphe prémisse de départ. Pour ce qui de l'endiguement faible, l'information dérivée remplace

celle de départ. Dans la section 4.2.1(page 152), nous avons montré que la relation d'endiguement (ED) n'est pas applicable dans le cas où les index des documents correspondent à des graphes conceptuels simples. Nous avons également montré que la propriété de restriction de concept est implicitement prise en compte par l'opérateur de projection. Reste donc à savoir comment intégrer la propriété de spécialisation des relations dans les décisions de pertinence. Il s'agit d'exploiter les relations de spécialisation/généralisation du treillis des types de relations $\mathcal{T}_r$. Vu la taille que peut avoir ce treillis, il sera plus facile dans ce cas de considérer un opérateur de restriction de relations sur les graphes que de représenter toutes ces relations de spécialisation par des règles de $\mathcal{R}_{M_{Rel}}$ (voir figure 4.25). L'idée est d'étendre l'opérateur de projection par un raisonnement sur le treillis $\mathcal{T}_r$. Pour ce faire, nous adoptons l'extension suivante de la définition de la projection:

Définition 4.4.16 (Projection Relationnelle) *Il existe une projection d'un graphe h sur un graphe g si, et seulement si, il existe une application π de h vers g satisfaisant les conditions suivantes:*

- *pour chaque concept c de h, $\pi(c)$ est soit une spécialisation de c, soit identique à c.*
- *pour chaque relation conceptuelle r de h, $\pi(r)$ est une relation dans g, tel que $\pi(r) \leq_\sigma r$, où $\leq_\sigma$ est la relation d'ordre partiel que nous avons défini sur les relations.*
- *si le i^{eme} arc de r est lié à un concept c dans g, alors le j^{eme} arc de $\pi(r)$ doit être lié à $\pi(c)$ dans g, avec $j = \sigma(i)$ où σ est la fonction injective associée à la relation d'ordre sur les relation $\leq_\sigma$.*

Un graphe conceptuel g peut être dérivé à partir d'un autre graphe h, si une opération de généralisation est appliquée à une ou à plusieurs relations dans h. Cette dérivation trouve son duale en termes de graphes conceptuels par le biais d'une nouveau opérateur de restriction:

Définition 4.4.17 (Restriction sur les Relations) *Soit u un graphe conceptuel. Chaque relation r de u peut être remplacée par une relation r', tel que $r' \leq_\sigma r$. Ce changement doit se faire avec une double précaution:*

(1) *Le type de concept du i^{eme} noeud concept voisin de r dans u doit être en relation de spécialisation avec le j^{eme} argument de la signature de r', où $j = \sigma(i)$, i.e., $u_i(r) \leq \Sigma_{\sigma(i)}(r')$. Dans le cas contraire, le type de concept du i^{eme} noeud concept de r doit être restreint au j^{eme} argument de la signature de r'. Cette restriction de type doit aussi vérifier la relation de conformité, si le i^{eme} noeud concept de r est un concept individuel. Si cette relation de conformité n'est pas vérifiée, la restriction de la relation r à la relation r' n'est pas autorisée.*

(2) *Si les relations r et r' n'ont pas la même arité, en plus de la condition (1), le graphe u est "complété' au niveau de la relation r', en associant un noeud de concept générique à tous les types de concepts de la signature de r' pour lesquels il n'existe pas d'antécédent avec la fonction σ. Ces concepts sont ensuite liés à la relation r' en ajoutant les arcs nécessaires.*

Exemple 41 *Si on reprend l'exemple des deux relations spatiales* "a-droite" *et* "entre" *tels que* Σ_{entre}= (entre, 3, A', B, C) *et* $\Sigma_{a-droite}$= (a-droite, 2, A, B), *alors le graphe:*

[A" : a] $\rightarrow$ (a-droite) $\rightarrow$ [B]

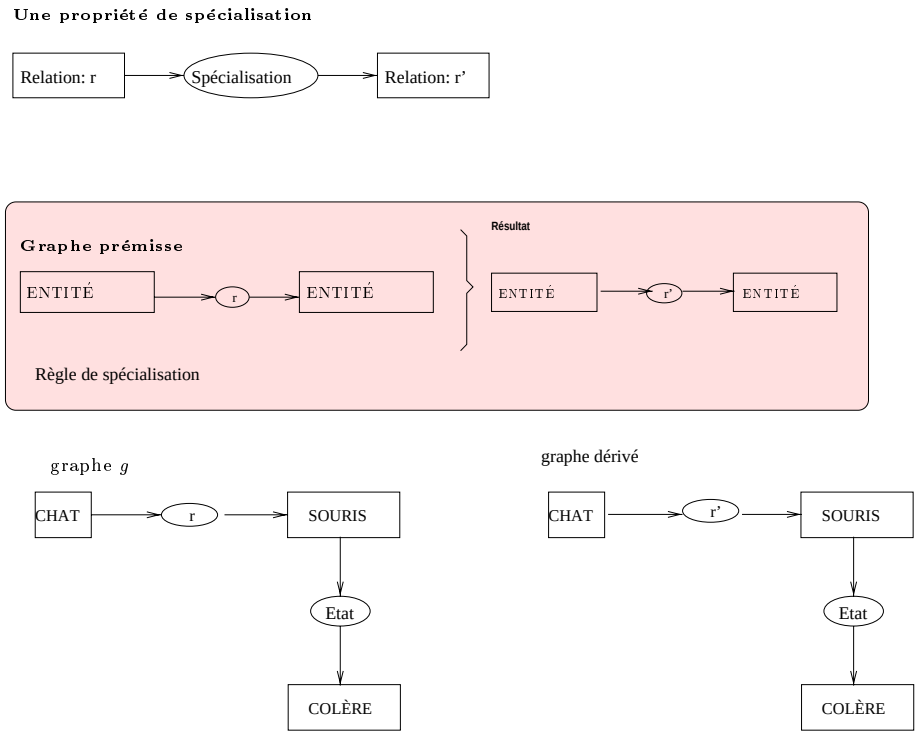


Figure 4.25. Application de la propriété de spécialisation

peut être restreint au graphe:

$$\begin{array}{c} [\text{A''} : a] \rightarrow (entre) \rightarrow [\text{B}] \\ \downarrow \\ [\text{C}] \end{array}$$

uniquement dans le cas où $A'' \leq A'$.

Par rapport à l'opérateur de translation ϕ de Sowa, il est facile de vérifier que tout comme les quatre opérateurs de construction de graphes, l'opérateur de restriction sur les relations vérifie la propriété suivante:

Propriété 10 *Si le graphe conceptuel* v *est obtenu à partir d'un graphe* u *par un opérateur de restriction de relation, alors* $v \leq u$ *et* $\phi(v) \rightarrow \phi(u)$

L'ensemble des règles $\mathcal{R}_{M_{Rel}}$ se limitera donc à la prise en compte des propriétés de l'endiguement avec retenue et de l'endiguement équivalent. Plus formellement, une règle de $\mathcal{R}_{M_{Rel}}$ est définie de la façon suivante:

Définition 4.4.18 (Les Règles de Dérivation de $\mathcal{R}_{M_{Rel}}$) *Une règle de dérivation sur les graphes* $label_R$*:* $G_1 \Rightarrow G_2$*, est un couple de lambda-expressions* $(\lambda x_{11} \ldots x_{1n} G_1,\ \lambda x_{21} \ldots x_{2n} G_2)$*. Les* x_i*,*

$i = 1, n$ constituent des variables correspondant à des liens de coréférence entre des concepts apparaissant simultanément dans G_1 et dans G_2. Nous avons alors x_{1i} est coréférent de x_{2i}. Quant à $label_R$, il désigne le label de la règle en question.

Le label de la règle peut être utilisé comme nous l'avons vu à la fin du chapitre précédent pour justifier et analyser les réponses fournies par le système. Le raisonnement sur les relations suit alors l'approche classique du chaînage avant. Il s'agit d'enrichir les graphes par des informations implicites contenues dans la base de connaissances $\mathcal{K}_s$. Soit g un graphe conceptuel et r_1 une règle ayant le label $label_{r_1}$. Si g remplit la prémisse de r_1, alors la conclusion de cette dernière peut être ajoutée à g. Comme nous l'avons vu dans la section 4.2.1, l'ajout d'informations en graphes conceptuels correspond à l'application d'une jointure, et est guidé par la règle de *Monotonicité Prudente*. La définition suivante fixe comment s'effectue cet ajout prudent d'informations.

Définition 4.4.19 (Application d'une Règle de Dérivation) *Une règle $label_R$: $G_1 \Rightarrow G_2$ s'applique à un graphe conceptuel g, s'il existe une projection π de G_1 sur g. Le graphe résultat de l'application de cette règle, noté g_{etendu} (pour graphe étendu), est construit à partir de g et G_2 en joignant chaque x_{2i} de G_2 avec $\pi(x_{1i})$, où $\pi(x_{1i})$ est l'image du coréférent x_{1i} par π. En termes d'opérateurs de formation de graphes, l'ajout correspond aux étapes suivantes:*

- *Dans G_2 restreindre tous les labels des coréférents x_{2i} aux labels des $\pi(x_{1i})$*
- *Joindre chaque coréférent x_{2i} de G_2 avec $\pi(x_{1i})$ dans g.*

Par exemple, le graphe "graphe dérivé" de la figure 4.24 précédente illustre le résultat de l'application de la règle r_1 de label $label_{r_1}$ au graphe g.

Dans [SM96], il est prouvé qu'un chaînage avant sur les graphes est correct vis-à-vis de la logique de premier ordre, c'est-à-dire si un graphe g_{etendu} est obtenu à partir de g par application de un ou de plusieurs règles de dérivation de graphes, alors nécessairement $g_{etendu} \leq g$. De plus, étant donnée l'incomplétude de l'opérateur ϕ, le chaînage avant n'est complet que dans le cas des graphes conceptuels sous forme normale (voir section 4.1.3).

Par conséquent nos règles de dérivation relationnel $\mathcal{R}_{M_{Rel}}$ s'ajouteront à l'opérateur de projection lors de l'élaboration de la correspondance entre les documents et la requête. Nous verrons dans la section suivante comment un système à base de graphes conceptuels fonctionnera dans ce cas.

4.4.6 Un Modèle Relationnel pour les Graphes Conceptuels

Nous avons vu dans la section 4.2.1 que l'opérateur de projection n'est pas suffisent à lui seul pour la prise en compte des propriétés des relations. En effet, par rapport à notre modèle relationnel du chapitre 3, plusieurs postulats intégrant les connaissances sur les relations ne sont pas modélisés par les décisions de pertinence associées à la projection. Nous avons ainsi tenté, tout au long des sections précédentes, d'introduire des définitions et des méthodes permettant justement de combler les limites de cet opérateur. Dans la suite nous proposons un système logique étendu à base de graphes conceptuels.

Nous avons ainsi présenté dans la section 4.4.4 (page 186) le langage d'indexation relationnelle $\mathcal{LR}$ fondé sur le formalisme des graphes conceptuels. Ce langage comprend un alphabet de base et un ensemble de règles de construction syntaxiques. Si nous analysons ce langage dans le cadre du

modèle relationnel que nous avons proposé, nous pouvons alors l'assimiler au langage d'indexation de la Définition 3.2.1 (page 81). Plus précisément [18], le langage d'indexation relationnelle dans ce cas est confondu avec le langage des relons. En effet, les graphes conceptuels utilisés pour l'indexation étant simples, nous ne pouvons avoir d'imbrication de graphes [19].

Plus concrètement, le relon n'est autre ici qu'un graphe isomorphe à l'un des graphes signatures de $\mathcal{S}_R$. Une fois que le canon de l'application établi, le nombre des relons sera limité. En effet, de part les contraintes sur les signatures, les divers treillis de types et le fait que la dénotation d'un type est forcément fini [20], le langage d'indexation relationnelle associé aux graphes conceptuels est fini.

Toujours par rapport à ce langage d'indexation relationnelle, un graphe index n'est pas forcément normalisé [21] dans notre approche relationnelle. Les règles de construction syntaxiques $\mathcal{R}_{\mathcal{LR}}$ pouvant générer des graphes index contenant le même élément de $Ism(\mathcal{S}_r)$ plusieurs fois.

Cependant, considérons un graphe index extrême contenant au moins une fois tous les éléments de $Ism(\mathcal{S}_r)$. Si un autre élément de $Ism(\mathcal{S}_r)$ est dérivable par les règles $\mathcal{R}_{M_{Rel}}$ à partir de cet index, alors deux cas peuvent se présenter: (i) si cet élément n'est pas un graphe de $Ism(\mathcal{S}_r)$, alors le graphe obtenu ne sera pas bien formé, et ne pourra donc jamais être dérivé par les règles de dérivation sur les graphes, où la canonicité du graphe résultat est toujours assuré. (ii) cet élément est un graphe de $Ism(\mathcal{S}_r)$, il existe donc déjà une ou plusieurs fois dans le graphe index extrême. De ce fait, de part les propriétés de la relation de préférence $\prec_R$ sur les index introduite dans le chapitre précédent (non-réflexivité), il constituera une information redondante. En effet, l'index obtenu est dans ce cas équivalent en termes des graphes conceptuels [22] à celui déjà existant. L'information dérivée ne sera pas ajoutée puisqu'elle contredit la relation de préférence sous-jacente au modèle relationnel proposé dans le chapitre 3.

La relation de préférence est donc fortement liée à la notion d'équivalence des graphes (voir définition 6, page 147). L'ajout d'une information à un graphe index par le biais de l'application d'une ou de plusieurs règles de dérivation relationnelles dans $\mathcal{R}_{M_{Rel}}$ ne sera autorisé que si le graphe conceptuel dérivé n'est pas équivalent au sens des graphes conceptuels à l'index de départ. Nous énonçons alors la propriété suivante:

Propriété 11 (Relation de Préférence Vs. Graphes Équivalents) *Soit $\chi(d)$ un graphe index de $\mathcal{G}$. La conséquence de l'application d'une relation de dérivation relationnelle $\mathcal{R}_{M_{Rel}}$ sur $\chi(d)$ ne sera effective, que si le graphe conceptuel dérivé $\chi'(d)$ n'est pas équivalent à celui du départ: $\chi'(d) \not\equiv \chi'(d)$, i.e., $trans(\chi'(d))$ [23] $\prec_R$ $trans(\chi(d))$. Dans le cas contraire, aucune information ne sera ajoutée à $\chi(d)$.*

Ainsi, en utilisant le théorème 4 du chapitre précédent (page 115), et en conformité avec la propriété 11 ci-dessus, nous pouvons conclure au théorème de décidabilité suivant:

Théorème 17 (Décidabilité) *Un système logique à base de graphes conceptuels et bâti sur une approche relationnelle est décidable.*

18. Voir également la section 4.2.1.
19. Il n'est pas possible d'avoir des relons de relons.
20. Par conséquent l'ensemble des marqueurs sera fini.
21. Voir définition dans la section 4.1.3.
22. Il le sera également en termes de la théorie des situations.
23. "*trans*" est la fonction de translation des graphes conceptuels en situations (voir section 4.2.1, page 152).

Par ailleurs, afin de réaliser ce système logique intégrant un raisonnement sur les relations, nous devons spécifier notre langage de dérivation de pertinence. Dans le chapitre 3, ce langage s'articule autour du langage d'indexation relationnelle tout en intégrant les diverses relations d'endiguement connues par le système (Définition 3.4.2, page 108). En termes de graphes conceptuels, il s'agit d'inclure les connaissances $\mathcal{K}_s$ dans le langage de dérivation. Les dérivations de pertinence, quant à elles, sont menées en ayant pour support les règles $\mathcal{R}_{M_{Rel}}$. Comparées au système de dérivation relationnel que nous avons proposé dans le chapitre 3, ces règles correspondent à celles établissant l'*Équivalence Logique à Gauche*, l'*Endiguement Élémentaire* et l'*Endiguement avec Retenue*. En effet, elles exploitent les connaissances du domaine en ajoutant aux graphes conceptuels, les informations implicites qu'ils véhiculent. Cet ajout s'effectue en gardant les informations (les sous-graphes) de départ. Ces trois règles ne sont pas supportées par la projection comme nous l'avons vu dans la section 4.2.1(page 152).

Afin de construire un système logique à base de graphes conceptuels selon les lignes directrices du modèle relationnel, il faudra ajouter aux règles $\mathcal{R}_{M_{Rel}}$ toutes celles restantes et mentionnées par ce modèle. En particulier, il s'agit au moins d'ajouter des règles de graphes correspondant à l'axiome de *Réflexivité* et aux deux règles du *Cut* et de la *Monotonicité Prudente*[24]. Ces règles sont heureusement supportées par la projection (voir page 152) à travers les opérateurs de formation de graphes. Pour ce qui est de la règle d'*Endiguement Faible*, nous avons vu dans la section précédente qu'elle peut être supportée par l'introduction d'un opérateur algébrique de restriction de relations.

Enfin, nous savons que les règles de d'*Équivalence*, de *Disjonction* et de *Bouclage* du modèle relationnel sont également supportées par l'opérateur de projection. Ainsi, toutes les décisions de pertinence requises par le modèle relationnel peuvent être maintenant supportées par le modèle des graphes conceptuels. Pour arriver à ce résultat, nous avons proposé de nombreuses extensions aux formalisme de base de Sowa. Ces extensions apparaissent en grisé dans la figure 4.26 suivante.

Cette figure résume les différentes étapes requises pour la construction d'un système relationnel à base de graphes conceptuels. Le premier niveau de la figure décrit la nouvelle ontologie proposée pour le modèle des graphes conceptuels. Cette ontologie n'est plus réduite à une base minimale de graphes canoniques associée à des treillis de types de concepts et de relations, mais regroupe en plus un ensemble d'endiguements $\mathcal{E}$ et un ensemble minimal de connaissances $\mathcal{K}_{s_{Min}}$ permettant de décrire les propriétés des relations du domaine. De plus, c'est l'ensemble des graphes signatures $\mathcal{S}_r$ qui constitue la base canonique minimale. Le fond blanc sur lequel est illustré cet ensemble mentionne le fait qu'il s'agit plus d'une notion raffinée que d'une extension au formalisme de base. Les relations et leurs signatures sont en effet suffisantes pour la construction d'un langage d'indexation relationnelle à base de graphes conceptuels. Lors de l'établissement des décisions de pertinence, le langage d'indexation relationnelle sera complété par l'ensemble des relations d'endiguement contenues dans $\mathcal{K}_s$. Ces dernières sont obtenues à partir de l'ensemble $\mathcal{K}_{s_{Min}}$ en appliquant les règles $\mathcal{R}_{R_{Spe}}$. Enfin l'indexation des documents à proprement parler se fait par le biais de la fonction de description $\chi()$, en partant du langage d'indexation relationnelle $\mathcal{LR}$. Le résultat est l'ensemble $\mathcal{G}$ des graphes index. Chaque graphe index est associé à un document du corpus.

Il est alors facile à ce stade de faire l'analogie avec le modèle logique de Rijsbergen. En effet, nous savons que dans ce modèle le processus de dérivation d'une requête par rapport au corpus est dépendant d'un ensemble de connaissances. Si l'on pose l'hypothèse que la projection associée à l'ensemble des règles $\mathcal{R}_{M_{Rel}}$ assure le processus de dérivation et que l'ensemble des connaissances

24. Nous savons en effet que tout système logique devra inclure une forme de monotonicité en plus de l'axiome de réflexivité et de la règle du cut.

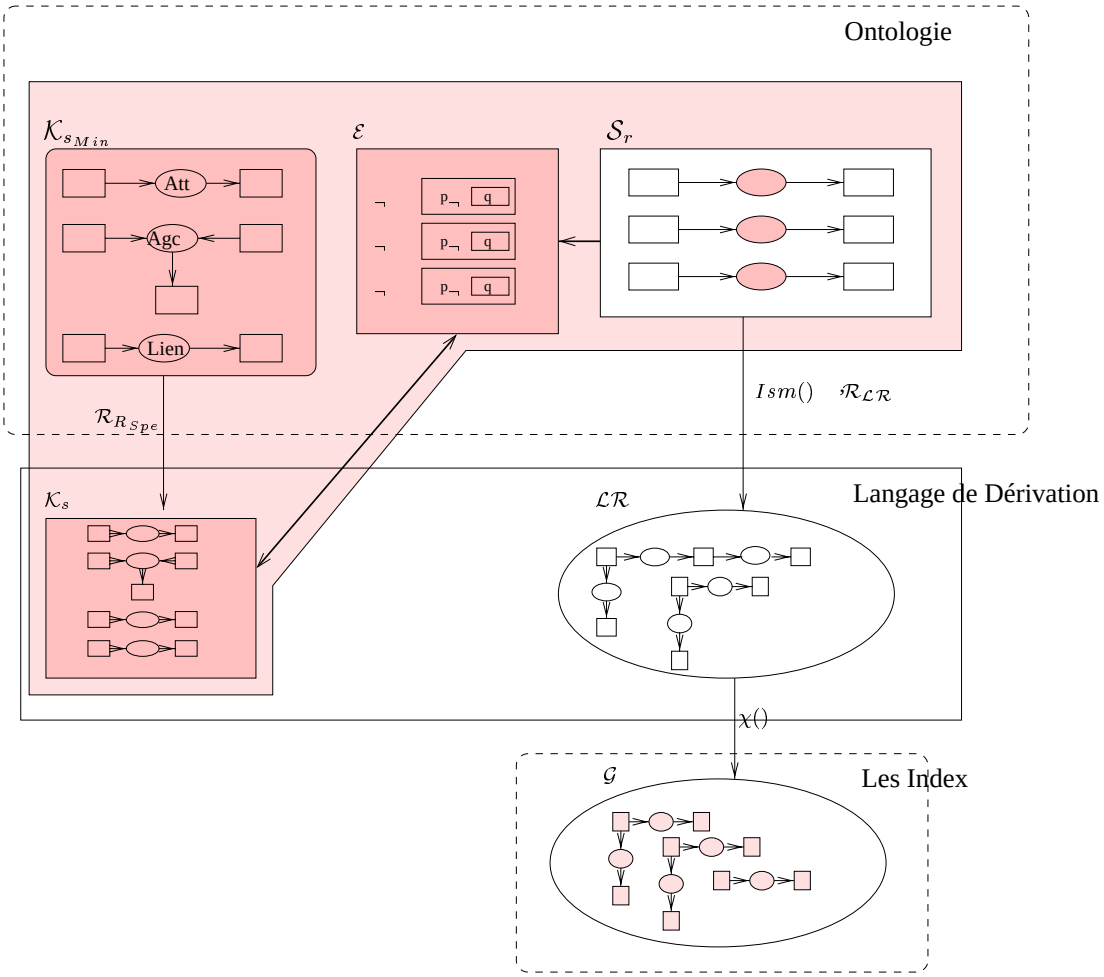


Figure 4.26. Modèle étendu des Graphes Conceptuels

n'est autre que la partie en grisé foncé de la figure précédente, il sera facile d'établir le système logique à base de graphes conceptuels de la figure 4.27.

Reste à savoir comment peut être implanté un processus de dérivation logique à base de graphes conceptuels. Deux démarches sont en effet possibles:

- Appliquer les règles $\mathcal{R}_{M_{Rel}}$ aux documents du corpus avant de lancer l'opérateur de projection. Dans ce cas, tout se passe comme si les graphes du corpus sont étendus par les informations implicites connues du système. Ces informations regroupent principalement les caractéristiques et le comportement des relations. Ces dernières ne pouvant être directement déduites par la projection classique, elles sont ajoutées aux index des documents.
- Appliquer d'abord la projection sur les documents du corpus. Cette dernière divisera ainsi le corpus en deux parties. Les règles $\mathcal{R}_{M_{Rel}}$ seront ensuite appliquées à celle regroupant les documents sur lesquels il n'a pas été possible de projeter la requête.

Le choix d'une stratégie d'implantation dépend de la complexité de l'opérateur de projection

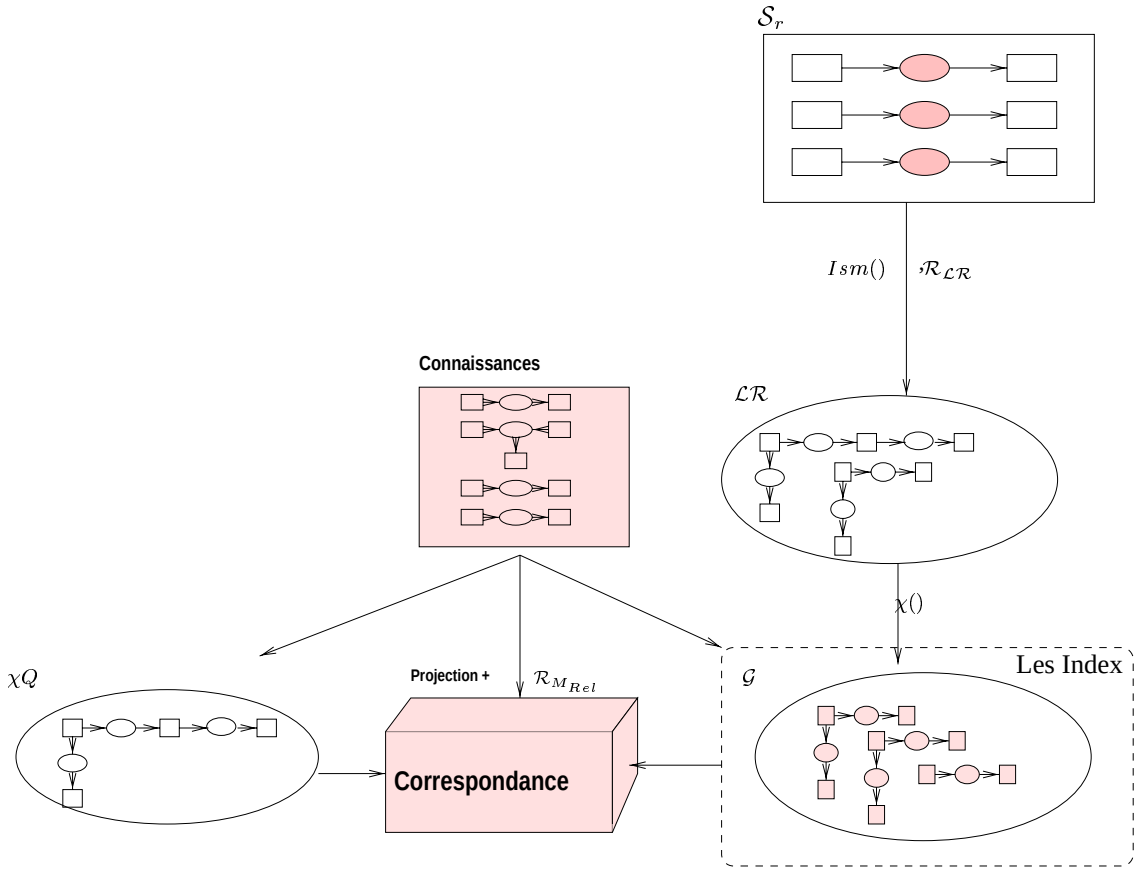


Figure 4.27. Un système logique à base de graphes conceptuels

classique et de celle de l'application des règles de $\mathcal{R}_{M_{Rel}}$. Ainsi, l'opérateur de projection est connu pour avoir une complexité exponentielle [Gui96a] [25]. De plus, la projection est une fonction orientée vers la précision des réponses. De ce fait, très peu de documents contiennent une projection de la requête. Ainsi, la taille de la partie des documents pour lesquels il n'a pas été possible d'appliquer une projection sera sensiblement égale à la taille du corpus. Par conséquent, il est plus intéressant d'adopter une démarche dans laquelle les règles de dérivation $\mathcal{R}_{M_{Rel}}$ sont appliquées en premier lieu à tous les documents du corpus. C'est l'approche que nous avons adopté lors de l'implantation de notre système RELIEF [OP97b].

4.4.7 La Couverture d'un Graphe Conceptuel

Nous introduisons enfin une notion qui nous sera utile par la suite pour la réalisation d'une approche algébrique pour les graphes conceptuels. Il s'agit de la notion de *couverture de graphe*, une notion classique en théorie des graphes [ST81]. Dans le cadre du formalisme des graphes conceptuels, elle a été déjà introduite dans [MC93b, Khe95]. Nous l'adaptons ici par contre à notre propre démarche d'indexation relationnelle.

25. La suite de ce chapitre sera consacrée à l'étude de ce problème de complexité.

Informellement, la couverture d'un graphe g est l'ensemble des sous-graphes connexes de g, tel que chaque noeud et chaque arc de g apparaît dans au moins un de ces sous-graphes. Plus formellement, la définition suivante est introduite dans [MC93b]:

Définition 4.4.20 (Couverture de Graphe) *Un ensemble de sous-graphes* $\{g_1, g_2, \ldots, g_n\}$ *d'un graphe conceptuels* g *constitue une couverture de* g *si, et seulement si, le graphe* $(\cup_i C_i, \cup_i R_i, \cup_i A_i, ord, label)_{1 \leq i \leq n}$ *est égal à* g*, où* C_i*,* R_i *et* A_i *dénotent respectivement les noeuds concepts, les noeuds relations et les arcs de* g_i*.*

Exemple 42 *Considérons la figure 4.28 suivante. l'ensemble des trois graphes* g_1*,* g_2 *et* g_3 *est une couverture du graphe conceptuel* g*.*

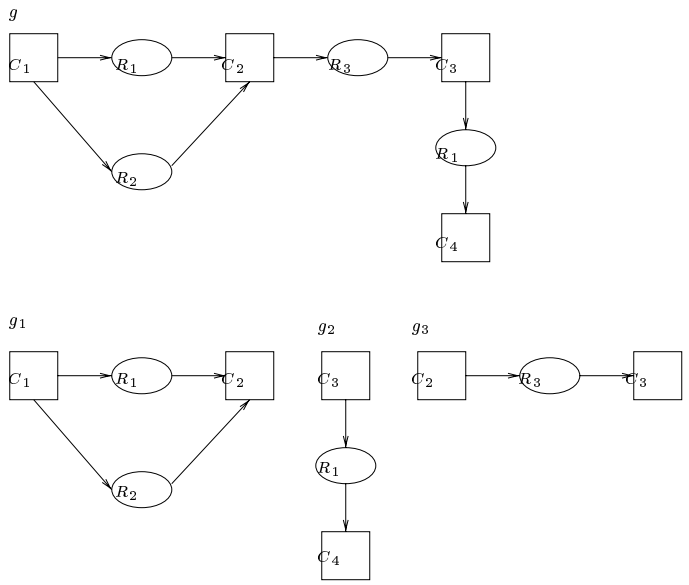


Figure 4.28. L'ensemble $\{g_1, g_2, g_3\}$ constitue une couverture de g

Pour un graphe conceptuel donné, il peut exister plusieurs couvertures possibles. En effet, selon les sous-graphes choisis pour couvrir les noeuds concepts et les noeuds relations du graphe de départ, il peut exister plusieurs façon de le couvrir. Comme dans [Khe95], nous introduisons une notion de couverture par rapport à un langage.

Définition 4.4.21 (Couverture par Rapport à $\mathcal{LR}$) *Soit* g *un graphe index de* $\mathcal{G}$*. Nous appelons par* couverture du graphe g par rapport au langage d'indexation relationnelle $\mathcal{LR}$*, l'ensemble des expressions de* $\mathcal{LR}$ *constituant des sous-graphes de* g*. On définit alors la fonction* $COUVERTURE_{\mathcal{LR}}$*:*

$$COUVERTURE_{\mathcal{LR}} : \mathcal{G} \rightarrow 2^{\mathcal{LR}}$$

où $2^{\mathcal{LR}}$ *désigne l'ensemble des parties de* $\mathcal{LR}$*.*

Une expression de $\mathcal{LR}$ peut être construite à partir des graphes isomorphes aux graphes signatures de $\mathcal{S}_r$ et des règles de construction $\mathcal{R}_{\mathcal{LR}}$ (jointure, simplification et Copie). L'utilisation de la jointure permet donc à une expression de $\mathcal{LR}$ de contenir plusieurs relations. Par conséquent, il existe également plusieurs manières de couvrir un graphe de $\mathcal{G}$ par des expressions du langage d'indexation relationnelle $\mathcal{LR}$. D'un point de vue pratique, cette multiplicité des couvertures que peut avoir un graphe conceptuel n'arrange pas des comparaisons directes (du point de vue de la relation d'ordre $\leq$) entre graphes.

Notre objectif est alors de chercher une technique permettant de couvrir de façon unique un graphe index. Dans notre démarche relationnelle, nous avons considéré qu'un graphe index doit contenir au moins une relation (Définition 8, page 187). Ainsi, la plus petite unité indexable est tout graphe isomorphe à un graphe signature de $\mathcal{S}_r$. L'ensemble des plus petites unités indexables est donc l'union des graphes isomorphes aux graphes signatures, à savoir l'alphabet relationnel $\mathcal{A}_{\mathcal{LR}}$ $= Ism(\mathcal{S}_r)$. Nous pouvons ainsi envisager la couverture d'un graphe conceptuel par rapport aux plus petites unité indexables, c'est-à-dire par rapport à $\mathcal{A}_{\mathcal{LR}}$.

Propriété 12 *Soit g un graphe index de $\mathcal{G}$. Par rapport à l'alphabet du langage d'indexation relationnelle $\mathcal{A}_{\mathcal{LR}} = Ism(\mathcal{S}_r)$, il existe une et une seule couverture possible pour g. Nous notons cette couverture $COUVERTURE_{\mathcal{A}_{\mathcal{LR}}}(g)$, ou simplement $COUVERTURE(g)$, quand le langage d'indexation relationnelle est connu.*

Preuve 16 *Soient $\mathcal{C}_1$ et $\mathcal{C}_2$ deux couvertures différentes d'un graphe h par rapport à $\mathcal{A}_{\mathcal{LR}}$. On peut alors trouver un graphe g_1 de $\mathcal{C}_1$ n'appartenant pas à $\mathcal{C}_2$ ou l'inverse (la démonstration est équivalente aux notations près). Si $g_1 \in \mathcal{C}_1$ alors nécessairement il existe un unique graphe signature $g_{\Sigma_r \in \mathcal{S}_r}$ correspondant à une relation r tel que $g_1 \in Ism(g_{\Sigma_r})$. Comme la relation r est certainement couverte dans $\mathcal{C}_2$, alors il existe $g_2 \in \mathcal{C}_2$ tel que $g_2 \in Ism(g_{\Sigma_r})$. Les graphes de $\mathcal{C}_1$ et $\mathcal{C}_2$ étant tous bien formés (ils appartiennent à $\mathcal{A}_{\mathcal{LR}}$), quand un graphe couvre une relation alors il couvre nécessairement les concepts qui sont rattachés à cette relation dans le graphe. Ceci en respectant la direction des arcs et les labels de concepts. Par conséquent, une relation r apparaissant dans un graphe h ne peut être couverte que par un seul graphe bien formé. Les graphes g_1 et g_2 sont donc identiques. Ainsi, il n'existe pas de graphe g_1 dans $\mathcal{C}_1$ n'ayant pas un identique dans $\mathcal{C}_2$, d'où $\mathcal{C}_1 = \mathcal{C}_2$.*

La figure 4.29 suivante illustre une couverture d'un graphe index par rapport à l'alphabet $\mathcal{A}_{\mathcal{LR}}$. Afin de trouver la seule et unique couverture d'un graphe g par rapport à cet alphabet, nous adoptons l'algorithme qui suit. Il est inspiré d'une approche de vérification de la canonicité[26] d'un

26. Il s'agit de s'assurer qu'un graphe conceptuel donné est canonique ou pas.

graphe présentée dans [MC93b]:

Algorithme 2 (Couverture d'un Graphe)
COUVERTURE*(un graphe index g, l'ensemble $\mathcal{S}_r$)*

// R est l'ensemble des noeuds relations de g
Début
Rel := *R*
COUVERTURE := $\emptyset$
TANT QUE Rel $\neq \emptyset$ *FAIRE*
    soit r $\in$ *Rel*
    POUR TOUT $g_{\Sigma_r} \in \mathcal{S}_r$ *FAIRE*
        ResProj := *PROJECTION*(g_{Σ_r}, *g*)
        SI ResProj $\neq \emptyset$ *ALORS*
        COUVERTURE := COUVERTURE $\cup$ *ResProj*
        Rel := *Rel* - {*r*}
        Exit
        FINSI
    SI r $\in$ *REL ALORS afficher(" g n'est pas un graphe index) FINSI*
    FIN FAIRE
    Retourner COUVERTURE
FIN FAIRE
Fin

L'algorithme donne pour chaque graphe index $g = (C, R, A, ord, label)$ de $\mathcal{G}$ sa couverture. La démarche consiste à partir de l'ensemble R des relations du graphe et à faire projeter pour chaque $r \in R$, le graphe signature de r sur g. Les divers résultats possibles[27] de la projection étant tous des éléments de la couverture de g. En effet, une relation peut apparaître plusieurs fois dans un graphe. Il y a ainsi autant d'apparition d'une relation r que de possibilités de projeter son graphe signature g_{Σ_r} sur le graphe dans lequel elle apparaît. Soit donc la fonction $PROJECTION(g_{\Sigma_r}, g)$, la fonction qui retourne les différentes projections possibles du graphe g_{Σ_r} sur g. Cette fonction retourne l'ensemble vide $\emptyset$ quand il n'est pas possible d'appliquer une telle projection.

La complexité de cet algorithme dépend de la fonction $PROJECTION(g_{\Sigma_r}, r, g)$. En effet, la complexité de cet algorithme est $|R| \times |\mathcal{S}_r| \times O(PROJECTION)$, où $|E|$ désigne la cardinalité de l'ensemble E et $O(PROJECTION)$ désigne la complexité de la fonction $PROJECTION$. Malgré de nombreux travaux ayant trait à l'étude de la complexité de la projection dans le formalisme des graphes conceptuels [Mug93, MC93a, PC95, LB94, Gui96a], la classe de complexité de cette dernière n'est pas encore exactement connue. Tout au plus, certains algorithmes polynomiaux ont été proposés pour des graphes conceptuels de type particuliers (par exemple le cas où les graphes correspondent à des arbres [Mug93]). Cependant, à l'heure actuelle, il n'existe pas d'algorithme polynomial traitant de la projection d'une façon générale sur n'importe quel type de graphe conceptuel. La complexité de la recherche de la couverture d'un graphe n'est donc pas polynomiale. Toutefois, à chaque fois que l'algorithme de projection est polynomial, la fonction $COUVERTURE$ l'est également.

Dans la suite, nous proposerons un algorithme polynomial pour la projection. Cet algorithme fonctionne pour tout type de graphe conceptuel simple et a l'importante caractéristique d'inclure les

27. Nous rappelons qu'il peut exister plusieurs résultats possibles à l'opérateur de projection (voir page 145).

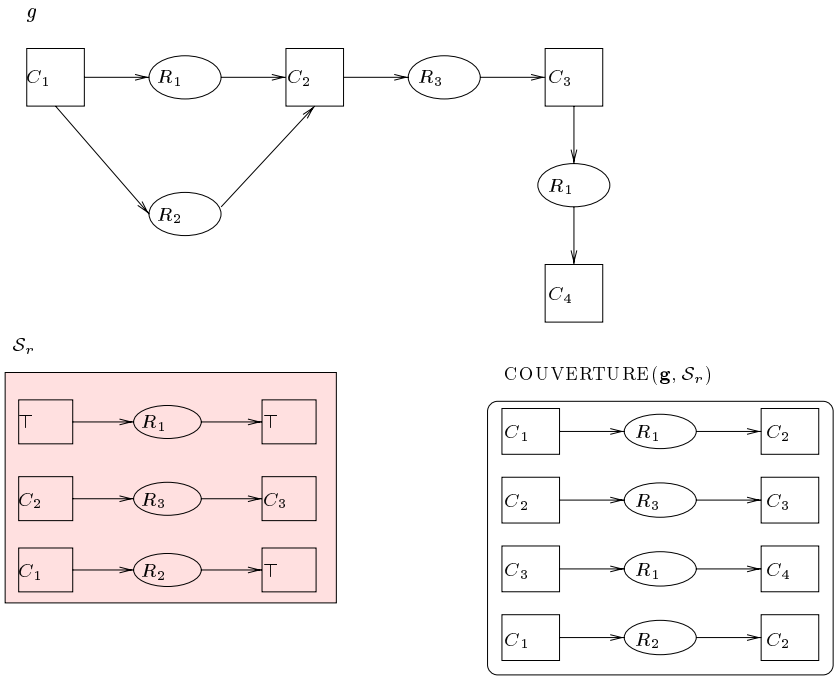


Figure 4.29. Couverture relationnelle du graphe index g

raisonnements sur les relations. Pour la définition de cet algorithme, la recherche de la couverture des graphes index est une tâche primordiale. Le processus d'indexation des images que nous avons adopté dans notre système RELIEF engendre ainsi de façon directe les couvertures qui leur sont associées par rapport à $\mathcal{A}_{\mathcal{LR}}$. Pour la définition de ce nouveau algorithme, nous utilisons une extension de l'interprétation logique associée aux graphes conceptuels. La section suivante présente cette extension.

4.5 Une Interprétation Logique pour les CGs

J. Sowa définit un opérateur ϕ qui transforme chaque élément du modèle des graphes conceptuels en un élément de la logique du premier ordre. Cette interprétation donne une sémantique particulière au marqueur générique $\star$. Dans la page 142 de [Sow84], l'affirmation suivante est énoncée:

> Any formula in first-order logic can be expressed with simply nested contexts and lines of identity.

Un peu plus loin, à la page 173 Sowa affirme dans un théorème que les règles d'inférence des graphes conceptuels sont complètes (il est possible de faire dériver toutes les tautologies). Cependant, dans [Wer95], Wermelinger démontre que ϕ est loin de vérifier l'affirmation et que le

théorème, par ailleurs non prouvé par Sowa, est faux. Wermelinger énonce alors le résultat suivant:

> The Sowa's original definition of the mapping is incomplete, incorrect, inconsistent, and unintuitive, and the proof system is incomplete too.

Le sens de dérivation des graphes étant opposé à celui des formules logiques associées, l'opérateur ϕ n'est pas assez intuitif: quand un graphe g_1 est dérivé à partir d'un graphe g_2 par l'application d'un ou de plusieurs opérateurs de construction de graphes, alors la formule logique associée à g_1 implique celle relative à g_2 [Sow84].

L'interprétation existentielle donnée par ϕ au marqueur générique $\star$ explique ce constat. Dans la section 4.1.3, page 146, nous avons ainsi contesté cette interprétation puisqu'elle confond le quantificateur existentiel avec le quantificateur universel. Les sections 4.3.3 (page 167) et 4.4.2 (page 183) de ce chapitre étaient dédiées à rectifier les imprécisions du modèle de base de Sowa en spécifiant la sémantique des marqueurs. Dans la suite nous établirons d'abord un nouveau opérateur ϕ pour les graphes conceptuels. Ce dernier est censé résoudre les problèmes posés par l'opérateur originel introduit par Sowa. En particulier nous donnons une interprétation universelle au marqueur $\star$. Nous montrons ensuite que l'opérateur de Sowa n'est autre qu'un cas particulier de ce que devrait être la sémantique d'un graphe conceptuel en termes de la logique de premier ordre. Enfin, nous précisons encore la sémantique associée aux graphes conceptuels, en utilisant la technique des témoins, une technique de la logique de premier ordre qui permettra par la suite de réaliser une implantation efficace du formalisme.

4.5.1 Extension Universelle de l'Opérateur Φ

Commençons tout d'abord par une discussion préliminaire relative à l'utilisation des graphes conceptuels. Prenons donc une pièce élémentaire $[C_1 : \star]\rightarrow(Rel)\rightarrow[C_2 : \star]$, où C_1 et C_2 sont des concepts et *Rel* une relation binaire qui les lie ensemble. La fonction d'ordre *ord* introduite dans le canon détermine lequel des concepts est le premier argument de la relation.

Il est utile de spécifier à ce niveau que les relations lient des individus du domaine de discours. La raison est qu'elles sont basées sur ce qu'on perçoit dans la réalité. Comme la perception est souvent liée à des individus ou à des instanciations, il est logique de considérer que les relations sur les types ou les concepts ne sont au fait que des généralisations des relations sur les individus [vdB93].

Nous n'observons pas le fait que le type LIVRE a pour propriété qu'il est composé de PAGES, mais nous observons plusieurs fois au niveau des individus que *chaque livre que nous voyons, admet des pages*. L'énoncé *les livres contiennent des pages* représenté par le graphe [LIVRE]→(*Compose*)→[PAGE] est une généralisation par induction effectuée par les humains. Cette distinction comme le signale Brachman [Bra83] pose un certain nombre de problèmes puisqu'elle module la compréhension du sens de la relation:

> "But understanding what is on either side of the link is also the key to understanding the import of the link"

Pour résumer, typiquement les relations s'appliquent à des individus. Les relations entre types sont simplement considérées comme des relations dérivées. La relation *Rel* de la pièce élémentaire de graphe ci-dessus n'est donc pas traitée comme une propriété de second ordre entre concepts mais

comme une relation de premier ordre entre des individus particuliers du domaine de discours. Ces individus doivent être des éléments des dénotations des types de concepts C_1 et C_2 (ou conformes à ces types de concepts). Ainsi, une des remarques concernant la lecture d'un graphe conceptuel est que les variables et les quantificateurs pour les individus sont implicitement introduits dans le graphe. Dans ce qui suit nous avons à élaborer d'autres hypothèses pour la manipulation des variables dans les graphes conceptuels. Notre but étant de définir une fonction de translation ϕ entre les graphes conceptuels et la logique du premier ordre.

a) La Notion d'Arch

Appelons *arch*[28], la plus petite unité d'information $[C_1 : \star] \rightarrow (Rel) \rightarrow [C_2 : \star]$ d'un graphe conceptuel. Elle correspond à une structure sémantique contenant une seule relation. L'arch est ainsi un élément du vocabulaire terminal $\mathcal{A}_{LR}$, c'est-à-dire un élément de l'ensemble des graphes isomorphes aux graphes signatures du canon. Par souci de simplification des formules, nous considérons dans ce qui suit le cas des relations binaires. Cependant, il est facile de généraliser l'approche à des relations n-aires arbitraires.

Le marqueur générique $\star$ est plus grand que tous le reste des marqueurs. Il dénote donc tous les individus du domaine. Ainsi, pour tous les individus i_1 et i_2 du domaine de discours, si ces individus sont en relation de conformité avec C_1 et C_2 respectivement, alors nécessairement ils doivent appartenir à la dénotation de la relation *Rel* qui les lie. En effet comme nous l'avons souligné, une relation dans un graphe conceptuel est traitée comme une propriété de premier ordre sur les individus du domaine, pourvu qu'ils soient en conformité avec leur type de concept associé. Nous proposons alors la transformation logique suivante pour un arch donné:

Définition 4.5.1 (La formule de graphe) *La formule de graphe est une conjonction logique d'un ensemble d'archs, chacun d'eux ayant la forme suivante:*

$$\Phi([C_1 : \star] \rightarrow (Rel) \rightarrow [C_2 : \star]) = \forall x_1 \forall x_2 (C_1(x_1) \wedge C_2(x_2) \rightarrow \mathrm{Rel}(x_1, x_2)) \tag{4.1}$$

Cette interprétation comme on le verra dans la section 4.5.3 est plus générale que celle donnée par Sowa. Elle est aussi plus intuitive et donne une signification plus adéquate au marqueur générique $\star$.

b) La Translation d'un Graphe Conceptuel

Supposons que nous avons deux ou plusieurs graphes conceptuels et que nous voulons définir un opérateur permettant de les combiner logiquement. Il est naturel de considérer dans ce cas que l'union des graphes est une conjonction logique de leurs formules de graphes associées. Cependant en logique, afin de substituer facilement les individus à des variables, il est important que la *formule de graphe* ait une forme prenexe. Une *formule prenexe* est une formule dans laquelle tous les

28. Ce mot est emprunté de l'anglais. Il est différent du mot français "arche".

quantificateurs sont déclarés au début de cette formule. Nous nous basons pour cela sur l'hypothèse suivante:

Hypothèse 9 (Renommée de Variables) *Chaque nom de concept dans la formule de graphe est uniquement associé à un ensemble fini de variables. Soient* $C_1, C_2, \ldots, C_n, \ldots$, *l'énumération des noms de concepts et* $x_1, x_1^1, \ldots, x_1^{i_1}, x_2, \ldots, x_2^{i_2}, \ldots, x_n, \ldots, x_n^{i_n}, \ldots$ *un ensemble de variables. Si* C_i *apparaît dans un graphe conceptuel alors uniquement* $C_i(x_i^j)$ *pour un* j *donné peut apparaître dans la formule de graphe. Si* C_i *apparaît* k *fois dans un certain arch* A *alors il apparaîtra avec* k *variables* $x_i, \ldots, x_i^{k-1}$ *dans la formule de graphe.*

Muni de cet hypothèse, l'opération qui consiste à faire passer les quantificateurs au début des formules est alors correcte et complète vis-à-vis de la logique de premier ordre. Pour prouver ce fait, le théorème qui suit est utilisé [Men64]:

Théorème 18 (Mendelson 64) *$(\forall x_i \phi(x_i)) \rightarrow \phi(x_i))$ est vrai quand la variable x_i est libre dans ϕ.*

Ce théorème est un cas particulier de l'axiome logique ci-dessous cité, où t est un terme libre dans ϕ pour x.

$$(\forall x \phi(x)) \rightarrow \phi(t) \tag{4.2}$$

Ce dernier signifie qu'une fois la substitution de la formule t à x réalisée, les occurrences des variables dans t doivent être libres, c'est-à-dire qu'elles ne tombent pas sous la portée d'aucun quantificateur, devenant ainsi des variables liées.

