Contribution à l'étude des relations d'inférence non-monotone combinant inférence classique et préférences

Marie-Christine LAGASQUIE-SCHIEX

Thèse présentée à l'Université Paul Sabatier de Toulouse
en vue de l'obtention du titre de
Docteur de l'Université Paul Sabatier
Spécialité : Informatique

Soutenue le 8 décembre 1995 devant la commission :
Mme C. CAYROL (examinateur)
Mr M. CAYROL (examinateur)
Mr M. COOPER (examinateur)
Mr F. LÉVY (rapporteur)
Mr H. PRADE (examinateur)
Mme MC. VILAREM (rapporteur)

Numéro d'ordre 2217

Remerciements

Je tiens à remercier tout particulièrement :

- Claudette Cayrol pour avoir su orienter mes recherches et pour avoir eu le courage de se plonger dans chacune des démonstrations de cet ouvrage ;
- Michel Cayrol pour m'avoir accueillie dans son équipe ;
- Marie-Catherine Vilarem et François Lévy pour avoir réussi à lire ce document en entier et en détail sans succomber aux charmes de Morphée dissimulés entre quelques pages ;
- Martin Cooper et Henri Prade pour avoir accepté de faire partie de mon jury de soutenance.

Je remercie aussi Thomas dont le soutien et la présence ont su tout à la fois me motiver et me "déstresser".

Et enfin, un grand merci à toute l'équipe du 3ème étage avec laquelle, de repas au RU en cérémonie du thé en passant par quelques bridges, le temps a passé bien vite.

Résumé

Le domaine abordé ici est le raisonnement non-monotone et plus particulièrement les formalismes modélisant la capacité de déduction d'un tel raisonnement. Ce travail ne porte pas sur le développement de nouveaux formalismes de raisonnement non-monotone, mais propose une étude comparative dans un cadre unificateur. À partir d'une base de croyances, on choisit quelques formalismes combinant un principe d'inférence p et un mécanisme de génération m de sous-ensembles de croyances consistants issus de la base initiale et pouvant être ordonnés ou pas à l'aide de préférences. On identifie ainsi deux classes de relations d'inférence :

- celles du type $(E, <) \mid\!\sim^{p,m} \Phi$, notées p-m ;
- et celles du type $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$, notées G-p-m.

Plusieurs relations d'inférence non-monotone peuvent être définies selon le mécanisme de génération m et le principe d'inférence p retenus. Une comparaison théorique de ces processus est alors effectuée selon trois points de vue distincts : la complexité temporelle, la prudence (réalisme d'une relation du point de vue du nombre de conclusions à obtenir), les propriétés de déduction (approche axiomatique de l'inférence non-monotone). Les résultats obtenus sont les suivants :

- la complexité dans le cas propositionnel général est catastrophique, et seules des restrictions extrêmement contraignantes amènent à une complexité polynomiale ;
- le critère de prudence est très subjectif par rapport au contexte ;
- les résultats concernant les propriétés de déduction confirment la prédominance de certaines relations due essentiellement à la subjectivité des propriétés étudiées ; ils prouvent aussi que les propriétés dites essentielles sont insuffisantes (non détection de problèmes importants).

Cette étude théorique est suivie d'une partie pratique portant sur l'algorithmique de certaines de ces relations. On présente ainsi quelques algorithmes et leurs résultats.

Table des matières

1 Introduction et motivations générales **1**
- 1.1 Problématique . . . 1
- 1.2 Cadre des travaux . . . 2
- 1.3 Guide de lecture . . . 3

2 Définition des relations d'inférence non-monotone **5**
- 2.1 Processus de création d'une relation d'inférence non-monotone . . . 5
- 2.2 État de l'art sur les mécanismes de sélection . . . 7
  - 2.2.1 Les sous-bases consistantes maximales pour l'inclusion et les ordres associés 8
  - 2.2.2 Les sous-bases consistantes et l'ordre lexicographique . . . 14
  - 2.2.3 Les extensions de la logique des défauts . . . 16
  - 2.2.4 Les sous-bases consistantes et l'ordre "Best-Out" . . . 19
  - 2.2.5 Les ensembles de Gärdenfors et Makinson . . . 24
- 2.3 Choix des mécanismes et motivations de ces choix . . . 25
- 2.4 État de l'art sur les principes d'inférence . . . 26
  - 2.4.1 La conséquence forte . . . 27
  - 2.4.2 La conséquence faible . . . 27
  - 2.4.3 La conséquence argumentative . . . 27
  - 2.4.4 La conséquence forte avec préférences . . . 28
  - 2.4.5 La conséquence faible avec préférences . . . 28
  - 2.4.6 La conséquence argumentative avec préférences . . . 28
  - 2.4.7 Les autres principes . . . 29
- 2.5 Choix des principes d'inférence et motivations de ces choix . . . 29
- 2.6 Différentes classes de relations d'inférence . . . 29
  - 2.6.1 La première définition pour les relations de classe 2 . . . 30
  - 2.6.2 La seconde définition pour les relations de classe 2 . . . 31
  - 2.6.3 Caractérisation des sous-ensembles préférés dans la première définition . . . 31
  - 2.6.4 Comparaison des deux définitions . . . 43
  - 2.6.5 Définition utilisée pour les relations de classe 2 . . . 45
  - 2.6.6 Lien entre les deux classes de relations d'inférence non-monotone . . . 46
- 2.7 Récapitulatif des relations à étudier . . . 46