Si la variable x_i est utilisée dans deux archs différents $\forall x_i \forall \vec{y} \phi(x_i, \vec{y}) \wedge \forall x_i \forall \vec{z} \phi'(x_i, \vec{z})$, alors x_i est libre dans la formule de graphe $\forall \vec{y} \phi(x_i, \vec{y}) \wedge \forall \vec{z} \phi'(x_i, \vec{z})$. De part l'hypothèse 9, x_i ne peut pas être lié à un quantificateur et ses occurrences sont donc toutes libres. Nous pouvons ainsi appliquer une généralisation sur la variable x_i et obtenir ainsi la formule valide suivante: $\forall x_i (\forall \vec{y} \phi(x_i, \vec{y}) \wedge \forall \vec{z} \phi'(x_i, \vec{z}))$. Par une simple récurrence sur les variables, nous pouvons faire passer tous les quantificateurs au début de la formule des graphes.

Reste maintenant à prouver que la formule prenexe obtenue de la façon ci-dessus est équivalente à la formule du départ. Supposons pour cela que nous sommes en présence d'un graphe conceptuel contenant deux archs ayant un concept $[C_1]$ en commun. Supposons ainsi qu'il admet pour formule prenexe l'expression

$$\forall x_1 \forall x_2 \forall x_3 (C_1(x_1) \wedge C_2(x_2) \rightarrow Rel_1(x_1, x_2)) \wedge (C_1(x_1) \wedge C_3(x_3) \rightarrow Rel_2(x_1, x_3))$$

Les variables x_1, x_2 et x_3 étant libres dans $(C_1(x_1) \wedge C_2(x_2) \rightarrow Rel_1(x_1, x_2)) \wedge (C_1(x_1) \wedge C_3(x_3) \rightarrow Rel_2(x_1, x_3))$, cette dernière peut être dérivée par l'application du théorème 18:

$$(C_1(x_1) \wedge C_2(x_2) \rightarrow Rel_1(x_1, x_2)) \wedge (C_1(x_1) \wedge C_3(x_3) \rightarrow Rel_2(x_1, x_3)) \tag{4.3}$$

Par l'application d'une tautologie de la logique de premier ordre, les deux expressions logiques suivantes peuvent être dérivées à partir de (4.3):

$$(C_1(x_1) \wedge C_2(x_2) \rightarrow Rel_1(x_1, x_2)) \tag{4.4}$$
$$(C_1(x_1) \wedge C_3(x_3) \rightarrow Rel_2(x_1, x_3)) \tag{4.5}$$

Les occurrences des variables x_i étant toutes libres dans (4.4) et (4.5), nous pouvons appliquer une généralisation à ces expressions logiques pour dériver $\forall x_1 \forall x_2 (C_1(x_1) \wedge C_2(x_2) \rightarrow Rel_1(x_1, x_2)) \wedge \forall x_1 \forall x_3 (C_1(x_1) \wedge C_3(x_3) \rightarrow Rel_2(x_1, x_3))$. Cette dernière formule constitue la formule de graphe. L'équivalence entre la forme normale prenexe et la formule de graphe peut ainsi être prouvée.

Il est donc toujours possible de mettre la formule de graphe sous une forme prenexe équivalente. Le corollaire suivant est énoncé:

Corollaire 3 (Forme Prenexe) *Pour toute formule $\phi(g)$ associée à un graphe conceptuel g, il existe une forme normale prenexe $\phi'(g)$ telle que $\phi(g) \equiv \phi'(g)$*

L'un des intérêts de cette forme prenexe est de faciliter les dérivations sur les formules de graphes associées en exhibant clairement leurs conditions de vérité.

4.5.2 Adéquation et Complétude de ϕ

Nous avons présenté dans la section précédente un nouveau opérateur de translation pour les graphes conceptuels. Chaque graphe conceptuel se voit ainsi assigné une formule de graphe. Il reste à vérifier si notre fonction de translation est adéquate et complète par rapport aux dérivations sur les graphes conceptuels. Une notion fondamentale pour tout raisonnement concernant les graphes conceptuels est la relation d'ordre $\leq$. Elle est engendrée par les opérateurs canoniques de formation de graphes. Ainsi, si un graphe g est une spécialisation d'un graphe h ($g \leq h$), alors nous pouvons passer de h à g par une séquence de ces opérations canoniques. De plus Sowa montre que si $g \leq h$ alors la formule logique associée à g implique celle relative à h. Comme nous l'avons vu au début de la section 4.5, la transformation logique définie par Sowa [Sow84] n'est pas intuitive, puisque les implications logiques vont dans le sens contraire des dérivations sur les graphes (voir figure 4.30).

$$\begin{array}{ccccccccc} h & \vdash_{GC} & h_1 & \vdash_{GC} & h_2 & \vdash_{GC} & \ldots & \vdash_{GC} & g \\ \phi(h) & \longleftrightarrow & \phi(h_1) & \longleftrightarrow & \phi(h_2) & \longleftrightarrow & \ldots & \longleftrightarrow & \phi(g) \end{array}$$

Figure 4.30. Le sens des implications est opposé à celui des dérivations

Nous verrons dans la suite que le nouveau opérateur de translation ϕ proposé pour les graphes conceptuels est plus intuitif, et est adéquat par rapport aux dérivations sur les graphes conceptuels. Pour cela nous reprenons une à une les opérateurs de formation de graphes et nous essayons de définir quel type de transformation logique peut leur correspondre.

a) L'Interprétation des Opérateurs Canoniques

Les opérateurs de formation de graphes canoniques ont pour but de construire des graphes plus complexes à partir d'autres graphes. Ils permettent ainsi de spécialiser un graphe conceptuel de départ.

La copie: La copie d'un graphe conceptuel est un graphe conceptuel. Dans notre transformation, cet opérateur a simplement pour conséquence logique la duplication de la formule associée au graphe de départ (voir figure 4.31). Cette opération est valide puisqu'elle correspond à la *loi de l'idempotence* qui constitue une tautologie de la logique de premier ordre:

$$\varphi \leftrightarrow \varphi \wedge \varphi$$

La copie n'est autre qu'un moyen préparant les graphes conceptuels à une éventuelle opération de jointure. Elle n'est autre qu'une manière graphique exprimant l'opération de renommage et d'unification de variables dans la logique de premier ordre:

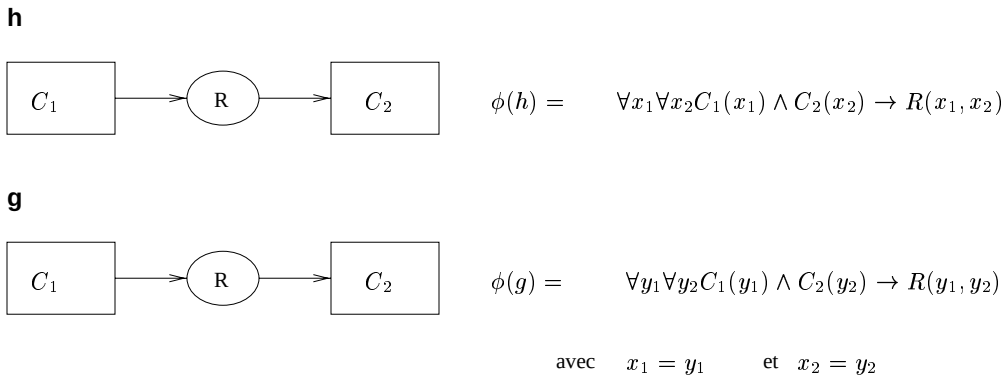


Figure 4.31. Copie d'un arch

L'utilisation de *loi de l'idempotence* permet de prouver la validité de l'opération de copie: si $\vdash_{copie} g \leq h$ alors $\models_{\phi} \phi(h) \rightarrow \phi(g)$. En particulier nous avons ici, $\phi(h) \equiv \phi(g)$.

La restriction: Le graphe résultant d'une opération de restriction est obtenu par remplacement du label d'un concept par un label plus spécifique. Il s'agit de remplacer un type de concept par un de ses sous-types, ou d'un marqueur générique par un marqueur individuel, dans le cas où cela ne viole pas la relation de conformité.

Dans le cas de la restriction d'un type de concept C_1 vers un sous-type S_1 (voir figure 4.32), la formule suivante est valide en logique de premier ordre:

$$\forall x_i (C_1(x_i) \rightarrow \varphi(x_i)) \rightarrow \forall x_j (S_1(x_j) \rightarrow \varphi(x_j)) \tag{4.6}$$

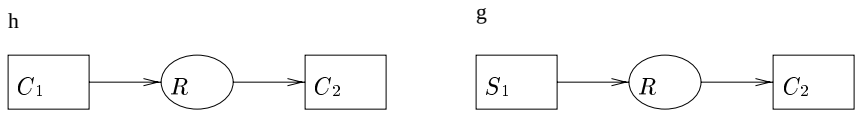


Figure 4.32. La restriction d'un graphe conceptuel ($S_1 \leq C_1$)

En d'autres termes, s'il est vrai que pour tous les individus x_i de la dénotation de C_1, la formule $\varphi(x_i)$ peut être dérivée, alors cette formule peut être dérivée à partir d'un sous-ensemble de la dénotation de C_1, à savoir les éléments de S_1. En effet, si $S_1 \leq C_1$ alors tous les éléments de S_1 sont aussi des éléments de C_1, c'est-à-dire nous avons nécessairement $\forall x S_1(x) \rightarrow C_1(x)$.

La formule logique associée au graphe h de la figure 4.32 étant $\forall x_1 \forall x_2 (C_1(x_1) \wedge C_2(x_2) \rightarrow R(x_1, x_2))$, l'utilisation de l'implication (4.6) permet de faire dériver l'expression $\forall x_j \forall x_2 (S_1(x_j) \wedge C_2(x_2) \rightarrow R(x_j, x_2))$, correspondant au formule du graphe g [29].

De la même façon, s'il s'agit plutôt d'ajouter un marqueur individuel m à un type de concept C_1, l'expression suivante est valide dans le cas où m est conforme au type de concept C_1, c'est-à-dire quand m appartient à la dénotation de C_1:

$$\forall x_i (C_1(x_i) \rightarrow \varphi(x_i)) \rightarrow (C_1(m) \rightarrow \varphi(m)) \tag{4.7}$$

Si pour tous les éléments x_i de la dénotation de C_1, il est possible de dériver $\varphi(x_i)$, alors cette formule est valide pour un élément particulier de cette dénotation. Ce dernier étant conforme au type de concept C_1. La relation de conformité étant respectée lors de la dérivation des graphes, il est facile d'établir que l'expression $\forall x_1 \forall x_2 (C_1(x_1) \wedge C_2(x_2) \rightarrow R(x_1, x_2))$ dérive $\forall x_j (C_1(m) \wedge C_2(x_j) \rightarrow R(m, x_j))$

Dans les deux cas de restriction possibles, l'opérateur de restriction est valide: si $\vdash_{restriction} g \leq h$ alors $\models_\phi \phi(h) \rightarrow \phi(g)$.

N.B: Il est à noter que l'opérateur d'expansion d'une relation R' vers une relation R est également correcte. En effet, puisque si $\forall x \ldots \forall y R'(x, y) \rightarrow R(x, y)$ alors nous pouvons immédiatement inférer $\forall x_1 \forall x_2 (C_1(x_1) \wedge C_2(x_2) \rightarrow R(x_1, x_2))$ à partir $\forall x_1 \forall x_2 (C_1(x_1) \wedge C_2(x_2) \rightarrow R'(x_1, x_2))$.

La jointure: Lorsque deux graphes g_1 et g_2 possèdent respectivement un concept $[s]$ et un concept $[t]$ ayant des labels égaux, le graphe g_3 résultant de la fusion de g_1 et g_2 par identification de $[s]$ et $[t]$ est le résultat de l'application d'un opérateur de jointure entre g_1 et g_2 (voir figure 4.33).

29. Cette dérivation logique peut être comprise en faisant l'analogie avec les signatures de relations: si une relation peut lier deux types de concepts C_1 et C_2, alors nécessairement elle peut lier tous les sous-types de C_1 et C_2 respectivement.

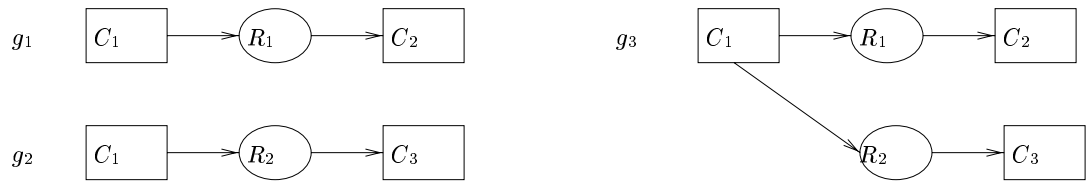


Figure 4.33. Jointure de deux graphes conceptuels

L'application de notre opérateur de translation au graphe g_3 permet de générer la formule prenexe suivante:

$$\forall x_1 \forall x_2 \forall x_3((C_1(x_1) \wedge C_2(x_2) \rightarrow R_1(x_1, x_2)) \wedge (C_1(x_1) \wedge C_3(x_3) \rightarrow R_2(x_1, x_3)))$$

D'après le corollaire 3, cette formule est équivalente à la conjonction des formules logiques associées aux deux archs de g_3:

$$\forall x_1 \forall x_2 (C_1(x_1) \wedge C_2(x_2) \rightarrow R_1(x_1, x_2)) \wedge \forall x_1 \forall x_3 (C_1(x_1) \wedge C_3(x_3) \rightarrow R_2(x_1, x_3))$$

Cette formule est aussi équivalente à l'expression (4.8) en renommant la variable x_1 par x_j:

$$\forall x_1 \forall x_2 (C_1(x_1) \wedge C_2(x_2) \rightarrow R_1(x_1, x_2)) \wedge \forall x_j \forall x_3 (C_1(x_j) \wedge C_3(x_3) \rightarrow R_2(x_j, x_3)) \quad (4.8)$$

Cette seconde formule n'indique pas de lien entre les noeuds des archs de g_3. Ainsi l'opérateur de jointure de graphes sur des concepts génériques de la forme $[C : \star]$ n'est autre ici que l'opération de renommage et de déplacement de variables décrite dans la page 207. Cette opération comme nous l'avons déjà vu dans la preuve du corollaire 3 est valide en logique de premier ordre.

Les deux expressions $\forall x_1 \forall x_2 (C_1(x_1) \wedge C_2(x_2) \rightarrow R_1(x_1, x_2))$ et $\forall x_j \forall x_3\ (C_1(x_j) \wedge C_3(x_3) \rightarrow R_2(x_j, x_3))$ sont valides en logique de premier ordre puisqu'elles correspondent à des graphes conceptuels bien formés. Elles ont donc toujours une interprétation évaluée à vraie. La fonction de translation ϕ fait associer à g_1 la formule de graphe $\phi(g_1) = \forall x_1 \forall x_2\ (C_1(x_1) \wedge C_2(x_2) \rightarrow R_1(x_1, x_2))$, et à g_2 l'expression $\phi(g_2) = \forall x_1 \forall x_3\ (C_1(x_1) \wedge C_3(x_3) \rightarrow R_2(x_1, x_3))$. Dans ce cas, étant donné le fait que le renommage des variables libres est valide en logique de premier ordre, il s'en suit l'implication suivante:

$\forall x_1 \forall x_3 (C_1(x_1) \wedge C_2(x_3) \rightarrow R_1(x_1, x_3)) \rightarrow$

$\forall x_1 \forall x_2 (C_1(x_1) \wedge C_2(x_2) \rightarrow R_1(x_1, x_2)) \wedge \forall x_j \forall x_3 (C_1(x_j) \wedge C_3(x_3) \rightarrow R_2(x_j, x_3))$ pour $\phi(g_1) \rightarrow \phi(g_3)$, et l'implication

$\forall x_1 \forall x_3 (C_1(x_1) \wedge C_3(x_3) \rightarrow R_2(x_1, x_3)) \rightarrow$

$\forall x_1 \forall x_2 (C_1(x_1) \wedge C_2(x_2) \rightarrow R_1(x_1, x_2)) \wedge \forall x_j \forall x_3 (C_1(x_j) \wedge C_3(x_3) \rightarrow R_2(x_j, x_3))$

pour $\phi(g_2) \rightarrow \phi(g_3)$.

L'opérateur de jointure est donc valide: c'est-à-dire si $\vdash_{jointure} g \leq h$ alors $\models_\phi \phi(h) \rightarrow \phi(g)$. Dans la figure 4.33 précédente, on a bien $g_3 \leq g_1$ et $g_3 \leq g_2$ avec $\phi(g_2) \rightarrow \phi(g_3)$ et $\phi(g_1) \rightarrow \phi(g_3)$.

La simplification: La simplification consiste à faire supprimer d'un graphe les relations redondantes. Ainsi, si deux noeuds relations d'un même graphe sont de même type et lient exactement les mêmes concepts du graphe, le graphe résultant de l'opération de simplification est obtenu par la suppression de l'un des deux noeuds relations. L'opération de simplification est illustrée sur la figure 4.34, où le graphe g est obtenu par suppression d'une occurrence de relation double dans h.

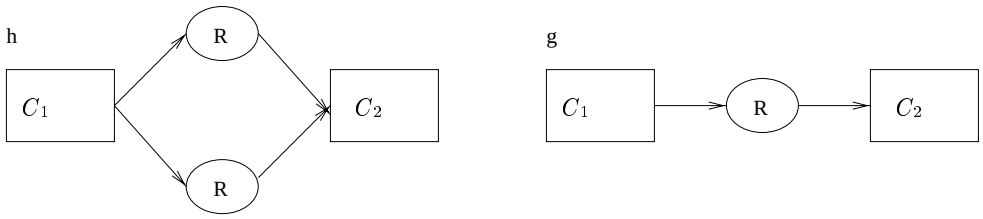


Figure 4.34. Simplification de la relation R

La validité de l'opérateur de simplification par rapport à notre fonction de translation ϕ est facile à prouver. En effet, elle constitue l'opérateur dual de la copie. Sa validité repose, comme pour la copie, sur la loi de l'idempotence ($\varphi \leftrightarrow \varphi \wedge \varphi$). Si nous simplifions c'est que nous sommes en présence de deux ou de plusieurs archs qui sont exactement les mêmes. Leur translation par la fonction ϕ a pour effet de réécrire la même formule d'arch plusieurs fois, en utilisant la conjonction logique $\wedge$. Nous avons alors, si $\vdash_{simplification} g \leq h$ alors $\models_\phi \phi(h) \rightarrow \phi(g)$.

La validité des opérateurs de formation de graphes démontrée tout au long de cette section permet d'énoncer le théorème suivant:

Théorème 19 (Correction de ϕ) *Soit $\mathcal{G}$ l'ensemble de graphes index. La nouvelle fonction de transformation ϕ est correcte, c'est-à-dire, pour tous g, $h \in \mathcal{G}$: si $g \leq h$ alors $\models_\phi \phi(h) \rightarrow \phi(g)$*

Il est intéressant de remarquer que notre fonction de translation ϕ est plus intuitive que celle définie par Sowa. En effet, les dérivations des graphes canoniques vont maintenant de paire avec les dérivations de leur formules logiques associées: si un graphe g est dérivé à partir d'un graphe h par le biais des quatre opérateurs de formation de graphes alors $\phi(h) \rightarrow \phi(g)$ (voir figure 4.35).

$$\begin{array}{ccccccccc} h & \vdash_{GC_{can}} & h_1 & \vdash_{GC_{can}} & h_2 & \vdash_{GC_{can}} & \ldots & \vdash_{GC_{can}} & g \\ \phi(h) & \Leftrightarrow & \phi(h_1) & \Leftrightarrow & \phi(h_2) & \Leftrightarrow & \ldots & \Leftrightarrow & \phi(g) \end{array}$$

Figure 4.35. Un sens de dérivation plus intuitif

b) La Complétude

Dans la section précédente nous avons discuté de l'adéquation des opérateurs de copie, de simplification, de jointure et de restriction par rapport au nouveau opérateur de translation ϕ que nous avons associé aux graphes conceptuels. Nous n'avons pas encore prouvé la complétude de cet opérateur, d'autant plus qu'il est justement facile de vérifier que l'opérateur ϕ n'est pas complet vis-à-vis de la relation de spécialisation $\leq$.

Reprenons ainsi l'exemple de la page 148 (figure 4.4). L'incomplétude de l'opérateur ϕ est facile à vérifier sur cet exemple. En effet, les deux graphes de cette figure ont la même translation en termes de notre nouveau opérateur ϕ:

$$(C_1(m_1) \wedge C_2(m_2) \rightarrow R_1(m_1, m_2)) \wedge (C_1(m_1) \wedge C_2(m_2) \rightarrow R_2(m_1, m_2))$$

Pourtant, si à partir du graphe h de la même figure, il est possible d'obtenir le graphe g ($g \leq h$) en appliquant une jointure sur le concept [C_2: m], il n'en est pas de même pour l'autre sens. Il est en effet impossible avec les seuls opérateurs de formation de graphes de déduire le graphe h à partir du graphe g.

Nous remarquons toutefois que ces deux graphes sont E-isomorphes (Définition 4.4.12, page 186). Il suffit en effet d'éclater le concept [C_2: m_2] dans le graphe g. Ainsi, si l'éclatement d'un noeud concept dans un graphe est autorisé par le formalisme, il est possible de prouver la complétude du formalisme. Nous introduisons alors l'opérateur d'éclatement suivant:

Définition 4.5.2 (Éclatement) *Si un concept [c] est lié à plus d'une relation dans un graphe conceptuel u, alors ce concept peut être dupliqué dans le graphe u en ayant pour arcs un sous-ensemble des arcs auxquels il était lié au départ.*

La formule logique associée à un graphe conceptuel v obtenu par éclatement de concepts est *équivalente* à celle du graphe conceptuel u de départ. En effet, comme la loi d'idempotence et le renommage des variables sont valides en logique de premier ordre, l'éclatement n'est autre qu'une opération de réécriture sur les graphes conceptuels. En particulier nous avons:

$$\text{si} \quad \vdash_{clatement} \quad v \leq u \quad \text{alors} \quad \models_{\phi} \phi(u) \rightarrow \phi(v)$$

Il est à noter que l'application de l'opérateur d'éclatement peut engendrer des graphes non connexes. Ce constat nous permet d'imaginer une forme normale pour un graphe conceptuel. Ainsi, l'opérateur d'éclatement peut être appliqué récursivement sur un graphe conceptuel jusqu'à l'obtention de tous ses archs (cela revient à une recherche de sa couverture). Les duplications d'un même arch peuvent être supprimées en utilisant les opérateurs de jointure et de simplification. Nous définissons alors la *forme standard* ou la *table* d'un graphe conceptuel:

Définition 4.5.3 (Table d'un Graphe Conceptuel) *La table d'un graphe conceptuel u est constituée de l'ensemble des archs obtenus par application récursive de l'opérateur d'éclatement à tous les noeuds concepts de u. Les archs dupliqués (identiques en termes de graphes) étant supprimés.*

Exemple 43 *Les tables associées aux deux graphes g et h de la figure 4.4 (page 148) sont explicitées dans la figure 4.36 suivante.*

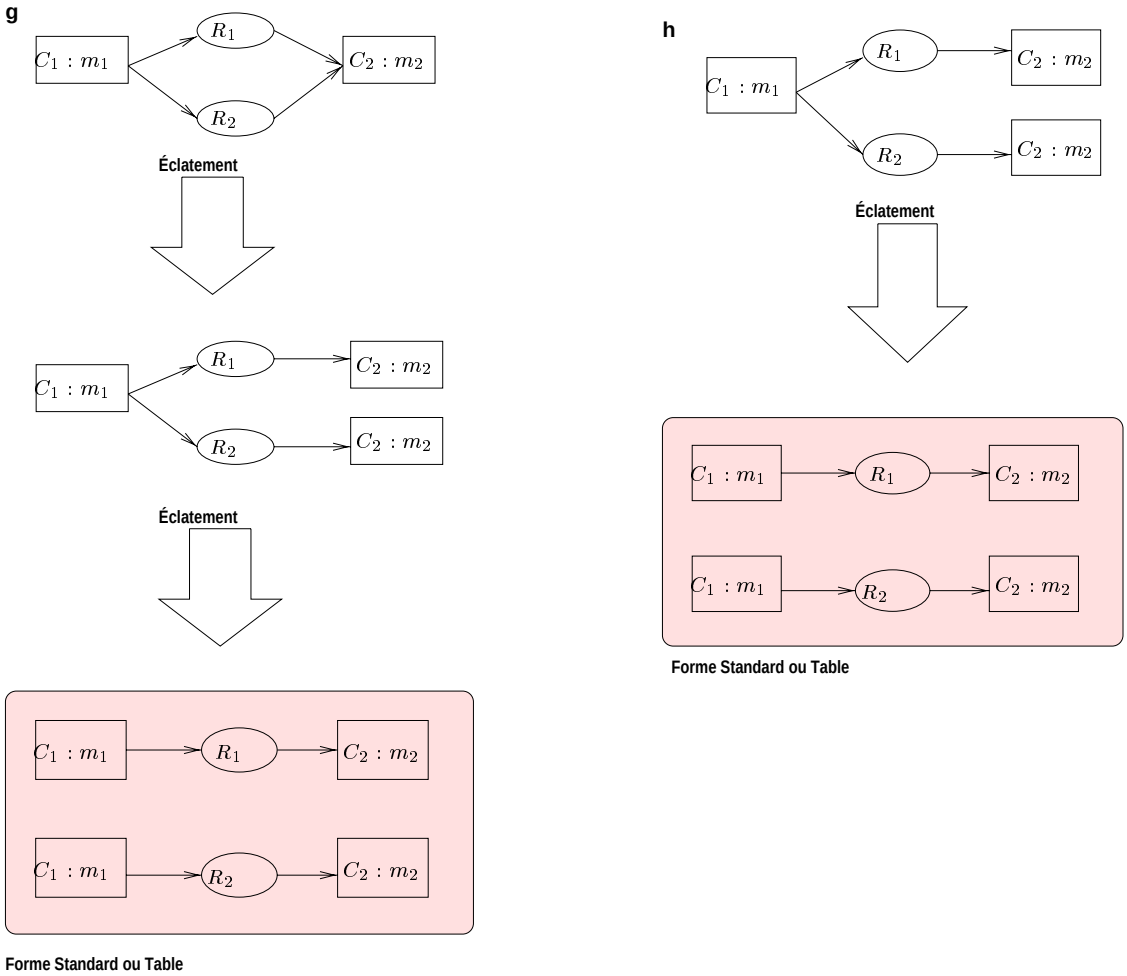


Figure 4.36. Détermination de la forme standard des graphes

Au niveau logique, si l'on admet la convention établie dans l'hypothèse 9, on dénote chaque arch de cette table comme suit: $\forall \star \forall \star\ (C_1(\star) \wedge C_2(\star) \rightarrow Rel_1(\star,\star))$ ou plus brièvement $C_1(\star) \wedge C_2(\star) \rightarrow Rel_1(\star,\star)$ or $(C_1 : \star)(C_2 : \star).Rel_1(\star,\star)$. Le marqueur $\star$ réfère à deux variables différentes dans C_1 et C_2 de part l'hypothèse 9. Si l'arch contient des concepts individuels, alors le symbole $\star$ sera remplacée par un marqueur individuel dans la formule.

Exemple 44 *Considérons les deux graphes conceptuels de la figure 4.37. Ils admettent les deux tables suivantes:*

$$\begin{array}{ll} (HOMME : \star)(FILLE : Mary). & Enft(John, Mary) \\ (FILLE : Mary)(AIMER : \star) & Obj(Mary, \star) \\ (AIMER : \star)(GARCON : Mary) & Agnt(\star, \star) \end{array}$$

pour le graphe $\chi(d)$, et

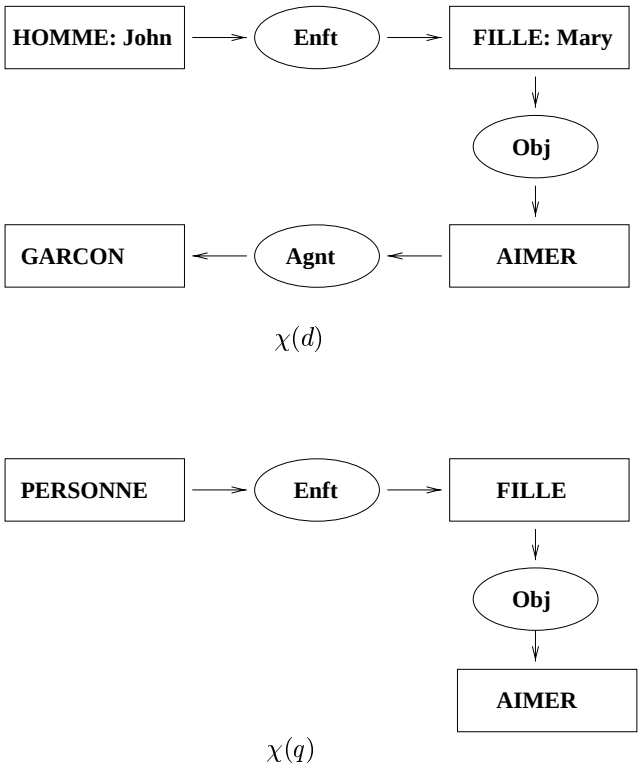


Figure 4.37. Exemple de graphes conceptuels

$(PERSONNE : \star)(FILLE : \star)$. $\quad Enft(\star, \star)$

$(FILLE : \star)(AIMER : \star) \quad Obj(\star, \star)$

pour le graphe $\chi(q)$

Mise sous cette forme, la forme standard du graphe conceptuel est dite *table logique*.

Il est alors possible de prouver que la projection est complète par rapport aux quatre opérateurs de formation de graphes étendu par l'opérateur d'éclatement:

Théorème 20 *g_1 et g_2 sont isomorphes par rapport à la projection, i.e. il existe une projection $\pi_1 : g_1 \Leftrightarrow\!\!\rightarrow g_2$ et une projection $\pi_2 : g_2 \Leftrightarrow\!\!\rightarrow g_1$ si, et seulement si, ils ont la même table.*

Preuve 17 *Considérons les projections $\pi_{s_i} : t_i \Leftrightarrow\!\!\rightarrow g_i$ de la table t_i sur g_i et la projection naturelle $\pi_{t_i} : g_i \Leftrightarrow\!\!\rightarrow t_i$, avec $i = 1, 2$. La composition de deux projections est une projection. Ainsi, $\pi_1 \odot \pi_{s_2} : g_1 \Leftrightarrow\!\!\rightarrow t_2$ est une projection de g_1 sur t_2. Par conséquent $\pi_{s_1} \odot \pi_1 \odot \pi_{t_2} : t_1 \Leftrightarrow\!\!\rightarrow t_2$ est une projection de t_1 sur t_2. Similairement, $\pi_{s_2} \odot \pi_2 \odot \pi_{t_1} : t_2 \Leftrightarrow\!\!\rightarrow t_1$ est une projection de t_2 sur t_1. Ces projections doivent être une-à-une. Dans le cas contraire, il doit exister deux éléments dans la table t_1 qui auront pour image le même élément de t_2. Si on va et vient entre $\pi_{s_2} \odot \pi_2 \odot \pi_{t_1}$ et $\pi_{s_1} \odot \pi_1 \odot \pi_{t_2}$ nous aurons une chaîne infinie d'éléments.*

4.5.3 Discussion sur l'Utilisation des Quantificateurs dans les GCs

Jusqu'à présent, nous nous sommes intéressés au problème de l'interprétation du marqueur générique $\star$. L'interprétation logique que nous proposons pour les graphes conceptuels ne sera complète que si une transformation logique adéquate est donnée aux marqueurs existentiels ou individuels des concepts.

a) Le quantificateur Existentiel

Prenons par exemple l'arch suivant:

$$\boxed{GARCON : \exists} \rightarrow \textcircled{Parle} \rightarrow \boxed{FILLE : \star}$$

Plusieurs interprétations logiques possibles peuvent être données à ce graphe:

$$(\forall y \exists x)(GARCON(x) \wedge FILLE(y) \rightarrow Parle(x, y)) \tag{4.9}$$
$$(\exists x \forall y)(GARCON(x) \wedge FILLE(y) \rightarrow Parle(x, y)) \tag{4.10}$$
$$(\forall y)(\exists x)(GARCON(x) \wedge (FILLE(y) \rightarrow Parle(x, y)) \tag{4.11}$$
$$(\exists x) GARCON(x) \wedge (\forall y)(FILLE(y) \rightarrow Parle(x, y)) \tag{4.12}$$

Les interprétations 4.9 et 4.11 stipulent qu'il existe un garçon qui adresse la parole à toutes les filles. Cela correspond à la notion du Dragueur. Par contre, les interprétations 4.10 et 4.12 affirment que toute fille admet un Copain.

Les interprétations 4.11 et 4.12 ont, en plus, la particularité de tenir compte de la relation de conformité $Conf$. Elles ne seront évaluées à vrai, que si l'individu x est un élément de la dénotation du type de concept GARCON. Cette condition ne sera vérifiée que si la variable x est choisie parmi la liste des marqueurs individuels définie par la relation de conformité du canon. La transformation 4.11 est difficile à gérer. En effet, elle établie une dépendance fonctionnelle entre les individus x et y tel que $x = f(y)$. Seule l'interprétation 4.12, considérant la relation de conformité et plus facile à administrer, semble ainsi être utile pour notre formalisation. Nous étendons l'opérateur de transformation ϕ aux marqueurs existentiels de la façon suivante:

$$\Phi([C_1 : \exists] \rightarrow (Rel) \rightarrow [C_2 : \star]) = \exists x_1 (C_1(x_1) \wedge \forall x_2 (C_2(x_2) \rightarrow Rel(x_1, x_2)) \tag{4.13}$$
$$\Phi([C_1 : \exists] \rightarrow (Rel) \rightarrow [C_2 : \exists]) = \exists x_1 \exists x_2 (C_1(x_1) \wedge C_2(x_2) \wedge Rel(x_1, x_2)) \tag{4.14}$$

La dernière interprétation 4.14 correspond bien à celle donnée par Sowa. Elle confirme le fait que Sowa adopte une interprétation existentielle pour le marqueur générique $\star$. Notre nouvel opérateur ϕ englobe bien la vue logique de Sowa, tout en faisant une nette distinction entre le marqueur générique $\star$ et le marqueur existentiel $\exists$. Ce nouveau opérateur ϕ étend donc l'interprétation de Sowa par un quantificateur universel.

Par analogie avec le marqueur existentiel "$\exists$", si l'arch contient un marqueur individuel a, l'interprétation associée est la suivante:

$$\Phi([C_1 : a] \rightarrow (Rel) \rightarrow [C_2 : \star]) = C_1(a) \wedge \forall x_2(C_2(x_2) \rightarrow Rel(a, x_2)) \tag{4.15}$$

Quand le marqueur individuel a est en relation de conformité avec le type de concept C_1, l'expression $\forall x_2(C_2(x_2) \rightarrow Rel(a, x_2))$ est logiquement interprétée comme vraie. Ainsi, la restriction de l'arch $[C_1 : \star] \rightarrow (Rel) \rightarrow [C_2 : \star]$ vers l'arch $[C_1 : a] \rightarrow Rel \rightarrow [C_2 : \star]$ est valide si $C_1(a)$ est interprété vrai par la relation de conformité $Conf$.

Cependant, la restriction de l'arch

$$[C_1 : \star] \rightarrow (Rel) \rightarrow [C_2 : \star]$$

vers l'arch

$$[C_1 : \exists] \rightarrow Rel \rightarrow [C_2 : \star]$$

n'est pas valide. Il n'est pas ainsi dit que la variable x_1 correspondant au marqueur existentiel $\exists$ peut être substituée par un individu de la dénotation du type de concept C_1. En effet, rien ne présuppose l'existence d'un élément du domaine de discours w conforme à C_1, et pouvant se substituer à la variable x_1 de façon à rendre l'opération de restriction valide. La notion de témoin nous sera utile pour cet effet. Elle rend valide l'opération de restriction ci-dessus citée.

b) La Notion de Témoin

La méthode des constantes définie dans [CK73] stipule que nous pouvons introduire pour chaque quantificateur existentiel d'une formule logique ϕ, une nouvelle constante w qui témoigne de la validité de cette formule existentielle. Soit par exemple, la formule $\exists x Personne(x) \wedge Ami(x, \text{Jean})$, indiquant qu'*il existe une personne x qui est l'ami de Jean.* Si cette formule est valide, alors il existe nécessairement une personne i du domaine de discours telle que cette personne est l'ami de *Jean*:

$$\exists x Personne(x) \wedge Ami(x, \text{Jean}) \equiv Personne(i) \wedge Ami(i, \text{Jean})$$

où i est un élément du domaine de discours. Cet individu se comporte ainsi comme un *témoin* de la validité de la formule. Il en découle de cette approche la théorème suivant:

Axiome 1 (Axiome des Témoins) *Soit $\mathcal{D}^A$ le domaine de discours. Si une formule logique est de la forme $\exists x \alpha(x)$, alors nous pouvons substituer à toutes les occurrences libres de la variable x dans α, une nouvelle constante unique $w \in \mathcal{D}^A$ tel que:*

$$\exists \alpha(x) \rightarrow \alpha(w)$$

Notons que l'implication inverse, $\alpha(w) \rightarrow \exists\alpha(x)$, est aussi valide en logique de premier ordre. L'axiome des témoins est ainsi ajouté à notre opérateur de translation ϕ.

Expliquons à travers un exemple comment les témoins sont associés aux graphes conceptuels. Considérons ainsi le graphe de la figure 4.38.

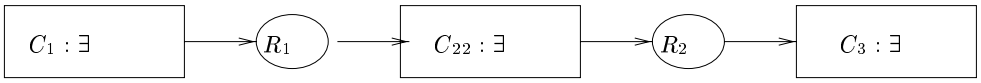


Figure 4.38. Un exemple de graphe conceptuel existentiellement marqué

La formule logique de premier ordre associée à ce graphe par notre fonction ϕ est la suivante:

$$\exists x_1 \exists x_2 \exists x_3 (C_1(x_1) \wedge C_{22}(x_2) \wedge R_1(x_1, x_2) \wedge C_{22}(x_2) \wedge C_3(x_3) \wedge R_2(x_2, x_3))$$

L'utilisation de l'axiome des témoins permet d'associer à cette translation, la formule logique équivalente suivante:

$$C_1(w_1) \wedge C_{22}(w_2) \wedge R_1(w_1, w_2) \wedge C_{22}(w_2) \wedge C_3(w_3) \wedge R_2(w_2, w_3)$$

Cette équivalence n'est valide qu'après l'application de l'axiome 1. C'est-à-dire quand les constantes w_1, w_2, et w_3 sont toutes, des nouveaux éléments dans le domaine de discours. Chaque constante substituant toutes les apparitions libres de la variable x à laquelle elle est associée. Si une formule logique ne contenant pas de témoins peut être dérivée par notre nouvelle fonction de translation ϕ associée aux graphes conceptuels, alors cette formule peut être aussi obtenue par l'opérateur de translation standard de Sowa. Inversement, si une formule sans témoins peut être dérivée avec la fonction de translation standard de Sowa, alors cette formule peut aussi être inférée par notre translation étendue. Il en découle le théorème suivant:

Théorème 21 (Complétude Existentielle) *L'extension des graphes conceptuels par la notion des témoins dans le cas des marqueurs existentiels ne modifie pas l'interprétation originale de Sowa:*

$$\models \psi \equiv \models \psi^w$$

où ψ^w est la formule de graphe associée à un graphe par notre translation étendue par les témoins, tandis que ψ est la formule originale de Sowa.

La preuve de ce théorème est exposée dans la section suivante.

c) Preuve de Complétude

Avant de prouver ce théorème rappelons que l'axiome des témoins substitue à toutes les occurrences libres des variables existentielles, des nouvelles constantes dans le domaine du discours. En d'autres termes, si nous désignons par ψ l'interprétation logique associée à un graphe dont les marqueurs sont existentiellement quantifiés, alors la formule ψ^w correspondant à l'interprétation étendue aux témoins est définie de la façon suivante:

$$\psi^w = S^{x_1 x_2 \ldots x_n}_{w_1 w_2 \ldots w_n} \psi$$

où $S^{x_1 x_2 \ldots x_n}_{\theta_1 \theta_2 \ldots \theta_n} \psi$ est le résultat de la substitution simultanée des termes $\theta_1, \theta_2, ..., \theta_n$ (non nécessairement distinctes) aux variables $x_1, x_2, ..., x_n$, dans la formule ψ (les atomes x_i étant tous distincts). La notion de substitution en général d'un terme à une variable n'est valide que si toutes les occurrences des variables x sont libres dans ψ [Riv89]. On parle de *substitution libre*:

Définition 4.5.4 (Variable Librement Substituable) *y est* librement substituable *à x dans ψ si, et seulement si, x n'a pas d'occurrence libre dans ψ à l'intérieur de la portée d'un quantificateur $\forall y$.*

Cette définition suffit pour notre fragment de logique de premier ordre (où les seuls termes sont les variables et les constantes individuelles). En particulier, dans notre cas θ est une constante individuelle du domaine de discours. Cependant, pour des langages logiques où figurent des termes complexes construits à partir des symboles de fonction, la définition plus générale suivante est considérée:

Définition 4.5.5 (Terme Librement Substituable) *θ est* librement substituable *à x dans ψ si, et seulement si, x n'a pas d'occurrence libre dans ψ à l'intérieur de la portée d'un quantificateur $\forall y$, où y est une variable ayant une occurrence dans θ.*

Les variables existentielles x des formules de graphes étant toutes libres et vérifiant la définition 4.5.4, la substitution correspondant à l'axiome des témoins est valide puisqu'elle est libre. D'une façon générale, la substitution se définie comme suit:

Définition 4.5.6 (Substitution) *Soit L une logique donnée. La notation $S^x_\theta \psi$ est utilisée pour désigner la formule qui résulte de ψ par remplacement de toutes les occurrences de x par le terme θ (qui peut être une variable ou une constante). L'opération de substitution est définie de manière inductive:*

1. *si ψ est une formule atomique, $S^x_\theta \psi$ est l'expression qui résulte du remplacement de toutes les occurrences de x par θ dans ψ*

2. *si $\psi = \neg\chi$ alors:*

   $$S^x_\theta \psi = \neg S^x_\theta \chi$$

   où χ est une formule de la logique L en question.

3. *si* $\psi = \chi\, c\, \sigma$ *alors:*

$$S^x_\theta \psi = S^x_\theta \chi\, c\, S^x_\theta \sigma$$

où c *est un connecteur de la logique* L*, et* χ *et* σ *deux formules dans la dite logique.*

4. *si* $\psi = \forall y \chi$ *alors :*

$$S^x_\theta \psi = \begin{cases} \forall y \chi, & si\ x = y \\ \forall y (S^x_\theta \chi), & si\ x \neq y \end{cases}$$

À partir de la définition 4.5.6, nous pouvons énoncer le corollaire suivant pour les substitutions simultanées [Riv89]:

Corollaire 4 (Substitution Simultanée) *Le résultat d'une substitution simultanée peut toujours être obtenu par une suite de substitutions successives de la manière suivante: soient* $y_1, y_2, ..., y_n$*, les* n *premières variables dans l'ordre alphabétique n'ayant pas d'occurrence dans les termes* $\theta_1, \theta_2, \ldots,$ θ_n *qui sont à introduire dans la formule initiale* ψ*; nous substituons alors successivement* y_1 *à* x_1*,* y_2 *à* x_2*, ... ,* y_n *à* x_n*, puis* θ_1 *à* y_1*,* θ_2 *à* y_2*, ... ,* θ_n *à* y_n*.*

L'axiome des témoins se comporte comme une substitution simultanée des constantes individuelles (les témoins) $w_1, w_2, \ldots, w_n$ aux variables existentielles $x_1, x_2, \ldots, x_n$. Cette substitution n'est dans ce cas qu'une substitution d'une constante à la variable qui lui est associée par l'axiome ajoutée. En effet, nous sommes en présence d'un cas simpliste où les θ sont des constantes individuelles.

Théorème 22 (Théorème de Remplacement) *Soient* ϕ *une formule et* ψ *une sous-formule de* ϕ*; soit* ψ' *une formule logiquement équivalente à* ψ*; si* ϕ' *est obtenue à partir de* ϕ *par remplacement d'une ou plusieurs occurrences de* ψ *par* ψ'*, alors:* $\models (\psi \leftrightarrow \psi')$

Cela nous ramène à la preuve de la complétude de notre translation existentielle par rapport à celle de Sowa. Cette preuve utilisera le théorème 22 de remplacement, valide en logique de premier ordre [CK73]:

Preuve 18 (Complétude Existentielle) *Il s'agit de prouver que si* $\models \psi$*, alors* $\models \psi^w$ *et vice-versa. Rappelons que* $\psi^w = S^{x_1 x_2 \ldots x_n}_{w_1 w_2 \ldots w_n} \psi$*.*

Si $\models \psi$**, alors nécessairement** $\models \psi^w$ *: cette première implication est une conséquence directe du théorème de substitution libre. Ce théorème [Riv89] énonce que si* ψ *est une formule comportant les variables librement substituables* $x_1, x_2, \ldots, x_n$*, et que*

$$\psi^\star = S^{x_1 x_2 \ldots x_n}_{\theta_1 \theta_2 \ldots \theta_n} \psi$$

alors: si $\models \psi$*, alors* $\models \psi^\star$*.*

Ce théorème découle du fait que la substitution libre préserve la validité des formules. Dans notre cas, ψ *est le résultat de la translation existentielle standard* ϕ *de Sowa et les* θ_i *sont les témoins* w_i*. La substitution correspondant à l'ajout de l'axiome des témoins génère* ψ^w *tout en préservant la validité.*

Si $\models \psi^w$, alors nécessairement $\models \psi$ *: L'axiome des témoins (Cf. Axiome 1) est une tautologie en logique de premier ordre. En particulier, l'inverse de l'axiome $\exists\alpha(x) \rightarrow \alpha(w)$ est aussi valide en logique de premier ordre. Il est ainsi possible de faire associer à chaque constante témoin w_i dans ψ^w une variable x_i, quantifiée existentiellement, telle que cette variable n'existe pas déjà dans la formule de graphe ψ^w. Une constante témoin w_i, n'étant présente que dans un seul arch à la fois (ceci est une conséquence logique de l'hypothèse 9, l'expression logique $\psi\_a^{w_i}$ correspondant à l'arch du graphe formule correspondant est équivalente, d'après l'axiome des témoins, à l'expression logique $\psi\_a^{x_i}$ obtenue en remplaçant le témoin w_i par une variable existentielle x_i, non utilisée auparavant. La formule de l'arch $\psi\_a^{w_i}$ étant une sous-formule de ψ^w, la preuve se fait alors par récurrence sur les témoins de la formule de graphe ψ^w et en utilisant le théorème de remplacement: puisque $\psi\_a^{w_i}$ est équivalent à $\psi\_a^{x_i}$, la formule f obtenue à partir de ψ^w par remplacement de $\psi\_a^{w_i}$ par $\psi\_a^{x_i}$ est équivalente à ψ^w. La nouvelle formule à prendre en considération sera ainsi f. Le processus finira quand il n'y aura plus de témoins à faire remplacer par des variables existentielle dans la formule ψ^w. Dans ce cas, ψ^w sera équivalent à l'interprétation standard existentielle de Sowa. Nous avons alors: si $\models \psi^w$, alors nécessairement $\models \psi$.*

d) Les Graphes Conceptuels avec Témoins

Pour tous les marqueurs existentiels $\exists$ présents dans un graphe conceptuel, nous considérons une nouvelle constante w, jamais utilisée auparavant tel que à partir d'un arch de la forme

$$[C_1 : \exists] \rightarrow (Rel) \rightarrow [C_2 : \exists]$$

nous obtenons

$$[C_1 : w] \rightarrow (Rel) \rightarrow [C_2 : \exists]$$

où le prédicat $C_1(w)$ est interprété vrai par la relation de conformité $Conf$. Ainsi, l'hypothèse sous-jacente à l'interprétation des marqueurs existentiels dans un graphe conceptuel est d'introduire un nouveau individu w pour chaque concept $[C : \exists]$ et d'étendre la relation de conformité avec $C(w)$. Il n'y a ainsi aucune différence pratique entre les individus, à proprement parler, dans ζ (cf. le canon page 162) et les témoins w. La seule différence est qu'on peut régulièrement ajouter, à travers la relation binaire $=$ introduite dans la section 4.3.4, page 169, l'interprétation $a = w$ dès lors qu'on rencontre un marqueur individuel a. Nous supposons que $=$ est une relation arbitraire. Elle peut être exprimée en graphes conceptuels par la substitution de a par w, pourvu que $a = w$ est dans la relation $=$, et que $C(w)$ est interprété vrai par la relation de conformité.