3 État de l'art sur quelques critères de comparaison théoriques **49**
- 3.1 Liste de quelques critères possibles, choix et motivations de ces choix . . . 49
- 3.2 Complexité . . . 49
  - 3.2.1 Qu'est-ce-qu'un problème ? Comment le formaliser ? . . . 50
  - 3.2.2 Les machines de Turing et les classes de problèmes . . . 51
  - 3.2.3 Les exemples de problèmes . . . 53
  - 3.2.4 Représentation de la hiérarchie polynomiale . . . 53
  - 3.2.5 Les conjectures de la théorie de la complexité . . . 54
- 3.3 Prudence . . . 54
- 3.4 Propriétés de déduction . . . 55

3.4.1 Les propriétés de Kraus, Lehmann et Magidor et leur sémantique 55
3.4.2 Les propriétés de Gärdenfors et Makinson et leur sémantique 67
3.4.3 Quelle est la classe de relations concernée par l'étude des propriétés de déduction ? 71

4 Résultats de la comparaison théorique 73
4.1 Complexité 73
4.1.1 Champ de l'étude de complexité 73
4.1.2 Étude de complexité dans le cas général 75
4.1.3 Étude de complexité des différentes relations d'inférence dans des cas particuliers 110
4.1.4 Synthèse des résultats et conclusion 144
4.2 Prudence 151
4.3 Propriétés de déduction 153
4.3.1 Les résultats déjà existants et leurs incidences 153
4.3.2 Les nouveaux résultats 158
4.3.3 Conclusion sur les résultats 175
4.4 Réunion des trois critères de comparaison théorique 180
4.4.1 La complexité 181
4.4.2 La prudence 181
4.4.3 Les propriétés de déduction 182
4.4.4 Les trois points de vue à la fois 182
4.4.5 Conclusion générale 190

5 Application à des problèmes de révision 193
5.1 Deux problématiques liées : inférence plausible, révision de bases de croyances . . . 193
5.2 Quelques résultats 194

6 Mise en œuvre de relations d'inférence non-monotone 199
6.1 Liste de quelques travaux sur l'algorithmique 199
6.2 Les spécificités algorithmiques des relations étudiées 199
6.3 Les choix 200
6.4 Les algorithmes 201
6.4.1 L'algorithme naïf 201
6.4.2 L'algorithme utilisant Max-Sat 202
6.4.3 L'approche utilisant les BDD 204
6.5 Les résultats expérimentaux 212
6.5.1 L'implémentation des algorithmes 212
6.5.2 Les jeux de tests 214
6.5.3 Conclusion sur l'étude algorithmique 218

7 Conclusions et perspectives 225

Bibliographie 229

A Théorèmes et définitions utilisés pour les démonstrations de complexité 235
A.1 Les théorèmes et définitions généraux 235
A.2 Les théorèmes et définitions pour la révision 237
A.2.1 Complexité de Fmr 237
A.2.2 Complexité de Sbr 240
A.2.3 Complexité de Pbr 244
A.2.4 Complexité de Ubr 245

B Tableau récapitulatif des transformations polynomiales utilisées 249

Index **261**

Table des figures

2.1 Processus de création d'une relation d'inférence non-monotone 6
2.2 Mécanisme de sélection E : Arbre des extensions 18
2.3 Mécanisme de sélection Bo 36
2.4 Mécanisme de sélection E* : Arbre de recherche des extensions de E 43
2.5 Mécanisme de sélection E* : Arbre de recherche des extensions de $\Psi \oplus E$ 44

3.1 Exemple d'un modèle cumulatif 58
3.2 Exemple d'un modèle cumulatif ordonné 59
3.3 Exemple d'un modèle préférentiel 60
3.4 Exemple d'un modèle rangé 62
3.5 Exemple d'un modèle simplement cumulatif 63
3.6 Exemple d'un modèle simplement préférentiel 64
3.7 Les systèmes de Kraus, Lehmann et Magidor – Classement par ordre de puissance. 65