Inversement, nous pouvons remarquer que la dérivation suivante est également valide. Elle consiste à appliquer la controverse de l'axiome des témoins.

$$[C_1 : w] \rightarrow (Rel) \rightarrow [C_2 : \exists]$$
$$[C_1 : \exists] \rightarrow (Rel) \rightarrow [C_2 : \exists]$$

0ù $C_1(w)$ est interprété vrai par la relation de conformité. Les dérivations suivantes sont également valides:

$$[C_1 : a] \rightarrow (Rel) \rightarrow [C_2 : \exists]$$
$$[C_1 : \exists] \rightarrow (Rel) \rightarrow [C_2 : \exists]$$
$$[C_1 : w] \rightarrow (Rel) \rightarrow [C_2 : \exists]$$

L'égalité $a = w$ faisant partie de l'extension de la relation $=$. Notre convention, ici, est que chaque fois que nous rencontrons un marqueur existentiel dans un concept du graphe, nous l'interprétons comme un individu avec un nouveau nom, et nous supposons que ce nom est également dans le concept. Cet individu est supposé aussi être un élément de la dénotation du type de concept en question. La notion de témoin est donc simplement une technique permettant d'interpréter correctement l'utilisation des marqueurs existentiels et individuels dans les graphes conceptuels. Nous verrons dans la suite que cette technique nous permettra d'avoir un algorithme de complexité polynomiale pour la projection.

4.5.4 Quels Quantificateurs pour la Recherche d'Informations?

Dans les sections précédentes, nous avons distingué le marqueur générique $\star$ du marqueur existentiel $\exists$ (ou encore #). Il est toutefois intéressant dans le cadre d'une indexation à base de graphes conceptuels de déterminer la nature des quantificateurs utilisés.

Ainsi, bien que les corpus proviennent de domaines diverses, ils ont généralement un contenu *descriptif*. C'est-à-dire que la recherche d'informations consiste à retrouver des "objets" (qui peuvent être des documents, des logiciels, des images, de la vidéo, etc...) tels que *décrits* dans le corpus. On n'a donc pas affaire ici à des histoires ou à des scénarios, mais à des descriptions d'objets complexes.

Cette nature descriptive peut permettre certaines simplifications au niveau de la représentation. On peut souvent conclure à l'unicité et l'existence des concepts et ainsi éviter le problème de la quantification. Par exemple, dans un corpus de comptes-rendus médicaux, quand on parle d'"un patient" ou de "poumon cancéreux", on peut conclure qu'il s'agit de concepts existants et uniques: "patient" représente le patient décrit par le compte-rendu, et "poumon cancéreux", le poumon du patient.

De ce fait, si dans une image apparaît un *homme en position assise*, alors dans le graphe [HOMME]$\Leftrightarrow$(*Position*)$\Leftrightarrow$[ASSIS] associé, nous assumons l'existence d'un homme dans l'image ayant une position précise de type ASSIS. Les deux concepts génériques [HOMME] et [ASSIS] sont donc censés être existentiellement quantifiés:

[HOMME : $\exists$] $\Leftrightarrow$ (*Position*) $\Leftrightarrow$ [ASSIS : $\exists$]

Les concepts dans les graphes index sont donc existentiellement quantifiés, à défaut d'être individuels (ayant un élément de l'ensemble ζ comme référent[30]). Dans la suite, nous considérons que tous les concepts génériques des graphes conceptuels sont existentiellement quantifiés.

4.5.5 Conclusion: Quel(s) Algorithme(s) pour la Projection?

Nous avons proposé une extension de l'opérateur de translation logique de Sowa. Outre l'avantage qu'elle garde intacte les propriétés de la fonction d'interprétation de Sowa pour le cas des

30. Voir section 4.4.1, page 181.

marqueurs individuels et existentiels (ou interprétés en tant que tels), notre translation ϕ fait en sorte que le marqueur universel puisse être pris en charge avec des graphes simples. Notre interprétation permet ainsi au formalisme des graphes conceptuels de couvrir un plus grand fragment de la logique de premier ordre que celui initialement cerné par l'opérateur ϕ de Sowa.

Un graphe conceptuel simple peut donc inclure, des concepts individuels, des concepts existentiellement quantifiés mais aussi des concepts universellement quantifiés, faisant améliorer ainsi l'expressivité du modèle des graphes conceptuels simples. D'un point de vue logique, grâce à notre fonction de translation, le formalisme est capable d'exprimer un plus grand nombre d'énoncés de la logique de premier ordre. Une des conséquences de cette extension est qu'il faut adapter les opérateurs de jointure et de restriction en marge de la possibilité d'inclure en même temps dans un graphe des concepts existentiels et universels.

En particulier, selon que le graphe conceptuel est uniformément quantifié ou qu'il inclue les deux types de quantifications possibles[31], l'algorithme de la projection n'est pas le même. En collaboration avec Gianni Amati, nous avons relevé dans [AO97] trois types de projection possibles, selon les quantificateurs utilisés:

(1) Si les graphes utilisés sont universellement quantifiés, c'est-à-dire que chaque concept générique $[C]$ est interprété $[C : \forall]$, l'algorithme de projection revient à une comparaison de tables. Ainsi, les graphes sont mis sous la forme d'une table logique (voir section 4.5.2b), page 216), et l'établissement de la projection d'un graphe $\chi(q)$ sur un graphe $\chi(d)$ revient à chercher une implication $\phi(\chi(q)) \rightarrow \phi(\chi(d))$[32]. Cela revient à vérifier si la table logique de $\chi(q)$ est incluse dans la table logique de $\chi(d)$ à des restrictions de types et des *unifications de variables* près. Dans l'exemple 44 de la page 215, une telle inclusion est vérifiée.

(2) Si tous les concepts génériques des graphes sont existentiellement quantifiés, l'établissement de la projection d'un graphe $\chi(q)$ sur un graphe $\chi(d)$ revient à chercher une implication $\phi(\chi(d)) \rightarrow \phi(\chi(q))$[33]. Par rapport aux tables logiques, le processus de comparaison est un peu plus compliqué que celui du cas précédent. Toutefois, il est possible de l'implanter avec un moteur d'inférence d'une logique linéaire, si les tables étaient par exemple représentées en des clauses de prolog. La démarche algébrique équivalente que nous présentons dans la suite de ce chapitre permet d'éviter la complexité des traitements tout en gardant intacte la logique véhiculée.

(3) Si dans les graphes, il est possible d'avoir des concepts existentiels et des concepts universels, la projection est plus dure à modéliser en termes d'une implication logique. Le problème est que dans ce cas, les opérateurs de formation de graphes appliqués aux concepts existentiellement quantifiés font en sorte que les dérivations sur les graphes vont dans le sens opposé aux implications logiques sous-jacentes, alors que ces mêmes opérateurs appliqués aux concepts universellement quantifiés ramènent les implications logiques au sens des dérivations sur les graphes. Pour pouvoir prendre en compte cet aspect, on peut envisager l'utilisation de certaines techniques d'abduction et d'induction.

Comme nous l'avons vu dans la section précédente, en recherche d'informations les graphes index sont existentiellement quantifiés (cas (2) ci-dessus). Nous nous attachons donc dans la section

31. C'est-à-dire qu'il inclue des concepts existentiellement quantifiés et des concepts universellement quantifiés.

32. En effet de part l'interprétation logique assignée aux concepts universels, le sens des dérivations logiques suit le sens des dérivations des graphes: à partir du graphe $\chi(q)$ on dérive le graphe $\chi(d)$, et donc $\phi(\chi(q)) \rightarrow \phi(\chi(d))$.

33. Dans ce cas, le sens des dérivations logiques est bien entendu le même que celui établi par l'opérateur de translation de Sowa. Nous avons ainsi prouvé que notre interprétation coïncide avec la sienne dans ce cas.

suivante à proposer un algorithme de projection en conformité avec l'interprétation proposée. Cet algorithme est équivalent à celui de Sowa, puisque dans ce cas notre interprétation logique coïncide avec son opérateur ϕ (voir théorème 21, page 219). Cependant, de part la technique des témoins que nous avons associé à notre interprétation, il est possible d'adopter une approche algébrique pour les graphes conceptuels. Le résultat est que les raisonnements sont plus rapides, en particulier la complexité de la projection sera polynomiale. Dans la section suivante, nous présentons cette démarche algébrique.

4.6 Une Approche Algébrique pour les Graphes Conceptuels

Dans cette section nous présentons une implantation algébrique pour les GCs. Cette implantation donne lieu à un algorithme de projection polynomial. L'interprétation logique associée aux graphes conceptuels étant celle proposée dans la section précédente. Elle considère que les concepts génériques dans les index $\chi(d)$ sont existentiellement interprétés.

4.6.1 Logique Vs. Algèbre

Une autre forme de logique d'inspiration mathématique, allait voir le jour au milieu du XIXe siècle. C'est au mathématicien George Boole (1815-1864) qu'on en doit la formalisation, même si des logiciens comme DeMorgan et Leibniz avaient déjà initié l'idée. Cependant ce qui était, chez ces logiciens, une idée fragmentaire et vague, reçoit avec Boole une première réalisation. Ce mathématicien présente un système achevé qui apporte des solutions aux problèmes auxquels se heurtaient les logiques classiques, à savoir des procédures de décision, permettant des *calculs efficaces* et remplaçant les démonstrateurs de preuves. Cela a donné lieu à l'algèbre de Boole, qui depuis a toujours continué à être développée sur des bases mathématiques, donnant lieu notamment au théorème de représentation de Stone en 1930. Ce dernier montre que l'algèbre de Boole fournit un cadre excellent pour l'analyse des circuits électroniques.

Or s'il revient à Boole l'idée de construire une logique selon l'esprit des mathématiques, c'est à Frege, reconnu par les logiciens comme le fondateur de la logique [Sow97], que revient les concepts fondamentaux, les cadres et les premiers éléments fixant les fondements de l'édifice logico-mathématique. L'idée de base de Boole était qu'au lieu d'aboutir à la logique par un procédé d'abstraction à partir des démarches effectives de la pensée, il la traite comme une construction formelle, pour laquelle on cherchera seulement ensuite une interprétation. L'approche de départ consiste à dire que toutes les opérations du langage, considéré comme un instrument de raisonnement, peuvent être menées par un système mathématique composé des éléments suivants [BD96]:

1. Des symboles littéraux, tels que x, y, z, ... , représentant les choses qui font l'objet de nos conceptions;

2. Des symboles d'opérations, tels que $+, \Leftrightarrow, \times, \ldots$, représentant les opérations de l'esprit pour lesquelles les conceptions des choses sont combinées ou résolues, de façon à former des conceptions nouvelles enveloppant les mêmes éléments;

3. Le signe de l'identité, $=$.

Nous pouvons saisir dès lors de nombreuses analogies entre les lois de la syntaxe algébrique et celles de la syntaxe logique, celle qui règle la composition de nos conceptions. La table 4.1 en énumère quelques exemples.

Syntaxe algébrique	*Syntaxe logique*
$xy = yx$	bonnet blanc = blanc bonnet
$x + y = y + x$	hommes et femmes = femmes et hommes
$z(x + y) = zx + zy$	Les Européens (homme et femmes) = les Européens hommes et les Européens femmes
$z(x \Leftrightarrow y) = zx \Leftrightarrow zy$	Les Européens (sans les femmes) = les Européens hommes, mais non les Européens femmes

Table 4.1. Syntaxe algébrique Vs. Syntaxe logique

Boole avait toutefois décelé un défaut dans une telle approche: dans la pensée logique ordinaire, la loi $x^n = x$ est valable. En effet, si nous prenons la classe des Français, par exemple, combinée avec la classe des Français, elle ne donne rien de plus en logique que la classe des Français, puisque "Français $\wedge$ Français = Français". Ceci n'est pas le cas en algèbre où, l'élévation aux puissances donne généralement autre chose que la classe de départ, sauf dans le cas particulier de $x^2 = x$, admettant deux racines, qui sont 0 et 1:

$$0^2 = 0$$

$$1^2 = 1$$

Nous retrouvons ici l'analogie avec la loi de la logique propositionnelle. Boole a ainsi conclu que la logique peut être assimilée à une espèce particulière d'algèbre, une algèbre dans laquelle les symboles numériques ne seraient susceptibles de recevoir d'autres valeurs que les valeurs 0 et 1. Toutefois, une telle algèbre n'aurait d'intérêt en logique que si l'on peut lui trouver une interprétation en termes de la logique. Ainsi, aux symboles littéraux, Bool associait des concepts qui, interprétés en extension, correspondent à des classes d'individus; les symboles des opérations fondamentales qui sont + (pour addition) et $\times$ (pour la multiplication), conviennent pour la somme logique (réunion de deux classes) et pour le produit (intersection de deux classes); enfin le symbole de l'égalité signifie que les deux classes dont les symboles l'entourent ont la même extension dans un domaine de discours, c'est-à-dire qu'elles s'incluent mutuellement. Quand aux symboles 0 et 1, Boole donnait l'interprétation suivante: 1 symbolise la classe universelle $\top$, celle qui dénote tous les éléments du domaine de discours, et 0 la classe vide ou absurde, $\perp$, symbolisant la non-existence, c'est-à-dire ne dénotant aucun élément du domaine de discours.

L'algèbre ci-dessus construite munie de l'interprétation qui lui est associée permet de simuler le comportement de la logique propositionnelle, tout en offrant des procédures de décisions plus faciles à gérer que les dérivations logiques. Elles correspondent en effet à des opérations ensemblistes simples. On énonce le principe de dualité suivant:

Principe de Dualité 4.6.1 *(Idée de Boole) Soit une logique $\mathcal{L}$. Soit son correspondant algébrique*

$\mathcal{A}_{\mathcal{L}} = \langle \mathcal{X}, \mathcal{F} \rangle$ où $\mathcal{X} = \{\mathcal{S}_{\Delta}\}$ est une collection d'ensembles non vides $\mathcal{S}_{\Delta}$ dénotant des classes du domaine de discours Δ, et $\mathcal{F}$ un ensemble fini d'opérations sur $\mathcal{X}$. À chaque expression primitive de $\mathcal{L}$ correspond un ensemble dans $\mathcal{X}$ et à chacun de ses connecteurs c correspond un ou plusieurs opérations de $\mathcal{F}$, tels que les théorèmes de $\mathcal{L}$ seront simulés, dans toute leur étendue, avec les raisonnements de l'algèbre associée.

Ce principe de dualité permettra de transformer les dérivations en logique à un calcul algébrique simple muni de procédures de traitements efficaces et faciles à implanter. La solution d'un problème en logique se déroulera en trois temps: traduire les données du problème logique dans le vocabulaire du calcul algébrique, effectuer dans ce calcul les opérations convenables, retraduire les résultats du calcul dans le langage logique initial.

Les graphes conceptuels simples de Sowa forment à ce titre une algèbre de graphes, où les opérateurs algébriques peuvent simuler un raisonnement logique. Les raisonnements dans un fragment de la logique de premier ordre peuvent en effet se faire par le biais d'opérateurs graphiques, une fois le problème transformé en graphes conceptuels. Malheureusement, l'adoption d'une algèbre spéciale pour une logique donnée se heurte aux deux problèmes suivants:

- établir les lois de cette algèbre telle qu'elle corresponde aux lois de la logique.
- trouver une interprétation logique acceptable aux symboles et opérateurs proposés par cette algèbre.

Il n'est pas toujours évident d'apporter des solutions à ces deux points. Il est ainsi parfois difficile de trouver des significations logiques à des opérateurs comme la jointure maximale dans les graphes conceptuels [CM95]. Inversement, un connecteur logique comme la négation n'a pas d'analogue exact dans le symbolisme algébrique, sauf par le biais de certains artifices d'écriture [BD96]. Une solution consiste à ignorer la correspondance entre la logique et l'algèbre pendant le déroulement du calcul algébrique. Ainsi, il suffit que cette correspondance s'établit au point de départ (lors de l'énoncé du problème) et au point d'arrivée (après la fin des traitements algébriques). C'est l'approche généralement adoptée pour toute dualité entre algèbre et logique, en particulier dans la dualité entre les graphes conceptuels et le fragment de la logique de premier ordre correspondant.

4.6.2 Algèbre et Systèmes d'Informations

Nous avons vu dans la section précédente, qu'une algèbre munie d'une interprétation adéquate peut simuler un raisonnement logique. Les raisonnements algébriques ont la propriété de permettre des calculs efficaces et rapides. Ceci, comme nous l'avons souligné dans l'introduction, est à l'opposé des démonstrateurs de théorèmes dont les traitements sont complexes et lourds à mettre en oeuvre. Ainsi à chaque fois que cela fut possible, les systèmes d'informations opérationnels ont opté pour une implantation algébrique des logiques sous-jacentes à leur modèle.

a) Les Bases de Données Relationnelles

Dans les modèles de bases de données relationnelles, le langage de manipulation des données a reçu une attention toute particulière. Comme cela a été signalé dans [DA82], il existe principalement deux classes de langages d'interrogation proposées pour les bases de données relationnelles: d'une

part les langages algébriques, où l'expression d'un besoin d'un utilisateur se fait à l'aide d'opérations dont les opérandes sont les relations, et d'autre part les langages prédicatifs, où l'utilisation des prédicats permet de sélectionner l'ensemble des n-uplets souhaités. Il a été montré que toutes les questions qui recevaient une réponse dans un langage étaient en mesure d'en recevoir la même dans un autre langage. Autrement dit, l'équivalence entre ces deux types de langage a été prouvée : Si A est une expression algébrique, alors il existe une formule prédicative équivalente à A dans le cas des modèles de bases de données relationnelles.

Si dans la pratique, certains systèmes relationnels ont été développés à base de langage prédicatifs (par exemple, QUERY-BY-EXAMPLE), il n'en reste pas moins que l'approche consistant à démontrer un théorème au sens de la logique mathématique a été vite abandonnée par les systèmes commerciaux. En effet, du point de vue de l'évaluation d'une réponse par rapport à une requête, la mise en oeuvre de méthodes systématiques relevant de la démonstration de théorèmes n'est pas concevable en base de données vu le volume des axiomes. En effet, les énoncés élémentaires (les faits de la base) et les propriétés générales sont assimilés à des axiomes de la logique de premier ordre[34]. De plus, le choix d'une sémantique propre aux bases de données (comme celle que nous avons adopté pour la recherche d'information au début du chapitre 3) n'est pas une tâche aisée. De l'adoption d'une approche à monde ouvert ou fermé, dépend la façon dont sont traitées les inconsistances générées par les règles de déduction [DA82]. Il s'agit donc soit de spécifier toutes les informations négatives, c'est-à-dire qu'il faut non seulement indiquer ce que l'on considère comme vrai, mais aussi ce que l'on considère comme faux, soit il s'agit de considérer que tout ce qui n'est pas énoncé est faux. Dans le premier cas, l'approche contribue à accroître le volume des axiomes. Dans le deuxième cas, il faudra prendre des précautions particulières pour gérer les inconsistances.

Cette équivalence entre les langages prédicatifs et les langages algébriques, ajoutée aux performances opérationnelles qu'on connaît des démonstrateurs de preuves ont ramené les entreprises commerciales en base de données à préférer l'adoption d'une approche algébrique lors de la conception de leur système de gestion de base de données (ORACLE, INFORMIX). L'efficacité de ces systèmes, leur capacité à gérer de très grandes quantités d'informations et la rapidité de leurs réponses n'est plus à démontrer et justifient amplement le choix algébrique.

b) Le Modèle Booléen et les Fichiers Inverses

En recherche d'informations, le modèle Booléen [SM83, vR79] basé sur une logique propositionnelle fut ainsi implanté grâce à l'algèbre de Boole. Un document dans ce modèle est représenté par une conjonction de mots-clés. Un répertoire de tous les mots indexant le corpus doit être maintenu. Toutes les références aux documents contenant chaque mot est stockée dans une structure de *fichier inverse* (voir table 4.2).

La requête est une expression booléenne combinant des mots-clés à l'aide de trois opérateurs: ET, OU et NON. Pour une requête, il est nécessaire d'établir la liste des références à tous les documents concernés. La résolution d'une requête revient ainsi à combiner par les opérations ensemblistes associées aux opérateurs booléens (union pour le OU, intersection pour le ET, différence pour le NON), les ensembles de références correspondant à chaque terme de la requête.

Par exemple, une requête *Recherche* ET *Information* appliquée au fichier inverse de la figure

34. Pour des plus amples détails sur les rapports entre la logique mathématique et les bases de données, on pourra consulter [GMN84].

Mots-clés	Documents
Système	d_1, d_3
Recherche	d_1, d_2, d_3, d_6
Information	d_1, d_6, d_8
Données	d_3

Table 4.2. Une structure de fichier inverse où le document d_1 est indexé par la conjonction Système $\wedge$ Recherche $\wedge$ Information

précédente sera traitée par la procédure suivante:

- rechercher dans le fichier inverse, les documents indexés par le mot-clé *Recherche*. Le résultat est l'ensemble $S_1 = \{d_1, d_2, d_3, d_6\}$.
- rechercher dans le fichier inverse, les documents indexés par le mot-clé *Information*. Le résultat est l'ensemble $S_2 = \{ d_1, d_6, d_8\}$.
- déterminer les documents contenus aussi bien dans l'ensemble S_1 que dans l'ensemble S_2. Il s'agit d'appliquer une intersection sur les deux ensembles. Le résultat est l'ensemble $S_3 = \{d_1, d_6\}$.
- retourner comme réponse les documents dont l'identificateur est un élément de S_3.

Le grand avantage d'une telle implantation est son efficacité, vu que les requêtes peuvent être résolues avant d'accéder aux contenus des documents. Seuls les documents sélectionnés sont traités par le système.

Cette solution fondée sur des fichiers inverses n'est cependant utilisable que pour des systèmes ayant un vocabulaire d'indexation assez stable, c'est-à-dire dans le cas où les mises à jour des documents ne sont pas très fréquentes. En effet, la génération du fichier inverse requiert une analyse détaillée du contenu sémantique de tous les documents du corpus, nécessitant par la même occasion des traitements de contrôle assez délicats à mener et très coûteux en temps de calcul. Ainsi, si les données du corpus changent fréquemment, le coût total de la mise à jour du fichier affectera négativement le processus de recherche en sa totalité.

La fréquence des mises à jour des corpus en recherche d'informations dépend du domaine d'application traité. Ainsi, si les corpus sont stables dans le cas d'une base bibliographique ou d'une base d'images historiques, il en devient autrement si l'on prend par exemple le cas du Web où les ajouts des pages HTML se font à un rythme endiablé. Cependant, même dans ce dernier cas, il n'est pas impératif de traiter ces mises à jours en temps réel. Par exemple, un moteur de recherche comme Altavista, ne remet à jour ses index des pages Web qu'une fois tous les mois, voir plus, à en juger la fréquence de ses accès aux serveurs pour la ré-actualisation de ses données[35]. D'ailleurs les moteurs de recherche sur le Web utilisent des structures d'accélération similaires aux fichiers inverses. En recherche d'informations, il est donc envisageable d'adopter des mises à jour périodiques, une fois

35. Donnée établie à la base de ses accès aux serveurs de l'IMAG.

par jour, une fois par semaine voir même une fois par mois. En effet, très rare sont les domaines d'application où un utilisateur est intéressé par un document qui vient juste d'être crée.

Cette caractéristique de la recherche d'informations explique pourquoi les fichiers inverses sont de nos jours largement utilisés par les systèmes commerciaux [SM83]. Les temps de réponses de ces systèmes sont au-dessus de toute concurrence. Le choix de ces structures s'avère être efficace, même pour des grands corpus.

4.6.3 Motivations d'une Approche Fichier Inverse

Malgré le succès relatif des systèmes de recherche d'informations Booléens auprès des utilisateurs, leur capacité à gérer convenablement des corpus de très grande taille, l'efficacité des fichiers inverses à leur assurer des temps de réponses instantanés, nous continuons encore à penser que ces systèmes sont inadéquats dès lors qu'il s'agit d'assurer la précision des réponses.

Conscients toutefois de l'importance des délais de réponse fournis par le système comme un critère de satisfaction primordial pour les utilisateurs, nous nous proposons dans la suite de présenter une approche permettant d'assurer aussi bien la précision des réponses que la rapidité du processus de recherche.

Ainsi, même si la projection semble être un opérateur algébrique plus efficace du point de vue de la rapidité des traitements que les démonstrateurs de théorèmes, il n'en reste pas moins que le problème de l'existence d'une projection d'un graphe sur un autre est un problème NP-complet [CM92, MC96][36]. En prenant pour hypothèse que la projection est un problème NP-complet, de nombreux travaux se sont intéressés à trouver des algorithmes rapides pour le traitement de la projection. Un bon recensement de ces travaux est présenté dans la thèse de Guinaldo [Gui96b]. Nous énumérons ici les principales solutions utilisées:

- utiliser des techniques de "Backtracking" pour optimiser la recherche de la projection [McG82]
- tirer profit du domaine de l'application [MLL91].
- limiter le langage des graphes conceptuels à des classes pour lesquelles le problème de l'existence d'une projection devienne polynomial [Mug93, LB94, Wil95, PC95, Gui96a].
- maintenir la base des graphes conceptuels dans l'ordre lié à la relation de spécialisation $\leq$ [Ell93, EL94].

Nous nous inspirons ici de la technique des fichiers inverses pour proposer un algorithme de recherche polynomial pour n'importe quelle classe de graphes conceptuels simples. Cet algorithme utilise une organization des graphes en un fichier inverse pour effectuer une recherche rapide. L'idée

36. Contrairement à la relation de subsomption en logiques terminologiques, il n'existe pas beaucoup d'études sur la complexité de l'opérateur de projection. D'ailleurs les preuves de complexité fournies dans les deux références citées sont laconiques et manquent de rigueur. Elles utilisent des résultats de la théorie des graphes mais n'explicitent pas le lien avec les graphes conceptuels. Nous avons cependant vérifié dans [AO98] que l'opérateur de projection en général est effectivement NP-Complet. Nous avons toutefois montré qu'en utilisant l'interprétation logique de la section 4, et la forme logique résultante des graphes, le problème de dérivation entre graphes conceptuels peut être réduit à un problème propositionnel de question-réponse comme dans Datalog [DEGV97, JL77, Var82, Imm86]. Dans ce cas, la projection admet une complexité polynomiale.

consiste à pré-calculer des parties complexes de l'opérateur de projection lors de la phase d'indexation. Ces parties sont ensuites stockées puis associées au fichier inverse. En recherche d'informations, la rapidité d'un système est évaluée lors de la phase d'interrogation. Il s'agit ainsi de faire en sorte que ce qui reste à calculer de l'opérateur de projection lors de cette phase soit polynomial. C'est ce dernier calcul qui constituera notre algorithme de recherche. Cela nous permettra d'implanter un système de recherche d'informations orienté vers la précision des réponses, mais également muni d'une fonction de correspondance très rapide.

Nous voulons toutefois argumenter que cet algorithme de recherche ne serait pas envisageable sans l'utilisation de la technique des témoins et donc de la fonction d'interprétation ϕ proposée dans la section 4.5. En effet, c'est cette interprétation logique qui nous garantie que l'opérateur de recherche implanté est correct et compplet vis-à-vis de l'interprétation logique de Sowa.

Notre objectif ici est de montrer que l'on peut indexer des documents par des graphes conceptuels, et que les raisonnements sur ces graphes peuvent aussi être simulés par des opérations ensemblistes faciles à calculer. Le principe est le même que pour le modèle Booléen. Dans ce dernier, il est possible pour un mot-clé de lui associer toutes les références aux documents contenant ce mot-clé. Supposons donc que pour un graphe conceptuel g_1, il est possible de lui associer les documents dans lesquels il apparaît[37]. Pour aboutir à une approche algébrique, il suffirait qu'à chaque fois qu'un graphe conceptuel est dérivé à partir d'un autre graphe par les quatre opérateurs de construction classiques, il est possible de trouver un équivalent ensembliste de cette dérivation. L'ensemble des documents associé au graphe de départ sera modifié en conséquence:

$g_1 \mapsto \{\text{ensemble de documents}\}$

$\oplus_{graphe}(g_1) \mapsto \oplus_{ensembliste}(\{\text{ensemble de documents}\})$

Le problème avec les graphes conceptuels est qu'une telle approche est difficile à mettre en oeuvre du fait de l'utilisation des treillis des types de concepts et des spécificités de l'opérateur de jointure. Ce n'est donc pas un hasard que la seule différence en termes des décisions de pertinence entre un système Booléen et un système à base de graphe conceptuels a trait à ces deux particularités. Bien entendu, la différence majeure entre ces deux approches est à chercher du côté de l'expressivité du langage sous-jacent. Autant le formalisme des graphes conceptuels est expressif et robuste, autant la représentation plate des mots-clés dans le modèle Booléen n'avantage pas une description précise du contenu des documents.

Ainsi, nous avons présenté dans la section 4.2 les décisions de pertinence élaborées par le modèle des graphes conceptuels. À partir d'un travail semblable appliqué au modèle Booléen [HOC96], il en ressort que les décisions de pertinence du modèle des graphes conceptuels en recherche d'informations sont, à deux règles près, équivalentes à celles supportées par le modèle Booléen:

Définition 4.6.1 (Décisions de Pertinence du Modèle Booléen) *Le système de dérivation de pertinence $\mathcal{B}$ correspondant aux décisions de pertinence du modèle Booléen est défini comme l'ensemble des postulats {Réflexivité, Monotonicité, Équivalence Ensembliste, Cut}.*

Contrairement au modèle des graphes conceptuels, le modèle Booléen adopte une Monotonicité non prudente. Les graphes conceptuels étant connexes[38], un graphe ne peut être "ajouté" à un

37. C'est-à-dire les documents pour lesquels il constitue un sous-graphe de leur index.

38. Dans [MC96], cette contrainte a été levée, entraînant des changements profonds au niveau du formalisme de base.

autre graphe que s'il existe un concept commun entre ces deux graphes. Autrement dit, s'il est possible d'appliquer une jointure entre ces deux graphes.

De plus, le modèle Booléen ne supporte pas la règle d'endiguement présente dans le formalisme des graphes conceptuels. Il s'agit de la règle qui tient compte du treillis des types de concepts. Un utilisateur intéressé par le concept ANIMAUX sera satisfait par les documents traitant des OISEAUX.

Par conséquent, un modèle Booléen muni d'un thésaurus aura un comportement décisionnel de pertinence presque équivalent à celui du modèle des graphes conceptuels. D'ailleurs plusieurs systèmes booléens intègrent réellement cet élément. Le système STAIRS d'IBM [SM83], par exemple, introduit un dictionnaire permettant la détection des relations de synonymie entre certains mots-clés. Au niveau de l'implantation, un fichier inverse est toujours utilisé, sauf que cette fois ci, à ce dernier on associe une table dictionnaire, où chaque ligne est une entrée constituée d'un terme et d'un ensemble de pointeurs vers des entrées du fichier inverse. Ces dernières correspondent à des mots-clés synonymes au terme de la table.

L'idée est que pour chaque terme de la requête, il y a lieu d'accéder à la table du dictionnaire, de détecter l'ensemble des mots-clés synonymes à ce terme et de considérer que les documents du fichier inverse dans lesquels apparaissent ces synonymes sont potentiellement pertinents pour cette requête. Pour un terme donné, on établie ainsi l'union de toutes les références aux documents contenant ce terme mais aussi les synonymes de ce dernier.

Cette démarche peut être étendue aux relations de spécialisation/généralisation, ou à toute autre relation pouvant être modélisée dans un thésaurus. La démarche consiste à faire en sorte que chaque fois qu'on a besoin d'effectuer un traitement sur les termes d'indexation d'un système, il y a lieu de le faire dans une autre structure, qui sera par la suite associée au fichier inverse principal du corpus.

Pour pouvoir aboutir à une structure de fichier inverse dans le cas du formalisme des graphes conceptuels, nous adoptons une démarche progressive où nous tâchons d'abord d'intégrer des termes plus complexes comme entrées d'un fichier inverse. On s'inspirera ensuite de la démarche de STAIRS pour proposer une technique permettant la prise en compte des divers treillis de types de l'application. Cette approche permettra alors d'imaginer un fichier inverse pour les graphes conceptuels, où les termes en entrée correspondent à des archs. On montrera alors que l'utilisation de la technique des témoins permet la construction d'une véritable approche algébrique pour les graphes conceptuels analogue à celle du modèle Booléen.

4.6.4 Extension Relationnelle du Fichier Inverse

Nous avons proposé dans [OP97a], une extension à l'approche classique des fichiers inverses, de façon à pouvoir considérer des structures sémantiques comme terme d'indexation. Ainsi, au lieu des simples mots-clés, nous utilisons les relons du chapitre précédent, c'est-à-dire des structures de la forme $\langle\langle \mathit{Relation}, c_1, c_2, \ldots, c_n ; 1\rangle\rangle$, où $c_1, c_2, \ldots, c_n$ sont des (types de) concepts. À chaque type de relation est associé une signature.

Afin de montrer comment fonctionne l'approche, considérons trois images de la collection test PARISYS [CM97]. Un index simplifié de ces trois images pourrait être les ensembles des termes

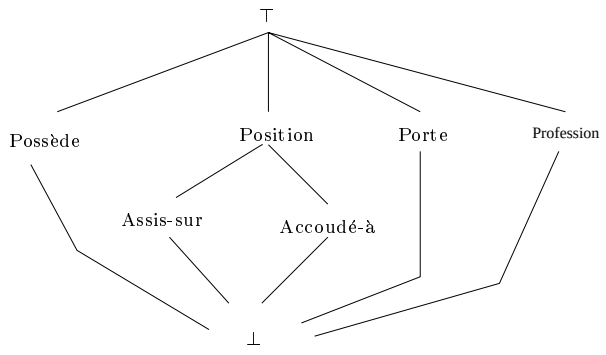


Figure 4.39. Le treillis des types de Relations

d'indexation suivants:

$\chi(d_1)$ = { ⟨⟨Profession, PERSONNE, TISSERAND ; 1⟩⟩, ⟨⟨Porte, PERSONNE, CHAPEAU ; 1⟩⟩, ⟨⟨Assis-sur, PERSONNE, CHAISE ; 1⟩⟩},

$\chi(d_2)$ = { ⟨⟨Profession, PERSONNE, COMPOSITEUR ; 1⟩⟩, ⟨⟨Porte, PERSONNE, COUVRE-CHEF ; 1⟩⟩, ⟨⟨Assis-sur, PERSONNE, FAUTEUIL ; 1⟩⟩, ⟨⟨Accoudé-à, PERSONNE, TABLE ; 1⟩⟩, ⟨⟨Possède, PERSONNE, BARBE ; 1⟩⟩}

$\chi(d_3)$ = { ⟨⟨Profession, PERSONNE, ECRIVAIN ; 1⟩⟩, ⟨⟨Accoudé-à, PERSONNE, TABLE ; 1⟩⟩, ⟨⟨Assis-sur, PERSONNE, FAUTEUIL ; 1⟩⟩, ⟨⟨Possède, PERSONNE, BARBE ; 1⟩⟩}

Considérons que le treillis des types de relations est celui de la figure 4.39. Considérons également le treillis des types de concepts de la figure 4.41. Supposons que les signatures associées à ces divers types de relations sont les suivantes:

$\Sigma_{Profession}$ = (Profession, 2, PERSONNE , TRAVAIL)

Σ_{Porte} = (Porte, 2, PERSONNE , CHAPEAU)

$\Sigma_{Assis-sur}$ = (Assis-sur, 2, PERSONNE , CHAISE)

$\Sigma_{Accoud\text{é}-\text{à}}$ = (Accoudé-à, 2, PERSONNE , MEUBLE)

$\Sigma_{Poss\text{è}de}$ = (Possède, 2, PERSONNE , CARACTERISTIQUE)

Notre fichier inverse étendu aux structures sémantique est construit selon l'algorithme suivant:

Algorithme 3 (Construire Fichier Inverse)

// $\mathcal{D}$ est l'ensemble des documents indexés du corpus
// $\mathcal{FI}$ désigne la table du fichier inverse
CONSTRUIRE$_{\mathcal{FI}}$($\mathcal{D}$, $\mathcal{FI}$) **Début**
POUR chaque document d_i FAIRE
    POUR chaque relon r dans $\chi(d_i)$ FAIRE
        Res := TROUVELIGNE($\mathcal{FI}$, r)
        SI Res = $\emptyset$ ALORS
        AJOUTERLIGNE($\mathcal{FI}$, r, [d_i])
        SINON
        AJOUTERDOCUMENT($\mathcal{FI}$, r, 'd_i')
        FINSI
    FINFAIRE
FINFAIRE
Retourne $\mathcal{FI}$
Fin

Il s'agit de parcourir séquentiellement les index des documents, de lire leurs relons et, pour chacun d'eux, de vérifier si le relon en question existe déjà ou pas dans le fichier inverse. S'il existe déjà comme entrée du fichier, l'identificateur du document contenant ce relon est ajouté à la liste des références associées à ce dernier dans le fichier inverse. Dans le cas contraire, on crée une nouvelle ligne dans le fichier inverse ayant pour entrée le relon en question, et pour liste de références le seul identificateur du document correspondant. Cet algorithme, appliqué aux trois index de notre exemple donne comme résultat le fichier inverse de la figure 4.40 suivante.

Ligne	**Relon**	**Docs.**
1	$\langle\langle$Profession, PERSONNE, TISSERAND ; 1$\rangle\rangle$	[d_1]
2	$\langle\langle$Porte, PERSONNE, CHAPEAU ; 1$\rangle\rangle$	[d_1]
3	$\langle\langle$Assis-sur, PERSONNE, CHAISE; 1$\rangle\rangle$	[d_1]
4	$\langle\langle$Profession, PERSONNE, COMPOSITEUR ; 1$\rangle\rangle$	[d_2]
5	$\langle\langle$Porte, PERSONNE, COUVRE-CHEF ; 1$\rangle\rangle$	[d_2]
6	$\langle\langle$Assis-sur, PERSONNE, FAUTEUIL ; 1$\rangle\rangle$	[d_2,d_3]
7	$\langle\langle$Accoudé-à, PERSONNE, TABLE ; 1$\rangle\rangle$	[d_2,d_3]
8	$\langle\langle$Possède, PERSONNE, BARBE ; 1$\rangle\rangle$	[d_2,d_3]
9	$\langle\langle$Profession, PERSONNE, ECRIVAIN ; 1$\rangle\rangle$	[d_3]

Figure 4.40. Fichier Inverse $\mathcal{FI}$ de l'exemple

Considérons une requête utilisateur cherchant les images montrant "*des personnes ayant une*

barbe et assises sur un fauteuil". Cette requête sera représentée par deux relons, à savoir $\chi(q_1) = \{\langle\langle$Possède, PERSONNE, BARBE ; 1$\rangle\rangle$, $\langle\langle$Assis-sur, PERSONNE, FAUTEUIL ; 1$\rangle\rangle\}$. Si une procédure similaire à celle présentée pour les mots-clés est utilisée, le système retournera les documents ayant pour identificateurs $\{d_2, d_3\} \cap \{d_2, d_3\} = \{d_2, d_3\}$.

Le cas de recherche ci-dessus présenté est particulièrement simple, puisqu'il ne nécessite l'utilisation d'aucun treillis de types de concepts ou de relations. Considérons ainsi une autre requête q_2 où l'utilisateur souhaite retrouver les images montrant "*des artistes assis sur une chaise*". Cette requête peut être représentée avec les relons suivants: $\chi(q_2) = \{\langle\langle$Profession, PERSONNE, ARTISTE ; 1$\rangle\rangle$, $\langle\langle$Assis-sur, PERSONNE, CHAISE ; 1$\rangle\rangle\}$. Si seules les données du fichier inverse sont utilisées, la réponse à cette requête sera vide. Pourtant les images correspondants aux index $\chi(d_2)$ et $\chi(d_3)$ constituent des bonnes réponses à la requête. Le treillis des types de concepts doit être utilisé pour cet effet.

Comme dans STAIRS, nous utilisons un autre ensemble de tables, appelées *tables d'accélération*, dont le rôle est de pré-calculer toutes les spécialisations d'un relon donné. Ainsi, pour un relon donné, il est possible de connaître la position de ses spécialisations dans le fichier inverse. Il existe une table d'accélération pour chaque type de relation. Chaque ligne de ces tables a deux champs: le premier constitue un *concept*, alors que le second contient une liste de références aux lignes du fichier inverse dans lesquels une spécialisation du concept est contenue dans le relon correspondant.

Concept	**Liste des no. de ligne de $\mathcal{FI}$**
PERSONNE	[1,4,9]
TRAVAIL	[1,4,9]
OUVRIER	[1]
TISSERAND	[1]
ARTISTE	[4,9]
COMPOSITEUR	[4]
ECRIVAIN	[9]

Table 4.3. Table d'accélération pour la relation *Profession*

L'algorithme suivant détaille la procédure de construction d'une table d'accélération dans le cas où les index ne sont constitués que de relons binaires. Cet algorithme peut facilement être étendu aux relons n-aires. On notera par $\Sigma_r = (\mathrm{r}, 2, C_{sign_1}, C_{sign_2})$, la signature d'un type de relation

binaire r donné.

Algorithme 4 (Construire Tables d'Accélération)

$\mathcal{T}_c$ est le treillis des concepts
$\mathcal{FI}$ désigne le fichier inverse
TA désigne l'ensemble des tables d'accélération, une pour chaque relation
CONSTRUIRE$_{TA}$($\mathcal{FI}$, $\mathcal{T}_c$)

POUR chaque Ligne de $\mathcal{FI}$ FAIRE
  $R(C_1, C_2)$:= EXTRAIRE-RELON (Ligne)
  Accéder à la table d'accélération TA_R de R
  trouve Ligne dans table TA_R ayant C_1 pour premier champs
    SI il n'existe pas de telle Ligne
      ajouter nouvelle ligne dans TA_R
      champs 'concept' de Ligne := C_1
      champs 'références' de Ligne := liste vide
    FINSI
  ajouter 'Ligne' au champs 'références' de toutes
  les lignes de TA_R où le champs 'concept' est C_i,
  avec $C_1 \leq C_i \leq C_{sign_1}$.
  trouve Ligne dans table TA_R ayant C_2 pour premier champs
    SI il n'existe pas de telle Ligne
      ajouter nouvelle ligne dans TA_R
      champs 'concept' de Ligne := C_2
      champs 'références' de Ligne := liste vide
    FINSI
  ajouter 'Ligne' au champs 'références' de toutes
  les lignes de TA_R où le champs 'concept' est C_i,
  avec $C_2 \leq C_i \leq C_{sign_2}$.
FINFAIRE
Retourne TA

Appliqué au treillis des types de concepts de la figure 4.41, cet algorithme produit la table d'accélération 4.3, pour le type de relation *Profession*. L'application du même algorithme donne la table d'accélération 4.4 suivante pour la relation *Assis-sur*.

Par exemple dans la table d'accélération 4.3 précédente, deux lignes du fichier inverse, 4 et 9, correspondent au concept ARTISTE (voir figure 4.40). Dans la ligne 4 de ce fichier inverse, nous avons ⟨⟨Profession, PERSONNE, COMPOSITEUR ; 1⟩⟩ comme relon. Le type COMPOSITEUR est une spécialisation de ARTISTE, c'est pourquoi la ligne 4 apparaît devant le concept ARTISTE dans cette table d'accélération.

Pour le treillis des relations, un raisonnement similaire peut être appliqué. Seulement, dans ce cas il est plus facile d'étendre le fichier inverse. Il se verra ainsi augmenté de tous les relons pouvant être dérivés à partir de ceux déjà existants. Ainsi, puisque les deux relations *Assis-sur* et *Accoudé-à* sont des spécialisations de la relation *Position*, chaque fois qu'ils apparaissent dans un document, ils véhiculent implicitement l'existence d'une relation *Position* dans ce document. Nous pouvons ajouter des lignes au fichier inverse exploitant ce fait (voir la figure 4.42 suivante).

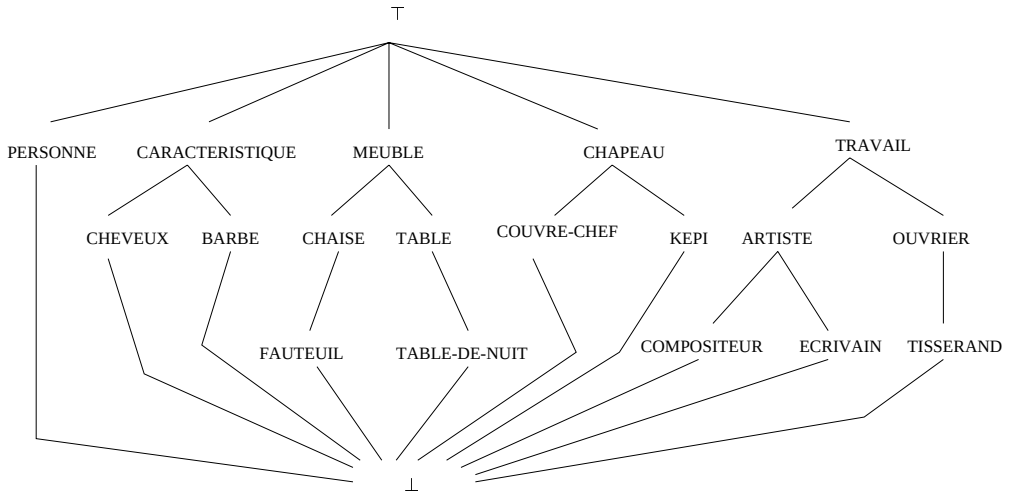


Figure 4.41. Le treillis des types de Concepts

Concept	**Liste des no. de ligne de $\mathcal{FI}$**
PERSONNE	[3,6]
CHAISE	[3,6]
FAUTEUIL	[6]

Table 4.4. Table d'accélération pour la relation *Assis-sur*

Ligne	**Relon**	**Docs.**
10	$\langle\langle$Position, PERSONNE, CHAISE, ; 1$\rangle\rangle$	$[d_1]$
11	$\langle\langle$Position, PERSONNE, FAUTEUIL, ; 1$\rangle\rangle$	$[d_2,\ d_3]$
12	$\langle\langle$Position, PERSONNE, TABLE, ; 1$\rangle\rangle$	$[d_2,\ d_3]$

Figure 4.42. Extension à base du treillis des relations

Il est clair qu'il faudra d'abord ajouter au fichier inverse les relons découlants du treillis des types de relations avant de lancer la construction des tables d'accélération. Ces dernières seront utilisées par la suite lors de la phase de correspondance. L'algorithme de recherche fonctionne de

la façon suivante:

Algorithme 5 (Recherche)
RECHERCHE($\chi(q)$, $\mathcal{FI}$, TA)
cette fonction retourne les document pertinents

Résultat := liste de tous les documents
POUR chaque relation R(C_1,C_2) $\in \chi(q)$ FAIRE
    Liste1 := CHERCHERÉFÉRENCES*(TA_R, C_1)*
    Liste2 := CHERCHERÉFÉRENCES*(TA_R, C_2)*
    Liste := intersection(Liste1,Liste2)
    Résultat-tempo := liste vide
        POUR chaque e $\in$ Liste, constituant une ligne dans $\mathcal{FI}$ FAIRE
            Résultat-tempo := union(Résultat-tempo, champs 'références' de cette ligne)
        FINFAIRE
    Résultat := intersection(Résultat, Résultat-tempo)
    SI (Résultat = liste vide)
      retourne liste vide
    FINSI
FINFAIRE
Retourne Résultat

Reprenons maintenant notre requête q_2. Le premier relon dans la description de cette requête est $\langle\langle$Profession, PERSONNE, ARTISTE ; 1$\rangle\rangle$. La fonction de correspondance accède ainsi à la table d'accélération de la relation *Profession* (voir table 4.3). Les lignes du fichier inverse contenant une spécialisation d'un relon, ayant *Profession* pour relation et PERSONNE et ARTISTE pour concepts, sont obtenus en effectuant une intersection entre [1,4,9] (pour PERSONNE) et [4,9] (pour ARTISTE). Le résultat de l'intersection des lignes est ainsi [4,9] pour le premier relon (voir algorithme 5). L'union des identificateurs des documents dans les lignes 4 et 9 du fichier inverse donne $[d_2] \cup [d_3]$ = $[d_2,d_3]$. Ainsi, le premier relon peut être satisfait par les documents d_2 et d_3.

De la même manière, pour le second relon de la requête, $\langle\langle$Assis-sur, PERSONNE, CHAISE ; 1$\rangle\rangle$, la table d'accélération de la relation *Assis-sur* donne [3,6] $\cap$ [3,6] = [3,6] pour les concepts PERSONNE et CHAISE (voir figure 4.4). L'accès au fichier inverse permet de déduire que $[d_1,d_2,d_3]$ constituent des documents pertinents pour ce second relon. Par conséquent, seuls les documents vérifiant l'intersection $[d_2,d_3] \cap [d_1,d_2,d_3] = [d_2,d_3]$ sont pertinents pour la requête. Ainsi, tout ce que fait l'algorithme dans ce cas est d'appliquer des opérations ensemblistes sur les tables, ce qui le rend polynomial. En effet, les opérations d'union et d'intersection sont connues pour être polynomiales. Plus précisément, la complexité de cet algorithme est de l'ordre de $O(n^2)$.

Nous pouvons ainsi constater que la technique des fichiers inverses peut être utilisée pour des formalismes plus riches que ceux basés sur les simples mots-clés. Ces formalismes peuvent bénéficier d'une technique largement utilisée dans les systèmes de recherche commerciaux, et connue pour sa praticabilité. Un parallèle peut être fait entre les relons $\langle\langle$Rel, C_1, C_2, ... , C_n ; 1$\rangle\rangle$que nous avons utilisé dans cette section et la notion d'arch introduite dans la section 4.5. En particulier un relon binaire $\langle\langle$Rel, C_1, C_2; 1$\rangle\rangle$peut se transformer en un arch $[C_1]\rightarrow$(Rel)$\rightarrow[C_2]$. Or, les graphes conceptuels peuvent être vus comme un ensemble d'archs liés par application d'une jointure. Afin d'aboutir à un fichier inverse pour les graphes conceptuels, il faudra donc pouvoir représenter ces jointures. La technique des témoins, comme nous allons le montrer dans la section suivante, peut être utilisée pour cet effet.

4.6.5 Utilisation d'un Fichier Inverse en GCs

Nous nous proposons dans cette section de montrer comment une approche algébrique analogue à celle adoptée pour le modèle Booléen peut être utilisée pour les graphes conceptuels. Nous nous inspirons ici de l'extension relationnelle que nous avons appliqué dans la section précédente à la technique des fichiers inverses. Nous verrons que notre approche permet la prise en compte de toutes les caractéristiques de la vision relationnelle que nous avons adopté pour les graphes conceptuels. Ainsi, outre la prise en compte des caractéristiques classiques de la projection, le raisonnement sur les relations est également supporté dans cette approche. Ceci est en conformité avec le système logique à base de graphes conceptuels présenté dans la section 4.4.6. La correction et la complétude de cette démarche sont le résultat de l'application de la technique des témoins pour l'interprétation des concepts existentiels et individuels des graphes.

Dans la suite, nous donnons l'idée générale de notre démarche algébrique, et de l'algorithme qui s'en suit. Une présentation détaillée de la démarche adoptée est ensuite présentée. Un exemple sur un corpus de 5 graphes illustrera le fonctionnement de cette démarche, de la construction du fichier inverse à l'application de la fonction de projection.

a) Idée Générale de l'Approche

Notre but ici est de construire une structure de fichier inverse vérifiant deux conditions. La première est que la procédure de construction doit être réversible, c'est-à-dire que nous devons être capable de reproduire les graphes de départ à partir du fichier inverse. Ce fichier sera composé des parties élémentaires des graphes, à savoir les archs.

Nous utilisons la technique des témoins telle qu'elle a été présentée dans la page 222. Nous adoptons en particulier la convention donnée à la fin de la section correspondante. Cette convention consiste non pas à remplacer les marqueurs quantifiés[39] des concepts par des témoins distincts, mais plutôt à ajouter ces derniers aux concepts en question. Cela permet de reproduire chaque graphe après éclatement. Au niveau du fichier inverse, chaque entrée correspondra à un arch muni des témoins préalablement associés à ses concepts.

Le résultat de notre algorithme de projection sera donc un ensemble de témoins. Cependant, en recherche d'informations, l'objectif est de retrouver l'ensemble des documents pertinents. Afin d'interpréter les résultats de façon plus simple, nous utilisons la notation w_i^j pour les témoins, où i est un identificateur unique du document et j un identificateur unique associé au concept $[C]$ dans i, avec $C(w_i^j)$ interprété vrai par la relation de conformité $Conf$. Grâce à cette démarche, les témoins obtenus restent distincts comme cela est imposé par la technique des témoins. De plus, nous sommes capables de déduire pour chaque témoin, l'index $\chi(d_i)$ auquel il appartient.

La deuxième caractéristique est que le fichier inverse servira de base à définir un opérateur de projection π équivalent à celui de Sowa[40], mais qui ne soit pas appliqué de la requête $\chi(q)$ à chaque index $\chi(d_i)$ du corpus, mais seulement une seule fois à tout le corpus, c'est-à-dire à tout le fichier inverse.

Pour chaque index $\chi(d_i)$, nous associons des témoins w_i^j à ses concepts comme cela a été indiqué ci-dessus. Nous éclatons les graphes index en cherchant leur couverture par rapport au vocabulaire

39. Comme nous l'avons vu dans la la définition 4.3.1, page 162, cette notion regroupe le marqueur générique $\star$, les quantificateurs individuels et le quantificateur $\exists$ lui-même.

40. De part la preuve de complétude du théorème 21, page 219.

terminal $\mathcal{A}_{\mathcal{LR}}$ du langage d'indexation relationnelle. Le résultat est un ensemble d'archs qui seront organisés sous la forme d'un fichier inverse.

Le fichier inverse étant construit, l'algorithme de projection reposera ensuite sur la structure de la requête pour établir les décisions de pertinence. Cette structure peut facilement être identifiée grâce à l'utilisation des témoins:

- De la même manière que pour les index, le graphe de la requête est interprété par la technique des témoins. Ce graphe est ensuite éclaté en un ensemble de sous-graphes constituant chacun une sous-requête. L'éclatement est guidé par les *témoins joints* de la requête, c'est-à-dire les témoins associés à des concepts joints dans la requête. Chaque sous-graphe est constitué des archs auxquels appartient le témoin joint.
- Pour chacun de ces sous-graphes de la requête, nous cherchons leurs spécialisations dans le fichier inverse, dans le but d'obtenir des réponses partielles.
- Nous exprimons alors des conditions de recomposition des réponses partielles obtenues, afin de ne générer que des réponses de projection exactes. Les conditions de recomposition sont basées sur la structure de la requête.

Dans la section suivante, nous donnons une vue d'ensemble de cet algorithme. Ses fonctionnalités ainsi que ses caractéristiques calculatoires seront sommairement présentés.

b) Vue d'Ensemble sur l'Algorithme de Projection

Le fichier inverse est construit à la base du processus d'interprétation logique associant des témoins w_i^j aux concepts de chaque index $\chi(d_i)$, comme cela a été mentionné dans la section précédente. Le même processus d'interprétation logique est appliquée à la requête. Les opérations effectuées par notre algorithme de projection sont données dans la figure 4.43 suivante.

Pour guider le processus de présélection dans le fichier inverse, l'algorithme s'appuie sur les témoins Wit_q^i, qui apparaissent dans plusieurs archs de la requête. Ces archs forment un sous-graphe de la requête, et sont identifiés par la fonction TrouveArchsJoints. Pour chacun de ces archs, où Wit_q^i apparaît, la fonction TrouveSpécialisations permet de spécifier les lignes du fichier inverse ayant pour entrée une spécialisation de l'arch en question. Chaque ligne indique, par le biais des témoins, les documents susceptibles de contenir une spécialisation de cet arch. C'est la fonction ExtraireTémoins qui permet d'accéder à ces témoins dans le fichier inverse. L'application d'une opération d'union sur ces témoins permet, pour chaque arch de la requête, de cerner tous les documents susceptibles de contenir une spécialisation de ce dernier. Afin de trouver, pour chaque témoin joint Wit_q^i, les documents susceptibles d'être à la fois pertinents pour tous les archs joints sur ce témoin, une opération d'intersection est appliquée aux ensembles des témoins retournés pour chacun de ces archs.

À partir des résultats précédents, la fonction TrouveProjectionsPartielles permet de constituer les diverses projections partielles de la requête sur les index des documents. Ainsi, selon le nombre des témoins joints pour lesquels un index donné est en mesure de satisfaire tous les archs correspondants, il est possible de constituer diverses classes de projections partielles. Un document est d'autant plus pertinent pour la requête, qu'il compte le maximum de sous-graphes sur lesquels les archs associés aux témoins joints de la requête se projettent. Un document sera

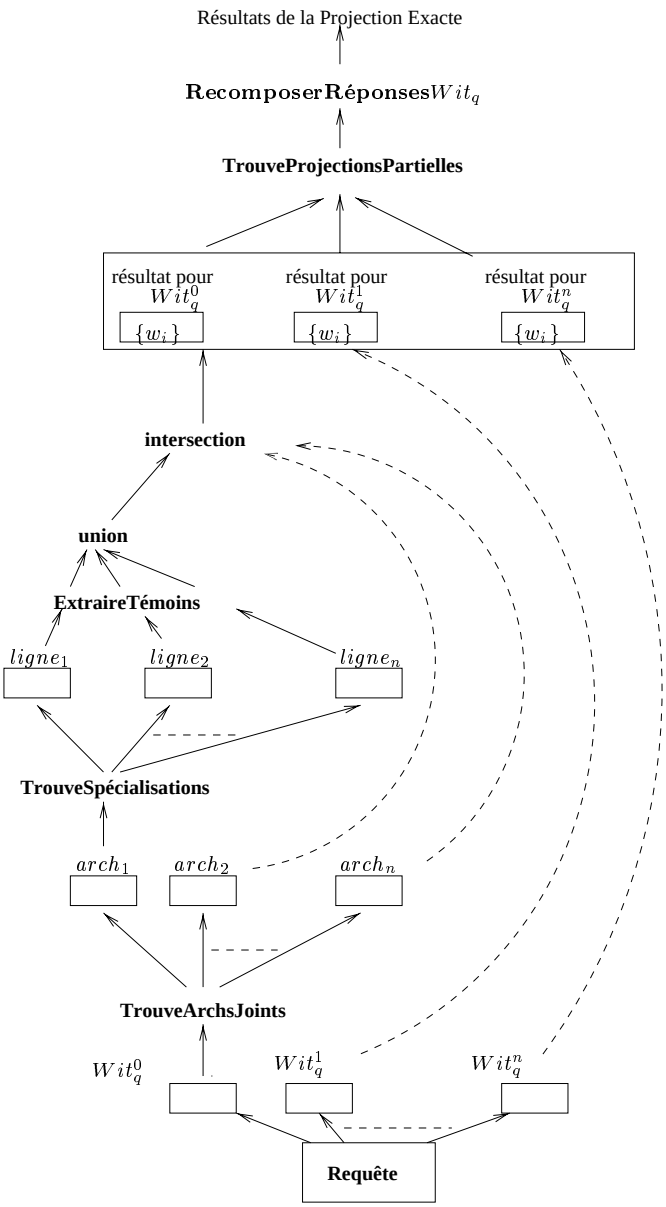


Figure 4.43. Description sommaire du fonctionnement de l'algorithme de projection

totalement pertinent, si tous les sous-graphes correspondants aux témoins joints de la requête ont un spécifique dans le document et s'ils sont disposés dans l'ordre donné par la structure de la requête. C'est la procédure de recomposition (RecomposerRéponsesWit) qui vérifie ce fait.

À la lumière de la figure 4.43, nous pouvons déjà constater que l'algorithme de projection appliqué n'est autre qu'une combinaison d'unions et d'intersections ensemblistes. Ce qui lui confère sa complexité polynomiale. Cette complexité est donnée par:

$(\, |Q| \times \mathcal{O}(\text{TrouveArchsJoints}) \times |A| \times$

$\times\ \mathcal{O}(\text{TrouveSpécialisations}) \times |R| \times$

$\times\ \mathcal{O}(\text{ExtraireTémoins}) \times \mathcal{O}(\text{union}) \times \mathcal{O}(\text{intersection})\ ) +$

$+\ \mathcal{O}(\text{TrouveProjectionsPartielles}) + \mathcal{O}(\text{RecomposerRéponsesWit})$

où par $|Q|$ et $|A|$ nous désignons la cardinalité des ensembles des témoins de la requête et des archs joints par chacun d'entre eux, respectivement. Enfin, $|R|$ est la cardinalité de l'ensemble des lignes du fichier inverse qui contiennent une spécialisation possible de chaque arch. La complexité des deux dernières fonctions est négligeable comparée au premier terme de la somme. Pour l'union et l'intersection, la complexité est au maximum égale à $\mathcal{O}(n^2)$ [41], alors que les autres fonctions constituent des opérations d'accès et ont donc une complexité insignifiante. Nous pouvons ainsi d'ores et déjà constater la complexité polynomiale de l'algorithme de projection proposé.

c) La Prise en Compte des Propriétés des Relations

Nous avons préféré, pour l'implantation de notre système logique à base de graphe conceptuels, appliquer les règles de dérivation relationnelle $\mathcal{R}_{M_{Rel}}$ avant de lancer la projection à proprement dite (voir section 4.4.6). Nous avons justifié notre choix dans la page 200 de la même section. Au fait comme ces propriétés sont intrinsèques aux index, il sera moins coûteux en termes de complexité de calcul de les inclure directement dans le fichier inverse au lieu d'appliquer des règles de dérivation logiques en temps réel, c'est-à-dire lors de l'interrogation. Toutefois, pour la prise en compte réelle de ces propriétés dans une approche algébrique, deux choix se présentent:

- Étendre d'abord les graphes index par application des règles de dérivation $\mathcal{R}_{M_{Rel}}$, et construire ensuite le fichier inverse correspondant;
- Construire le fichier inverse à partir des graphes index originaux, puis appliquer les règles $\mathcal{R}_{M_{Rel}}$ sur le fichier inverse lui-même;

Nous adoptons dans ce qui suit la seconde alternative. Cela permettra d'éviter la modification des index de départ, avec les inconvénients qui en découlent à savoir la perte à jamais du contenu originel de ces index, l'impossibilité de revenir sur certains choix de propriétés des relations, etc.

Ainsi, une fois le fichier inverse construit à partir des index originaux des documents, il lui est appliqué des règles équivalentes à $\mathcal{R}_{M_{Rel}}$, mais ayant la propriétés en plus de tenir compte des témoins associés aux entrées du fichier inverse.

Comme nous l'avons vu dans la section 4.4.6 (propriété 11, page 198), l'application de ces règles se fait en vérifiant que l'information susceptible d'être ajoutée au fichier inverse n'existe pas déjà dans ce dernier. De plus, le système de dérivation basé sur ces règles étant décidable (voir, théorème 17, page 198), il est garantie que l'extension du fichier inverse prendra fin à un moment donné.

d) Présentation Détaillée de l'Approche

Dans cette section nous présentons les structures et l'algorithme utilisés pour la recherche. Nous reprenons en détail les éléments mentionnés dans les trois sections précédentes. Notre approche

41. Par exemple, un ordre sur les ensembles auxquels s'appliquent ces opérations permet de diminuer cette complexité.

algébrique suit les étapes suivantes:

Éclatement des Graphes

Afin de comprendre, d'un point de vue pratique, les raisons qui nous amènent à introduire des témoins, considérons les deux graphes de la figure 4.44.

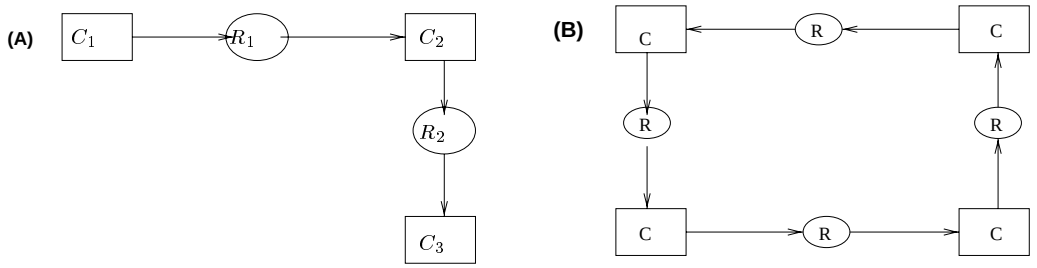


Figure 4.44. Pourquoi les témoins?

La couverture du graphe (A) comporte deux archs. Nous avons ainsi: COUVERTURE(A) = $\{[C_1]\rightarrow(R_1)\rightarrow[C_2], [C_2] \rightarrow (R_2) \rightarrow [C_3]\}$. Ces deux archs sont joints sur le concept $[C_2]$. Dans ce cas, il semble qu'il n'y a pas besoin d'utiliser les témoins. Dans cette couverture, $[C_2]$ apparaît deux fois. Nous pouvons ainsi supposer qu'un *concept commun* à plusieurs archs est un noeud de jointure pour ces derniers. Cependant, cette convention n'est pas valide dans tous les cas. Considérons ainsi le graphe (B) de la même figure. Si nous cherchons la couverture de ce graphe, nous obtenons quatre archs identiques. Il existe plusieurs graphes qui puissent être construits à partir de ces archs. Pour éliminer cette ambiguïté dans la re-construction du graphe de départ, l'utilisation des témoins s'avère nécessaire. Si des témoins distincts sont associés aux divers concepts du graphe (B), il n'y aura qu'une seule solution possible pour construire ce graphe. Il suffit en effet de joindre les archs selon les témoins: si deux témoins sont égaux, leurs concepts associés sont joints. Il s'agit d'une conséquence directe de la technique des témoins et de l'utilisation de l'hypothèse 9 (page 208).

Pour chaque index du corpus, il est ainsi appliqué une opération de recherche de sa couverture. Une fonction $Path$ est utilisée pour cet effet. Cette fonction fournira une stratégie permettant de couvrir le graphe dans un certain ordre, de façon à associer des témoins de la forme w_i^j, où i est l'identificateur du document associé à l'index et j un entier fourni par $Path$.

La procédure d'éclatement des divers graphes index étant achevée, nous obtenons une collection d'archs dont les concepts sont munis de témoins distincts. Pour chaque concept, nous savons exactement le graphe dans lequel il apparaît (c'est le suffixe i dans la notation du témoin). L'avantage de cette représentation est qu'il n'est pas nécessaire de traiter chaque graphe séparément. La collection des graphes index sera considérée globalement.

Construction du Fichier Inverse

Par analogie avec l'approche basée sur les mots-clés, pour identifier dans quels documents de la collection un arch apparaît, au lieu de le chercher dans tout le corpus, nous groupons ensemble

les archs qui sont syntaxiquement égaux, c'est-à-dire qui partagent des concepts communs. Ainsi, étant donné un arch, nous pouvons immédiatement localiser les index $\chi(d)$ contenant cet arch. Chaque ligne du fichier inverse est un arch avec un ensemble de témoins w_i^j associés à ses concepts. Par exemple, pour le graphe (B) de la figure 4.44, au lieu d'avoir quatre archs syntaxiquement identiques $[C]\rightarrow(R)\rightarrow[C]$ avec un nouveau témoin pour chaque occurrence du concept $[C]$ dans le graphe, nous appliquons une union des témoins associés au premier et second concept de chacun de ces archs respectivement. Le fichier inverse contiendra dans ce cas une seule ligne avec un arch $[C]\rightarrow(R)\rightarrow[C]$ et l'union des témoins pour chacun de ses concepts. Un exemple de construction d'un fichier inverse suivra cette section.

Extension du Fichier Inverse

Une fois tous les archs des index placés dans le fichier inverse avec leurs témoins respectifs, ce dernier est étendu par les propriétés syntaxiques et sémantiques des relations. L'extension se fait en conformité avec le choix précédemment établi dans la page 242, c'est-à-dire qu'elle est appliquée au fichier inverse et non pas aux index originaux eux-mêmes.

Les règles $\mathcal{R}_{M_{Rel}}$ permettent la prise en compte de toutes les propriétés des relations contenues dans $\mathcal{K}_s$. À cet effet, l'ensemble $\mathcal{E}$ des relation d'endiguement du canon est utilisé. À chaque fois qu'il existe un ensemble de relations dans le fichier inverse qui impliquent une autre relation, cette dernière est ajoutée au fichier inverse avec ses ensembles de témoins associés. Par exemple, pour prendre en compte le fait qu'une relation binaire R est symétrique, chaque fois qu'on trouve un arch $a = [C_1]\rightarrow(\mathrm{R})\rightarrow[C_2]$ contenant R dans le fichier inverse, nous créons un nouveau arch b dans le quel on prend R comme relation, mais pour lequel les concepts de l'arch a seront inversés, autrement dit $b = [C_2]\rightarrow(\mathrm{R})\rightarrow[C_1]$. Les témoins associés aux concepts de l'arch a dans le fichier inverse suivent cette inversion. Si l'arch b existe déjà dans le fichier inverse, les témoins associés à ses concepts seront ajoutés à la ligne correspondante dans l'ordre, autrement une nouvelle ligne sera créée dans le fichier inverse.

De façon similaire, le treillis des types de relations $\mathcal{T}_r$ peut être utilisé pour enrichir les décisions de pertinence de la projection classique. Pour cet effet, un opérateur d'expansion [42] des archs peut être appliqué aux diverses entrées du fichier inverse. Il s'agit d'ajouter pour chaque arch qui s'y trouve, tous les archs construits en remplaçant la relation sous-jacente avec celles qui lui sont plus générales dans $\mathcal{T}_r$. Ceci en conformité avec les propriétés et les caractéristiques de la relation d'ordre partiel $\leq_\sigma$. Par exemple, si dans $\mathcal{T}_r$ nous avons deux relations R_1 et R_2 tel que $R_2 \leq_\sigma R_1$, alors pour chaque ligne du fichier inverse où un arch a avec la relation R_2 apparaît, nous créons une nouvelle ligne avec un nouveau arch b construit de la façon suivante: R_2 est substituée par sa généralisation R_1; Désignons par $g_j^c(R_2)$, le concept associé au j^{eme} type de concept de l'arch g [43]; nous créons un nouveau concept $b_i^c(R_1)$ dans la nouvelle ligne, auquel nous lui associons l'ensemble des témoins que nous trouvons pour $a_{\sigma(i)}(R_2)$, où σ est la fonction injective associée à $\leq_\sigma$ (i = 1..arité de R_1). Si l'arch b existe déjà dans le fichier inverse, les témoins associés à ses concepts seront ajoutés à la ligne correspondante, autrement une nouvelle ligne sera créée dans le fichier inverse.

Enfin, rappelons que dans les deux cas, l'extension du fichier inverse se fait en conformité avec la propriété 11 de la page 198, c'est-à-dire que l'ajout des témoins et des archs ne peut se confirmer qu'après avoir vérifier leur non-existence dans le fichier inverse en construction. Cette procédure

42. C'est l'opérateur dual à la restriction sur les relations (Voir définition 4.4.17, page 195.
43. Nous rappelons que $g_i(r)$ désigne le type de concept du i^{eme} noeud concept voisin de r dans g.

d'extension prend fin lorsqu'il est vrai qu'aucune information (témoins ou archs) supplémentaire ne peut être inférée par les règles de dérivation. Comme nous l'avons déjà signalé, de part la décidabilité de notre approche, il est certain que ce processus prendra fin.

Construction des Tables d'Accélération

La notion de table d'accélération introduite dans la section 4.6.4 est appliquée ici de la même manière. Ces tables sont introduites pour rendre disponible, étant donné un arch, l'ensemble de toutes ses spécialisations apparaissant dans le fichier inverse. Tout comme le fichier inverse, ces tables sont construites lors du processus d'indexation du corpus.

Il existe une table d'accélération pour chaque type de relation r dans $\mathcal{T}_r$. Elle contient n colonnes, où n est l'arité de r. Pour chaque arch a du fichier inverse contenant r, nous cherchons les graphes qui constituent des généralisations de ce dernier, mais qui soient conformes à la signature Σ_r de r. Ces généralisation tiennent compte aussi bien du treillis des types de concepts $\mathcal{T}_c$, que du treillis sur l'ensemble $\mathcal{M}$ des marqueurs.

Chaque graphe résultant g est ajouté à la table d'accélération comme une structure de la forme $(g_1^c(r)[k],\ g_2^c(r)[k],\ \ldots,\ g_n^c(r)[k])$ formée par les concepts $g_i^c(r)$ qui lui sont associés et par leur référence à la ligne k du fichier inverse dans laquelle l'arch g apparaît comme entrée. Le i^{eme} $g_i^c(r)[k]$ est ajouté à la i^{eme} colonne de la table. La fonction ord associée aux graphes conceptuels assure le bon ordre. Si le concept $g_i^c(r)$ existe déjà dans la i^{eme} colonne de la table, la référence [k] est ajoutée à l'ensemble des références déjà associées à ce concept.

Pour un arch b donné ayant comme relation r et comme concepts $(b_i^c(r),\ b_2^c(r),\ \ldots,\ b_n^c(r))$, les archs pertinents qui lui correspondent dans le fichier inverse (ceux qui constituent une spécialisation de b) sont déterminés en appliquant une intersection sur les références associées à chaque $b_i^c(r)$ dans la table d'accélération du type de relation r.

Il est à noter que la procédure de construction des tables d'accélération suit dans l'ordre chronologique celle relative à l'extension du fichier inverse. Il s'agit la de s'assurer que les nouvelles lignes introduites dans le fichier inverse sont prises en compte au niveau des tables d'accélération.

Implantation de la Projection

Comme nous l'avons déjà signalé dans la page 240, notre algorithme de projection a deux caractéristiques: (i) il est guidé par la requête, puisque la structure de cette dernière est utilisée pour diriger la correspondance. (ii) il ne consiste que de combinaisons d'opérateurs ensemblistes (des intersections et des unions), ce qui le rend polynomial (voir figure 4.43).

Quand un utilisateur entre sa requête q, un témoin Wit_q^i est associé à chaque concept $[C_q]$ de $\chi(q)$. Cette requête est éclatée de la même manière que $\chi(d)$, en utilisant la fonction $Path$[44]. Nous obtenons ainsi un ensemble d'archs associés à la requête.

La stratégie de recherche est élaborée selon la structure de cette requête. Si cette dernière consiste d'un seul arch, la recherche des documents pertinents pour cette requête est une procédure facile: à partir de la table d'accélération associée à la relation de cet arch, nous obtenons les lignes

44. Cette opération est généralement réalisée sans coût au niveau de l'interface d'interrogation, puisqu'elle est appliquée au fur et à mesure de l'introduction de la requête [Mec95c, MMFB97].

du fichier inverse contenant une spécialisation de l'arch en question. Ces archs correspondent ainsi à des documents pertinents. Pour identifier ces derniers, nous extrayons l'identificateur i de chaque témoin associé à un concept de ces divers archs. Nous appliquons une union sur ces identificateurs, permettant ainsi d'obtenir les documents pertinents. Dans ce cas l'approche est équivalente à celle utilisée pour les mots-clés.

Si la requête contient deux archs ou plus, la stratégie de recherche est guidée par les témoins joints de la requête, c'est-à-dire par les témoins qui apparaissent dans plus d'un arch de celle-ci. Pour chaque témoin joint Wit_q^i, l'utilisation des tables d'accélération permet d'identifier tous les archs du fichier inverse constituant une spécialisation de chaque arch qu'il joint. Nous avons cependant besoin d'appliquer une plus forte sélection, puisque ces archs constituent une condition nécessaire mais non suffisante pour obtenir une projection d'une partie de la requête. Il faut en effet que chaque jointure entre des archs de la requête ait un correspondant dans le document. Nous appliquons alors une intersection des ensembles des témoins associés à ces archs selon la façon dont ils sont joints. Cette intersection étant effectuée, pour chaque témoin joint Wit_q^i de la requête, nous aurons un ensemble de témoins apparaissant dans des documents, considérés alors comme des candidats potentiels à la projection. En effet, pour chaque Wit_q^i, nous obtenons les sous-graphes des index correspondant à la projection d'une partie de la requête. Cette partie correspond aux archs de la requête dans lesquels Wit_q^i apparaît.

Algorithme 6 (Algorithme de Projection)

Début
1) POUR chaque témoin joint Wit_q^i *de la requête FAIRE*
  $ProjectionPartielle_{Arch_q}$ = *tous les témoins*
  $ArchsJoints_q$ = TROUVEARCHSJOINTS(Wit_q^i)
  2) POUR chaque $Arch_q \in ArchsJoints_q$ *FAIRE*
    $LignesSpécifiquesDeArch_q$ = TROUVESPÉCIALISATIONS$(Arch_q)$
    TémoinsAssociés = $\emptyset$
    3) POUR chaque $LigneArch_q \in LignesSpécifiquesDeArch_q$ *FAIRE*
      $Témoins_{LigneArch_q}$ = EXTRAIRETÉMOINS$(LigneArch_q)$
      $TémoinsAssociés = Témoins_{LigneArch_q} \cup TémoinsAssociés$
    FINFAIRE
    $ProjectionPartielle_{Arch_q} = ProjectionPartielle_{Arch_q} \cap TémoinsAssociés$
  FINFAIRE
$ProjectionPartielle_q$ = TROUVEPROJECTIONSPARTIELLES$(ProjectionPartielle_{Arch_q})$
$ProjectionExacte_q$ = RECOMPOSERRÉPONSESWIT$_q(ProjectionPartielle_q)$
FINFAIRE
Fin

Une fonction TROUVEPROJECTIONSPARTIELLES permet de retrouver à partir des résultats précédents toutes les projections partielles possibles de la requête sur les index des documents. Ceci est important en recherche d'informations, puisque nous pouvons alors introduire une notion de degré de pertinence selon le nombre de parties de la requête qui se projettent sur les index des documents, et selon le nombre et la taille des archs qui les constituent.

Enfin pour retrouver les documents qui répondent à la totalité de la requête, c'est-à-dire pour lesquels il existe une projection totale de $\chi(q)$, nous utilisons une fonction appelée RECOMPOSER-

RÉPONSESWIT$_q$ permettant de vérifier que les témoins obtenus sont bien liés ensemble dans les documents de façon conforme à la structure de la requête. Nous adoptons alors l'algorithme 6. Cet algorithme utilise les symboles suivants:

TrouveArchsJoints(Wit$_q^i$): procédure pour retrouver tous les archs a de la requête dans lesquels Wit$_q^i$ apparaît. Cette procédure donne aussi le concept $a_i^c(r)$ de l'arch a associé à Wit$_q^i$.

TrouveSpécialisations(Arch$_q$): procédure pour retrouver, dans la table d'accélération de la relation de Arch$_q$, les lignes du fichier inverse contenant des spécialisations de Arch$_q$.

ExtraireTémoins(LigneArch$_q$): fonction qui extrait les témoins associés au concept $a_i^c(r)$ de l'arch a donné dans LigneArch$_q$.

TrouveProjectionsPartielles(ProjectionPartielle$_{Arch_q}$): fonction qui extrait les identificateurs de document des témoins correspondant à l'ensemble ProjectionPartielle$_{Arch_q}$.

RecomposerRéponsesWit$_q$(ProjectionPartielle$_q$:) fonction qui vérifie que les témoins obtenus sont liés dans le document dans le bon ordre déterminé par la structure de la requête.

Comme cela a été auparavant mentionné, la complexité de notre algorithme est fonction de la complexité des opérations d'intersection et d'union. Si aucune optimisation n'est appliquée à ces opérateurs (il peut s'agir de trier les ensembles manipulés), la complexité de notre algorithme est de l'ordre de $\mathcal{O}(n^4)$.

Afin d'illustrer comment fonctionne notre approche algébrique, nous donnons dans la section suivante un exemple de corpus de 5 documents. La phase d'indexation de ce corpus regroupera les procédures de construction du fichier inverse, de son extension et enfin l'élaboration des tables d'accélération. Le corpus ainsi indexé, il pourra être interrogé par les utilisateurs. La fonction de correspondance s'appuiera sur l'algorithme de projection ici présenté.

e) Un Exemple d'Exécution de l'Algorithme

Supposons que nous sommes en présence d'un ensemble de cinq documents dont les graphes index sont illustrés dans la figure 4.46 suivante.[45] Supposons que les treillis $\mathcal{T}_c$ et $\mathcal{T}_r$ sont ceux de la figure 4.45. De plus, considérons les signatures des types de relations suivantes:

- $\Sigma_{R_0} = (R_0, 2, C_2, C_3)$
- $\Sigma_{R_1} = (R_1, 2, C_1, C_2)$
- $\Sigma_{R_3} = (R_3, 2, C_1, C_2)$.

La phase d'indexation regroupe la phase d'éclatement des graphes, la construction du fichier inverse, l'application des règles de dérivation relationnelles à ce fichier inverse, et enfin la construction des tables d'accélération.

45. C'est pour des raisons de présentation que nous avons préféré donner comme exemple des graphes abstraits et non pas des index d'une collection réelle. En effet, dans ce dernier cas, les tableaux qui vont suivre seront trop chargés.

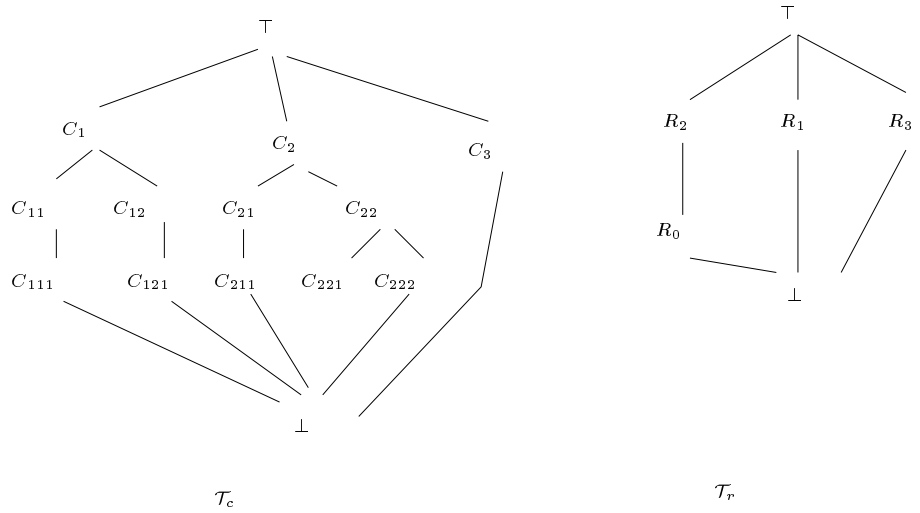


Figure 4.45. Les treillis des types de concepts et de relations

Éclatement des Graphes

La procédure d'éclatement transforme les cinq graphes conceptuels en un ensemble d'archs. À chaque concept est associé un témoin distinct. Par exemple, pour le premier graphe index $\chi(d_1)$, nous avons le témoin w_1^1 pour le concept $[C_1]$, le témoin w_1^2 pour le concept $[C_{22}]$, et le témoin w_1^3 pour le concept $[C_3]$ (voir figure 4.46).

Pour des raisons de clarté de présentation, nous traitons les témoins par rapport aux documents dans lesquels ils apparaissent. Ainsi, par la notation $d_i[j]$, nous désignons le témoin w_i^j. Par exemple, nous représentons l'ensemble des témoins de $\chi(d_1)$, $\{w_1^1, w_1^2, w_1^3\}$, par $d_1 : [1, 2, 3]$. Cette convention de présentation adoptée, nous pouvons construire notre fichier inverse.

Construction du Fichier Inverse

Afin d'illustrer plus clairement la construction de notre fichier inverse, nous adoptons une approche pédagogique, où les résultats partiels de cette construction sont donnés pour les index $\chi(d_1)$ et $\chi(d_2)$.

Dans le fichier inverse, il existe une ligne pour chaque arch a. Les concepts $a_i^c(r)$ de l'arch a et leurs témoins associées sont représentés verticalement, selon leur position dans l'arch: dans l'ordre donné par la fonction *ord* (voir section 4.4.1, page 182). Nous rappelons que $a_i^c(r)$ désigne le concept associé au i^{eme} type de concept du noeud voisin à r dans l'arch a.

Nous supposons que les graphes sont éclatés dans l'ordre, de $\chi(d_1)$ vers $\chi(d_5)$. Pour l'index $\chi(d_1)$, nous avons une couverture COUVERTURE($\chi(d_1)$) constituée de deux archs. Ces derniers étant différents, il leur sera assigné chacun une ligne dans le fichier inverse. Ces deux archs sont ainsi ajoutés au fichier inverse. De plus, à chaque concept, nous ajoutons les témoins qui lui correspondent. Le premier arch est le suivant: $[C_1] \rightarrow (R_1) \rightarrow [C_{22}]$. Les témoins sont d_1:[1] pour $[C_1]$,

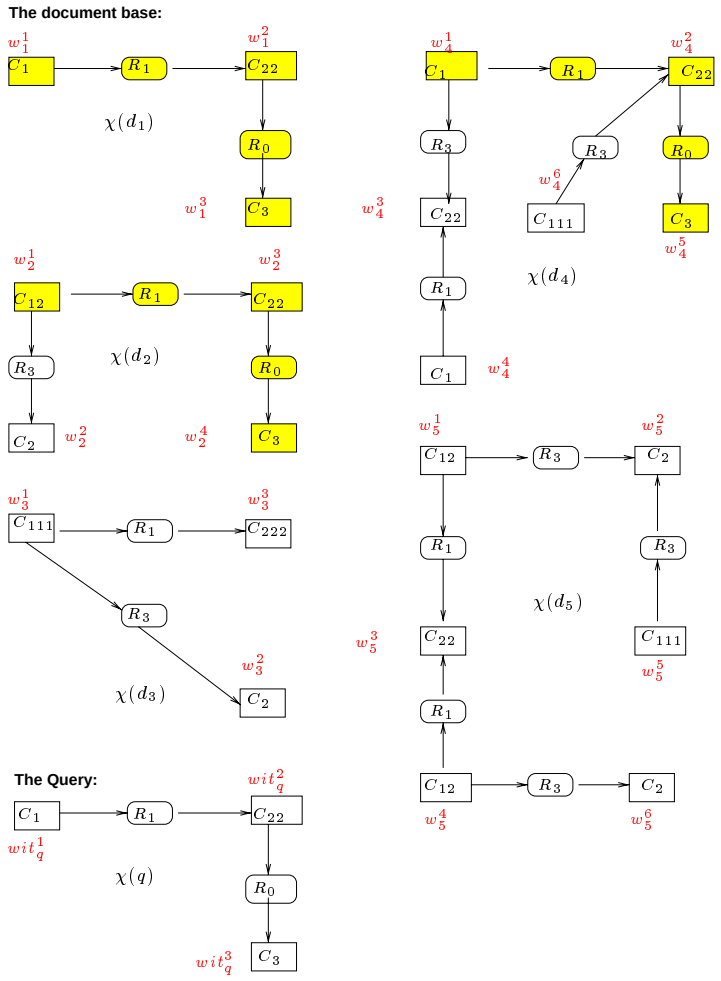


Figure 4.46. Un corpus de cinq documents

et d_1:[2] pour $[C_{22}]$. Pour le second arch, $[C_{22}] \rightarrow (R_0) \rightarrow [C_3]$, nous avons les témoins d_1:[2] pour $[C_{22}]$ et d_1:[3] pour $[C_3]$. (voir table 4.5).

Numéro Ligne	*Relation*	*Concepts*	*Témoins*
1	R_1	C_1:	d_1 : [1]
		C_{22}:	d_1 : [2]
2	R_0	C_{22}:	d_1 : [2]
		C_3:	d_1 : [3]

Table 4.5. Fichier Inverse après analyse des archs du premier index

Le second index $\chi(d_2)$ admet une couverture de trois archs. Seules deux lignes seront ajoutées au fichier inverse partiel de la figure 4.5 précédente, puisque l'arch $[C_{22}]\rightarrow(R_0)\rightarrow[C_3]$ est déjà présent dans cette structure intermédiaire. Nous obtenons alors la table 4.6 suivante.

Numéro Ligne	*Relation*	*Concepts*	*Témoins*
1	R_1	C_1:	$d_1 : [1]$
		C_{22}:	$d_1 : [2]$
2	R_0	C_{22}:	$d_1 : [2], d_2 : [3]$
		C_3:	$d_1 : [3], d_2 : [4]$
3	R_1	C_{12}:	$d_2 : [1]$
		C_{22}:	$d_2 : [3]$
4	R_3	C_{12}:	$d_2 : [1]$
		C_2:	$d_2 : [2]$

Table 4.6. Fichier inverse après analyse des archs des deux premiers documents

Nous considérons ensuite le reste des graphes index $\chi(d_3)$, $\chi(d_4)$ et $\chi(d_5)$. Le fichier inverse final des cinq graphes index du corpus est illustré dans la table 4.7.

Extension du Fichier Inverse

Comme nous l'avons précédemment expliqué, l'extension du fichier inverse se base d'une part sur le treillis des types de relations, et d'autre part sur les connaissances contenues dans $\mathcal{K}_s$. Afin de ne pas trop surcharger cet exemple, nous nous contentons ici de montrer le principe sur une seule propriété. Par exemple, du treillis $\mathcal{T}_r$ de la figure 4.45, la propriété $R_0 \leq_\sigma R_2$ peut être déduite. De plus, considérons que la signature de R_2 est donnée par $\Sigma_{R_2} = (R_0, 2, C_2, C_3)$ (vérifiant ainsi les propriétés de la relation d'ordre partiel $\leq_\sigma$). Ces deux relations ayant la même arité, une nouvelle ligne exploitant cette relation de spécialisation peut être ajoutée au fichier inverse de la façon indiquée dans la table 4.8.

Plus généralement, le fichier inverse complet est construit à la suite de trois actions. La première prend en compte les archs de tous les index $\chi(d_i)$ du corpus., la seconde action exploite le treillis des types de concepts pour enrichir le fichier déjà existant. Enfin, la troisième action affine encore plus ce fichier inverse en exploitant les connaissances de l'ensemble $\mathcal{K}_s$.

Construction des Tables d'Accélération

Les tables d'accélération sont construites à partir des archs apparaissant dans le fichier inverse étendu, du treillis des types de concepts $\mathcal{T}_c$ et des graphes signatures de $\mathcal{S}_r$. Il existe une table d'accélération pour chacun des types de relation, R_0, R_1, R_3. Nous analysons les archs de chaque ligne du fichier inverse. Considérons la première ligne de ce fichier. Elle consiste en un arch $a = [C_1]\rightarrow(R_1)\rightarrow[C_{22}]$. Le graphe signature $g_{\Sigma_{R_1}}$ est le graphe conceptuel bien formé le plus général contenant la relation R_1. En d'autres termes, il n'existe pas de graphe conceptuel bien formé

Numéro Ligne	*Relation*	*Concepts*	*Témoins*
1	R_1	C_1:	$d_1 : [1], d_4 : [1, 4]$
		C_{22}:	$d_1 : [2], d_4 : [2, 3]$
2	R_0	C_{22}:	$d_1 : [2], d_2 : [3], d_4 : [2]$
		C_3:	$d_1 : [3], d_2 : [4], d_4 : [5]$
3	R_1	C_{12}:	$d_2 : [1], d_5 : [1, 4]$
		C_{22}:	$d_2 : [3], d_5 : [3]$
4	R_3	C_{12}:	$d_2 : [1], d_5 : [1, 4]$
		C_2:	$d_2 : [2], d_5 : [2, 6]$
5	R_1	C_{111}:	$d_3 : [1]$
		C_{222}:	$d_3 : [3]$
6	R_3	C_{111}:	$d_3 : [1], d_5 : [5]$
		C_2:	$d_3 : [2], d_5 : [2]$
7	R_3	C_1:	$d_4 : [1]$
		C_{22}:	$d_4 : [3]$
8	R_3	C_{111}:	$d_4 : [6]$
		C_{22}:	$d_4 : [2]$

Table 4.7. Fichier inverse final des graphes index du corpus

Row number	*Relation*	*Concepts*	*Witnesses*
9	R_2	C_{22}:	$d_1 : [2], d_2 : [3], d_4 : [2]$
		C_3:	$d_1 : [3], d_2 : [4], d_4 : [5]$

Table 4.8. Une nouvelle ligne prenant en compte la propriété $R_0 \leq_\sigma R_2$

qui soit une généralisation de $g_{\Sigma_{R_1}}$. Tous les graphes constituant une spécialisation de $g_{\Sigma_{R_1}}$ et une généralisation de l'arch a peuvent se projeter sur ce dernier. Ces graphes sont obtenus en remplaçant chaque concept $[C_i]$ de l'arch a par ses généralisations, en suivant le treillis $\mathcal{T}_c$ [46]. La limite maximale de généralisation étant $g_{\Sigma_i(R_1)}$, le i^{eme} type de concept du graphe signature $g_{\Sigma_{R_1}}$: pour $[C_1]$ nous obtenons $[C_1]$, alors que pour $[C_{22}]$ nous avons comme généralisations possibles $[C_{22}]$ et $[C_2]$. Dans ce cas, les graphes résultats sont: $[C_1] \rightarrow (R_1) \rightarrow [C_{22}]$ et $[C_1] \rightarrow (R_1) \rightarrow [C_2]$ (voir table 4.9).

46. Dans cet exemple, les graphes conceptuels ne sont munis que de concepts génériques. Nous pouvons toutefois bien sûr avoir des concepts individuels ou encore des concepts dont le marqueur est une variable. Dans ce cas, la recherche des généralisations de chaque concept $[c] = [Type_c, m]$ se fait en exploitant le treillis des types de concepts $\mathcal{T}_c$, mais aussi le treillis des marqueurs. En particulier, $\forall m \in \zeta$, $m \leq \star$. Dans tous les cas, le concept résultant doit être conforme aux signatures des relations, mais vérifiant aussi la relation de conformité.

$a_1(R_1)$	*Liste des lignes du fichier inv.*	$a_2(R_1)$	*Liste des lignes du fichier inv*
C_1	[1]	C_2	[1]
		C_{22}	[1]

Table 4.9. Table d'accélération partielle pour la relation R_1 (analyse de la première ligne du fichier inverse)

La deuxième occurrence de la relation R_1 apparaît à la troisième ligne du fichier inverse. L'arch dans ce cas est $[C_{12}] \rightarrow (R_1) \rightarrow [C_{22}]$. Les graphes constituants des spécialisations du graphe signature $g_{\Sigma_{R_1}}$ et des généralisations de l'arch associé sont: $[C_{12}] \rightarrow (R_1) \rightarrow [C_{22}]$, $[C_1] \rightarrow (R_1) \rightarrow [C_{22}]$, $[C_{12}] \rightarrow (R_1) \rightarrow [C_2]$ et $[C_1] \rightarrow (R_1) \rightarrow [C_2]$ (voir table 4.10).

$a_1(R_1)$	*Liste des lignes du fichier inv.*	$a_2(R_1)$	*Liste des lignes du fichier inv.*
C_1	[1,3]	C_2	[1,3]
C_{12}	[3]	C_{22}	[1,3]

Table 4.10. Table d'accélération partielle pour R_1 (analyse des trois premières lignes du fichier inverse)

La dernière apparition de la relation R_1 dans le fichier inverse se trouve au niveau de la cinquième ligne. En adoptant le même processus que celui effectué pour la première et troisième ligne, nous obtenons la table d'accélération finale de la table 4.11 suivante pour la relation R_1.

$a_1(R_1)$	*Liste des lignes du fichier inv.*	$a_2(R_1)$	*Liste des lignes du fichier inv.*
C_1	[1,3,5]	C_2	[1,3,5]
C_{12}	[3]	C_{22}	[1,3,5]
C_{11}	[5]	C_{222}	[5]
C_{111}	[5]		

Table 4.11. Table d'accélération pour la relation R_1

Après analyse de toutes les lignes du fichier inverse, la table d'accélération de la relation R_0 est obtenue à partir de la ligne 2 du fichier inverse. (voir table 4.12).

Enfin la table d'accélération de la relation R_3 est obtenue après analyse des lignes 4, 6, 7 et 8 du fichier inverse (voir table 4.13).

$a_1(R_0)$	*Liste des lignes du fichier inv.*	$a_2(R_0)$	*Liste des lignes du fichier inv.*
C_2	[2]	C_3	[2]
C_{22}	[2]		

Table 4.12. Table d'accélération pour la relation R_0

$a_1(R_3)$	*Liste des lignes du fichier inv.*	$a_2(R_3)$	*Liste des lignes du fichier inv.*
C_1	[4,6,7,8]	C_2	[4,6,7,8]
C_{12}	[4]	C_{22}	[7,8]
C_{11}	[6,8]		
C_{111}	[6,8]		

Table 4.13. Table d'accélération pour la relation R_3

Le Processus de Recherche

Les tables d'accélération et le fichier inverse seront utilisés lors du processus de recherche, comme cela a été présenté dans la section précédente. Supposons ainsi qu'un utilisateur pose la requête $\chi(q)$ indiquée dans la figure 4.46. Il existe un seul témoin joint q : [2] apparaissant à la fois dans l'arch $[C_1]\rightarrow(R_1)\rightarrow[C_{22}]$ et dans l'arch $[C_{22}]\rightarrow(R_0)\rightarrow[C_3]$.

La procédure TROUVEARCHSJOINTS appliquée au témoin q : [2] analyse chaque arch de $\chi(q)$, dans lequel ce témoin apparaît. Le premier arch dans lequel le témoin joint q : [2] apparaît est le suivant: $A = [C_1]\rightarrow(R_1)\rightarrow[C_{22}]$. Le témoin q : [2] est associé dans ce cas au concept $[C_{22}]$.

Afin de trouver toutes les spécialisations de l'arch A, nous utilisons la table d'accélération de la relation contenue dans cet arch, à savoir la table d'accélération de la relation R_1. La fonction d'ordre *ord* nous permet d'associer les concepts $A_i^c(R_1)$ à leur correspondant $a_i^c(R_1)$ de la table d'accélération. Nous trouvons ainsi que les archs sur lesquels le concept $[C_1]$ peut se projeter se trouvent dans les lignes 1, 3 et 5 du fichier inverse. Pour le concept $[C_{22}]$, nous trouvons de la même manière les lignes [1,3,5].

Il ne reste plus alors qu'à appliquer une intersection sur ces lignes ainsi obtenues: $[1,3,5] \cap [1,3,5] = [1,3,5]$. La sémantique de l'intersection est la suivante: trouver toutes les lignes du fichier inverse contenant des archs g, pour lesquels $g_1^c(R_1) \leq [C_1]$ et $g_2^c(R_1) \leq [C_{22}]$. Ces archs correspondent à ceux sur lesquels l'arch A de la requête $\chi(q)$ peut se projeter.

Le deuxième arch de la requête dans lequel le témoin q : [2] apparaît est l'arch $B = [C_{22}] \rightarrow (R_0) \rightarrow [C_3]$. De la même façon que pour l'arch A, nous cherchons dans la table d'accélération de la relation R_0, les lignes du fichier inverse contenant des archs sur lesquels le concept $[C_{22}]$ peut se projeter. Nous trouvons dans ce cas que seule la ligne [2] du fichier inverse vérifie une telle caractéristique. Similairement, cette même ligne [2] est la seule qui contient un arch sur lequel le concept $[C_3]$ peut se projeter. En faisant l'intersection $[2] \cap [2] = [2]$, nous déduisons qu'il existe une

seule ligne du fichier inverse qui contient un arch sur lequel $[C_{22}]\rightarrow(R_0)\rightarrow[C_3]$ peut être projeté.

Au niveau de la requête, le témoin joint q : [2] est associé au concept $[C_{22}]$. Il s'agit donc du concept $A_2^c(R_1)$ pour l'arch A et $B_1^c(R_0)$ pour l'arch B. En d'autres termes, le concept $[C_{22}]$ correspond au deuxième concept de l'arch A et au premier concept de l'arch B dans l'ordre donné par la fonction *ord*. Nous tiendrons compte de cette jointure dans la suite de l'application de l'algorithme de projection.

Pour trouver la liste des documents dont le graphe index comporte des archs sur lesquels ceux contenus dans la requête peuvent se projeter, nous effectuons en premier lieu une union sur les listes des témoins associées au concept $A_2^c(R_1)$, dans les lignes 1, 3 et 5 du fichier inverse. Ces lignes ont été trouvées pour l'arch A de la requête:

$[d_1 : [2], d_4 : [2, 3]] \cup [d_2 : [3], d_5 : [3]] \cup [d_3 : [3]] =$

$[d_1 : [2], d_2 : [3], d_3 : [3], d_4 : [2, 3], d_5 : [3]]$.

Pour le deuxième arch B de la requête, nous avons obtenu une seule ligne du fichier inverse contenant un arch sur lequel B peut se projeter. Il s'agit de la ligne [2]. La fonction EXTRAIRE-TÉMOINS permet d'extraire la liste des témoins correspondant au concept $B_1^c(R_0)$ dans la ligne [2]. Nous obtenons alors la liste suivante:

$[d_1 : [2], d_2 : [3], d_4 : [2]]$.

Seul l'ensemble des documents pour lesquels il existe une jointure sur le concept $[C_{22}]$ associé à q : [2] seront considérés comme des réponses potentiellement correctes pour la requête $\chi(q)$. Pour trouver ces documents pertinents possibles, c'est-à-dire les projections partielles correspondant aux archs de la requête dans lesquels le témoin joint q : [2] apparaît, nous avons seulement à appliquer une intersection des deux dernières listes:

$[d_1 : [2], d_2 : [3], d_3 : [3], d_4 : [2, 3], d_5 : [3]] \cap [d_1 : [2], d_2 : [3], d_4 : [2]] =$

$[d_1 : [2], d_2 : [3], d_4 : [2]]$.

Comme il n'existe pas d'autres témoins joints dans la requête, nous obtenons le résultat final de l'algorithme de projection, à savoir que les documents d_1, d_2 et d_4 sont renvoyés à l'utilisateur.

Analyse des Résultats

Il est intéressant à ce niveau d'observer les résultats de notre opérateur de projection. Reprenons ainsi le résultat de la dernière intersection, i.e, $[d_1 : [2], d_2 : [3], d_4 : [2]]$. Les témoins permettent d'identifier le résultat de la projection de la requête sur les index des documents correspondants.

Par exemple, d_1 : [2] a l'interprétation suivante: la projection de la requête $\chi(q)$ sur $\chi(d_1)$ existe et elle correspond au graphe formé par les archs de $\chi(d_1)$ qui contiennent le concept ayant pour témoin d_1 : [2]. Dans la figure 4.46, les résultats de la projection de la requête $\chi(q)$ sur les divers index du corpus sont donnés en grisé.

Plus généralement, un résultat de la forme d_i:$[n_1, n_2, \ldots, n_n]$, indique que le résultat de la projection sur le graphe index du document d_i est construit à partir des archs dans lesquels les divers n_j apparaissent, et en établissant les jointures qui s'imposent.

Si au niveau de l'indexation, nous faisons en sorte que les témoins soient associés à une partie ou à un passage d'un document, il sera possible de retourner à l'utilisateur uniquement les parties

qui répondent à sa requête, c'est-à-dire celles correspondant à un témoin retourné par notre algorithme. Ceci est très utile puisque l'utilisateur s'attend souvent à ce qu'on lui retourne directement l'information recherchée, et non pas le document qui la contient. Dans les applications hypermédias, une telle caractéristique est nécessaire.

À Propos de l'Algorithme de Projection

Plusieurs approches dans la littérature proposent un algorithme de projection basé sur une décomposition de la requête en sous-requêtes, puis un mécanisme de recomposition des réponses selon les résultats obtenus pour chacune de ces sous-requêtes. Citons comme exemple, l'algorithme de M. Mechkour [Mec95c], ou encore celui de B. Carbonneill [Car96]. Dans ces algorithmes, il est toujours question d'appliquer une projection de chaque sous-requête sur toute la base de faits. L'objectif étant de retrouver les graphes potentiellement pertinents. Cette opération est coûteuse en temps de calcul. Dans notre algorithme, pour chaque sous-requête, nous identifions directement les graphes qui lui sont pertinents. La différence en termes de temps de traitement est donc à ce niveau.

Il est par ailleurs intéressant de signaler que l'algorithme de projection que nous avons proposé fonctionne pour n'importe quel type de graphes. Aucune restriction n'est ainsi faite sur les propriétés ou les caractéristiques que doivent présenter les graphes conceptuels considérés par cet algorithme. Ceci est important, puisque dans la littérature, à chaque fois qu'un algorithme polynomial est proposé, le résultat est tributaire du type des graphes utilisés. Ainsi, dans [Mug93], il a été montré qu'une complexité polynomiale peut être atteinte avec une forme arborescente du graphe conceptuel à projeter. Notre algorithme est donc intéressant de ce point de vue.

De plus, dans l'approche algébrique que nous avons exposé, nous avons considéré la notion d'arch comme le type d'entrée du fichier inverse. Toutefois, il reste toute à fait envisageable de remplacer cette entrée par n'importe quel graphe conceptuel. Tout ce qu'il faudra faire est de modifier les tables d'accélération en conséquence, de façon à pouvoir identifier directement les spécialisations d'un sous-graphe donné.

4.7 Résumé et Conclusion

Dans ce chapitre, nous avons appliqué notre modèle d'indexation relationnel au formalisme des graphes conceptuels. La notion de canon a été ainsi adaptée à une approche d'indexation relationnelle. L'intérêt d'une telle extension est de pouvoir améliorer les décisions de pertinence de ce formalisme, en exploitant toutes les propriétés sémantiques des relations représentées au niveau du canon. Le résultat serait un modèle relationnel pour les graphes conceptuels, dont les caractéristiques permettraient d'appréhender les nouvelles applications auxquelles est dédiée la recherche d'informations d'aujourd'hui.

Plusieurs éléments ambiguës du modèle des graphes conceptuels ont été précisés tout au long de cette démarche. Ainsi, un nouveau opérateur ϕ a été ainsi introduit, permettant de différentier les marqueurs dans un graphe conceptuels. L'opérateur ϕ permet particulièrement la prise en compte d'une extension universelle aux graphes conceptuels simples, permettant ainsi de couvrir un plus grand fragment de la logique de premier ordre.

Un système logique à base de graphe conceptuels a été aussi proposé, intégrant notre démarche

d'indexation relationnelle du chapitre précédent. Dans ce système, la recherche de la projection d'un graphe sur un autre, exploite outre les treillis des types de concepts et de relations, les agencements possibles entres les relations, leurs propriétés mathématiques ainsi que tout autre lien sémantique pouvant les relier ensemble. Des règles de dérivation sur les graphes ont été alors proposées pour mettre en place une telle approche.

L'expansion ainsi réalisée de l'opérateur de projection aurait eu toutes les chances de se répercuter automatiquement sur la complexité des traitements associés. Cependant, l'instanciation du nouveau opérateur ϕ aux seuls concepts individuels ou existentiellement quantifiés permet d'envisager une implantation algébrique de ce formalisme. L'instanciation en question s'avère être complète et correcte par rapport à celle initialement introduite par Sowa. Elle utilise la technique des témoins de la logique de premier ordre.

Grâce à cette technique, les graphes conceptuels ont pu être organisés en une structure algébrique de fichier inverse. Les opérations sur les graphes reviennent dans ce cas à des opérateurs ensemblistes de complexité polynomiale. L'approche algébrique intègre toutes les caractéristiques du modèle relationnel, en adoptant en plus un opérateur de projection de complexité polynomiale.

Chapitre 5

Application à un Corpus d'Images

Ce chapitre présente le système de recherche d'images RELIEF [1]. Il se base sur une plate-forme de gestion de graphes conceptuels appelée ROGER, dont la conception est conforme à la vision relationnelle des graphes que nous avons adopté dans le chapitre précédent. Le système RELIEF profite des dernières avancées dans le domaine des bases de données orientées objet ainsi que des facilités JAVA pour l'interaction avec le Web.

5.1 Introduction

La définition d'un modèle logique relationnel basé sur les graphes conceptuels, conduit au développement de la plate-forme de gestion des graphes conceptuels ROGER intégrant les aspects théoriques de ce modèle. Cette plate-forme a été utilisée pour l'implantation du système de recherche d'informations RELIEF que nous décrivons dans ce chapitre. Le but de cette réalisation est de montrer le bien-fondé et la faisabilité de notre approche théorique.

Si le système dans sa version actuelle est dédié à la recherche d'images, c'est principalement à cause de la disponibilité d'une collection test d'images indexée par le biais du formalisme des graphes conceptuels. Par rapport à cette collection, il est ainsi possible d'évaluer le système de façon objective. Il est toutefois inutile de mentionner que notre approche est applicable pour n'importe quel type de média, pourvu que le processus d'indexation des documents fournit comme résultat des graphes conceptuels.

Le système RELIEF est construit au-dessus du Système de Gestion de Base de Données Orienté Objet O_2 [Oa91, Tec96, AC93]. L'interface du système est accessible par n'importe quel butineur du Web (Netscape, Internet Explorer, etc), ayant des facilités JAVA.

Ce chapitre est composé de deux parties. La première décrit la conception du système RELIEF et présente ses différents modules fonctionnels. En particulier, nous montrons comment les caractéristiques du modèle relationnel développé ont été intégrées à une plate-forme de gestion de graphes conceptuels. Dans une seconde partie, nous évaluons les performances du système RELIEF. Cette évaluation est réalisée dans une double optique. La première tend à montrer la rapidité de l'algorithme de projection imaginé dans le chapitre précédent. La seconde compare les performances qualitatives de RELIEF, d'abord par rapport au système vectoriel SMART [SM83], puis par rapport

1. Pour a RElational Logical approach based on InvErted File.

au système EMIR_2, un système de recherche d'images basé sur les graphes conceptuels.

5.2 Le système RELIEF

5.2.1 Conception Générale du Système

La conception du système RELIEF peut être globalement illustrée par le schéma simplifié qui suit, utilisant les notations de la méthode orienté objet OMT [RBE+94].

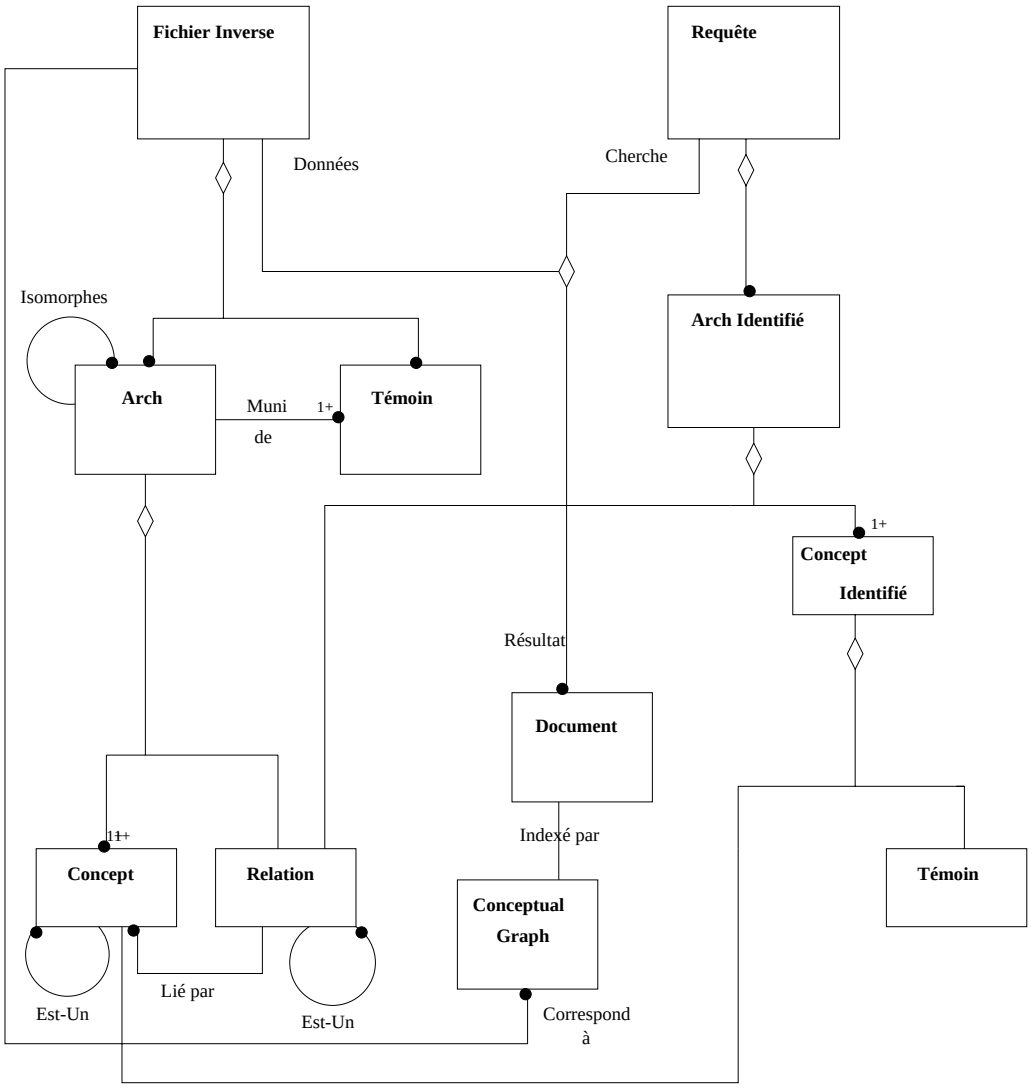


Figure 5.1. Conception générale du Système RELIEF

Certaines classes situées au niveau le plus bas du schéma expriment des éléments de base du formalisme des graphes conceptuels: *Concept*, *Relation*, *Graphe Conceptuel*. En recherche d'infor-

mations, des notions comme le *Document* et la *Requête* s'imposent. Enfin, la notion d'*Arch*, de *Témoin* et de *Fichier Inverse* constituent des classes propres à la démarche que nous avons adopté dans le chapitre précédent.

Le schéma de conception reprend les deux fonctionnalités classiques d'un système de recherche d'informations, à savoir l'indexation et l'interrogation. Le processus d'indexation est le processus qui permet d'extraire les informations mettant en évidence le contenu sémantique des documents. Dans notre cas, le formalisme utilisé pour la représentation de ces informations n'est autre que les graphes conceptuels. L'indexation associe donc un *Graphe Conceptuel* à chaque *Document* du corpus. Ce dernier fait est modélisé au niveau du schéma par une association "*Indexé par*" entre les deux classes *Graphe Conceptuel* et *Document*: un document a un unique graphe conceptuel comme index.

Notre démarche algébrique du chapitre précédent introduit le *Fichier Inverse* comme une technique d'organisation des graphes conceptuels. Le fichier inverse vérifie la condition du chapitre précédent, à savoir qu'il doit permettre de reconstruire les graphes index originaux. L'association "*Correspond à*" modélise cette condition: à un ensemble de graphes conceptuels index (le corpus indexé) correspond un seul fichier inverse permettant d'organiser les informations de ces index de manière adaptée aux objectifs de la recherche d'informations.

Le *Fichier Inverse* tel qu'il a été introduit dans le chapitre précédent est composé de l'ensemble des archs des graphes index, une fois que ces archs ont été *munis de* leurs témoins associés. Pour chaque *Arch*, il est possible d'identifier les archs qui lui sont *isomorphes* par le biais d'une association illustrée dans le schéma 5.1.

L'arch est lui-même une agrégation d'une *Relation* avec un ou plusieurs *Concepts*. Les concepts sont liés ensemble par la relation en question. Les concepts et les relations constituent les données de base du formalisme des graphes conceptuels. Ils sont tous les deux organisés en une structure de treillis: la relation $\leq$ qui lie ensemble un super-concept et un sous-concept est une relation *Est-Un*. De même les relations sont organisées en un treillis grâce au même type d'association.

La notion de témoin est un édifice que nous avons imaginé pour associer une interprétation logique adéquate au formalisme des graphes conceptuels. L'implantation du système RELIEF devra donc refléter cette interprétation logique. Comme nous l'avons vu dans le chapitre précédent, la technique des témoins associe un témoin unique à chaque concept existentiellement quantifié. De plus, l'utilisation de la relation $=$ permet également d'associer un témoin à un concept muni d'un référent. Ainsi, dans ces deux derniers cas, une nouvelle entité, nommée *Concept identifié* est créée. Un ensemble d'*Archs identifiés* est ainsi spécifié.

Le processus d'interrogation est guidé par le besoin en informations de l'utilisateur. Ce besoin est exprimé par une requête, qui constitue elle-même une agrégation d'archs identifiés. Le système utilise alors le *Fichier Inverse* pour déterminer les documents pertinents (zéro ou plusieurs), considérés alors comme le résultat de l'opérateur de projection. L'association ternaire apparaissant dans le schéma de conception exprime ce processus.

Pour l'implantation de ce schéma de conception, nous nous sommes basé sur une extension de la plate-forme de gestion des graphes conceptuels PGGC développée dans notre équipe depuis 1994. Les détails relatifs à cette nouvelles plate-forme appelée ROGER sont explicitées dans la section suivante.

5.2.2 De la Plate-forme PGGC vers la Plate-forme ROGER

La Plate-forme PGGC [Mec95c] (Plate-forme de Gestion de Graphes Conceptuels) est un environnement de développement d'applications manipulant le formalisme des graphes conceptuels. Développée par M. Mechkour depuis 1994, elle a permis l'implantation de la plupart des systèmes de recherche d'informations à base de graphes conceptuels développés par l'équipe MRIM, notamment le système de recherche d'informations médicales PRIME [MMFB97].

La plate-forme PGGC est constituée d'un ensemble de classes O_2 offrant des représentations simplifiées mais suffisantes pour la manipulation de la plupart des éléments du modèle des graphes conceptuels, ainsi que les méthodes nécessaires à leur manipulation. Ainsi, seuls les graphes conceptuels simples sont pris en considération par cette plate-forme. Il n'est donc pas possible d'avoir des graphes contextes ni des référents pluriels. La deuxième simplification du modèle concerne les relations conceptuelles. En effet, par souci de simplicité, PGGC ne considère que des relations binaires. Ceci est d'autant plus justifié dans le cadre de cette plate-forme que Sowa dans [Sow84] mentionnait la facilité de transformer une relation n-aire en un ensemble de relations binaires. Dans le chapitre 2 et 3, nous avons cependant indiqué l'importance en recherche d'informations de considérer certaines relations n-aire de façon directe. De plus, notre démarche dans le chapitre précédent était générale et permettait la manipulation de n'importe quel type de relation. Pour cette raison, nous avons préféré adopter une version de la plate-forme PGGC étendue au traitement des relations n-aires.

Toutefois, PGGC offre en plus des opportunités non fournies dans le modèle de base de Sowa. Il s'agit principalement de l'ajout de la notion de base canonique minimale (voir chapitre précédent) ainsi que l'introduction d'incertitude sur les concepts et les relations des graphes [WM93].

Afin de situer notre nouvelle plate-forme ROGER (Reasoning On Graphs with an Extended Relational approach) par rapport à PGGC, nous rappelons dans ce qui suit les principales fonctionnalités de PGGC.

a) Les Fonctionnalités de PGGC

La plate-forme PGGC est dédiée à la gestion des bases de connaissances utilisant le formalisme des graphes conceptuels. Une base de connaissances est définie par la donnée d'un ensemble de graphes canoniques représentant les faits d'un domaine particulier, et d'un canon permettant d'assurer le contrôle de la construction des graphes. Chaque application est pourvue de sa propre base de connaissances, nécessaire à son fonctionnement. Une base de connaissances (KB), est définie par le triplet suivant:

KB = [nom, canon, {graphe}]

PGGC offre d'abord un module pour la manipulation des différentes collections des bases de connaissances. Elle permet, entre autres, la création d'une nouvelle base de connaissances, la suppression d'une base de connaissances déjà existante, l'affichage des composants d'une base de connaissances et la sélection d'une base de connaissances courante[2].

L'objectif du développement de la plate-forme PGGC est qu'elle puisse être intégrée à n'importe quelle application O_2 sous forme d'un schéma importé. L'idée est donc de proposer à des personnes souhaitant développer une application de recherche d'informations à base de graphes conceptuels

2. Dans sa version actuelle, PGGC ne gère pas de droits d'accès sur les différentes bases de connaissances créées. La gestion de ces droits est actuellement laissée à la charge des applications utilisant ces bases de connaissances.

de créer leur propre base de connaissances et de l'exploiter avec toutes les opérations proposées par la plate-forme.

Au niveau de chaque base de connaissances, une interface graphique à base de menus permet d'utiliser toutes les fonctionnalités offertes par la plate-forme. Cette interface, permet surtout la saisie du canon de la base, sa modification, ainsi que la manipulation des graphes à proprement parler. Toutes les opérations sont appliquées sur la base courante sélectionnée par l'application. Pour le chargement d'une base de connaissances, Mechkour [Mec95c] a développé une syntaxe linéaire pour les graphes conceptuels. La définition d'un graphe est donnée sous forme d'un ensemble de sous-graphes linéaires (voir exemple 45 ci-dessous). Le référent d'un concept peut être vide, i.e, $\star$, une variable identifiée par une chaîne précédée par un #. La variable est utilisée pour identifier des concepts intervenant plusieurs fois dans la forme linéaire (concepts joints). Le référent d'un concept individuel est représenté par une chaîne précédée par un @.

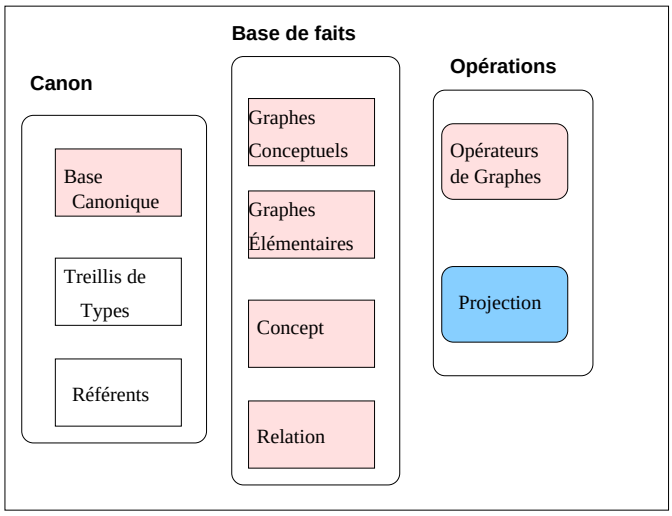


Figure 5.2. Extension de la Plate-forme PGGC

Exemple 45 *Le graphe de la figure 5.3 suivante est définie dans PGGC en utilisant la forme linéaire suivante:*

[CHAT:@Minou]→(chr)→[COULEUR:@noir]

[CHAT:@Minou]→(sur)→[CANAPE:#ca]

[CANAPE:#ca]→(a-côté)→[TABLE]

La figure 5.2 montre une vue macroscopique d'une base de connaissances au niveau de la plate-forme PGGC. Les classes sur fond grisé clair ont été modifiées puis étendues dans notre plate-forme ROGER. Les classes sur fond blanc ont été réutilisées, alors que les classes sur fond grisé foncé ont été supprimées dans la nouvelle plate-forme.

PGGC implémente la notion de canon définie par Sowa [Sow84]. Il est donc constitué d'un ensemble de graphes canoniques défini *a priori*, des treillis des types de concepts et de relations ainsi

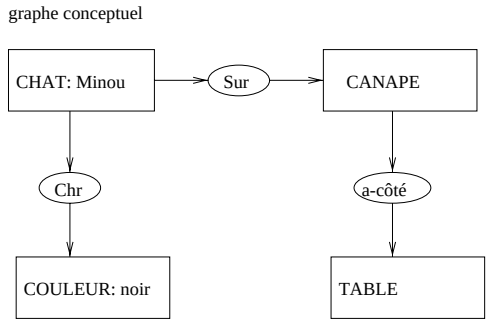


Figure 5.3. Transformation linéaire d'un graphe conceptuel

que d'un ensemble de référents. L'ensemble des graphes canoniques constitue une base canonique minimale. Cette condition de minimalité est maintenue par la plate-forme, en un sens que l'insertion d'un nouveau graphe dans la base n'est acceptée par PGGC que si le nouveau graphe ne peut être généré par les opérateurs de construction de graphes à partir de ceux déjà existants dans la base. PGGC offre des opérations de base permettant l'insertion et la suppression de nouveaux types de concept et de relations, en respectant les propriétés des relations d'ordre sur les types. La suppression d'un type de concept ou de relation entraîne automatiquement la suppression des graphes de la base canonique et de ceux de la base de faits le référençant.

b) La Plate-forme Étendue ROGER

La plate-forme ROGER reprend les fonctionnalités de PGGC au niveau de la gestion des bases de connaissances utilisant le formalisme des graphes conceptuels, à ceci près que le canon de chaque base de connaissances (KB) est dans ce cas conforme à la définition 4.3.1 du chapitre précédent (voir page 162). En particulier, il regroupe les graphes signatures ($\mathcal{S}_r$), jouant ainsi le rôle de la base canonique minimale et intégrant aussi bien des relations binaires que des relations n-aire, un ensemble d'endiguement ($\mathcal{E}$), un ensemble de graphes conceptuels permettant de représenter les propriétés des relations ($\mathcal{K}_s$) et enfin un treillis sur les marqueurs $\mathcal{M}$. Par conséquent la plate-forme ROGER offre également des méthodes permettant de gérer les nouvelles classes intégrées et d'assurer la consistance de leurs données.

Par exemple, notre nouvelle plate-forme vérifie que l'ajout d'une relation d'ordre entre deux types de relations R_1 et R_2 est cohérent avec les signatures qui leurs sont associées (voir page 166). Ainsi, si R_1 est introduit comme une spécialisation de R_2 dans le treillis des types de relation, et que les signatures de R_1 et R_2 sont définies tels que $\Sigma_{R_1} = (R_1, 2, C_1, C_2)$ et $\Sigma_{R_2} = (R_2, 2, C_3, C_4)$, ROGER est censé vérifier que dans le treillis des types de concepts, $C_1 \leq C_3$ et $C_2 \leq C_4$. Il existe également une autre contrainte d'intégrité maintenue par la nouvelle plate-forme. Il s'agit de s'assurer que les propriétés introduites pour les relations dans $\mathcal{K}_s$ sont valides au regard des signatures (voir page 174). Ainsi, si R est présentée comme étant une relation binaire transitive, ROGER vérifie que les types de concepts maximaux dans la signature de R sont bien égaux, autrement dit, $\Sigma_R = (R, 2, \text{TYPE}_c, \text{TYPE}_c)$, avec TYPE_c un élément du treillis des types de concepts. Enfin, l'ajout d'un graphe signature dans $\mathcal{S}_r$ n'est possible que si le type de relation associé au graphe

en question existe déjà dans le treillis des types de relations, et que ce type de relation n'est pas déjà muni d'une signature. L'extension du canon a été suivie au niveau de ROGER par la modification de l'interface permettant le chargement du canon. En particulier, une syntaxe pour la saisie des propriétés des relations présentes dans $\mathcal{K}_s$ ainsi que les graphes contextes de $\mathcal{E}$ a été imaginée.

Plus concrètement, mise à part les classes des treillis des types de concepts et de relations et les méthodes qui leurs sont associées (ajout/suppression d'un type, vérification des propriétés de la relation d'ordre, recherche des sous-types et des super-types d'un type particulier, etc ..), ROGER de part qu'il reflète le modèle relationnel que nous avons développé pour les graphes conceptuels utilise un canon entièrement révisé par rapport à celui considéré dans PGGC.

Dans PGGC, un graphe conceptuel est représenté par un objet de la classe *Graphes Conceptuels*. Chaque graphe est composé d'un ensemble de *Graphes Élémentaires*, définis comme des triplets $[C_1 : r_1]\rightarrow(\mathrm{R})\rightarrow[C_2 : r_2]$, où $[C_1 : r_1]$ et $[C_2 : r_2]$ sont des objets de la classe *Concept*, et R un objet de la classe *Relation*. Ce choix s'explique par le fait que seules des relations binaires sont utilisées dans le cadre de la plate-forme PGGC. Dans notre démarche, nous avons préféré l'adoption de la notion d'arch, un graphe élémentaire isomorphe à l'un des graphes signatures du canon. Un arch peut donc contenir une relation n-aire. Il correspond à un triplet uniquement dans le cas où la relation considérée est binaire. Nous avons donc modifié les classes de PGGC en conséquence.

La classe relation a été modifiée pour la prise en compte des relations n-aire. Les méthodes nécessaires à la manipulation des relations n-aires ont été ajoutées, parmi elles les méthodes relatives à la gestion de la fonction injective σ. La classe *Graphe Élémentaires* a été remplacée par la classe *Arch*. De plus, pour refléter l'interprétation logique que nous avons associé aux graphes conceptuels, la classe *Concept* a été modifiée pour prendre en compte les témoins.

Au niveau de la classe *Arch*, la plate-forme ROGER fournit les méthodes nécessaires pour l'application des règles de dérivation relationnelles $\mathcal{R}_{M_{Rel}}$ (voir page 192) permettant la prise en compte des propriétés des relations décrites dans $\mathcal{K}_s$.

La plate-forme ROGER considère ainsi des représentations très proches, de part leur généralité, du modèle de Sowa. Elle offre en plus l'avantage de permettre une prise en compte efficace des relations et leurs propriétés. Le schéma de la figure 5.4 suivante résume la représentation adoptée pour les graphes conceptuels.

Les principales méthodes définies sur la classe *Graphes Conceptuels* implantent les opérateurs canoniques de base définies par Sowa (copie, jointure, restriction, simplification). Ainsi, trois types de jointure sont disponibles dans PGGC:

- la jointure simple sur un concept commun à deux graphes conceptuels.
- la jointure maximale, effectuant une jointure de deux graphes conceptuels sur le sous-graphe maximal commun à ces deux graphes.
- la jointure maximale étendue, dans laquelle on considère tous les sous-graphes communs maximaux. Elle équivaut à l'ensemble des jointures simples possibles combiné avec une simplification du graphe produit.

Au niveau de la plate-forme ROGER, ces méthodes sont augmentées d'un opérateur de restriction sur les relations. Par ailleurs, elles sont modifiées de sorte qu'elles tiennent compte de l'interprétation associée au concepts existentiellement quantifiés (affectation des témoins).

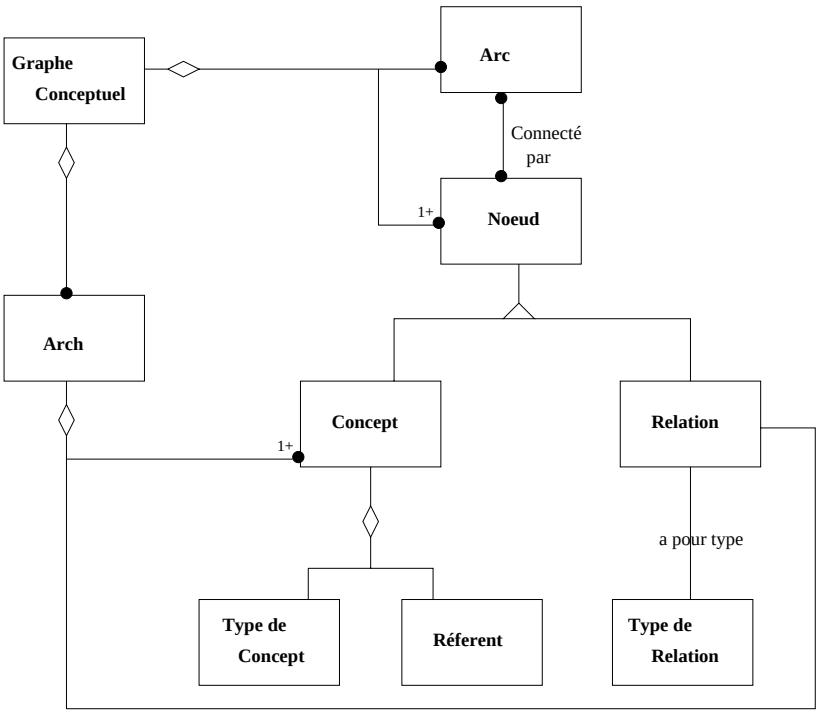


Figure 5.4. Modélisation détaillée des graphes conceptuels

La plate-forme PGGC propose également un certain nombre de méthodes permettant de définir l'opérateur de projection. Ces méthodes sont associées à la classe *Graphes Conceptuels*. Elles implantent la notion de projection partielle, la projection exacte, la recherche de toutes les projections possibles d'un graphes sur un autre, etc. Toutes ces projections peuvent tenir compte de l'extension incertaine associant des poids aux divers concepts et relations des graphes [WM93]. Toutes les projections définies au niveau de PGGC sont *injectives*. Dans la nouvelle plate-forme ROGER nous avons entièrement redéfini ces projections, de manière à ce qu'elles tiennent compte du treillis des types de relations (voir page 195). De plus, de part l'interprétation logique que nous avons associé aux graphes conceptuels, une interprétation entièrement équivalente à celle de Sowa dans le cas où les concepts génériques sont existentiellement quantifiés, la restriction de l'injectivité a été levée (voir page 219)[3]. Enfin pour pouvoir mener l'approche algébrique décrite à la fin du chapitre 4, toutes les méthodes nécessaires pour la construction du fichier inverse, des tables d'accélération ainsi que pour l'application de l'algorithme de projection (voir page 246) ont été implantées au niveau des classes *Fichier Inverse* et *Table accélération*, respectivement. Pour résumer, une base de connaissances dans la plate-forme ROGER admet l'architecture de la figure 5.5 ci-dessous.

En plus de la gestion des graphes conceptuels à proprement parler, la plate-forme PGGC intègre une interface d'interrogation permettant de guider l'utilisateur dans la formulation de sa requête puis de parcourir l'ensemble des documents triés par ordre décroissant de pertinence. L'interface

3. Signalons par ailleurs, que les algorithmes des diverses opérateurs de projections fournis par PGGC ont subi plusieurs améliorations en vue d'aboutir à des temps de réponses acceptables. Nous reviendrons à ce point dans la dernière section de ce chapitre.

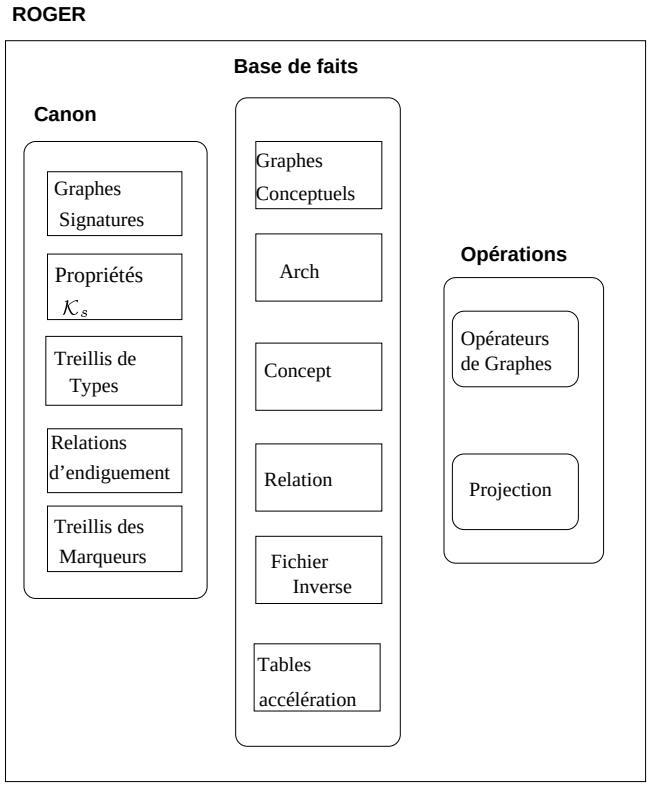


Figure 5.5. La Plate-forme ROGER

se charge de mettre la requête sous forme d'un graphe conceptuel avant le lancement du processus de correspondance. Au niveau de la nouvelle plate-forme ROGER, cette interface a été adaptée de sorte que le graphe requête généré soit sous forme plus adaptée à notre approche algébrique. Enfin, la plate-forme PGGC étend le SGBD O_2 par une extension multimédia offrant des fonctionnalités regroupant entre autres, des structures de données permettant la représentation raster des images et l'implantation des méthodes de traitements de ces images [Mec95c]. Au niveau de la plate-forme ROGER, ces modules développés par Cuzin [Cuz92] ont été repris.

5.2.3 Fonctionnalités et Architecture du Système RELIEF

Nous présentons dans cette section les principales fonctionnalités du système RELIEF. Ces fonctionnalités sont exportées à partir de la plate-forme ROGER. L'architecture de ce système est ensuite présentée.

a) Fonctionnalités de RELIEF

Le système RELIEF se base sur la plate-forme de gestion des graphes conceptuels ROGER pour implanter un système de recherche d'images. En particulier il se base sur les nouvelles structures permettant l'adoption d'une approche algébrique lors du processus de recherche. Dans la figure 5.6,

les fonctionnalités de ROGER utilisées par ce système sont présentées. Les modules correspondant à des structures de données sont représentés par des rectangles, alors que les ovales indiquent des traitements. Comme tout système de recherche d'informations, RELIEF divise ses fonctionnalités en deux parties: le processus d'indexation (représenté dans la même figure par des flèches en pointillés) et le processus de recherche (représenté par des flèches continues).

Les contraintes en termes de temps des traitements sont différentes selon qu'il s'agisse de l'indexation ou de l'interrogation. Ainsi, si le processus d'indexation ne requiert pas nécessairement des traitements rapides, il en ait autrement pour l'interrogation où l'utilisateur souhaite obtenir très vite les documents pertinents pour sa requête. En effet, le processus d'indexation est réalisé avant que les utilisateurs n'entrent en interaction avec le système. Pour ces derniers, les performances en temps de réponse du système se calculent par rapport à l'interrogation. La conception d'un système de recherche d'informations devra donc tenir compte de ces contraintes. L'usage du fichier inverse s'explique d'ailleurs par ce fait.

Le chargement des données, c'est-à-dire la saisie du canon et des index fait partie intégrante du processus d'indexation. Comme nous l'avons vu dans la section précédente, la plate-forme ROGER offre les fonctionnalités nécessaires à ce chargement. Pour changer le domaine d'application, il suffit d'introduire un nouveau canon et des nouveaux graphes index, le système reste quant à lui intacte.

Les index des documents (les données) sont organisés en un fichier inverse. Ce dernier étant le module principal de notre approche algébrique. La construction du fichier inverse intervient lors du processus d'indexation. Elle suit le procédé que nous avons exposé dans le chapitre précédent. Le moteur d'inférence est censé enrichir le mécanisme de dérivation des graphes conceptuels par un raisonnement sur les relations et leurs propriétés. Il s'agit d'utiliser les règles de dérivation $\mathcal{R}_{M_{Rel}}$ (voir page 192) pour étendre les graphes index par les données implicites engendrées par ces règles. L'effet de ces extensions est répercuté sur le fichier inverse originel.

L'obligation d'avoir des temps de réponses adéquats a engendré l'utilisation explicite des tables d'accélération. Leur construction est amorcée une fois le fichier inverse ayant été étendu par les propriétés des relations. Cela est réalisé par le processus "Isomorphes" correspondant à l'association du même nom dans la figure 5.1. Comme nous l'avons vu dans le chapitre précédent, le rôle de ces tables d'accélération est de déterminer à partir du fichier inverse, étant donné un arch de la requête, l'ensemble des documents contenant un isomorphe de cet arch et donc potentiellement pertinents.

La construction du fichier inverse étendu et des tables d'accélération achève le processus d'indexation. La fonction de correspondance de RELIEF est basé sur l'algorithme de projection que nous avons proposé dans le chapitre précédent. Ce dernier est implanté au niveau de ROGER de façon à s'appuyer sur le fichier inverse et les tables d'accélération pour retrouver les projections partielles et exactes satisfaisant la requête de l'utilisateur dans un processus entièrement guidé par la structure de cette dernière.

b) Architecture du Système RELIEF

La figure 5.7 montre comment se présente le système de recherche d'images RELIEF. Dans ce schéma chaque module groupe l'ensemble des classes impliquées pour l'accomplissement d'une tâche particulière, mais ne correspond pas pour autant à un processus puisque RELIEF est implanté comme un schéma dans la base de données O_2. Le choix d'une base de données orientée objet est surtout motivé par sa capacité à fournir un stockage sécurisé et un accès à une grande quantité

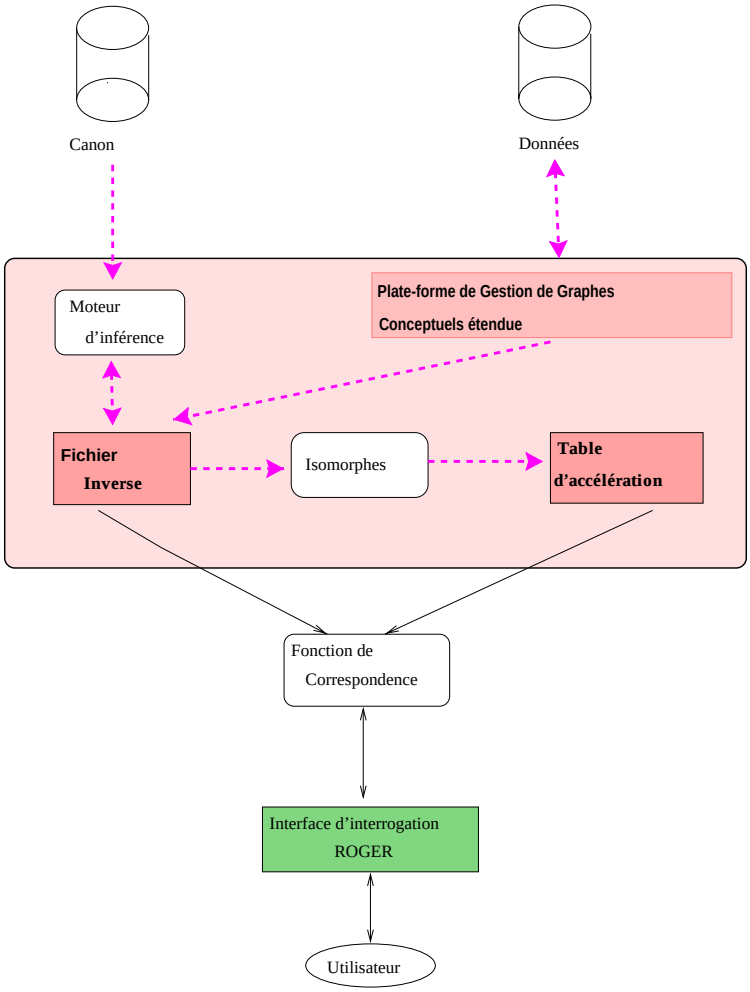


Figure 5.6. Fonctionnalités du système RELIEF

d'informations. De plus, une approche orientée objet supporte bien la complexité des données multimédia.

La plate-forme ROGER décrite dans la section précédente est le principal composant du système RELIEF. Le gestionnaire d'images, comme nous l'avons déjà signalé, a été développé dans le cadre d'une extension multimédia à O_2. Cette extension offre les fonctions de traitement d'images les plus importantes non disponibles dans les bases de données orientées objet. Le gestionnaire d'index permet d'associer à une image du corpus sa description en graphes conceptuels.

Le système RELIEF étant disponible sur le Web, n'importe quel butineur WWW peut être utilisé pour accéder à son interface d'interrogation. La partie interface dans RELIEF génère ainsi les pages HTML adéquates qui seront ensuite affichées par le navigateur WWW choisi. La passerelle $O2$Web fournie par la dernière version O_2 permet d'établir une connection entre O_2 et le Web. L'utilisation du langage JAVA pour le développement de cette interface a été choisi dans une optique d'une interaction agréable entre l'utilisateur et le système. Ceci facilite entre autres la formulation des

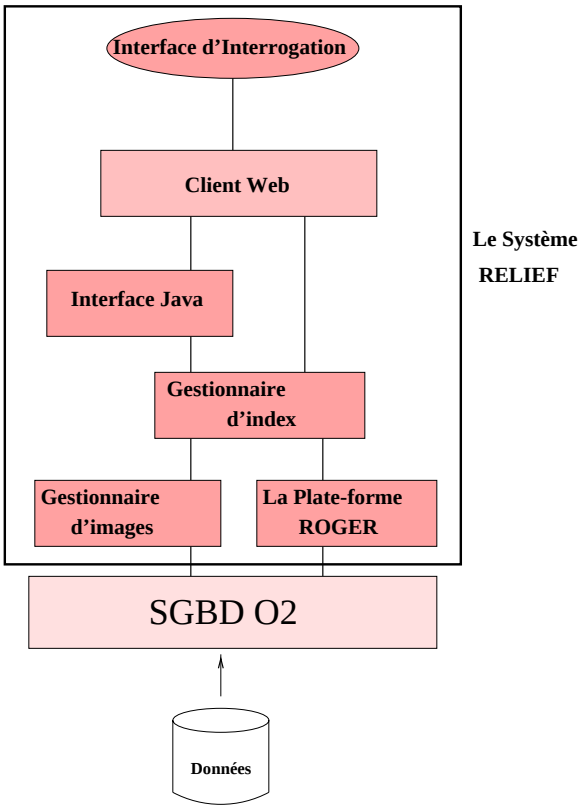


Figure 5.7. Architecture du système RELIEF

requêtes et la visualisation des documents.

Par ailleurs, il est à noter que les éléments de la plate-forme ROGER ont été développés en O_2C, un langage de programmation fourni par O_2, et constituant une extension du langage C pour intégrer les notions spécifiques au contexte objet.

5.3 Indexation du Corpus

5.3.1 Présentation de la Collection

Nous avons choisi d'appliquer notre système à une collection test d'images appelée PARISYS, fournie dans le cadre du projet Européen FERMI [Fer94]. Une description détaillée de cette collection se trouve dans [CM97]. Chaque partenaire de ce projet européen utilise son propre formalisme de représentation pour l'implantation d'une logique pour la recherche d'informations. Par exemple, des formalismes comme les logiques terminologiques [SS96], Datalog [Fuh95] ou même les mots-clés. L'intérêt de cette collection est qu'elle regroupe un ensemble homogène d'images pour lequel des requêtes et leur jugement de pertinence associé ont été définis, autorisant ainsi une évaluation objective des performances d'un système tout en présentant l'avantage de pouvoir le comparer à

d'autres systèmes.

La collection PARISYS est constituée de 663 images datant du début du siècle. Une première moitié environ de ces photographies illustre le vieux Paris, ses monuments, ses petits métiers, ses quartiers, des événements marquants de l'époque. L'autre moitié est plus hétéroclite, et représente des portraits d'hommes célèbres, des paysages, des objets d'art, etc. L'intérêt de ce corpus est donc d'ordre esthétique, historique, anecdotique, architectural, etc. De plus, la richesse sémantique des images, leur niveau de détail, le fait qu'elles étaient l'oeuvre de célèbres photographes français de l'époque, dont le fameux Atget résument l'intérêt de cette collection.

Le public potentiel est relativement varié, du journaliste qui cherche à illustrer un article, à l'étudiant en architecture qui se documente sur un type d'architecture particulier, en passant par l'amateur de vieilles photographies en général.

Cette collection a été indexée au départ par des documentalistes du Ministère français de la Culture. Les index ont été définis de manière très professionnelle par des spécialistes de l'image. Les descriptions des images sont munies d'un thésaurus qui couvre globalement tout le domaine sémantique associé à la collection. Ce thésaurus a été utilisé par les indexeurs pour décrire uniformément le contenu sémantique des images. L'indexation originelle a été bien entendue manuellement réalisée. Un des points intéressants de cette indexation est qu'elle adresse plusieurs facettes de l'image, c'est-à-dire qu'elle fournie une description pour chaque point de vue de l'image (voir Annexe B).

Cependant, si cette indexation a été rigoureusement et professionnellement menée, elle a été construite pour des systèmes de recherche d'informations orientés mots-clés. Il en résulte un certain nombre de limitations comparées aux modèles proposés dans la littérature pour la recherche d'images [Mec95c, Meg95]:

- Il n'existe pas d'identification des objets ou des régions de l'image. Par exemple, quand dans la facette Contenu, il est indiqué que l'image contient un bâtiment, et que dans une autre facette il est précisé (facette Précisions) que l'image montre le supermarché "La Samaritaine", l'association entre les mots-clés «Bâtiment» et «La-samaritaine» est entièrement laissée à la charge de l'utilisateur, c'est-à-dire que ce dernier devra identifier parmi les nombreux bâtiments de l'image celui qui est réellement "La Samaritaine".
- La vue structurelle et la vue spatiale sont totalement absentes. Par exemple, quand la facette Contenu de l'image, indique que cette dernière contient les deux mots-clés Homme et Table, il n'existe aucune indication quant à la position de l'homme par rapport à la table dans l'image.

De façon plus claire, il existe très peu d'informations sur les relations dans l'indexation originelle. Afin, de remédier à ce problèmes, il a été convenu au niveau du projet européen FERMI de ré-indexer la collection dans un sens où il sera possible de remplir les exigences d'une recherche d'image efficace. Le modèle d'image développé par Mechkour [Mec95c] (voir Chapitre 2) fut alors utilisé comme guide d'indexation. Le résultat de la nouvelle indexation sera donné en termes de graphes conceptuels. La transformation de ce formalisme en logiques terminologiques ou en logique de premier ordre étant facilement envisageable. De plus, une taxinomie de relations couvrant le domaine d'application a été proposée.

Par ailleurs, vu la qualité de l'indexation originelle, il a été convenu de s'en inspirer. L'objectif de la ré-indexation était ainsi de s'appuyer sur les images eux-mêmes pour compléter l'identification de leur contenu en se basant sur les descriptions fournies par les spécialistes du Ministère de la Culture.

5.3.2 Conversion en Graphes Conceptuels

Le processus de ré-indexation du corpus a été mené par 8 membres de l'équipe MRIM, dont l'auteur de ces lignes. Malgré la disponibilité d'un outil d'aide à l'indexation des images élaboré par Mechkour dans le cadre de la plate-forme PGGC [CM97], le processus d'indexation fut délicat et assez long à mener. Il a surtout fallu définir une stratégie d'indexation spécifiant un ensemble d'instructions devant être suivies par les divers indexeurs.

L'outil d'aide à l'indexation a été développé en JAVA sous le système UNIX. La tâche principale de l'interface d'indexation est de faciliter l'identification des objets images jugés pertinents dans l'indexation originelle, et de les décrire conformément au modèle d'image $EMIR_2$ de Mechkour. Par conséquent l'interface est organisée en trois fenêtres séparées:

- une fenêtre affichant l'image à indexer. Pour des raisons de présentation, cette fenêtre inclue également les outils nécessaires à la définition de la vue spatiale de l'image (figure 5.8)
- une fenêtre affichant la description originelle de l'image, telle qu'elle a été fournie par le Ministère de la Culture (figure 5.10).
- une dernière fenêtre permet de générer la vue structurelle et symbolique de l'image. Elle permet également l'introduction de certains attributs formatés (figure 5.9).

Le travail d'un utilisateur de l'interface consiste à identifier les objets correspondant aux mots-clés de l'indexation originelle. Il s'agit d'établir des contours sur l'image sous forme de polygones correspondant à ces objets. Ces contours apparaissent en blanc sur la figure 5.8. L'ensemble de ces objets est alors organisé en une hiérarchie de composition en rattachant chaque objet à ses composants (voir figure 5.9).

Une fois les objets de l'image identifiés, l'utilisateur pourra introduire les relations spatiales qui les lient. Pour ce faire, il dispose d'une liste prédéfinie de relations spatiales 2D et 3D. Il n'a alors qu'à cliquer sur les objets de l'image qui lui semblent être liés par une relation spatiale donnée. La relation spatiale nouvellement créée apparaît dans une fenêtre avec barre défilante (voir figure 5.8). Pour faciliter la présentation au niveau de l'interface, les crochets des concepts ainsi que les parenthèses des relations ont été omis. Par exemple, par FEMME→a-droite-de→RAMPE, l'interface indique que l'indexeur a asserté que l'objet FEMME se trouve à droite de l'objet RAMPE. Il est à noter que l'ordre de sélection des objets à lier est important, puisque l'outil d'indexation considère que le premier objet sélectionné correspond au premier argument du graphe isomorphe à la signature de la relation spatiale considérée. Le même outil d'indexation maintient également la contrainte de minimalité de la base de connaissances correspondant aux relations ajoutées. Par exemple, il est interdit d'utiliser une relation et son inverse entre les mêmes objets. Il s'agit d'éviter les redondances dans l'indexation (voir chapitre 3).

La dernière étape de l'indexeur consiste à préciser la vue symbolique de l'image, c'est-à-dire à spécifier le contenu sémantique de l'image. Cela se fait par le biais d'une panoplie de relations présentes dans le treillis des relations que nous avons ajouté. Afin de faciliter la tâche de l'indexeur, les relations ont été groupées en deux classes partageant les mêmes propriétés en ce qui concerne leurs opérandes. La première classe regroupe toutes les relations permettant d'associer un attribut ou un *qualificatif* à un objet. Par exemple, que l'image montre un paysage (IMAGE→Genre→PAYSAGE), que l'homme apparaît assis dans l'image (HOMME→Position→ASSIS), ou encore que telle statue

Figure 5.8. Vue spatiale et physique de l'image

est en or (STATUE→Matériel→OR). La seconde classe regroupe un ensemble de relations symboliques liant des objets de l'image. Par exemple, on pourra spécifier le fait que dans l'image, la femme porte une robe (FEMME→Portant→ROBE). Les relations ajoutées apparaissent dans la seconde moitié de la figure 5.9. La première fenêtre avec barre défilante affiche les qualificatifs associés aux objets de l'image, alors que la seconde résume les relations symboliques entre objets.

Selon le type du qualificatif, les valeurs possibles qui peuvent lui correspondre appartiennent soit à une liste contrôlée, c'est le cas par exemple du qualificatif "Position", soit ces valeurs peuvent être directement saisies par l'utilisateur. Dans les deux cas, ces valeurs correspondent à des types de concepts.

Les trois dernières étapes ainsi réalisées, le graphe index de l'image traitée pourra être généré selon le modèle d'images de Mechkour [Mec95c]. Dans la section suivante, nous présentons la sortie

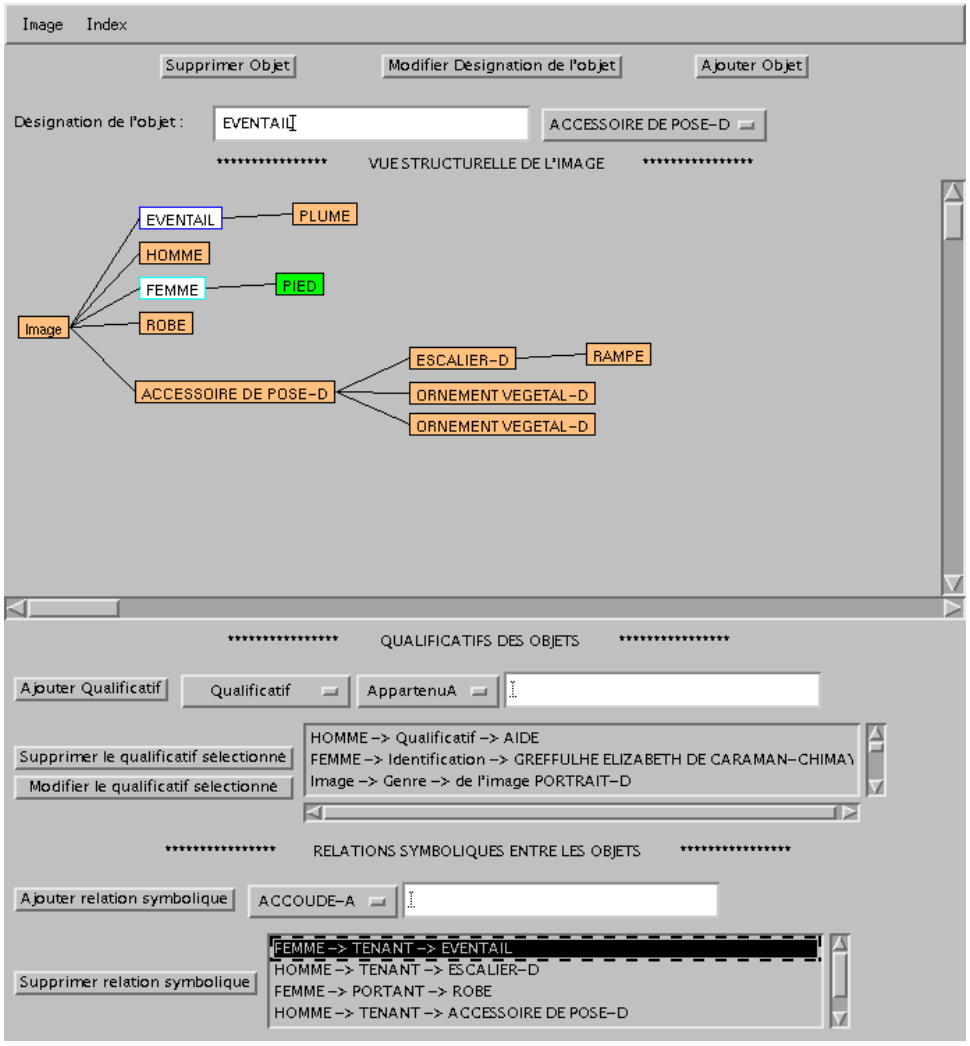


Figure 5.9. Vue structurelle de l'image

de l'indexation.

5.3.3 Le Modèle d'Indexation

L'outil d'indexation se charge de transformer le résultat du travail d'indexation de l'utilisateur en un graphe conforme au modèle d'images de Mechkour. Ainsi, chaque fenêtre d'affichage de l'interface d'indexation précédente génère une ou plusieurs vues de ce modèle:

- la fenêtre de la figure 5.9 permet de générer la vue structurelle et symbolique de l'image. La vue structurelle décrit la décomposition de l'image en objets. Elle revient à un ensemble d'archs de la forme: [Concept]→(Comp)→[Concept]. La vue symbolique permet la représentation du contenu sémantique des objets de l'image. Elle consiste surtout en une séquence d'archs de

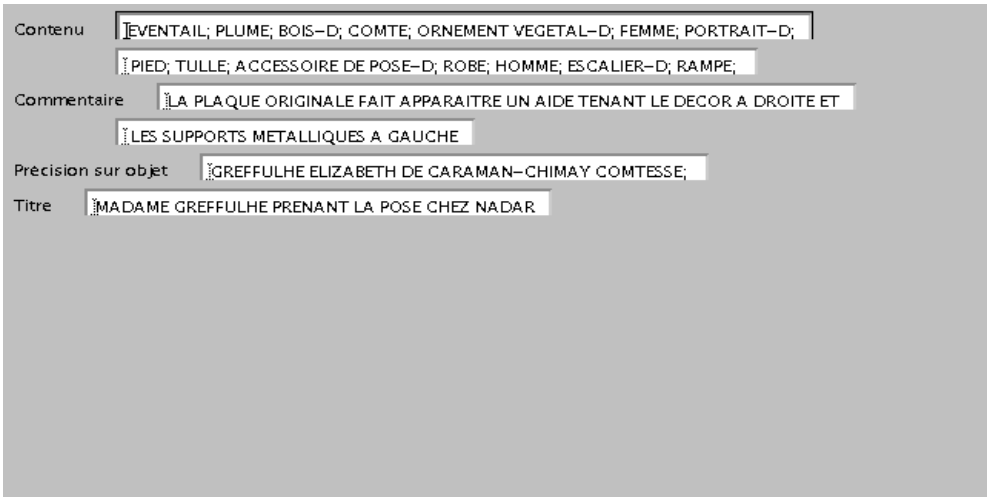


Figure 5.10. Description originelle de l'image

la forme: [ObjetImage]→(IsA)→[Concept-Spécifique], permettant l'identification des objets de l'image. La vue symbolique contient également les archs relatifs aux qualificatifs et aux relations symboliques indiqués dans la deuxième moitié de la figure 5.9.

– La fenêtre de la figure 5.8 correspond à la vue spatiale de l'image. La vue spatiale comporte les informations géométriques décrivant les objets spatiaux de l'image et les relations spatiales décrivant leurs positions relatives. Un objet image est spatialement décrit par le polygone correspondant au contour qui lui a été associé lors de l'indexation. En termes de graphes conceptuels, La vue spatiale est donnée par un ensemble d'archs de la forme: [ObjetImage]→(VueSpa)→[Polygone] pour la description de l'objet spatial et par un ensemble d'archs de la forme: [Polygone]→(Relation-Spatiale)→[Polygone] pour la spécification de leurs positions relatives.

Les graphes index générés ont une syntaxe linéaire conforme à celle adoptée dans la plate-forme ROGER (voir exemple 45). Ainsi, l'image de la figure 5.8 admet comme index, le graphe conceptuel ci-dessous. Le référent #i est introduit pour indiquer que le même concept apparaît plusieurs fois, connectant ainsi les archs auxquels il appartient. Par exemple, le concept [Image:#0] est le même dans les six premiers archs qui suivent.

[Image:#0]→(Identifiant)→[Numero:@4450]
[Image:#0]→(Comp)→[ObjetImage:#1]
[Image:#0]→(Comp)→[ObjetImage:#4]
[Image:#0]→(Comp)→[ObjetImage:#6]
[Image:#0]→(Comp)→[ObjetImage:#9]
[Image:#0]→(Comp)→[ObjetImage:#10]
[ObjetImage:#1]→(Comp)→[ObjetImage:#2]
[ObjetImage:#6]→(Comp)→[ObjetImage:#11]
[ObjetImage:#8]→(Comp)→[ObjetImage:#3]
[ObjetImage:#10]→(Comp)→[ObjetImage:#8]
[ObjetImage:#10]→(Comp)→[ObjetImage:#7]

[ObjetImage:#10]→(Comp)→[ObjetImage:#5]
[ObjetImage:#1]→(IsA)→[EVENTAIL:#1]
[ObjetImage:#2]→(IsA)→[PLUME:#2]
[ObjetImage:#3]→(IsA)→[RAMPE:#3]
[ObjetImage:#4]→(IsA)→[HOMME:#4]
[ObjetImage:#5]→(IsA)→[ORNEMENT-VEGETAL-D:#5]
[ObjetImage:#6]→(IsA)→[FEMME:#6]
[ObjetImage:#7]→(IsA)→[ORNEMENT-VEGETAL-D:#7]
[ObjetImage:#8]→(IsA)→[ESCALIER-D:#8]
[ObjetImage:#9]→(IsA)→[ROBE:#9]
[ObjetImage:#10]→(IsA)→[ACCESSOIRE-DE-POSE-D:#10]
[ObjetImage:#11]→(IsA)→[PIED:#11]
[HOMME:#4]→(Qualificatif)→[AIDE]
[FEMME:#6]→(Identification)→[GREFFULHE-ELIZABETH-DE-CARAMAN-CHIMAY]
[Image:#0]→(Genre-de-l'image)→[PORTRAIT-D]
[ESCALIER-D:#8]→(Represente)→[ACCESSOIRE-DE-POSE-D]
[EVENTAIL:#1]→(Qualificatif)→[ACCESSOIRE-DE-POSE-D]
[ROBE:#9]→(Materiau)→[TULLE]
[ESCALIER-D:#8]→(Materiau)→[BOIS-D]
[FEMME:#6]→(Qualificatif)→[COMTE]
[FEMME:#6]→(LieA)→[NADAR]
[FEMME:#6]→(TENANT)→[EVENTAIL:#1]
[HOMME:#4]→(TENANT)→[ESCALIER-D:#8]
[FEMME:#6]→(PORTANT)→[ROBE:#9]
[HOMME:#4]→(TENANT)→[ACCESSOIRE-DE-POSE-D:#10]
[ObjetImage:#1]→(VueSpa)→[Polygone:#1]
[ObjetImage:#2]→(VueSpa)→[Polygone:#2]
[ObjetImage:#3]→(VueSpa)→[Polygone:#3]
[ObjetImage:#4]→(VueSpa)→[Polygone:#4]
[ObjetImage:#5]→(VueSpa)→[Polygone:#5]
[ObjetImage:#6]→(VueSpa)→[Polygone:#6]
[ObjetImage:#7]→(VueSpa)→[Polygone:#7]
[ObjetImage:#8]→(VueSpa)→[Polygone:#8]
[ObjetImage:#9]→(VueSpa)→[Polygone:#9]
[ObjetImage:#10]→(VueSpa)→[Polygone:#10]
[Polygone:#6]→(a-droite-de)→[Polygone:#3]
[Polygone:#1]→(au-dessus-2D)→[Polygone:#6]
[Polygone:#6]→(devant-2D)→[Polygone:#1]
[Polygone:#6]→(en-dessous-2D)→[Polygone:#5]
[Polygone:#7]→(a-gauche-de)→[Polygone:#6]
[Polygone:#6]→(sur)→[Polygone:#8]

Forme linéaire du graphe index de l'image de la figure 5.8

Lors de l'indexation des images du corpus, près de 7000 types de concepts et 350 types de relations ont été utilisés (voir un extraits du canon dans l'annexe D). En moyenne l'index d'une image comporte 35 archs soit autant de relations et plus de 70 concepts. La taille d'un graphe conceptuel index est ainsi importante.

5.4 Évaluation des Requêtes

L'évaluation de la correspondance entre les requêtes et les documents se fait en deux étapes comme cela a été expliqué dans le chapitre précédent. Il s'agit d'abord de chercher les projections partielles, ensuite de déterminer les projections exactes.

5.4.1 Interface d'Interrogation

L'utilisateur interagit avec le système RELIEF à travers une interface sur le Web[4]. Supposons par exemple qu'il cherche à trouver "*les portraits d'hommes assis à droite d'un meuble*". Cette requête est introduite en utilisant les trois régions de l'interface de formulation de requêtes (voir figure 5.11:

- La région principale de l'interface permet de spécifier la vue structurelle du graphe requête. Dans notre exemple, l'utilisateur cherche des images composées de deux objets: HOMME et MEUBLE. La construction de la requête est faite à travers un champ d'entrée (en blanc dans la figure 5.11), dans lequel les concepts peuvent être introduits. Si l'utilisateur n'a pas de connaissances sur les types de concepts utilisés dans l'application, il pourra consulter le treillis des types de concepts à travers le bouton "Hiérarchie des Objets".
- La deuxième région permet d'affiner la requête en associant des qualificatifs aux concepts précédemment introduits. Dans notre cas, l'utilisateur doit spécifier qu'il cherche à retrouver des images de type "portrait". De plus, il devra mentionner qu'un homme en position assise devra apparaître sur chacune des images à retrouver.
- La troisième région permet de spécifier le contenu sémantique devant être véhiculé par les images à retrouver. Des relations symboliques et spatiales peuvent ainsi être introduites. Elles permettent de mieux spécifier la requête. Dans notre exemple, l'utilisateur devra mentionner dans cette région l'existence d'une relation "à-droite-de" entre HOMME et MEUBLE.

L'interface fournit des facilités de "click-and-drag", ainsi que plusieurs boutons. La requête utilisateur est lancée en cliquant sur le bouton "Evaluer Requête". Une applet JAVA collecte alors les données à partir des 3 régions de l'interface et assemble ensuite la requête RELIEF. Un analyseur permettra ensuite de vérifier la syntaxe et la canonicité de cette requête, puis de la translater vers sa représentation interne:

[Image:#0]→(Comp)→[ObjectImage:#1]

[Image:#0]→(Comp)→[ObjectImage:#2]

[ObjectImage:#1]→(IsA)→[HOMME:#1]

[ObjectImage:#2]→(IsA)→[MEUBLE]

[HOMME:#1]→(Position)→[ASSIS]

[Image:#0]→(Genre)→[PORTRAIT]

[ObjectImage:#1]→(VueSpa)→[Polygone:#1]

[ObjetImage:#2]→(VueSpa)→[Polygone:#2]

[Polygone:#1]→(a-droite-de)→[Polygone:#2]

L'algorithme de projection est ainsi lancé. Le résultat est un ensemble de listes d'images qui sera retourné à l'interface, laquelle se charge alors de les transformer en des documents HTML puis de les afficher à l'utilisateur selon leur ordre de pertinence. La première liste regroupe les images pour lesquelles il existe une projection exacte. Les autres classes correspondent à des images dont

4. Cette interface a été développée par Mourad Mechkour dans le cadre du projet FERMI.

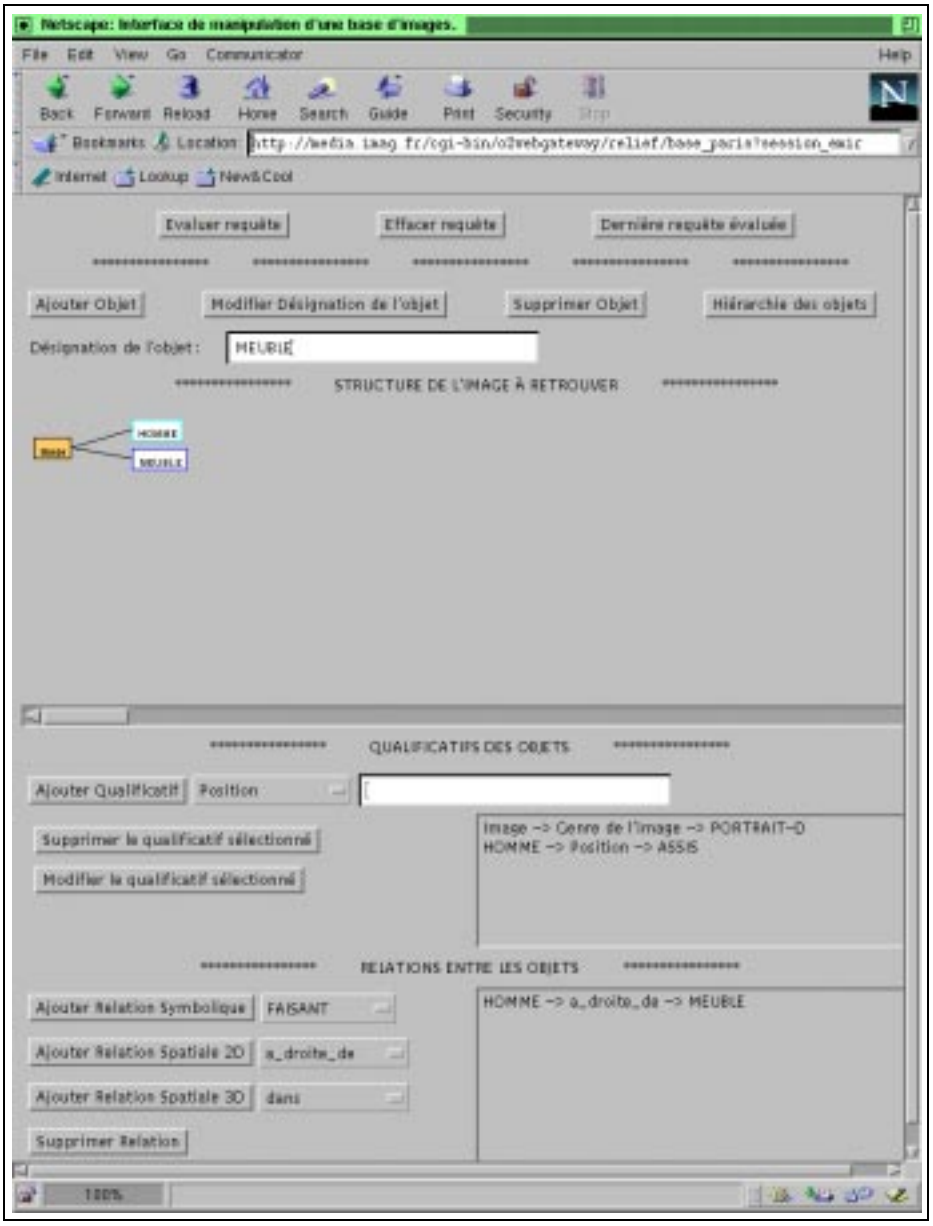


Figure 5.11. Interface de formulation de requêtes

la description vérifie une partie seulement de la requête. Il s'agit d'images dont le graphe index ne contient qu'une projection partielle de la requête. Le degré de pertinence d'une image croît avec l'importance du sous-graphe de la requête projeté sur son graphe index.

5.4.2 Visualisation des Réponses

La phase de visualisation consiste à afficher les réponses du systèmes en divers classes de pertinence, puis de permettre à l'utilisateur de consulter la description des images affichées. Les dif-

férentes classes de pertinence sont indiquées par des couleurs allant du rouge au blanc. La couleur rouge correspond aux images contenant une projection exacte. Autant la couleur est foncée, autant elle indique un degré de pertinence plus important.

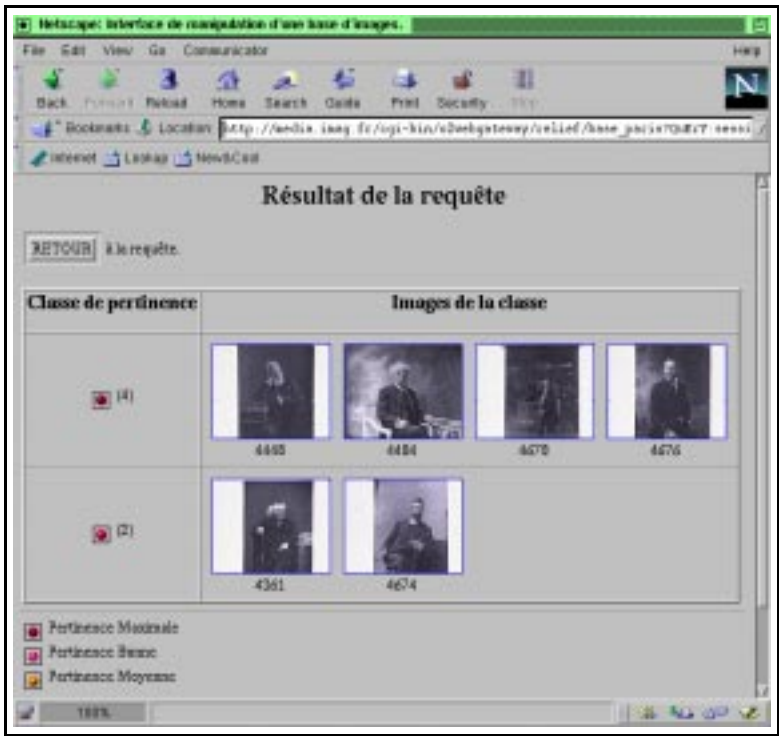


Figure 5.12. Interface pour la visualisation des réponses

Pour notre requête de la section précédente, le résultat de la correspondance donne deux classes de pertinence (voir figure 5.12). La première regroupe 4 images, alors que la seconde ne compte que 2 images. Les 4 premières images apparaissent dans la classe associée à la couleur rouge. Ils ont donc un degré de pertinence maximal, c'est-à-dire qu'ils impliquent logiquement la requête. Ainsi, la première image 4448 montre une photographie de Victor Hugo. Ce dernier est bien assis à droite d'une table dans ce portrait. Son graphe index contient une projection totale de la requête. Les 2 images affichées dans la seconde classes ne sont pas entièrement pertinentes à la requête. Leur description n'implique qu'une partie de la requête. Par exemple, l'image 4361 apparaît dans la seconde classe de pertinence, à cause du fait que la relation spatiale "a-droite-de" n'est pas satisfaite: l'homme dans cette image n'est pas à droite, mais plutôt à gauche du meuble (une table dans ce cas). Si cette relation est supprimée de la requête, le graphe index de l'image impliquera la partie résultante de la requête et l'image 4361 appartiendra à la première classe.

Toutes les images affichées dans l'interface de visualisation apparaissent sous forme d'ancres URL, permettant ainsi à l'utilisateur d'accéder facilement à une description plus détaillée de l'image (informations supplémentaires sur le contexte de l'image, ses aspects subjectifs, son auteur, etc ...). L'utilisateur peut également consulter le graphe index associé à l'image.

5.5 Analyse des Résultats

Nous discutons dans cette section des résultats de l'indexation et de la recherche du système RELIEF sur la collection PARISYS. Nous insistons particulièrement sur les performances du système aussi bien en termes de temps de réponse qu'en termes de qualité des documents fournis.

5.5.1 Les Performances du Système

Le fonctionnement du système RELIEF est tributaire du chargement des graphes index associés au corpus et du canon de l'application (les treillis de types, les graphes signatures, les propriétés des relations, etc.). Ceci est réalisé par le biais des fonctionnalités de la plate-forme ROGER (voir section 5.2.3, page 265). Dans l'ordre, le chargement du canon s'effectue en premier lieu. La plate-forme vérifie alors que les graphes index sont bien canoniques. Les graphes index ainsi que les éléments du canon doivent vérifier la syntaxe supportée par la plate-forme. La table 5.1 présente les résultats du chargement des données. La colonne *taille* donne la taille de la structure correspondante dans la base O_2, la colonne *ajout*, la différence par rapport aux structures originelles, enfin la dernière colonne donne le temps d'exécution approximatif[5] de la construction de la structure en question.

Si le stockage des éléments du canon dans la base O_2 ne requiert pas un espace disque important (moins de 200 Kilo-octets), les index originaux de la collection test occupent, quant à eux, 9.67 Méga-octets. La construction du fichier inverse correspondant à ces index suit l'approche présentée dans le chapitre précédent. Elle passe nécessairement par l'affectation des témoins aux concepts génériques existentiellement quantifiés. Le fichier inverse ainsi construit fait 69% de moins en taille que les index originaux. Cette compression notable s'explique principalement par le fait que seule la première occurrence des archs identiques est représentée dans le fichier inverse. Les autres apparitions se traduisent par un ajout de témoins dans la ligne correspondante. Ainsi, si les graphes index comptent environ 30 000 archs, le fichier inverse correspondant admet à peine 4200 entrées. Au fait, le modèle d'image utilisé n'est pas étranger à cette constatation. Par exemple, toutes les relations de composition se traduisent par deux entrées dans le fichier inverse: [Image]→(Comp)→[ObjetImage] ou [Image]→(Comp)→[ObjetImage].

Structure	*taille*	*ajout*	*temps CPU*
Canon	180 Ko	-	1 mn
index initiaux	9.67 Mo	-	10 h
fichier Inverse	2.92 Mo	-69%	8h
fichier Inverse étendu	3.52 Mo	-63%	1h30
table d'accélération	3.98 Mo.	-	1h
Nouvel Index	7.5 Mo	-23%	

Table 5.1. Chargement des Index

5. Le chargement a été réalisé sur SUN UltraSparc 2, OS Solaris 2.5, 256 Mo RAM. Les temps dépendent de la charge de la machine lors de l'exécution des programmes.

L'exploration des propriétés des relations fait augmenter la taille initiale du fichier inverse. Ce dernier fait tout de même 63% de moins en espace de stockage que les index originaux. L'extension des graphes index par les informations implicites engendrées à partir des propriétés des relations correspond également, le plus souvent, à des ajouts de témoins et non pas à des ajouts d'entrées dans le fichier inverse. En termes d'archs, le nombre de ces derniers qui était de l'ordre de 35 archs passe à 43 arch en moyenne, soit une augmentation de 22%.

Les tables d'accélération sont construites à partir du fichier inverse étendu. L'ensemble de ces tables occupe 3.98 Méga-octets dans la base O_2. En total, la taille des données du corpus correspondant à la somme de la taille du fichier inverse avec celle des tables d'accélération fait 7.5 Mo, soit 23% de moins que les index originaux. Le gain en stockage est ainsi intéressant, surtout qu'aucun effort d'optimisation n'a été jusqu'à présent mené lors de l'implantation des représentations des structures de notre approche. Il est donc fort à parier que ce gain peut largement être consolidé dans le futur.

Pour ce qui est des temps de chargements, les premiers résultats de la table 5.1 sont très mitigés pour une collection de taille moyenne comme PARISYS. Notre effort lors de la réalisation du système RELIEF s'était surtout concentré autour de l'optimisation de la phase de recherche et donc sur l'amélioration de l'opérateur de projection. Ainsi, entre la phase de chargement des index et la construction du fichier inverse étendu, en passant par l'établissement du lien entre les graphes index chargés et les images de la base auxquelles ils correspondent, pas moins de 19 heures sont requises.

Certaines de ces mauvaises performances sont dues au SGBD O_2 lui-même, et sa façon de gérer la mémoire lors du chargement des graphes index et leur images associées. D'autres par contre, comme la lenteur de la construction du fichier inverse nécessitant approximativement 9 heures, sont imputables à la programmation. Ainsi le temps de construction du fichier inverse peut être considérablement diminué, par exemple en ordonnant le fichier inverse et en utilisant des procédures de recherche optimisées lors de sa construction. En effet, chaque arch d'un graphe index devra être potentiellement introduit comme entrée dans le fichier inverse, à moins qu'il ne soit déjà présent. Dans ce dernier cas, seul un ajout de témoins est effectué. La présence de l'arch comme entrée dans le fichier inverse devra donc être vérifiée. Le temps requis pour entrer un arch dans le fichier inverse augmente avec le nombre d'archs déjà introduits. L'ordre de cet augmentation peut être réduit avec des algorithmes de recherche adéquats.

Si le chargement des données de la base O_2 apparaît coûteux à première vue, ceci n'est valable que pour le chargement de toutes les images du corpus en même temps. Ainsi, les opérations de modification des graphes index ou encore les ajouts de nouveaux documents n'ont pas la même ampleur en termes d'exécution. Par exemple, l'introduction de 5 nouvelles images dans la base ne prend en tout que 6mn, chargement et ré-actualisation du fichier inverse compris. De même la modification d'un graphe index dans le fichier inverse ne prend pas plus d'une minute. De tels délais restent acceptables dans un système opérationnel de recherche d'informations. Cependant, nous sommes toute à fait conscients du fait qu'il faudra optimiser les programmes relatifs à cette partie du chargement. Les marges de manoeuvre sont assez larges à ce niveau comme nous le verrons dans la suite de cette section.

Afin d'étudier le comportement des étapes de chargement sur différents documents, nous présentons à la figure 5.18 les temps d'exécution et l'espace de stockage en fonction du nombre des documents chargés. Les temps d'exécution (figure 5.18a) augmentent de façon linéaire quant à l'interprétation logique des graphes index (association des témoins aux concepts existentiellement

quantifiés). Seules 31 minutes s'avèrent être nécessaires pour l'interprétation logique des graphes index. Le coefficient de régression linéaire est de 0.98 dans ce cas.

Pour la construction des tables d'accélération, les résultats sont assez étranges. Ainsi jusqu'à 500 images, l'augmentation des temps d'exécution a été non seulement linéaire, mais très faible. La construction des tables d'accélération ne prend pas plus de 25 minutes pour un tel nombre de documents. Toutefois, à partir de 550 documents, nous avons constaté une assez forte ascension de la courbe d'exécution correspondante. À notre avis, ceci s'explique par le fait que comme lors de la construction du fichiers inverse, le parcours des tables d'accélération est séquentiel. En effet, vérifier qu'un arch existe déjà dans une table donnée revient à un parcours séquentiel, dont la complexité croît avec le nombre d'archs déjà existants. Un parcours plus optimisé (ex. dichotomique) permettra d'améliorer les délais obtenus.

Pour la construction du fichier inverse, très vite nous nous apercevons que les temps d'exécution augmentent de façon exponentielle. Dans un intervalle de 200 documents les temps de chargement doublent, triplent voire même quadruplent. Ainsi, pour 450 documents 1 heure 20 minutes est nécessaire pour la construction du fichier inverse. Ce chiffre passe à plus de 8 heures pour 650 documents. Comme nous l'avons déjà expliqué, ces chiffres peuvent être corrigés en optimisant les programmes réalisés.

Une même analyse de l'espace de stockage occupé par les diverses structures donne des résultats assez positifs. La figure 5.18b illustre ainsi le gain d'espace résultant de l'application de notre approche. Quelque soit le nombre des documents, la taille du fichier inverse ajoutée à celle des tables d'accélération reste inférieure à celle occupée par les index originaux. Comme aucune optimisation des structures des données n'a été spécialement menée lors de l'implantation. Il est donc encore possible de compresser les structures de notre approche.

5.5.2 Résultats sur les Requêtes

Notre approche devrait *a priori* favoriser les temps de réponse du système. Nous avons ainsi voulu comparer Relief au système de recherche d'images EMIR^2 de Mechkour [Mec95c]. Ce système se base sur l'algorithme de projection de Sowa [Sow84] comme fonction de correspondance. Rien de surprenant donc que les temps de réponse au départ dépassent les 3 minutes pour un corpus d'une cinquantaine d'images[6]. L'algorithme de projection utilisé est de complexité exponentielle. Ces dernières années, dans un effort d'optimisation intensif mené dans le cadre du projet FERMI [Fer94], un autre algorithme de projection a été utilisé dans EMIR^2. Il est basé sur des techniques de présélection utilisant entre autres l'approche des signatures[7] [Che92, Khe95]. Cet effort d'optimisation a été payant puisque la dernière version du système EMIR^2 présentée en Mai 97 dans la revue finale du projet FERMI montre un temps de réponse moyen de 20 secondes.

Dans la suite, EMIR^2 sera utilisé comme notre test de référence. La raison de ce choix est que parmi les divers systèmes opérationnels présentés dans le cadre du projet FERMI, EMIR^2 a montré les résultats les plus encourageants en termes de rapidité, de qualité de réponses mais aussi en ergonomie. Il serait donc intéressant de le comparer à Relief sur la même collection test, à savoir PARISYS.

6. Expérimentation dans le cadre du prototype de recherche médicale PRIME [MMFB97].

7. La technique des signatures consiste à définir une représentation très compacte de données complexes, à partir de laquelle une présélection séquentielle rapide (filtrage) est possible avec un faible coût additionnel en place occupée. Pour plus de détail on pourra consulter [Gau89].

Numéro Requête	*RELIEF* (*temps en secondes*)	*EMIR*^{2} (*temps en secondes*)
01	1	20
02	2	24
03	1	17
04	2	20
05	3	15
06	1	16
07	2	18
08	3	13
09	1	6
10	-	-
11	1	13
12	2	15
13	1	20
14	2	53
15	2	5
16	3	35
17	1	8
18	1	14
19	1	10
20	2	24
21	2	14
22	3	32
23	-	-
24	1	8
25	1	4
26	3	11
28	2	10
29	2	14
30	3	9
Moyenne	**1.78**	**18.82**

Table 5.2. Table comparative des temps de réponse

Les résultats obtenus dépassent à vrai dire nos espérances pour une première implantation de l'approche algébrique proposée. Ainsi, RELIEF s'avère être 10 fois plus rapide en moyenne que la dernière version de EMIR^2. Pour les 30 requêtes de PARISYS (voir Annexe C), le temps de réponse de notre système n'a jamais dépassé les 3 secondes (voir table 5.2). L'histogramme de la page 283 illustre la rapidité de RELIEF sur un certain nombre de requêtes.

Un résultat intéressant de cette expérimentation est que le temps de réponse ne dépend pas de

la complexité de la requête. En d'autres termes, le nombre d'archs contenus dans la requête n'a pas d'influence particulière sur son évaluation. Ainsi, partant d'une requête très simple (moins de 5 archs), l'ajout d'autres archs n'altère pas le délai de réponse. La figure suivante illustre cette constatation.

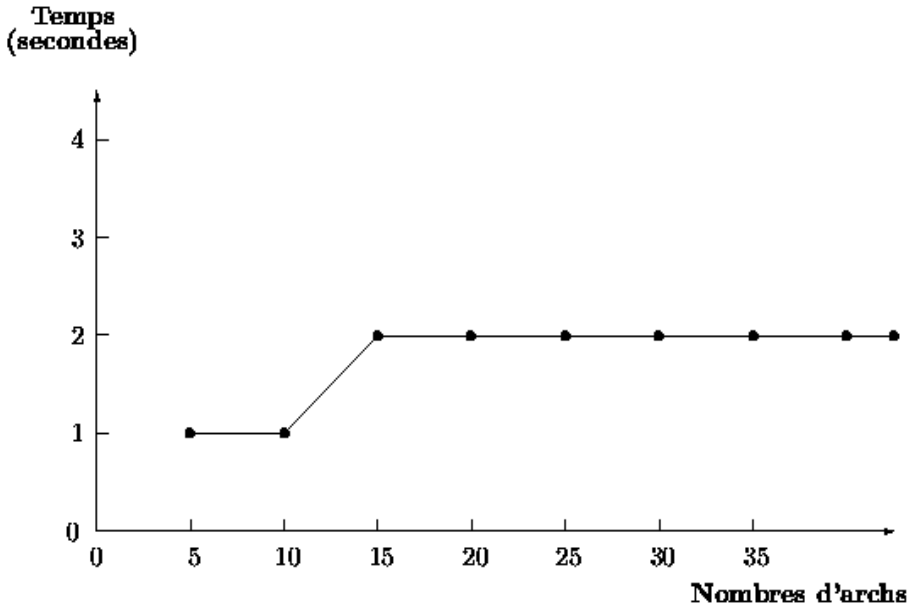


Figure 5.13. Temps estimé pour l'évaluation d'une requête

Par contre, le temps d'évaluation croît avec le degré de généricité de la requête. Ainsi, lorsque cette dernière contient des concepts trop génériques, l'étape de recherche des projections partielles nécessite plus de temps. Dans ce cas, les archs correspondant aux concepts génériques adressent un grand nombre d'entrées dans le fichier inverse, et de ce fait les opérations d'intersection et d'union sont très nombreuses, outre le fait qu'elles font intervenir un grand nombre de témoins. Ainsi, une requête contenant le seul concept universel [$\top$] nécessite 40 secondes de traitement dans RELIEF. La même requête requiert 5 minutes et 20 secondes de temps de traitements dans EMIR^2. Dans le cas de RELIEF ceci est lié à l'implantation. En effet, les opérations ensemblistes que nous avons développé ne sont pas optimisées. Elles ont été programmées sans utilisation des techniques de codage binaire classiques. Leur complexité pourrait être sensiblement diminuée. Il va sans dire que l'utilité de telles requêtes dans un contexte de recherche orienté vers la précision des réponses n'est pas envisageable. Il est rare qu'un utilisateur spécialiste pose des requêtes trop génériques.

Les résultats obtenus par RELIEF sont donc très encourageants. Si l'on admet l'hypothèse que le temps d'évaluation augmente proportionnellement à la taille de corpus, on peut prévoir que la version actuelle du système peut effectuer une évaluation de requête sur un corpus de 10 000 images en moins de 25 secondes. L'expérimentation nous permet donc de conclure que l'approche proposée dans le chapitre précédent est envisageable pour des grands corpus.

Relativement aux résultats obtenus, nous pouvons faire les remarques suivantes concernant des

points à développer dans le futur:

- L'optimisation des opérateurs ensemblistes implantés pourra sensiblement réduire le temps de réponse. Une telle réduction pourra se sentir dans des corpus comportant des dizaines de milliers de documents.
- L'évaluation des sous-requêtes par des techniques de présélection comme les signatures peuvent largement améliorer les performances.
- L'exploitation des caractéristiques des graphes conceptuels utilisés (ils sont conformes à un modèle d'images particulier) pourra favoriser les pré-sélections et donc la réduction des temps de calcul. Il s'agit de diminuer le nombre d'intersections et d'unions sur les témoins. En particulier, les entrées du fichier inverse peuvent contenir des sous-graphes à la place d'archs dans des cas à définir.

Le dernier point est particulièrement intéressant à étudier d'un point de vue théorique. Ainsi, le fait que les graphes conceptuels utilisés aient une syntaxe particulière pourra guider la construction d'un fichier inverse ayant une structure plus optimisée. Il ne s'agit pas donc de projeter deux graphes quelconques mais des graphes particuliers ayant des caractéristiques bien définies dont l'exploitation pourra favoriser l'obtention plus rapide des réponses.

Ainsi dans sa version actuelle, notre approche est non seulement performante, mais offre une très grande marge de manoeuvre pour des optimisations futures, permettant de l'envisager sans problèmes majeurs pour de très grands corpus.

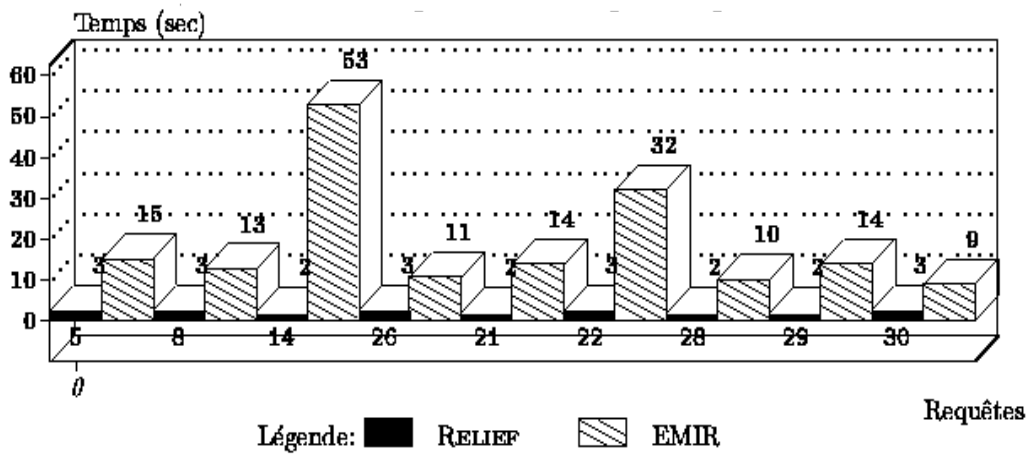


Figure 5.14. Histogramme comparatif des temps de réponses

5.5.3 Évaluation Qualitative

Nous utilisons pour l'évaluation, les jugements de pertinence associés aux 30 requêtes de la collection PARISYS. Pour chacune de ces requêtes, on dispose de quatre ensembles de jugements

de pertinence, donnés par 4 personnes différentes. Ainsi, chaque requête associe à chaque document de la collection un nombre compris entre 0 et 4, qui représente le nombre des personnes qui ont jugé ce document pertinent pour la requête. Pour nos mesures de rappel et de précision, nous supposons qu'une image doit être retrouvée par le système, si elle a été jugée pertinente par au moins 2 personnes.

Notre approche devrait améliorer les performances qualitatives de la recherche. Cela devrait toucher à la fois les taux de rappel et de précision. En effet, au niveau du rappel, notre approche par le fait qu'elle favorise la prise des décisions de pertinence en effectuant des raisonnements sur les relations, rend plus de documents pertinents que la seule considération de l'opérateur de projection. D'autre part, nous caractérisons de façon plus précise les connaissances du système, en représentant les propriétés des relations du domaine de l'application. Notre approche devrait donc également favoriser la précision.

Ces deux hypothèses ont été vérifiées en soumettant les requêtes de PARISYS à RELIEF. La figure 5.17 montre les courbes de rappel/précision relatives à trois systèmes de recherche d'informations: SMART, EMIR^2 et RELIEF. Par rapport au système de recherche à base de mots-clés SMART, le bien-fondé de l'utilisation des graphes conceptuels comme formalisme de représentation et de recherche est très net. Il apparaît ainsi clair que la recherche de documents complexes comme les images ne peut être efficacement menée par une approche classique de mots-clés.

La comparaison de notre système avec EMIR^2 pose plusieurs problèmes, essentiellement dus à notre souhait de comparer les deux systèmes sur des bases égales et justes, du point de vue des temps de réponses. En effet, autant les propriétés des relations représentées au niveau du canon d'un système sont nombreuses, autant le risque est grand de voir les graphes index étendus par un très grand nombre d'archs dérivés. Ainsi, le système sous-jacent verra sa projection s'appliquer sur des graphes conceptuels de plus grande taille. Cela pourra le défavoriser lors de l'estimation des temps de réponse. D'autre part, l'absence de données sur les relations dans l'ontologie fournie avec la collection test rend subjectif tout effort d'extension du canon originel. D'autant plus que la détermination de toutes les propriétés des relations demande une expertise dans le domaine de l'application. La définition de ces propriétés sans le recours à cette expertise affectera la crédibilité des résultats. Dès lors nous nous sommes contentés de représenter les propriétés mathématiques des relations ainsi que certaines inversions. Il s'agit des propriétés faciles à détecter. Cela mettra à pied d'égalité les deux systèmes, puisque EMIR^2 implante de façon *ad-hoc* la transitivité, la symétrie et l'inversion. Seules certaines relations de composition qu'on avait jugé triviales ont été ajoutées au canon.

Les résultats donnent quand même un léger avantage à notre système. Ceci s'explique surtout par le fait que la projection implantée dans EMIR^2 est injective. Notre système adopte par contre une projection plus générale, non nécessairement injective, conforme donc à la définition originale de Sowa. Une projection injective est forcément incomplète d'un point de vue logique, puisque certaines implications évidentes ne sont pas détectées. Ainsi, pour la requête PARISYS numéro 29, le système EMIR^2 ne fournit pas de réponses simplement parce qu'il n'existe pas de graphe index contenant à la fois les concepts [COMMERCE] et [ENSEIGNE]. Pourtant dans le treillis des types de concepts nous avons ENSEIGNE $\leq$ COMMERCE. De ce fait la requête devra se projeter sur tout document contenant le concept [ENSEIGNE]. Il suffit de projeter [COMMERCE] et [ENSEIGNE] sur le même concept [ENSEIGNE] apparaissant dans le document. Le système RELIEF détecte ces projections et affiche les documents pertinents à la requête 29.

Pour ce qui est des raisonnements sur les relations, le système EMIR^2 tient compte de toutes

les propriétés mathématiques des relations, en étendant les graphes index par les informations implicites engendrées par ces propriétés (voir chapitre 2). De ce point de vue, RELIEF n'élabore pas des décisions de pertinence supplémentaires. Au fait, les capacités de notre système n'ont pas été pleinement exploitées lors de cette première expérimentation. Les propriétés d'agencements sémantiques entre les relations n'ont pas été représentées dans le canon, et les règles de dérivation qui leurs sont associées n'ont pu être lancées. De plus, la collection PARISYS ayant été indexée uniquement par des relations binaires, les possibilités de raisonnement sur les relations n-aires n'ont pas été mises en oeuvre dans cette expérimentation. Pourtant, ces deux dernières fonctionnalités ont été implantés.

Par ailleurs, afin de montrer l'importance du raisonnement sur les propriétés des relations, nous avons essayé de lancer le système RELIEF sans la prise en compte du fragment des propriétés mathématiques considérées. La figure 5.15 présente le résultat de cette expérimentation.

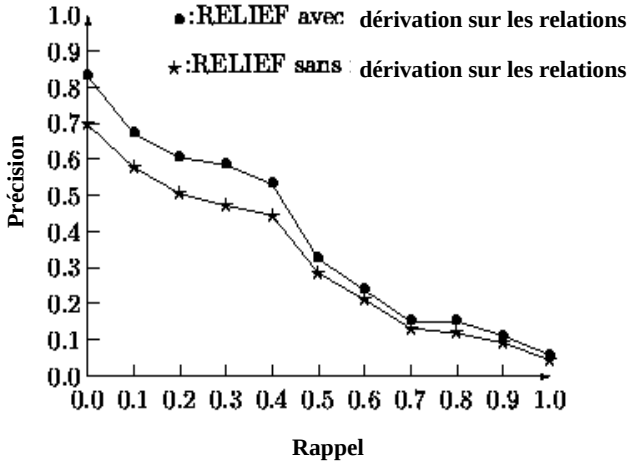


Figure 5.15. Importance des propriétés des relations

Dans cette figure, la courbe «RELIEF avec dérivation sur les relations» désigne les résultats du système RELIEF tels qu'ils ont été obtenus sur les requêtes de PARISYS. À l'opposé, la courbe «RELIEF sans dérivation sur les relations» désigne toujours les résultats obtenus par le système RELIEF sur la collection PARISYS, mais cette fois-ci sans l'utilisation de l'ensemble K_s des propriétés des relations. Ces résultats montrent que la seule utilisation des propriétés mathématiques des relations permet une amélioration nette des performances du système. Par rapport à un modèle de base des graphes conceptuels[8], l'utilisation, même très partielle, des propriétés des relations permet une amélioration de 9% des performances du système.

Durant ces expérimentations, nous avons remarqué que les valeurs de précision et de rappel sont faibles pour certaines requêtes. Ceci s'applique aussi bien au système RELIEF qu'à EMIR^2 (voir figure 5.16). Ces faibles taux peuvent s'expliquer par les raisons suivantes:

– Les limitations des index originaux de la collection test tels qu'ils ont été fournis par les

8. Nous rappelons que sans l'utilisation des propriétés des relations, la projection implantée par RELIEF est équivalente à celle de Sowa.

Requête	*RELIEF*		$EMIR^2$	
	(Rappel)	*(Précision)*	*(Rappel)*	*(Précision)*
05	1	0.8	0.8	0.8
26	0.05	1	0.05	1
14	0.5	0.16	0.18	0.18
21	0.34	0.72	0.34	0.72
22	0.46	0.95	0.46	0.95
29	0.51	0.69	0	0
08	0.45	0.625	0.45	0.625
28	0.13	0.42	0.13	0.42

Figure 5.16. Quelques mesures de Rappel/Précision

spécialistes du Ministère de la Culture. En effet, malgré la qualité de ces index, ils ne sont pas exemptés d'erreurs. Il existe des éléments d'images qui n'ont pas été représentés par ces spécialistes bien qu'ils peuvent être jugés importants par les utilisateurs. Comme le processus d'indexation à base de graphes conceptuels a été mené à partir de ces index, ces erreurs d'appréciation ont pu être propagés aux graphes index.

- Les personnes ayant décidé des images pertinentes associées à chacune des requête ont procédé à la base d'un trombinoscope où les 650 images du corpus apparaissaient dans un petit format. Certains détails des images ont dus leur échapper. De plus certaines requêtes sont ambiguës et peuvent avoir plusieurs interprétations pouvant donner lieu à des confusions lors de leur représentation en graphes conceptuels.

- Notre choix de considérer qu'une image devra être retrouvée par un système si au moins deux personnes l'on jugé pertinente est discutable puisqu'il ne se base sur aucune raison spéciale. De plus, les graphes conceptuels constituent un formalisme orienté vers la précision des réponses et notre choix ne favorise pas cette caractéristique implicite du formalisme.

5.6 Conclusion

Nous avons présenté dans ce chapitre une expérimentation qui intègre dans un système complet notre modèle relationnel pour les graphes conceptuels, et qui démontre qu'il peut être implanté de façon performante et efficace. En particulier le bien-fondé et la faisabilité de l'approche algébrique adoptée pour les graphes conceptuels ont été confirmés. La fonction de projection implantée est ainsi réellement rapide, ne dépassant pas en moyenne les 2 secondes pour un corpus de 650 images. Les résultats de cette expérimentation confirment donc notre idée de départ, à savoir qu'une bonne interprétation logique des graphes conceptuels est le point de départ de toute activité d'implantation efficace de ce formalisme.

Si le système décrit dans ce chapitre est dédié à la recherche d'images, il n'est en rien limité à ce média. Le choix d'une architecture basée sur une plate-forme indépendante (ROGER) permet

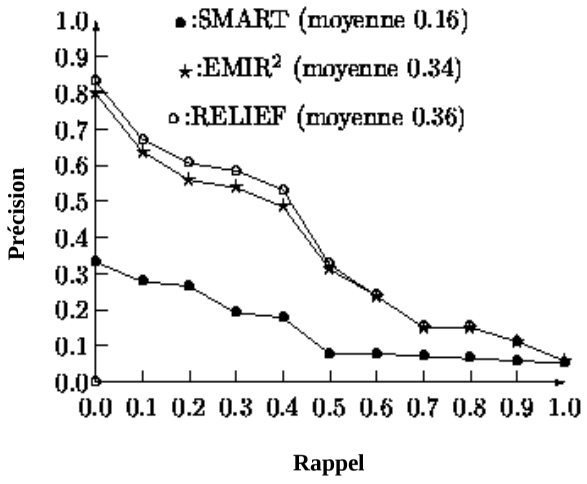


Figure 5.17. Comparaison des Résultats qualitatifs

de traiter n'importe quel type de document, incluant des documents hétérogènes (image, texte, graphiques, etc.), pourvu que l'indexation soit faite dans le formalisme des graphes conceptuels. Il est en effet possible de comparer une image et un texte, si les deux documents sont indexés par rapport au même canon.

Les résultats qualitatifs de notre approche ont été comparés au système SMART puis au système EMIR^2. Cette comparaison, même si elle a été partielle, puisqu'elle n'a porté que sur l'intégration des propriétés mathématiques ainsi que quelques relations évidentes d'inversion, montre l'importance de la prise en compte des relations et leurs propriétés dans le processus de correspondance. De plus, l'expérimentation montre le bien-fondé de l'adoption d'une approche logique pour la construction des systèmes de recherche d'informations. La complétude de la logique sous-jacente permet d'assurer toutes les déductions possibles et contribue ainsi à améliorer le taux de rappel. La requête numéro 29 de la collection test[9] confirme ce fait.

Il existe plusieurs directions pour étendre ce travail. D'abord en observant la figure 5.11, une personne peut être sceptique quant à la facilité de manipulation de cette interface, comparée par exemple à celles associées aux moteurs de recherche sur le Web. En effet, même si cette interface permet de guider l'utilisateur dans la construction de son graphe, il n'en reste pas moins qu'elle risque de décourager certains utilisateurs vu le temps que nécessite la saisie d'une requête simple. Une aide appréciable serait de permettre à ces derniers de poser leurs requêtes directement en langue naturelle et de fournir un système de traduction du texte vers les graphes conceptuels. Ainsi, dans le cadre du projet IOTA [Def86], des investigations récentes sont menées au niveau de notre équipe pour fournir une meilleure compréhension des requêtes en langue naturelle. En marge d'une participation aux expérimentations de TREC [Jon95], ce système a été repris et permet actuellement l'indexation automatique de très larges volumes d'informations. Les index consistent en des structures conceptuelles proches dans leur forme des clauses de Prolog. Elles peuvent donc être transformables en graphes conceptuels. Les résultats actuels de ce système permettent de

9. voir section précédente.

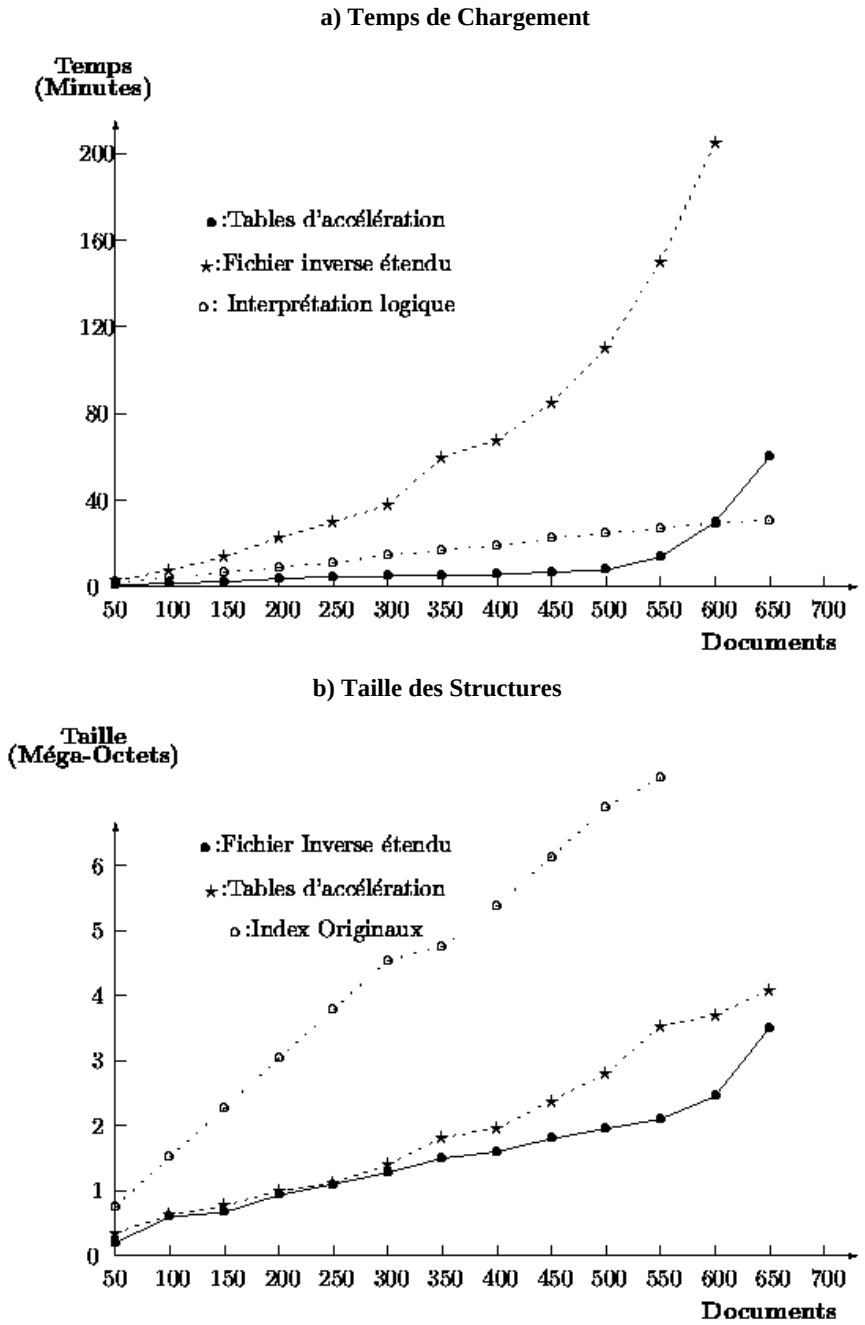


Figure 5.18. Chargement des Données

l'envisager que ce soit comme un système de traduction du texte vers les graphes conceptuels au niveau de l'interface, ou encore comme outil d'indexation automatique des documents textuels.

Toujours au niveau de l'interface, le langage des requêtes actuellement utilisé n'est pas assez expressif. Ainsi puisque le modèle des graphes conceptuels sur lequel se base ce langage ne permet

ni l'expression des négations (NOT), ni celles des disjonctions (le OR), certaines requêtes sont impossibles à formuler. C'est le cas par exemple de la requête 10 de la collection PARISYS (voir annexe C). Notre démarche algébrique du chapitre précédent, nous permet toutefois de résoudre ce problème. En effet, un opérateur comme le OR appliqué aux graphes conceptuels peut être implanté comme une union de témoins sur notre fichier inverse. De même, la négation NOT peut être implantée comme une différence ensembliste. D'un point de vue logique, il est facile de leur associer une sémantique claire. Ainsi, la disjonction de deux graphes peut être interprétée comme une disjonction de la conjonction des formules des graphes associées à leurs archs. La négation d'un graphe, quant à elle, peut être interprétée comme la négation de la conjonction des formules de graphes associées à ses archs. D'un point de vue pratique, nous pouvons envisager qu'un graphe non connexe [MC96] exprime une disjonction des sous-graphes qu'il contient. Pour la négation, nous pouvons utiliser les notations des graphes contextes. Les notations graphiques risquent donc de se complexifier, si un tel langage de requête est autorisé. Cela confirme ce que nous disions un peu plus haut, à savoir que la formulation des requêtes devra se faire de façon conviviale à défaut d'utilisation de la langue naturelle.

Enfin, du point de vue de l'implantation du système elle-même, nous sommes entrain d'étudier des méthodes permettant l'optimisation des algorithmes de construction des structures utilisées ainsi que la réduction de leur taille. Un travail similaire est également mené sur les méthodes liées à l'algorithme de projection proposée. Une telle amélioration nous permettra d'envisager sereinement des collections de très grande taille comme TREC.

Chapitre 6

Conclusion

6.1 Conclusions et Apports

L'importance d'un modèle formel pour la recherche d'informations n'est plus à démontrer. La qualité de ce modèle influence directement les performances du système. Beaucoup de travaux se sont intéressés à un modèle général de correspondance [vR86, Nie90, Hui96], spécifiant les décisions de pertinence que doit élaborer un système de recherche d'informations, d'autres plus récents se sont proposés de définir un modèle général pour l'indexation [Par96], décrivant essentiellement comment représenter les index. Aucun modèle cependant ne s'est attaché à proposer un modèle général d'indexation et de correspondance pour la recherche d'informations. Notre approche se veut un pas dans cette direction. L'originalité du travail décrit dans cette thèse se situe principalement à ce niveau. Notre modèle d'indexation relationnel spécifie les caractéristiques que doit présenter un système de recherche d'informations aussi bien au niveau de la représentation des index qu'au niveau de la correspondance.

Notre approche fait de la *relation*, une notion centrale dont la présence dans les index des documents est plus que nécessaire, à en juger les récentes applications de la recherche d'informations. Elle fait aussi en sorte que cette notion soit au coeur de toute décision de pertinence. Enfin, de part la richesse sémantique des relations, notre approche donne une place de choix à la représentation des nombreuses propriétés qui en découlent au niveau de la base de connaissances du système.

En se basant sur les applications auxquelles est dédiée la recherche d'informations d'aujourd'hui, le modèle proposé spécifie le degré d'expressivité minimal que doit présenter un langage en vue d'une bonne indexation du corpus. Il établie les connaissances que doit représenter un système de recherche d'informations, et fixe enfin les règles de dérivation permettant, en s'appuyant sur les connaissances, d'établir les décisions de pertinence. Le modèle proposé peut donc être vu comme une logique idéale pour la recherche d'informations. La complétude et la correction de cette logique, ainsi que le problème de décidabilité ont été étudiés par rapport à une sémantique entièrement dédiée à la recherche d'informations. La sémantique repose sur les modèles préférentiels de Kraus et al. [KLM90], pour définir une notion de relation de préférence sur les index des documents lors de la phase de correspondance.

L'intérêt de l'approche est que, par rapport à cette logique, il est possible d'analyser, de mieux comprendre puis d'étendre les systèmes de recherche d'informations opérationnels. Il suffit pour cela de les exprimer par rapport à cette logique, puis de détecter les éléments du modèle qu'ils ne

sont pas en mesure de représenter. Les limites d'un système opérationnel peuvent se caractériser au niveau de l'un des trois axes classiques de la recherche d'informations: indexation trop pauvre des index, prise en compte partielle des connaissances du domaine de l'application ou encore des décisions de pertinence incomplètes.

Par rapport à des travaux comme ceux de Huibers [Hui96] ou de Nie [Nie90], nous défendons l'idée que mise à part les règles de dérivation assurant le comportement logique du système sous-jacent, la notion de pertinence est nécessairement liée à des règles de dérivation sur les relations et leurs propriétés. De plus, le modèle proposé ne se contente pas d'étudier les systèmes au niveau de leurs décisions de pertinence, mais également au niveau de l'expressivité de leur langage d'indexation et des connaissances du domaine qu'ils prennent en compte.

Nous avons choisi le formalisme des graphes conceptuels comme application de notre modèle pour un double objectif. D'une part pour montrer un exemple d'utilisation du modèle, et d'autre part pour vérifier sa faisabilité et son bien-fondé. Il en ressort de notre étude que ce formalisme, tout en offrant un langage de représentation adéquat, souffre de certaines insuffisances au niveau de la représentation des connaissances du domaine et de l'élaboration des décisions de pertinence. En particulier, tout ce qui a trait aux propriétés des relations n'est pas pris en charge au niveau du modèle de base des graphes conceptuels.

Par conséquent, une adaptation du formalisme des graphes conceptuels aux besoins de la recherche d'informations a été menée. Un aspect important de ce travail est qu'il constitue une contribution au développement d'un système de représentation de connaissances et de raisonnement fondé sur les graphes conceptuels. Ainsi, la notion du canon a été entièrement révisée pour pouvoir représenter les propriétés des relations. Le modèle ainsi étendu permet de gérer un treillis de types de relations d'arités différentes. Enfin, la projection a été étendue de façon à assurer un raisonnement sur ce treillis.

De plus, en nous plaçant dans le cadre d'une approche relationnelle d'indexation, nous avons précisé plusieurs notions ambiguës du modèle de base des graphes conceptuels. Des règles de dérivations sur les graphes ont été proposées pour appréhender cette approche. Les caractéristiques du modèle des graphes conceptuels font que ces règles sont décidables. Nous avons ensuite proposé une nouvelle interprétation logique permettant, sans pour autant dépasser le cadre des graphes simples, d'étendre le fragment de la logique de premier ordre correspondant à ce formalisme. Il s'agit de pouvoir représenter des concepts universellement quantifiés.

Une technique classique de la logique de premier ordre a été utilisée pour pouvoir envisager des traitements rapides au niveau du formalisme ainsi étendu. Cette technique est une instanciation de la nouvelle interprétation logique aux seuls concepts individuels ou existentiellement quantifiés. L'utilisation de cette technique revient à une interprétation des graphes conceptuels correcte et complète vis-à-vis de la définition de la fonction de translation originelle ϕ de Sowa.

L'intérêt de cette approche est de pouvoir représenter les graphes conceptuels sous forme de structures algébriques, et de ramener ainsi les opérateurs sur les graphes à des opérations ensemblistes polynomiales. Une approche de fichier inverse a été ainsi introduite pour les graphes conceptuels. Une fonction de correspondance polynomiale permet de chercher dans cette structure la projection d'une requête sur un corpus de documents, cette projection étant étendue à la prise en compte des propriétés des relations. Contrairement à plusieurs approches existantes, ce résultat n'a pas été le fruit d'une restriction sur la nature des graphes considérés. En effet, n'importe quel graphe conceptuel simple peut être utilisé comme index dans notre démarche. Un autre aspect

positif de cette projection est qu'elle peut être appliquée sur tout le corpus en même temps.

Cette extension relationnelle du formalisme des graphes conceptuels a conduit à la réalisation d'une plate-forme de gestion de graphes conceptuels, implantée au-dessus d'un SGBD orienté objet. Cette plate-forme est générale, en un sens qu'elle ne dépend pas des applications et qu'elle n'est pas dédiée à la recherche d'informations. Elle a notamment servi à implanter le système de recherche d'images RELIEF. Appliquée à une collection test d'images, les résultats de comparaison avec d'autres systèmes de recherche montrent un net avantage pour notre approche que ce soit au niveau de la qualité des réponses qu'au niveau de la rapidité des traitements.

6.2 Perspectives

Au niveau du modèle relationnel proposé, si le langage d'indexation a été décrit, le processus de dérivation automatique des index à partir des documents n'a pas été explicité. Nous avons ainsi mentionné dans le chapitre 2, les difficultés d'une indexation relationnelle automatique. Si les techniques d'analyse sémantique de la langue naturelle sont aujourd'hui jugés trop coûteuses, nous pouvons envisager des approches plus pragmatiques visant à reconnaître certaines évidences textuelles permettant d'inférer plus ou moins fiablement la présence d'une structure sémantique dans le texte [TAA89, Par96]. Pour les documents multimédias, les travaux ont surtout porté sur les images. La difficulté de la tâche a orienté les travaux vers la recherche de techniques permettant d'intégrer l'indexation des images dans un cadre plus général en prenant en considération l'environnement. Ces travaux n'abordent plus la recherche dans des bases constituées uniquement d'images mais considèrent que l'image est un élément parmi d'autres dans le document et qu'il est donc souhaitable d'utiliser cet environnement pour indexer les images [HSD97].

Au niveau du modèle théorique lui-même, le plus important ajout concerne une meilleure intégration des extensions qui lui ont été appliquées à la fin du chapitre 3. Plus particulièrement, l'intégration des règles par défaut pour la modélisation des situations, l'utilisation des labels des règles pour justifier les réponses du système ou encore pour prendre en compte ses jugements de pertinence méritent un plus grand intérêt. Ces travaux s'inscrivent dans un effort plus général de modélisation *contextuelle* des systèmes de recherche d'informations, et intègrent des caractéristiques jusqu'à maintenant considérées comme *externes* au système. Nous projetons ainsi de lier ces extensions aux travaux de Denos [Den97] sur la modélisation de l'utilisateur et la prise en compte des situations de recherche dans laquelle il opère.

La notion de facette est également importante pour pouvoir assurer une indexation fidèle des documents complexes. Nous l'avons déjà appliqué au formalisme des graphes conceptuels dans [OC96b]. Une définition intuitive de la notion de facette a été donnée dans cette thèse. Des travaux plus poussées doivent cependant être menés pour caractériser les liens entre ces facettes et offrir un mécanisme d'aide à leur identification. Un autre aspect à considérer au niveau du modèle relationnel proposé consiste à étudier des notions logiques comme la *consistance* et la *contradiction*. En particulier, l'introduction des règles par défaut ou la prise en compte de la notion de non-pertinence imposent des travaux à ce niveau. La théorie des situations permet d'aborder ces problèmes dans un cadre formel adéquat [BE87].

Au niveau du formalisme des graphes conceptuels, il peut exister une multitude de suites aux travaux ici présentés. La première extension concerne l'introduction de l'incertitude au niveau de l'indexation des graphes mais aussi au niveau des dérivations. Les quatre opérateurs de construction

des graphes ainsi que la fonction de translation des graphes conceptuels doivent ainsi être modifiés en conséquence. L'intégration de l'incertitude permettra d'appliquer les extensions proposées au niveau du modèle relationnel. Nous avons commencé à étudier cet aspect [AO97], en nous basant sur les méthodes probabilistes utilisées dans les réseaux de croyances [Pea88, vdG90, Bru93]. Nous sommes entrain également de considérer l'utilisation de la théorie d'évidence de Dempster-Shafer [Lal96] pour une modélisation de l'incertitude au niveau des graphes conceptuels. Cette théorie a été déjà appliquée par Lalmas à un modèle de recherche de documents structurés et multimédias [1] développé par l'équipe MRIM [CFM96, Lal97]. Ce modèle utilise une logique propositionnelle pour la représentation des index. Le passage au formalisme des graphes conceptuels serait ainsi intéressant à envisager.

Dans cette thèse, nous avons bâti notre approche relationnelle sur le modèle de base des graphes conceptuels. Seuls des marqueurs simples ont été utilisés dans les graphes index. Il est cependant intéressant de pouvoir exploiter toute l'expressivité du formalisme. Les graphes conceptuels avec contextes (ou les *graphes emboîtés* [MC96, Car96]), peuvent ramener l'expressivité de ce formalisme à un niveau équivalent au langage d'indexation relationnelle que nous avons proposé au chapitre 3. Dans ce cas, il sera possible de représenter des structures récursives [2]. Ces structures s'avèrent notamment adaptées à la recherche des documents structurés [Khe95], même s'il est bien sûr toujours possible de ramener des graphes emboîtés à des graphes simples en ajoutant des relations conceptuelles spéciales [3]. Afin d'utiliser ces structures, il faudra réétudier les aspects formels du formalisme à ce niveau. En effet, Sowa n'a pas spécifié d'opérations sur ce type de graphes. De plus, il ne leur a été associé aucune interprétation logique. Nous avons entrepris des travaux à ce niveaux. Ainsi, s'il est facile de définir des opérateurs sur les graphes emboîtés, la recherche d'une interprétation logique correcte et complète par rapport à ces opérateurs est encore un problème ouvert. Nous nous sommes ainsi heurtés à la difficulté de trouver une forme logique prenexe à ce type de graphe. Nous entreprenons actuellement des travaux à ce niveau [AO97]. Une forme prenexe facilitera les dérivations sur les formules de graphes associées.

Par ailleurs, nous pouvons adopter la même démarche algébrique ici présentée pour le cas des graphes emboîtés. Une approche de fichier inverse peut être menée selon la même stratégie développée dans le chapitre 4. Cependant, sans une interprétation logique correcte et complète de ces graphes, nous ne pouvons pas prouver que l'algorithme implanté est réellement une projection d'un graphe emboîté sur un autre. Dans une telle optique la recherche d'une fonction de translation ϕ étendue aux graphes emboîtés s'impose.

L'étude formelle des graphes emboîtés peut s'inspirer des règles de dérivation de notre modèle d'indexation relationnel pour imaginer des opérateurs de formation de graphes adéquats. En effet, comme l'objectif de notre travail était de comparer ce dont a besoin les applications récentes de recherche d'informations et ce dont sont capables de fournir les formalismes opérationnels, il serait judicieux d'exploiter la différence pour proposer des améliorations en conséquence. Le même travail peut se faire au niveau d'autres formalismes comme les logiques terminologiques. L'extension de ces logiques par des opérateurs de formation de rôles, peut suivre notre démarche relationnelle. Nous espérons ainsi qu'en proposant ce modèle de façon entièrement basée sur des applications réelles, nous avons aidé à identifier les besoins et à encourager les spécialistes de ces formalismes à proposer des extensions dans ce sens.

1. Pour des plus amples détails sur ce modèle, voir chapitre 2.
2. Cela revient à avoir des relons de relons.
3. P. Mulhem [Mul93] avait proposé de telles relations pour faire passer les arborescences récursives du système RIME vers des graphes conceptuels simples.

Annexe A

Notations OMT

Nous nous sommes basés sur les notations données par Rumbaugh et al. dans [RBE+94]. La figure A.1 illustre celles utilisées dans le chapitre 5 de cette thèse.

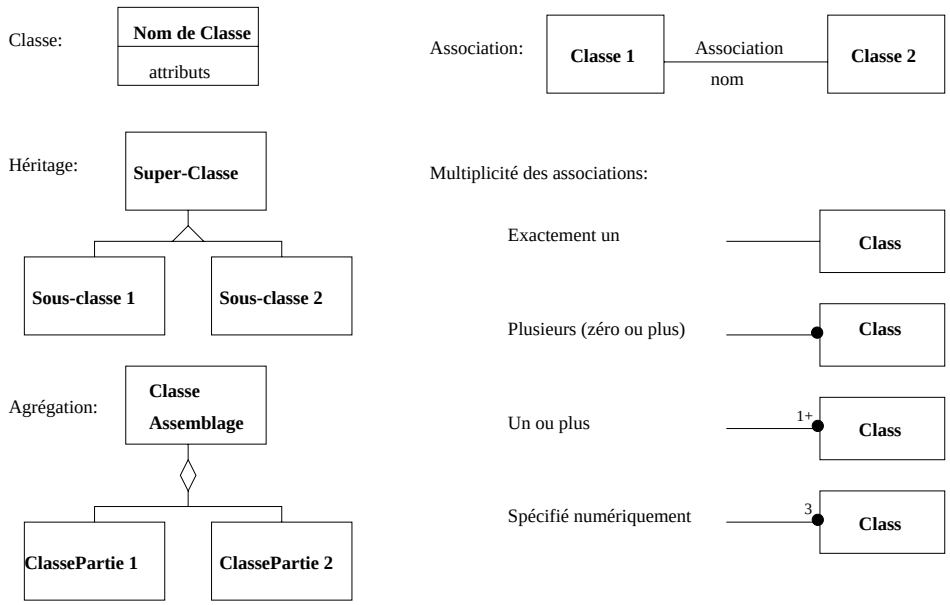


Figure A.1. Notations de la méthode OMT

Annexe B

La Collection Test

L'indexation dans le corpus originel fourni par le Ministère de la Culture recouvre plusieurs aspects des images, identifiés par plusieurs attributs : Titre, Auteur, Genre de l'image, Commentaire, Contenu, Précisions, Connotation, Morphologie, Lumière. Elle a été réalisée par des documentalistes selon un modèle attribut-valeur. L'attribut Numéro correspond simplement à l'identificateur de la photographie.

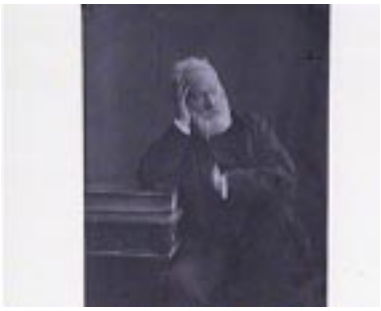


Figure B.1. Image 4448 de la collection test

La figure B.2 donne l'index originel de l'image 4448 précédente. Les attributs Auteur, Genre de l'image, Contenu, Connotation, ont des valeurs dans une liste de mots-clés. Les attributs Titre, Précision et Commentaire peuvent avoir pour valeurs des phrases en langue naturelle pouvant atteigner la taille d'un paragraphe.

L'attribut Contenu fait référence à la description du contenu sémantique de l'image. Ce contenu a été soigneusement choisi par les documentalistes spécialistes. Les attributs Commentaire et Précisions fournissent des informations contextuelles qui complètent la description du contenu sémantique de l'image. L'attribut Connotation donne un point de vue subjectif sur l'image. Les attributs Plan, Morphologie et Lumière décrivent des caractéristiques techniques de la photographie. Enfin, Numéro, Auteur et Titre constituent des attributs classiques de bases de données. Ils sont couramment appelés *Attributs Externes*

Attributs	**Valeurs**
Numéro	4448
Titre	Victor Hugo agé Accoudé à une table
Auteur	Nadar Atelier
Genre de l'image	Portrait
Commentaire	Hugo a une main passée dans son gilet, l'autre soutenant sa tête.
Contenu	Auteur Litteraire-D; Assis; Personne Agée; Rinceau; portrait-D; barbe; livre-D; homme; table-D;
Précisions	Victor Hugo est un écrivain français célèbre
Connotation	austère
Morphologie	plan rapproché; photographie en atelier; vue de trois-quarts

Figure B.2. Exemple d'image indexée

Annexe C

Les Requêtes FERMI

La collection PARISYS est munie de 30 requêtes données dans la table C.1.

Tab. C.1 – *Requêtes de la collection test*

01	militaire
02	femmes en habit contemporain
03	restaurants
04	mort
05	une personne à droite d'une table
06	des chevaux, et des escaliers ... où les chevaux sont à gauche des escaliers
07	une personne à droite d'une fontaine
08	des rues où l'on voit des enfants
09	une porte à droite d'escaliers
10	art religieux (mais pas de sculptures murales)
11	immeubles près de l'eau
12	premiers exemples d'architecture moderne
13	escaliers
14	gens qui travaillent
15	immeubles où le toit est clairement visible
16	transports
17	réflexion (reflet)
18	intérieurs
19	scènes de rue

20	animaux (réels et représentés)
21	gens assis
22	hommes avec une barbe (cad pas seulement une moustache)
23	vues depuis une position élevée
24	inondations
25	bord de mer/plage
26	gens portant des chapeaux
27	mots visibles (dans n'importe quelle langue)
28	bateaux
29	commerce avec enseignes
30	bois/campagne

Annexe D

Le Canon

D.1 Les Types de Concepts

L'extension -D ajoutée à certains types de concepts signifie «divers», et regroupe tous les termes spécifiques du père de ce terme, et qui ne sont pas répertoriés explicitement. Ainsi, ARTISAN DU CUIR-D dénote tous les artisans du cuir qui ne sont ni cordonniers, ni tanneurs, ni bottiers.

TAB. D.1 – *Extrait du treillis des types de concepts*

- o ARTISANAT
  - PROFESSION DE L'ARTISANAT
    - ARTISAN
      - ARTISAN DU CUIR
        - CORDONNIER
        - ARTISAN DU CUIR-D
        - TANNEUR
        - BOTTIER
      - ARTISAN DU BATIMENT
        - RAMONEUR
        - MARBRIER
        - ELECTRICIEN
        - MACON
        - PEINTRE EN BATIMENT
        - COUVREUR
        - PLOMBIER
        - CHARPENTIER
        - PLATRIER
        - ARTISAN DU BATIMENT-D
        - TAILLEUR DE PIERRE
        - VITRIER
        - MENUISIER
        - CHAUFFAGISTE
        - SERRURIER
      - VANNIER
      - BARBIER
        - CARDEUR
      - COMPAGNON
      - ARTISAN DU METAL
        - MARECHAL-FERRAND
        - CHARRON
        - ETAMEUR
        - FERRONNIER
        - FORGERON

FONDEUR
REMOULEUR
CHAUDRONNIER
ORFEVRE
FERBLANTIER
LAVANDIERE
MEUNIER
VERRIER
HORLOGER
TONNELIER
PALUDIER
REMPAILLEUR
POTIER
COIFFEUR
CHEF D'OEUVRE
OUTILLAGE ARTISANAL
OUTIL DE L'ARTISANAT
RABOT
ROUET
PINCE
CLE A MOLETTE
LIME
TENAILLE
TOURNEVIS
ETAU
QUENOUILLE
FUSEAU
BATTE
BOITE A ONGLET
GOUGE
ENCLUME
OUTIL DE L'ARTISANAT-D
CREUSET
BURIN
FOUR
METIER A TISSER
METIER A TISSER-D
NAVETTE
ETABLI
OUTILLAGE ARTISANAL-D
MEULE

D.2 Les graphes signatures

La figure suivante donne un extrait des signatures de relations utilisées dans notre expérimentation:

[*VueStructurelle*]→(Comp)→[ObjetImage];
[Concept]→(Date)→[TypeValeurDate];
[Image]→(Genre-de-l'image)→[TypeValeurGenre];
[Image]→(Identifiant)→[Numero];
[Image]→(Lieu-I)→[TypeValeurLieu];
[Concept]→(Represente)→[TypeValeurRepresente];
[Image]→(Saison)→[TypeValeurSaison];
[Concept]→(Theme)→[TypeValeurTheme];
[Concept]→(Qualificatif)→[TypeValeurQualificatif];
[Concept]→(Contient)→[TypeValeurContient];

[ObjetImage]→(IsA)→[ObjetSymbolique];
[ObjetImage]→(VueSpa)→[ObjetSpatial];
[ObjetSpatial]→(a-droite-de)→[ObjetSpatial];

[ObjetSpatial]→(a-gauche-de)→[ObjetSpatial];
[ObjetSpatial]→(au-dessus-2D)→[ObjetSpatial];
[ObjetSpatial]→(au-dessus-3D)→[ObjetSpatial];
[ObjetSpatial]→(dans)→[ObjetSpatial];
[ObjetSpatial]→(derriere-2D)→[ObjetSpatial];
[ObjetSpatial]→(derriere-3D)→[ObjetSpatial];
[ObjetSpatial]→(devant-2D)→[ObjetSpatial];
[ObjetSpatial]→(devant-3D)→[ObjetSpatial];
[ObjetSpatial]→(en-dessous-2D)→[ObjetSpatial];
[ObjetSpatial]→(en-dessous-3D)→[ObjetSpatial];
[ObjetSpatial]→(pres-de)→[ObjetSpatial];
[ObjetSpatial]→(sur)→[ObjetSpatial];

[ObjetSymbolique]→(ACCOUDE-A)→[ObjetSymbolique];
[ObjetSymbolique]→(AFFRONTE)→[ObjetSymbolique];
[ObjetSymbolique]→(ASPERGEANT)→[ObjetSymbolique];
[ObjetSymbolique]→(ATTAQUANT)→[ObjetSymbolique];
[ObjetSymbolique]→(BROUTANT)→[ObjetSymbolique];
[ObjetSymbolique]→(CARESSANT)→[ObjetSymbolique];
[ObjetSymbolique]→(CONDUISANT)→[ObjetSymbolique];
[ObjetSymbolique]→(CONTENANT)→[ObjetSymbolique];
[ObjetSymbolique]→(ENJAMBANT)→[ObjetSymbolique];
[ObjetSymbolique]→(ENVAHISSANT)→[ObjetSymbolique];
[ObjetSymbolique]→(FAISANT)→[ObjetSymbolique];
[ObjetSymbolique]→(FERMANT)→[ObjetSymbolique];
[ObjetSymbolique]→(GARER)→[ObjetSymbolique];
[ObjetSymbolique]→(HABILLANT)→[ObjetSymbolique];
[ObjetSymbolique]→(LEVANT)→[ObjetSymbolique];
[ObjetSymbolique]→(PECHANT)→[ObjetSymbolique];
[ObjetSymbolique]→(PORTANT)→[ObjetSymbolique];
[ObjetSymbolique]→(TENANT)→[ObjetSymbolique];
[ObjetSymbolique]→(TIRANT)→[ObjetSymbolique];

[ObjetSymbolique]→(AppartenuA)→[TypeValeurAppartenuA];
[ObjetSymbolique]→(Auteur)→[TypeValeurAuteur];
[ObjetSymbolique]→(Etat-de-l'objet)→[TypeValeurEtat];
[ObjetSymbolique]→(Identification)→[TypeValeurIdentification];
[ObjetSymbolique]→(Inscription)→[TypeValeurInscription];
[ObjetSymbolique]→(LieA)→[TypeValeurLieA];
[ObjetSymbolique]→(Lieu-O)→[TypeValeurLieu];
[ObjetSymbolique]→(Materiau)→[TypeValeurMateriau];
[ObjetSymbolique]→(OccupePar)→[TypeValeurOccupePar];
[ObjetSymbolique]→(Porte)→[TypeValeurPorte];
[ObjetSymbolique]→(Position)→[TypeValeurPosition];
[ObjetSymbolique]→(Profession)→[TypeValeurProfession];

Bibliographie

[AAD93] J. Allwood, L.G. Anderson, and O. Dahl. *Logic in Linguistics*. Cambridge University Press, London, 1993.

[AC93] M. Adiba and C. Collet. *Objets et bases de données, le SGBD O_2*. Hermès, 1993.

[Ago93] M. Agosti. Hypertext and information retrieval. *Information Processing & Management*, 29(3):283–285, 1993.

[AK92] G. Amati and S. Kerpedjiev. An information retrieval logical model: implementation and experiments. Technical report, REL 5b04892, Fondazione Ugo Bordoni, Roma, Italy, March 1992.

[All83] J.F. Allen. Maintaining knowledge about temporal intervals. *Communication of the ACM*, 26:832–843, 1983.

[AMC95] M. Agosti, M. Melucci, and F. Crestani. Automatic construction of hypermedia for information retrieval. *Multimedia Systems*, 1995.

[AO97] G. Amati and I. Ounis. Do we need joining and simplify in CGs: a polynomial algorithm for projection in conceptual graphs. MRIM, research report, CLIPS-IMAG, March 1997.

[AO98] G. Amati and I. Ounis. Interpreting conceptual graphs in first order logic: Projections, graph derivation systems and their complexity. Research report RAP98-001 (submitted), FUB (Rome), CLIPS-IMAG (Grenoble), 1998.

[AR89] D. L. McGrunero A.Borgida, R.J. Brachman and L. A Resvick. Classic: A structurel data model for object. *ACM SIGMOD*, pages 58–67, 1989.

[Bar89] J. Barwise. *The situation in logic*. Number 17 in Lecture Notes. Center for Study of Language and Information, 1989.

[BCd86] M.K. Di Benigno, G.R. Cross, and C.G. deBessonet. Corel- a conceptual retrieval system. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy*, pages 144–148, September 1986.

[BCHL93] C. Boksenbaum, B. Carboneill, O. Haemmerlé, and T. Libourel. Conceptual graphs for relational databases. In G. Mineau, W. Moulin, and J.F. Sowa, editors, *Proceedings of the third International Conference on Conceptual Structures, ICCS'93*,

volume 699 of *Lecture Notes in Artificial Intelligence*, pages 142–161, Quebec City, Canada, August 1993. Springer-Verlag.

[BD96] R. Blanché and J. Dubucs. *La Logique et son histoire.* Collection «U», série Philosophie. Armand Colin, Masson, Paris, 1996.

[BDFL88] A. Berard-Dugourd, J. Fargues, and M.C. Landau. Natural language analysis using conceptual graphs. In *Proceedings of the International Computer Science Conference, ICSC'88*, 1988.

[BE87] J. Barwise and J. Etchemendy. *The Liar, An Essay on Truth and Circularity.* Oxford University Press, 1987.

[BE90] J. Barwise and J. Etchemendy. Informations, infons, and inferences. In R. Cooper, K. Mukai, and J. Perry, editors, *Situation Theory and its Applications, volume I*, chapter 2, pages 33–78. CSLI, 1990.

[Bel77] N.D. Belnap. A useful four-valued logic. In G. Epstein and J.M Dunn, editors, *Modern uses of multiple-valued logic*, pages 8–37, Reidel, Dorecht, NL, 1977.

[Ber88] C. Berrut. *Une méthode d'indexation fondée sur l'analyse sémantique de documents spécialisés. Le prototype RIME et son application à un corpus médical.* PhD thesis, Université Joseph Fourier, 1988.

[Ber90] C. Berrut. Indexing medical reports: The rime approach. *Information Processing and Management*, 26(3):93–109, 1990.

[BGPT91] J. Barwise, J.M. Gawron, G. Plotkin, and S. Tutiya. *Situation Theory and its Applications, Volume II.* CSLI Lectures Notes, Number 26. CSLI, Stanford, 1991.

[BH95] P.D. Bruza and T.W.C. Huibers. How nonmonotonic is aboutness ? Technical Report, UU-CS-1995-09, Department of Computer Science, Utrecht University, March 1995.

[Bla86] W.J. Black. *Intelligent Knowledge Based Systems-An introduction.* Van Nostrand Reinhold, 1986.

[Bla90] D.C. Blair. *Language and representation in information retrieval.* Elsevier science publishers, second edition, 1990.

[Bla93] A.W. Black. *A situation Theoretic Approach to Computational Semantics.* PhD thesis, Department of Artificial Intelligencem, University of Edinburgh, UK, 1993.

[BP83] J. Barwise and J. Perry. *Situations and Attitudes.* Bradford Book, MIT Press, 1983.

[Bra77] R.J. Brachman. What's in a concept : structural foundations for semantic networks. *International Journal of man-machine studies*, 9:127–152, 1977.

[Bra79] R. Brachman. *On the epistemological status of semantic networks*, pages 3–50. Findler, Academic Press, 1979.

[Bra83] R.J. Brachman. What IS-A is what isn't: an analysis of taxonomic links in semantic networks. *IEEE Computers*, 16(10):30–36, 1983.

[Bri95] M. Brisebois. Modélisation de l'intentionnalité pour la recherche d'informations. Technical report (examen prédoctoral), Département d'informatique et de recherche opérationnelle. Université de Montréal, 1995.

[Bro95] E.W. Brown. Fast evaluation of structured queries for information retrieval. In *Proceedings the 18th Annual International ACM SIGIR Conference on Research and Development inInformation Retrieval, Seatle, USA*, pages 30–38, July 1995.

[Bru93] P.D. Bruza. *Stratified Information Disclosure, a Synthesis between Hypermedia and Information Retrieval.* PhD thesis, Katholieke Universiteit Nijmegen, March 1993.

[BS85] R.J. Brachman and J.G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, April 1985.

[BS92] S. Bergamaschi and C. Sartori. On taxonomic reasonning in conceptual design. *ACM Transaction on Database Systems*, 17(3):385–422, September 1992.

[BZ92] J. Bouaud and P. Zweigenbaum. A reconstruction of Conceptual Graphs on top of a production system. In *Proceedings of the 7th annual workshop on Conceptual Graphs*, 1992.

[CACS94] V. Christophides, S. Abiteboul, S. Cluet, and M. Scholl. From structured documents to novel query facilities. In *ACM SIGMOD*, Minneapolis, USA, May 1994.

[Car66] R. Carnap. *Probability and Content Measure*, pages 248–260. University of Minnesota Press, 1966.

[Car83] J. Caron. *Les régulations du discours, Psycholinguistique et pragmatique du langage.* Presses Universitaires de France, 1983.

[Car96] B. Carbonneill. *Vers un système de représentation de connaissances et de raisonnement fondé sur les graphes conceptuels.* PhD thesis, Janvier 1996.

[CC92] Y. Chiaramella and J.P. Chevallet. About retrieval models and logic. *The Computer Journal*, 35(3), 1992.

[CCO95] J.P. Chevallet, Y. Chiaramella, and I. Ounis. An approach to IR based on modal logic. In *Work Part 1 Deliverable: A Logic for Information Retrieval*, chapter 5, pages 143–154. ESPRIT BRA Project No. 8134 - FERMI, 1995.

[CFM96] Y. Chiaramella, F. Fourel, and P. Mulhem. Modelling multimedia structured documents. Technical report, FERMI BRA 8134, April 1996.

[CH88] R. Chaffin and D.J Herrmann. *The nature of semantic relations: A comparison of two approaches*, pages 261–288. Cambridge University Press, 1988.

[Che92] J.P. Chevallet. *Un Modèle Logique de Recherche d'Informations appliqué au formalisme des Graphes Conceptuels. Le prototype ELEN et son expérimentation sur un corpus de composants logiciels.* PhD thesis, Université Joseph Fourier, Grenoble, 1992.

[Che94] P.S. Chen. On inference rules of logic-based information retrieval systems. *Information Processing & Management*, 30(1):43–59, 1994.

[CK73] C.C. Chang and H.J. Keisler. *Model Theory.* North-Holland, 1973.

[CL92] H. Chen and K.J. Lynch. Automatic construction of networks of concepts characterizing documents databases. *IEEE Expert Intelligent Systems & their Applications*, 22:887–902, 1992.

[Cle91] C.W. Cleverdon. The significance of the Cranfield tests on index languages. In A. Bookstein, Y. Chiaramella, G. Salton, and V.V. Raghavan, editors, *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, Chicago, October 1991. ACM, ACM Press, New York.

[CM92] M. Chein and M.L. Mugnier. Conceptual graphs: fundamental notions. *Revue d'Intelligence Artificielle*, 6(4):365–406, 1992.

[CM93] M. Chein and M. L. Mugnier. Specialization: Where do the difficulties occur? In Pfeiffer and Nagle [PN93], pages 229–238.

[CM95] M. Chein and M.L. Mugnier. Conceptual graphs are also graphs. Research report 95-004, LIRMM, January 1995.

[CM97] Y. Chiaramella and M. Mechkour. Indexing an image test collection. Technical report, FERMI BRA 8134, April 1997.

[Con87] J. Conklin. Hypertext: An introduction and survey. *IEEE Computer*, 20(9):17–81, 1987.

[Con94] B. Conductier. *Recherche Conceptuelle d'Informations dans des Banques de Données en Ligne: Application au Projet INDUSCOPE.* PhD thesis, Université d'Aix-Marseille III, 1994.

[Coo71] W.S. Cooper. A definition of relevance for information retrieval. *Information Storage and Retrieval*, 7:19–37, 1971.

[Coo94] W.S. Cooper. The formalism of probability theory in IR: A foundation for an encumbrance? In W.B. Croft and C.J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, PA*, pages 242–247. Springer-Verlag, July 1994.

[CR94] V. Christophides and A. Rizk. Querying structured documents with hypertext links using oodms. In *ECHT '94*, September 1994.

[Cra86] T.C. Craven. *String Indexing.* Academic Press, Inc, 1986.

[Cro85] W.B. Croft. An expert assistant for a document retrieval system. In *Proceedings of RIAO'85, Grenoble France*, Mars 1985.

[Cro86] W.B. Croft. User-specified domain knowledge for document retrieval. In *Proceedings the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy*, pages 201–206, September 1986.

[Cro92] W.B. Croft. Approches to intelligent information retrieval. *Information Processing & Management*, 23(4):249–254, 1992.

[Cuz92] R. Cuzin. Visualisation multimédia. application aux dossiers médicaux. Mémoire d'ingénieur CNAM, LGI-IMAG, Octobre 1992.

[CvR95a] F. Crestani and C.J. van Rijsbergen. Information retrieval by logical imaging. *The Journal of Documentation*, 51(1):1–15, March 1995.

[CvR95b] F. Crestani and C.J. van Rijsbergen. Probability kinematics in information retrieval: a case study. In E.A. Fox, W.B. Croft, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seatle*, pages 291–299. ACM, ACM Press, July 1995.

[CY92] C. Crouch and B. Yang. Experiments in automatic statistical thesaurus construction. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 77–88, 1992.

[DA82] C. Delobel and M. Adiba. *Bases de données et systèmes relationnels.* Informatique. DUNOD, BORDAS, Paris, 1982.

[dB85] L. de Brabandere. *Les infoducs.* Editions Duculot, Gembloux, 1985.

[DBS90] P. Devanbu, R.J. Brachman, and P.G. Selfridge. LaSSIE: a classification based software information system. In *Proceedings of the International Conference on Spftware Engineering*, pages 349–261. IEEE, 1990.

[Def86] B. Defude. *Etude et Réalisation d'un Système Intelligent de Recherche d'Informations: Le prototype IOTA.* PhD thesis, Institut National Polytechnique de Grenoble, Juillet 1986.

[DEGV97] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. In *Proceedings 12th Annual IEEE Conference on Computational Complexity (CCC'97)*, Ulm, Germany, June 24-27, 1997, 1997.

[Den97] N. Denos. *Modéliser la pertinence en recherche d'information.* PhD thesis, Université Joseph Fourier, October 1997.

[Des87] J.P. Desclés. Réseaux sémantiques: La nature logique et linguistique des relateurs. *Langages*, 87:55–78, 1987.

[Des90] J.P. Desclés. *Langages applicatifs, langues naturelles et cognition.* Langue, Raisonnement, Calcul. Éditions Hermès, 1990.

[Dev90] K. Devlin. *Infons and types in an information-based logic.* CSLI Lectures Notes. CSLI, Stanford, 1990.

[Dev91] K. Devlin. *Logic and Information.* Cambridge University Press, 1991.

[Dre81] F.I. Dretske. *Knowledge and the Flow of Information.* Basic Blackwell Publisher, 1981.

[Dun91] M. Dunlop. *Multimedia Information Retrieval.* PhD thesis, University of Glasgow, 1991.

[EEM96] P.W. Eklund, G. Ellis, and G. Mann, editors. *Fourth International Conference on Conceptual Structures, ICCS'96*, volume 1115 of *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science*, Sydney, Australia, August 1996. Springer-Verlag.

[EL94] G. Ellis and F. Lehmann. Exploiting the induced order on type-labeled graphs for fast knowledge retrieval. In *Proceedings of the 2nd International Conference on Conceptual Structures*, volume 835 of *Lecture Notes in Artificial Intelligence*, pages 293–310, 1994.

[Ell93] G. Ellis. Efficient retrieval from hierarchies of objects using lattice operations. In *Proceedings of the 1st International Conference on Conceptual Structures*, volume 699 of *Lecture Notes in Artificial Intelligence*, pages 274–293, 1993.

[ELRS95] G. Ellis, R. Levinson, W. Rich, and J.F. Sowa, editors. *Third International Conference on Conceptual Structures, ICCS'95*, volume 954 of *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science*, Santa Cruz CA, USA, august 1995. Springer-Verlag.

[Eps90] R.L. Epstein. *The Semantic Foundations of Logic. Volume 1: Propositional Logics.* Nijhoff International Philosophy Series. Kluwer academemic publishers, 1990.

[Eps94] R.L. Epstein. *The Semantic Foundations of Logic. Predicate Logic.* Oxford University Press, 1994.

[Esc93] John W. Esch. The scope of coreference in conceptual graphs. In Pfeiffer and Nagle [PN93], pages 3–12.

[Far80a] J. Farradane. Relational indexing Part I. *Journal of Information Science*, 1(5):267–276, 1980.

[Far80b] J. Farradane. Relational indexing part ii. *Journal of Information Science*, 1(6):313–324, 1980.

[Far89] J. Fargues. CG information retrieval using linear resolution, generalization and splitting. In *Proceedings of the 4th annual workshop on conceptual structures*, 1989.

[Fer94] Fermi, Formalisation and Experimentation on the Retrieval of Multimedia Information. Technical annex, FERMI BRA 8134, 1994.

[Fil88] C. Fillmore. *The case for case*, pages 1–90. Bach & Harms, 1988.

[Fuh92] N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255, 1992.

[Fuh95] N. Fuhr. Probabilistic datalog - a logic for powerful retrieval methods. In E.A. Fox, W.B. Croft, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seatle*, pages 282–290. ACM, ACM Press, July 1995.

[Gab85] D.M. Gabbay. Theoretical foundations for non-monotonic reasoning in expert systems. In K.R. Apt, editor, *Logics and Models of Concurrent Systems*, pages 439–457. Springer-Verlag, 1985.

[Gab93] D.M. Gabbay. *What is A Logical System?* Oxford University Press, 1993.

[Gab94] D.M. Gabbay. LDS- Labelled Deductive Systems. Volume 1-foundations. Book, MPI-I-94-223, Max-Planck-Institut Für Informatik, Germany, May 1994.

[Gai93] B.R. Gaines. Representation, discourse, logic and truth : situating knowledge technology. In G. Mineau, W. Moulin, and J.F. Sowa, editors, *Proceedings of the third International Conference on Conceptual Structures, ICCS'93*, volume 699 of *Lecture Notes in Artificial Intelligence*, pages 36–63, Quebec City, Canada, August 1993. Springer-Verlag.

[Gau89] C.L. Jimenez Gaurin. *Opération d'accès par le contenu à une base de documents textuels. Application à un environnement de bureau.* PhD thesis, Juillet 1989.

[GB95] R. Green and C.A. Bean. Topical relevance relationships. *JASIS*, 46(9):646–662, 1995.

[GBS+94] C.A. Goble, S.K. Bechhofer, W.D. Solomon, A.L. Rector, W.A. Nowlan, and A.J. Glowinski. Conceptual, semantic and information models for medicine. In *Proceedings of the 4th European-Japanese Seminar on Information Modelling and Knowledge Bases, Stockholm*, May-June 1994.

[GFS94] W.I. Grosky, F. Fotouhi, and I.K. Sethi. Using metadata for the intelligent browsing of structured media objects. *SIGMOD RECORD*, 23(4):49–56, December 1994.

[GG90] P. Gochet and P. Gribomont. *Logique-méthodes pour l'informatique fondamentale*, volume 1. Editions Hermès, 1990.

[Gin87] M.L. Ginsberg. *AI and Nonmonotonic Reasoning.* Morgan-Kaufmann Publishers, 1987.

[GMN84] H. Gallaire, J. Minker, and J.M. Nicolas. Logic and database: A deductive approach. *Computing Surveys*, 16(2):153–185, June 1984.

[GR95] V.N. Gudivada and V.V. Raghavan. Content based image retrieval systems. *IEEE Computer*, pages 18–22, December 1995.

[Gre92] G. Grefenstette. Use of syntactic context to produce term association lists. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and development in Information Retrieval*, pages 89–97, 1992.

[Gue82] M. Le Guern. Les descripteurs dans un système documentaire. In *Colloque traitement automatique des langues naturelles et systèmes documentaires, Clermont-Ferrant*, Mai 1982.

[Gui96a] O. Guinaldo. Conceptual graphs isomorphism: Algorithm and use. In P.W. Eklund, G. Ellis, and G. Mann, editors, *Proceedings of the 4th International Conference on Conceptual Structures, ICCS'96*, volume 1115 of *Lecture Notes in Artificial Intelligence*, pages 160–174, Sydney, August 1996. Springer-Verlag, Berlin.

[Gui96b] O. Guinaldo. *Etude d'un gestionnaire d'ensembles de graphes conceptuels.* PhD thesis, Décembre 1996.

[Haw88] S. Hawking. *A brief History of Time, From the Big Bang to Black Holes.* Bantam Books, London, 1988.

[HB94] T.W.C. Huibers and P.D. Bruza. Situations, a general framework for studying information retrieval. *Information Retrieval: New systems and current research*, 2, 1994.

[Hir91] G. Hirst. *Semantics*, pages 1024–1029. John Wiley & Sons, 1991.

[HKNP94] J. Heinsohn, D. Kudenko, B. Nebel, and H.J. Profitlich. An empirical analysis of terminological representation systems. *Artificial Intelligence*, 68:367–397, 1994.

[HLMS95] R. Hjelsvold, S. Lang, R. Midtstraum, and O. Sandsta. Integrated video archive tools. In *ACM Multimedia '95*, San Fransisco, CA USA, 1995. ACM.

[HLvR96] T.W.C. Huibers, M. Lalmas, and C.J. van Rijsbergen. Information retrieval and situation theory. *SIGIR Forum*, 30(1):11–25, 1996.

[HM94] R. Hjelsvold and R. Midtstraum. Modelling and querying video data. In *20th VLDB Conference*, pages 686–694, Santiago, Chile, 1994.

[HM96] K.A. Han and S.H. Myaeng. Image organisation and retrieval with automatically constructed feature vectors. In *Proceedings of the 19 th ACM SIGIR Conference*, Zurich, Switzerland, 1996.

[HOC95] T.W.C. Huibers, I. Ounis, and J.P. Chevallet. Axiomatization of a conceptual graph formalism for information retrieval in a situated framework. Technical Report RAP95-004, Groupe MRIM of the Laboratoire de Génie Informatique in Grenoble, France, July 1995.

[HOC96] T.W.C. Huibers, I. Ounis, and J.P. Chevallet. Conceptual graphs aboutness. In P.W. Eklund, G. Ellis, and G. Mann, editors, *Proceedings of the 4th International Conference on Conceptual Structures, ICCS'96*, volume 1115 of *Lecture Notes in Artificial Intelligence*, pages 130–144, Sydney, August 1996. Springer-Verlag, Berlin.

[Hor97a] I. Horrocks. A comparaison of two terminological knowledge representation systems. Master thesis of computer science, Faculty of Science, University of Manchester, 1997.

[Hor97b] I. Horrocks. Optimisation techniques for expressive description logics. Technical Report Series, UMCS-97-2-1, Department of Computer Science, University of Manchester, 1997.

[HS70] J. Hintikka and P. Suppes. *Information and Inference.* Reidel Dodrecht, 1970.

[HSD97] V. Harmandas, M. Sanderson, and M.D. Dunlop. Image retrieval by hypertext links. In *Proceedings of ACM SIGIR'97*, pages 296–303, Philadelphia PA, USA, July 1997. ACM.

[Hui96] T.W.C. Huibers. *An Axiomatic Theory for Information Retrieval.* PhD thesis, Department of Computer Science, Utrecht university, The Netherlands, November 1996.

[ILE88] M.A. Iris, B.E. Litowitz, and M. Evens. *Problems of the part-Whole relation*, pages 289–334. Cambridge University Press, 1988.

[Imm86] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[Isr80] D.J. Israel. What's wrong with non-monotonic logic? In *Proceedings of the First National Conference on Artificial Intelligence*, pages 99–101, 1980.

[JL77] N. Jones and W. Laaser. Complete problems in deterministic polynomial time. *Theoretical Computer Science*, 3:105–117, 1977.

[Jon95] K.S. Jones. Reflections on TREC. *Information Processing and Management*, 31(3):291–314, 1995.

[Jou93] C. Jouis. *Contribution à la Conceptualisation et à la Modélisation des Connaissances à partir d'une analyse linguistique de textes. Réalisation d'un prototype: Le système SEEK.* PhD thesis, Université Paris-Sorbonne, 1993.

[Jr.71] J.J. Rocchio Jr. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART retrieval System–Experiments in Automatic Document Processing*, chapter 14, pages 313–323. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

[JV96] A.K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, 1996.

[Ker84] D. Kerkouba. *Une Méthode d'Indexation Automatique des Documents Fondée sur l'Exploitation de leurs Propriétés Structurelles. Application à un corpus technique.* PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France, 1984.

[Khe95] A. Kheirbek. *Modèle d'intégration d'un système de recherche d'informations et d'un système hypermédia basé sur le formalisme des graphes conceptuels. Application au système RIME.* PhD thesis, Université Joseph Fourier, Grenoble, 1995.

[KLM90] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.

[Kri63] S.A. Kripke. Semantical analysis of modal logic. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–97, 1963.

[Kri71] S.A. Kripke. *Semantical considerations on modal logic*, pages 63–73. Oxford University Press, 1971.

[Lal96] M. Lalmas. *Theories of Information and Uncertainty for the modelling of Information Retrieval: an application of Situation Theory and Dempster-Shafer's Theory of Evidence.* PhD thesis, Department of Computing Science, University of Glasgow, Scotland, April 1996.

[Lal97] M. Lalmas. Dempster-shafer's theory of evidence applied to structured documents: modelling uncertainty. In *Proceedings of ACM SIGIR'97*, pages 110–118, Philadelphia PA, USA, July 1997. ACM.

[Lan86] F. Landman. *Towards a Theory of Information.* Foris, Dordrecht, 1986.

[Lar94] O. Larouk. *Extraction de Connaissances à Partir de Documents Textuels.* PhD thesis, University of Claude Bernard-Lyon, 1994.

[LB84] H.J. Levesque and R.J. Brachman. The tractability of subsumption in framed-based description languages in knowledge representation and reasoning. In *National Conference on Artificial Intelligence of the American Association for Artificial Intelligence (AAAI), Austin, Texas*, pages 34–37, 1984.

[LB85] H.J. Levesque and R.J. Brachman. *A fundamental tradeoff in knowledge representation and reasoning*, pages 42–70. Morgan Kaufmann Publishers, Los Atlos, California, 1985.

[LB87] H.J. Levesque and R.J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3(2):78–93, May 1987.

[LB94] M. Liquière and O. Brissac. A class of conceptual graphs with polynomial iso-projection. In *Supplement Proceedings of the 2nd International Conference on Conceptual Structures*, pages 93–102. College Park, 1994.

[Len95] D.B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11), 1995.

[Les89] H. Lesca. *Information et adaptation de l'entreprise, mieux gérer l'information pour une entreprise plus performante.* Masson, 1989.

[Lev84] H.J. Levesque. A logic of implicit and explicit belief. In *Proceedings of AAAI-84, 4th Conference of the American Association for Artificial Intelligence*, pages 198–202, Austin, TX, 1984.

[LKT+95] B. Lutes, S. Kutschekmanesh, U. Thiel, C. Berrut, Y. Chiaramella, F. Fourel, H. Haddad, and P. Mulhem. Study on non-textbased information retrieval - state of the art -. Final report, ELPUB 106, 1995.

[LR95] A.Y. Levy and M.C. Rousset. CARIN: A representation language combining horn rules and description logics. In A. Borgida, M. Lanzerini, D. Nardi, and B. Nebel, editors, *International Workshop on Description Logics*, pages 44–51, Rome, Italy, 1995.

[LvR92] M. Lalmas and C.J. van Rijsbergen. A logical model of information retrieval based on situation theory. In British Computer Society, editor, *Proceedings of the BCS 14th Information Retrieval Colloquium*, pages 1–13. Springer-Verlag, April 1992.

[LvR93] M. Lalmas and C.J. van Rijsbergen. A model of an information retrieval system based on situation theory and dempster-shafer theory of evidence. In *Incompleteness and Uncertainty in Information Systems*, pages 102–116. Concordia University, 1993.

[MA96] P. Martin and L. Alpay. Conceptual structures and structured documents. In P.W. Eklund, G. Ellis, and G. Mann, editors, *Proceedings of the 4th International Conference on Conceptual Structures, ICCS'96*, volume 1115 of *Lecture Notes in Artificial Intelligence*, pages 145–159, Sydney, August 1996. Springer-Verlag, Berlin.

[Mac90] I.A. Macleod. Storage and retrieval of structured documents. *Information Processing & Management*, 26(2):197–208, 1990.

[Mac91] R. MacGregor. *The evolving technology of classification-based knowledge representation systems*, pages 385–400. In [Sow91], 1991.

[MAD92] TAÏGA: Traitement et Analyse d'Informations Géopolitique et d'Actualités. Plaquette de présentation, Paris, 1992.

[Mal91] K. Malmkjaer. *The linguistics Encyclopedia.* Routeledge, 1991.

[Mar67] M.E. Maron. *Relational Data File 1 : Design Philosophy*, pages 211–233. Academic Press, London, 1967.

[Mar77] M.E. Maron. On indexing, retrieval and the meaning of about. *Journal of the American Society for Information Science*, 28:38–43, 1977.

[MC93a] M. L. Mugnier and M. Chein. Polynomial algorithms for projection and matching. In Pfeiffer and Nagle [PN93], pages 239–251.

[MC93b] M.L. Mugnier and M. Chein. Characterization and algorithmic recognition of canonical conceptual graphs. In Mineau et al. [MMS93], pages 294–311.

[MC96] M.L. Mugnier and M. Chein. Représenter des connaissances et raisonner avec des graphes. *Revue d'Intelligence Artificielle*, 10(1), 1996.

[McG82] J.J. McGregor. Backtrack search algorithms and the maximal common subgraph problem. *Software-Practice and Experience*, 12:23–34, 1982.

[Mec95a] M. Mechkour. EMIR2. An extended model for image representation and retrieval. In *DEXA'95. Database and EXpert system Applications, London.*, pages 395–404, September 1995.

[Mec95b] M. Mechkour. EMIR2: an extended model for image representation and retrieval. Technical report, Basic Research Action FERMI, n. 8134, 1995.

[Mec95c] M. Mechkour. *EMIR2. Un Modèle étendu de représentation et de correspondance d'images pour la recherche d'informations. Application à un corpus d'image historiques.* PhD thesis, Université Joseph Fourier, Grenoble, 1995.

[Mec96] M. Mechkour. CGM. A Conceptual Graphics Model for Information Retrieval. Chapter of deliverable D4, FERMI BRA 8134, 1996.

[Meg95] C. Meghini. A model for image bases and its query facility. In *ACM SIGIR*, Seattle, USA, 1995.

[Men64] E. Mendelson. *Introduction to Mathematical Logic.* International Student Editions. Van Nostrand Reinhold Company, New York, 1964.

[MEN94] Consortium MENELAS. Menelas: accès à des comptes-rendus d'hospitalisation en langage naturel. In *Actes des 5èmes journées francophones d'Informatique Médicale*, 1994.

[Mil96] G.A. Miller. WORDNET: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1996.

[MKL94] S.H. Myaeng, C. Khoo, and M. Li. Linguistic processing of text for a large-scale conceptual information retrieval system. In Tepfenhart et al. [TDS94], pages 69–82.

[ML93] S.H. Myaeng and E. Liddy. Information retrieval with semantic representation of texts. In *Proceedings of Symposium on Document Analysis and Information Retrieval*, Las Vegas, USA, April 1993.

[MLL91] S.H. Myaeng and A. Lopez-Lopez. A flexible matching algorithm for matching conceptual graphs. In *Proceedings the 6th Annual Workshop on Conceptual Graphs*, pages 135–151, 1991.

[MMFB97] M. Mechkour, P. Mulhem, F. Fourel, and C. Berrut. Prime-gc. a medical information retrieval prototype on the web. In *RIDE'97. Seventh International Workshop on Research Issues in Data Engineering: High performance Database Management for Large Scale Applications., Birmingham, England*, April 7–8 1997.

[MMS93] G.W. Mineau, B. Moulin, and J.F. Sowa, editors. *First International Conference on Conceptual Structures, ICCS'93*, volume 699 of *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science*, Montreal, Canada, August 1993. Springer-Verlag.

[MNE92] J.A. Markowitz, J.T. Nutter, and M.W. Evens. *Beyond is-a and part-whole: More semantic networks links*, pages 377–390. Pergamon Press, 1992.

[Mon74] R. Montague. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, 1974.

[MP96] T.P. Minka and R.W. Picard. interactive learning using a "society of models". Technical Report 349, M.I.T Media Laboratory Perceptual Computing, 1996.

[MS96] C. Meghini and U. Straccia. Extending a description logic to cope with the completeness of multimedia documents. In W. Wahlster, editor, *Proceedings of the 12th European Conference on Artificial Intelligence, ECAI 96*. John Wiley & Sons, Ltd, 1996.

[MSST93] C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In R. Korfhage, E. Rassmussen, and P. Willet, editors, *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, PA*, pages 298–307. ACM, ACM Press, June 1993.

[MT94] A. Müller and U. Thiel. Query expansion in an abductive information retrieval system. In *Proceedings of RIAO'94, 4th International Conference on Intelligent Multimedia Information Retrieval Systems and Management*, pages 461–480, 1994.

[Mug93] M.L. Mugnier. *Contributions Algorithmiques pour les Graphes d'Héritage et les Graphes Conceptuels*. PhD thesis, Université Montpellier II, October 1993.

[Mug95] M.L. Mugnier. On generalization / specialization for conceptuals graphs. *Journal of Experimental and Theoretical Artificial Intelligence*, 7:325–344, 1995.

[Mul93] P. Mulhem. *La correspondance d'objets dans la recherche d'informations: une expérimentation sur un SGBD OO.* PhD thesis, Université Joseph Fourier, Grenoble, 1993.

[Mya92] S.H. Myaeng. Conceptual graphs as a framework for text retrieval. In T.E. Nagle, J.A. Nagle, L.L. Gerholz, and P.W. Eklund, editors, *Conceptual Structures: current Research and Practice*, England, Ellis Horwood Workshops, 1992.

[Nar95] A.D. Narasimhalu. Special section on content-based retrieval- introduction. *Multimedia Systems*, 3(1):1–2, February 1995.

[NBY96] G. Navarro and R. Baeza-Yates. Integrating contents and structure in text retrieval. *ACM SIGMOD Record*, March 1996.

[Neb88] B. Nebel. Computational complexity of terminological reasoning in BACK. *Artificial Intelligence*, 34(3):371–383, April 1988.

[Neb90a] B. Nebel. Reasoning and revision in hybrid representation systems. volume 422 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1990.

[Neb90b] B. Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43(2):235–249, May 1990.

[NECH92] D. Nanci, B. Espinasse, B. Cohen, and H. Heckenroth. *Ingénierie des systèmes d'information avec Merise*. Performance. Sybex, ISBN: 2-7361-0747-7, 1992.

[Nie88] J. Nie. An outline of a general model for information retrieval systems. In *Proceedings of SIGIR 88, Grenoble*, pages 495–506. ACM, 1988.

[Nie89] J. Nie. An information retrieval model based on modal logic. *Information Processing & Management*, 25(5):477–491, 1989.

[Nie90] J. Nie. *Un Modèle Logique général pour les systèmes de recherche d'informations. Application au prototype RIME.* PhD thesis, Université Joseph Fourier, Grenoble, 1990.

[NSHP88] H. Nakashima, H. Suzuki, P.K. Halvorsen, and S. Peters. Towards a computational interpretation of situation theory. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 489–498. Institute for New Generation Computer Technology, 1988.

[Oa91] Deux O. and al. The o2 system. *Communications of the ACM*, 34(10):34–48, 1991.

[OC96a] I. Ounis and J.P. Chevallet. Applying MIRLOG to a model of multimedia documents. Chapter of deliverable D8, FERMI BRA 8134, 1996.

[OC96b] I. Ounis and J.P. Chevallet. Using conceptual graphs in a multifaceted logical model for information retrieval. In R.R. Wagner and H. Thoma, editors, *Proceedings of the 7th International Conference on Database and EXpert Systems Applications, DEXA'96*, volume 1134 of *Lecture Notes in Computer Science*, pages 812–823, Zurich, Switzerland, September 1996. Springer.

[OH97] I. Ounis and T.W.C. Huibers. A logical relational approach for information retrieval indexing. In *19th Annual BCS-IRSG Colloquium on IR Research, Aberdeen, Scotland*. EWIC, Springer-Verlag, 8–9 April 1997.

[OP97a] I. Ounis and M. Pasca. An extended inverted file approach for information retrieval. In *International Database Engineering and Application Symposium (IDEAS'97), Montreal, Canada*, pages 397–402. IEEE Computer Society Press, August 1997.

[OP97b] I. Ounis and M. Pasca. The RELIEF retrieval system. In *The IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'97), Newport Beach, California, U.S.A.* IEEE Computer Society Press, November 1997.

[Oun94] I. Ounis. Logique terminologique pour la correspondance entre graphes de concepts dans le cadre d'un système de recherche d'informations. Rapport de dea, ENSIMAG, Institut National Polytechnique de Grenoble, France, 1994.

[Oun95] I. Ounis. Une dénotation pour les graphes conceptuels:comparaison avec les logiques terminologiques en recherche d'informations. In *Congrès INFORSID 95, Grenoble, France*, pages 147–164, May - June 1995.

[Oun96] I. Ounis. Étude des modèles logiques pour la recherche d'information. Rapport MRIM-SUR96-002, CLIPS-IMAG, Octobre 1996.

[Pal90] P. Palmer. *Etude d'un analyseur de surface de la langue naturelle: application à l'indexation automatique de textes*. PhD thesis, Université Joseph Fourier, Grenoble, 1990.

[Par95] F. Paradis. Modeling textual information. Technical report, Basic Research Action FERMI, n. 8134, 1995.

[Par96] F. Paradis. *Un modèle d'indexation pour les documents textuels structurés*. PhD thesis, Université Joseph Fourier, Novembre 1996.

[PC95] J. Poole and J.A. Campbell. A novel algorithm for matching conceptual and related graphs. In Ellis et al. [ELRS95], pages 293–307.

[Pea88] J. Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufman, San Mateo, 1988.

[Pei58] C.S. Peirce. *Collected Papers of C.S. Peirce*. 8 volumes. Harvard University Press, 1931-1958.

[PL94] L. Padgham and P. Lambrix. A framework for part-of hierarchies in terminological logics. In *Principals of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference (KR'94), Bonn, Germany*, pages 485–496. Morgan Kaufmann, 1994.

[PN93] Heather D. Pfeiffer and Timothy E. Nagle, editors. *Conceptual Structures: Theory and Implementation*, volume 754 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1993. Proceedings of the Seventh Annual Workshop on Conceptual Graphs, Las Cruces, USA, July 1992.

[PS84] P.F. Patel-Schneider. Small can be beautiful in knowledge representation. In *Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems*, pages 11–16, Denver, USA, CO, 1984.

[PS89] P.F. Patel-Schneider. A four-valued semantics for terminological logics. *Artificial Intelligence*, 38:319–351, 1989.

[PSOKK+90] P.F. Patel-Schneider, B. Owsnicki-Klewe, A. Kobsa, N. Guarino, R. McGregor, W.S. Mark, D.L. McGuiness, B. Nebel, A. Schmiedel, and J. Yen. Term subsumption languages in knowledge representation. *AI Magazine*, 11:16–23, 1990.

[PW91] H.J. Peat and P. Willett. The limitation of term co-occurence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science (JASIS)*, 42(5):378–383, 1991.

[RBE+94] J. Rumbaugh, M. Blaha, F. Eddy, W. Premerlan, and W. Lorensen. *OMT : modelisation et conception orientées objet*. Prentice Hall, 1994, 1994.

[RDBFPD96] M. Ribière, R. Dieng, M. Blay-Fornarino, and A.M. Pinna-Dery. Link-based reasoning on conceptual graphs. In P.W. Eklund, G.E., and G. Mann, editors, *Conceptual Structures: Knowledge Representation as Interlingua — Auxiliary Proceedings of the Fourth International Conference on Conceptual Structures*, pages 146–160, Sydney, Australia, 1996. University of New South Wales.

[Rei80] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[Riv89] F. Rivenc. *Introduction à la logique*. Petite Bibliothèque Payot. Éditions Payot, Paris, 1989.

[Rob73] D.D. Roberts. *The Existential Graphs of Charles S. Peirce*. Mouton, 1973.

[Rob77] S.E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33:294–304, 1977.

[Sab88] G. Sabah. *L'intelligence artificielle et le langage (volume 1)*. Représentation des connaissances. Hermès, Paris, 1988.

[Sal71] G. Salton. *The SMART project*. Prentice Hall, 1971.

[Sal89] G. Salton. *Automatic Text Processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Publishing Company, 1989.

[Sat95] U. Sattler. A concept language for engineering applications with part-whole relations. In A. Borgida, M. Lanzerini, D. Nardi, and B. Nebel, editors, *Proceedings of the International Workshop on Description Logics- DL'95*, pages 119–123, Rome, Italy, 1995.

[Sat96] U. Sattler. A concept language extended with different kinds of transitive roles. In G. Gorz and S. Holldobler, editors, *Deutsche Jahrestagung für Künstliche Intelligenz*, Lecture Notes in Artificial Intelligence number 1137. Springer Verlag, 1996.

[SB71] R. Simmons and B. Bruce. Some relations between predicate calculus and semantic net representation of discourse. In *3me Actes IJCAI*, pages 524–529, 1971.

[SB91] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[Sch72] R.C. Schank. Conceptual dependency: a theory of natural language understanding. *Cognitive Psychology*, 3:552–631, 1972.

[Sch75] R.C. Schank. *Conceptual Information Processing.* North-Holland, 1975.

[Sch89] J.G. Schmolze. Terminological knowledge representation systems supporting n-ary terms. In *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning, KR-89*, pages 432–443, Toronto, Ontario, 1989.

[Sch91] K.L. Schubert. *Semantic nets are in the eye of the beholder*, pages 95–107. In Sowa [Sow91], 1991.

[Sch92] M. Schröder. Knowledge-based analysis of radiological reports using conceptual graphs. In *Proceedings of the 7th annual workshop on Conceptual Graphs*, pages 213–222, 1992.

[Sch94] M. Schröder. Interpreting clinical questions- medical text analysis supports image presentation. Technical report, Philips Research Laboratories, Dept. Technical Systems, Hambourg, 1994.

[SDV95] P. Saint-Dizier and E. Viegas. *Computational Lexical Semantics.* Cambridge University Press, 1995.

[Seb94] F. Sebastiani. A probabilistic terminological logic for modelling information retrieval. In W.B. Croft and C.J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, PA*, pages 122–130. ACM, Springer-Verlag, July 1994.

[SG92] P. Schauble and U. Glavitsh. A system for retrieving speech documents. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and development in Information Retrieval*, 1992.

[SM83] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval.* Mcgraw Hill Book Company, New York, 1983.

[SM96] E. Salvat and M.L. Mugnier. Sound and complete forward and backward chainings of graph rules. In Eklund et al. [EEM96], pages 248–262.

[SOKK+90] P.F. Schneider, B. Owsnicki-Klew, A. Kobsa, N. Guarino, R. McGregor, D. McGuiness, B. Nebel, A. Schmiedel, and J. Yen. Term subsumption languages in knowledge representation. *The AI magazine*, 11(2):16–23, 1990.

[Sow84] J.F. Sowa. *Conceptual Structures : Information Processing in Mind and Machine.* Addison-Wesley Publishing Company, 1984.

[Sow91] J.F. Sowa. *Principles of Semantic Networks: Explorations in the Representation of Knowledge.* Morgan Kaufmann, Boston, Massachusetts, 1991.

[Sow93] J.F. Sowa. Relating diagrams to logic. In Mineau et al. [MMS93], pages 1–35.

[Sow97] J.F. Sowa. *Knowledge Representation, Logical, Philosophical, and Computational Foundations.* Series of textbooks on Artificial Intelligence. PWS publishing Company, Boston, Massachusetts, 1997. Draft Version Distributed in ICCS'96.

[SS96] U. Straccia and F. Sebastiani. The fermi logic. Part I of the deliverable D2 : A logic for information retrieval, FERMI BRA 8134, 1996.

[SSS91] M. Schmidt-Schauβ and G. Smolka. Attributives concept descriptions with complements. *Artificial Intelligence*, 48:1–29, 1991.

[ST81] M.N.S Swamy and K. Thulasiraman. *Graphs, Networks and Algorithms.* Library of Congress Cataloging in Publication Data. John Wiley & Sons, Inc., 1981.

[SvR93] T.M.T. Sembok and C.J. van Rijsbergen. Imaging: a relevance feedback retrieval with nearest neighbour clusters. In *Proceedings of the BCS Colloquium in Information Retrieval, Glasgow*, pages 91–107, March 1993.

[SW64] C.E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* Illini Book, 1964.

[Swa94] M. Swan. *Practical English Usage.* Oxford University Press, Honk Kong, 1994.

[TA93] E. Tin and V. Akman. BABY-SIT: A computational medium based on situations. In *Proceedings of the 9th Amsterdam Colloquium, Volume Part III*, pages 655–681. University of Amsterdam, Amsterdam, 1993.

[TAA89] R.M. Tong, V.N. Askman, and L.A. Appelbaum. A knowledge representation for conceptual information retrieval. *IJIS*, 4:259–283, 1989.

[TAAC87] R.M. Tong, L.A. Appelbaum, V.N. Askman, and J.F. Cunningham. Conceptual information retreival using RUBRIC. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA*, pages 247–253. ACM, ACM Press, June 1987.

[TDS94] W.M. Tepfenhart, J.P. Dick, and J.F. Sowa, editors. *Conceptual Structures: Current Practices. Proceedings of the Second International Conference on Conceptual Structures, ICCS'94*, volume 835 of *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science*, College Park, USA, August 1994. Springer-Verlag.

[Tec96] O2 Technology. The o2 user manual. Technical report, 1996.

[TV94] E. Tin and V.Akman. Computational situation theory. *ACM SIGART Bulletin*, 5(4):4–17, October-November 1994.

[TV95] E. Tin and V.Akman. Towards situation-oriented programming languages. *ACM SIGPLAN Notices*, 30(5):27–36, January 1995.

[Var82] M. Vardi. Complexity of relational query languages. In *Proceedings of the 14 th STOC Conference"*, pages 137–146, San Francisco, 1982.

[vdB93] H. van den Berg. *Knowledge Graphs and Logic.* PhD thesis, September 1993.

[vdG90] L.C. van der Gaag. *Probability-Based Models for Plausible Reasoning.* PhD thesis, 1990.

[VLS87] B. Peltason V. Luck, K. Nebel and C. Schmiedel. The anatomy of the BACK system. Kit report #41, Berlin, 1987.

[vR79] C.J. van Rijsbergen. *Information Retrieval.* Butterworths, London, 1979.

[vR86] C. J. van Rijsbergen. A new theoretical framework for information retrieval. In *ACM Conference on Research and development in Information Retrieval, Pisa*, pages 194–200, 1986.

[vRL96] C.J. van Rijsbergen and M. Lalmas. An information calculus for information retrieval. *Journal of the American Society of Information Science*, 47(5):385–398, 1996.

[WAL+94] J.K. Wu, Y.H. Ang, P. Lam, H.H. Loh, and A.D. Narasimhalu. Inference and retrieval of facial images. *Multimedia systems*, 2:1–14, 1994.

[WBHvdW96] B.C.M. Wondergem, P. Van Bommel, T.W.C. Huibers, and T. van der Weide. How is this document's relevancy derived? - personalizing inferences in preferential models. Technical Report CSI-R9710, Computing Science Institute, University of Nijmegen, Nijmegen, The Netherlands", 1996.

[Wer95] M. Wermelinger. Conceptual graphs and first-order logic. In Ellis et al. [ELRS95], pages 323–337.

[Wil73] Y. Wilks. An artificial intelligence approach to machine translation. In Schank & Colby, editor, *Computer models of thought and language*, pages 114–151. Freeman, 1973.

[Wil75] Y. Wilks. Preference semantics. In Keenan, editor, *Formal semantics of natural language*, pages 329–348. Cambridge university press, 1975.

[Wil87] Y. Wilks. Primitives. In S.C. Shapiro, editor, *Encyclopedia of Artificial Intelligence, Part II*, pages 759–761, 1987.

[Wil95] M. Willems. Projection and unification for conceptual graphs. In Ellis et al. [ELRS95], pages 278–292.

[WL94] M. Wermelinger and J.G. Lopes. Basic conceptual structures theory. In William M. Tepfenhart, Judith P. Dick, and John F. Sowa, editors, *Conceptual Structures: Current Practices. Proceedings of the Second International Conference on Conceptual Structures*, volume 835 of *Lecture Notes in Artificial Intelligence*, pages 144–159, College Park, Maryland, USA, 16–20 August 1994. Springer-Verlag.

[WM93] V. Wuwongse and M. Manzano. Fuzzy conceptual graphs. In *In ICCS' 93 International Conference on Conceptual Structures, Quebec City, Canada*, pages 430–449, August 1993.

[Won96] B.C.M. Wondergem. Preferential structures for information retrieval. Master's Thesis INF-SCR-96-21, Department of Computer Science, Utrecht University, Utrecht, The Netherlands, 1996.

[Woo75] W.A. Woods. *What's in a link: foundations for semantic networks.* Academic Press, New York, 1975.

[Woo91] W.A. Woods. *Understanding subsumption and taxonomy: a framework for progress*, pages 22–27. In [Sow91], 1991.

[Za93] P. Zweigenbaum and al. Linguistic and medical knowledge bases: An access system for medical records using natural langage. Technical report, MENELAS: deliverable 9, AIM Project A2023, 1993.

Index thématique

S_Σ, 170
Σ_R, 104
$\perp_r$, 107
χ, 82
$\cong$, 185
$\cong_E$, 186
$\models\!\equiv$, 84, 87
$\equiv_r$, 105
$\leq_\sigma$, 166
$\mathcal{G}$, 187
$\mathcal{S}_\Re$, 114
$\mathcal{S}_\Re^{Ext}$, 117
$\mathcal{T}_{Cspe}$, 172
$\mathcal{T}_{Rspe}$, 172
$\models$, 75
$\odot$, 77, 83
$\rightarrow$, 76
$\rightarrow_r^\oplus$, 106
$\rightarrow_r^f$, 106
$\prec_R$, 85
$\prec_R$, 87
$\prec_R^{Ext}$, 120
$\Box\!\!\rightsquigarrow$, 83
$\Box\!\!\rightsquigarrow_R$, 108
::, 168
$\mathcal{E}$, 175
$\mathcal{M}$, 167
Elen, 7, 30, 150–152, 154–161
$\mathcal{K}_s$, 172
$\mathcal{S}_r$, 170
$\mathcal{S}_\Sigma$, 170
$\mathcal{T}_c$, 163
$\mathcal{T}_r$, 164
Corel, 29
I3R, 32
Lassie, 31
MMIR, 46
Pif, 52
Relief, 33, 191, 205, 257
Rime, 7, 30, 39
Scisor, 31
Taïga, 32
EMIR^2, 3, 48
HyperRIME, 46
IOTA, 3
KALIPSOS, 150
KL-ONE, 38
MENELAS, 31, 150
MIRLOG, 53
MIRTL, 3, 30
PGGC, 260
ROGER, 257, 262
RUBRIC-TOPIC, 32
SMART, 257
Conf, 168

Égalité de Concepts, 183
Égalité de Graphes, 184
Équivalence, 105
Équivalence de Relons, 105
Équivalence de Situations, 153
Équivalence Logique à Gauche, 109

à-propos, 83

Affaiblissement à Droite, 109
Agencements de Relations, 92
Alias, 95

Bouclage, 112

CBIR, 135
chemins de transformation, 118
Complétude, 68
Complétude faible, 68
Complétude stricte, 68
Composition, 94
Concept bien formé, 182
Concepts Communs, 184

Connaissance Incertaine, 119
Conséquence Sémantique, 87
Correspondance et Préférence sur les Index, 85
Correspondance Unique, 100
Cut, 110
CYC, 32

Dénotation de Type, 142
degré de pertinence, 119
degré de plausibilité, 119
Dempster-Shafer, 134
Disjonction, 112
document complexe, 52

E-isomorphisme, 186
Endiguement, 98
Endiguement Élémentaire, 109
Endiguement Équivalent, 105
Endiguement avec Retenue, 106, 110
Endiguement Faible, 106, 110
Exclusivité à Droite, 101
Exclusivité à Gauche, 100
expressions d'index, 20
Extension d'une relation, 34
extraction automatique, 32

facette, 131
fichier inverse, 228
forme normale, 149
formule bien formée, 80

Graphe Conceptuel, 142
Graphe conceptuel, 182
graphe conceptuel bien formé, 183
Graphe Index, 187
Graphe Simple, 143
Graphe Situation, 154
Graphes Équivalents, 147
Graphes isomorphes, 184

Hiérarchie, 24
hypermédia, 45
hypertexte, 45

Indexation relationnelle, 8
infon, 74
infon négatif, 74
infon paramétré, 76
infon positif, 74
Ingrédience, 25
Intension d'une relation, 34
Interprétation des Expressions d'Indexation, 69
Inversion, 95

Juxtaposition, 93

label de règle, 119
Langage d'Indexation Relationnelle, 81
Langage de Dérivation, 108
langage des situations, 82
Localisation, 25

Méta-Propriétés, 101
Modèle d'indexation, 8
Modèle logique, 10
modèle relationnel préférentiel, 87
monde possible, 69
Monotonicité Circonspecte, 111
Monotonicité Prudente, 111
Monotonicité Rationnelle, 112
Monotonicité Restreinte, 111

O_2, 257
objectivisme, 22

Permutation, 92
pertinence logique, 83
pertinence relationnelle, 108
plausiblement pertinent, 119
polarité, 74
Préclusion, 96, 107
primitives sémantiques, 23
Profons, 79
Projection, 144
Projection Relationnelle, 195
Propriétés Mathématiques, 90

Règles par Défaut, 124
Réflexivité, 109
réseaux sémantiques, 35
raisonnement non-monotone, 107
Relation bien formé, 182
Relation d'Endiguement, 76, 105
relation de préférence, 85
Relation de Préférence Étendue, 120
Relation de Support, 75

relon concept, 153
relon marqueur, 153
relon relation, 153
Relons, 79–81
relons, 152
Repérage, 25
Restriction, 98

Séquentialité, 91
Signature d'une Relation, 104
Simultanéité, 96
Situations Identiques, 78
Sous-graphe, 184
Spécialisation, 97
SRI conceptuels, 29
SRI orientés précision des réponses, 2
strictement pertinent, 119
Structure sémantique, 7
Subsomption, 77
Subsomption de Situations, 84
Succession, 25
Système de Dérivation, 108
Système Multifacette, 131

table de composition de relations, 36
terme d'indexation, 14
Théorie des Situations, 72

union de situations, 78
union informationnelle, 77, 83

valeur de pertinence, 119

Index des citations

Adiba, M., 227, 257
Agosti, M., 45
Akman, V., 136
Allen, J.F., 25
Alpay, L., 141
Amati, G., 69, 131, 224, 294

Ballard, D.H., 135
Barwise, J., 72, 76, 293
Bechhofer, S.K., 92
Berard-Dugourd, A., 150
Berg (van den), H., 206
Berrut, C., 7, 74, 259, 280
Black, A.W., 136
Black, W.J., 128
Blaha, M., 258
Blair, D.C., 2, 6
Blanché, R., 225, 227
Boksenbaum, C., 18
Bommel (van), P., 85
Bouaud, J., 141, 150
Brachman, R.J., 5, 9, 36, 38, 140, 206
Brisebois, M., 17
Brissac, O., 204, 230
Brown, E.W., 5
Bruza, P.D., 18, 20, 74, 79, 102, 111, 125, 126, 151, 293

Campbell, J.A., 204, 230
Carbonneill, B., 18, 148, 162, 255, 294
Carnap, R., 73
Chaffin, R., 164
Chang, C.C., 221
Chein, M., 145–148, 162, 165, 201, 204, 227, 230, 231, 288, 294
Chen, H., 17
Chen, P.S., 125, 126
Chevallet, J.P., 3, 6, 7, 53, 62, 65, 67, 74, 78, 80, 89, 101, 131, 141, 146, 147, 150–152, 159, 161, 188, 231, 280, 293
Chiaramella, Y., 6, 52, 62, 65, 99, 102, 293
Cleverdon, C.W., 2
Cohen, B., 18
Collet, C., 257
Conductier, B., 128
Cooper, W.S., 66
Craven, T.C., 20
Crestani, F., 45, 62, 69
Croft, W.B., 3, 22
Crouch, C., 17
Cuzin, R., 264

Désclés, J.P., 26
Dantsin, E., 230
Defude, B., 287
Delobel, C., 227
Denos, N., 66, 293
Devlin, K., 23, 74
Dretske, F.I., 76
Dubucs, J., 225, 227
Dunlop, M., 45

Eddy, F., 258
Eiter, T., 230
Ellis, G., 230
Epstein, R.L., 68, 76, 80
Esch, J., 176
Espinasse, B., 18
Etchemendy, J., 72, 76, 293
Evens, M.W., 164

Fargues, J., 150
Farradane, J., 6, 18, 81
Fillmore, C., 17
Fourel, F., 52, 99, 102, 259, 280, 293
Fuhr, N., 5, 62

Gaag (van der), L.C., 293
Gabbay, D.M., 80, 83, 108, 110, 111, 128, 157
Gaines, B.R., 141

Gallaire, H., 228
Gawron, J.M., 72
Ginsberg, M.L., 114
Glowinski, A.J., 92
Goble, C.A., 92
Gochet, P., 68
Gottlob, G., 230
Grefenstette, G., 17
Griboment, P., 68
Guarino, N., 5, 140
Gudivada, V.N., 135
Guinaldo, O., 148, 165, 184, 200, 204, 230

Haemmerlé, O., 18
Halvorsen, P.K., 136
Heckenroth, H., 18
Heinsohn, J., 95
Herrmann, D.J., 164
Hintikka, J., 73
Hirst, G., 23
Horrocks, I., 92
Huibers, T.W.C., 2, 5, 62, 65, 72, 78–80, 82, 83, 85, 102, 108, 111, 113, 136, 141, 150–152, 159, 231, 292

Immerman, N., 230
Israel, D.J., 114

Jain. A.K., 135
Jimenez Gauri, C.L., 280
Jones, K.S., 2
Jones, N., 230
Jouis, C., 24, 26

Keisler, H.J., 221
Kerkouba, D., 3, 16
Kerpedjiev, S., 69, 131
Kheirbek, A., 7, 45, 94, 141, 184, 202, 280
Khoo, C., 141, 150
Kobsa, A., 5, 140
Kraus, S., 83, 85, 86, 109–111, 113, 291
Kripke, S.A., 41, 69
Kudenko, D., 95

Laaser, W., 230
Lalmas, M., 5–7, 62, 65, 67, 69, 72, 133, 293
Lambrix, P., 92
Landau, M.C, 150
Landman, F., 76, 107
Larouk, O., 15, 16, 102
Le Guern, M., 16
Lehman, D., 83, 85, 86, 109–111, 113, 291
Lehmann, F., 230
Lesca, H., 1
Levesque, H.J., 5, 9, 140
Li, M., 141, 150
Libourel, T., 18
Liddy, E., 141, 150
Liquière, M., 204, 230
Lopez-Lopez, A., 230
Lorensen, W., 258
Lynch, K.J., 17

Müller, A., 62
Magidor, M., 83, 85, 86, 109–111, 113, 291
Malmkjaer, K., 22
Manzano, M., 260
Mark, W.S., 5
Markowitz, J.A., 164
Maron, M.E., 82
Martin, P., 141
McGill, M., 230
McGill, M.J., 3, 5, 13, 80
McGregor, J.J., 230
McGregor, R., 5, 140
McGuiness, D., 140
McGuiness, D.L., 5
Mechkour, M., 3, 7, 25, 48, 90, 102, 132, 141, 255, 259, 261, 269, 280
Meghini, C., 3, 6, 7, 49, 62, 269
Melucci, M., 45
Miller, G.A., 17
Minka, T.P., 28
Minker, J., 228
Montague, R., 23
Mugnier, M.L., 145–148, 162, 165, 192, 197, 201, 204, 227, 230, 231, 255, 288, 294
Mulhem, P., 42, 52, 99, 102, 259, 280, 293, 294
Myaeng, S.H., 141, 150, 230

Nakashima, H., 136
Nanci, D., 18
Nebel, B., 3, 5, 9, 35, 43, 95, 140
Nicolas, J.M., 228
Nie, J., 2, 5, 7, 10, 41, 69, 98, 118, 125, 131, 292
Nowlan, W.A., 92

Nutter, J.T., 164

Ounis, I., 53, 67, 78, 80, 89, 101, 102, 125, 131, 141, 150–152, 159, 191, 224, 231, 232, 293, 294
Owsnicki-Klew, B., 140
Owsnicki-Klewe, B., 5

Padgham, L., 92
Palmer, P., 7, 16, 74, 102
Paradis, F., 8, 16, 25, 93, 102
Pasca, M., 191, 232
Patel-Schneider, P.F., 5, 140
Pearl, J., 293
Peat, H., 17, 34
Peirce, C.S., 175
Peters, S., 136
Piccard, R.W., 28
Plotkin, G., 72
Poole, J., 204, 230
Premerlan, W., 258
Profitlich, H.J., 95

Raghavan, V.V., 135
Rector, A.L., 92
Reiter, R., 114
Rijsbergen (van), C.J., 2, 3, 5, 10, 15, 62, 66, 67, 69, 72, 80, 118, 131, 133
Rivenc, F., 221
Roberts, D.D., 175
Robertson, S.E., 5
Rumbaugh J., 258

Sabah, G., 13, 17, 24
Saint-Dizier, P., 17, 94
Salton, G., 3, 5, 13–15, 33, 80, 230
Salvat, E., 192, 197
Schank, R.C., 7, 23, 39
Schmiedel, A., 5, 140
Schmolze, J.G., 5, 38
Schröder, M., 141
Sebastiani, F., 3, 6, 7, 62, 69
Sembok, T.M.T., 69, 131
Solomon, W.D., 92
Sowa, J.F., 9, 89, 95, 111, 139–141, 149, 162, 165, 176, 178, 181, 187, 205, 206, 209, 260, 261
Straccia, U., 3, 6, 7, 62
Suppes, P., 73
Suzuki, H., 136
Swain, M.J., 135
Swamy, M.N.S., 201
Swan, M., 103

Thanos, C., 3, 6, 7, 62
Thiel, T., 62
Thulasiraman, K., 201
Tin, E., 136
Tutiya, S., 72

Vailaya, A., 135
Vardi, M., 230
Viegas, E., 17, 94
Voronkov, A., 230

Weide (van der), T., 85
Wermelinger, M., 162
Wilks, Y., 23
Willems, M., 162, 230
Willit, P., 17, 34
Wondergem, B.C.M., 85
Woods, W.A., 36, 37
Wuwongse, V., 260

Yang, B., 17
Yen, J., 5, 140

Zweigenbaum, P., 141, 150