4.1 Liens de prudence entre les relations d'inférence non-monotone de classe 1 154
4.2 Liens de prudence entre les relations d'inférence non-monotone de classe 2 155
4.3 Lien entre quelques relations G-Uni-m (point de vue des propriétés) 180
4.4 Pré-ordre entre les relations d'inférence non-monotone (prudence – convention 1) . 183
4.5 Pré-ordre entre les relations d'inférence non-monotone (prudence – convention 2) . 184

6.1 Fonction booléenne – Arbre de décision – BDD 205
6.2 Les ROBDD R_E, R'_E et R''_E de l'exemple sur les pingouins 210
6.3 Les ROBDD avec arcs inversés R_E, R'_E et R''_E de l'exemple sur les pingouins . . . 211
6.4 Résultats des trois approches sur tests aléatoires – 5, 10 et 20 variables 219
6.5 Les différents temps obtenus avec les BDD sur tests aléatoires – 5, 10 et 20 variables 220
6.6 Les différentes tailles obtenues avec les BDD sur tests aléatoires – 5, 10 et 20 variables 221
6.7 Impact du nombre de variables sur le temps de réponse des trois méthodes 222

Table des tableaux

2.1 L'ordre "Best-Out" (exemple 1) . . . 20
2.2 L'ordre "Best-Out" (exemple 2) . . . 22
2.3 L'ordre "Best-Out" (exemple 3) . . . 23
2.4 Les relations d'inférence non-monotone de la classe 1 notées *p-m* . . . 47
2.5 Les relations d'inférence non-monotone de la classe 2 notées *G-p-m* . . . 47

3.1 Les systèmes de Kraus, Lehmann et Magidor – Récapitulation et sémantique. . . . 64
3.2 Les variantes du système P et des relations rationnelles . . . 66

4.1 Démonstration de complexité pour ARG-E : EXI-E $\propto$ ARG-E . . . 107
4.2 Démonstration de complexité pour ARG-E : co-EXI-E $\propto$ ARG-E . . . 109
4.4 Résultats existants sur les propriétés des relations *G-p-m* par rapport à Kraus, Lehmann et Magidor. . . . 159
4.5 Résultats existants sur les propriétés des relations *G-p-m* par rapport à Gärdenfors et Makinson. . . . 159
4.6 Récapitulatif des résultats du point de vue de Kraus, Lehmann et Magidor . . . 176
4.7 Récapitulatif des résultats du point de vue de Gärdenfors et Makinson . . . 177
4.8 Complexité des relations d'inférence non-monotone . . . 181
4.9 Prudence des relations d'inférence non-monotone – convention 1 . . . 182
4.10 Prudence des relations d'inférence non-monotone – convention 2 . . . 185
4.11 Propriétés de déduction des relations d'inférence non-monotone . . . 186
4.12 Pré-ordre entre les relations d'inférence non-monotone (prudence – convention 1) . 187
4.13 Pré-ordre entre les relations d'inférence non-monotone (prudence – convention 2) . 188

5.1 Liens postulats de la révision – propriétés de déduction . . . 195
5.2 Liens propriétés de déduction – postulats de la révision . . . 195
5.3 Liens révision – inférence non-monotone . . . 197

6.1 Exemple des pingouins : la base de croyances . . . 214
6.2 Exemple du salaire : la base de croyances (31 clauses) . . . 215
6.3 Exemple du salaire : la base de croyances (77 clauses) . . . 216
6.4 Résultats sur les BDD . . . 217
6.5 Comparaison des trois méthodes sur quelques exemples choisis . . . 217

Chapitre 1

Introduction et motivations générales

1.1 Problématique

L'automatisation du raisonnement de sens commun est un thème majeur en intelligence artificielle. Elle symbolise la tentative de modélisation de l'esprit humain au travers d'une machine. À ce titre, elle rencontre de nombreux problèmes dont la représentation des connaissances et la mécanisation de processus d'inférence exploitant ces connaissances.

Historiquement, la logique classique a été l'un des premiers outils développés dans ce cadre. Ainsi, on dispose actuellement de toute une batterie d'outils théoriques ou pratiques permettant de représenter et d'utiliser des connaissances. Toutefois, de nombreux travaux ont mis en évidence certaines limitations de la logique classique en tant que modèle de raisonnement. La plus importante est, sans conteste, le fait que l'inférence classique ne peut pas prendre en compte des *croyances* contradictoires, ce que le raisonnement humain paraît maîtriser.

Remarquons que le terme employé ici n'est plus "connaissances" mais "croyances". Je me situe donc implicitement dans le cadre de connaissances qui n'ont qu'un statut de croyances, c'est-à-dire qui peuvent être incomplètes, incertaines, révisables, parfois incohérentes entre elles.

Pour contourner les limites de la logique classique, de nombreuses propositions ont été faites.

Certaines approches consistent à définir un nouveau langage ainsi que des systèmes d'inférence associés. C'est, par exemple, le cas de la logique des défauts de [Rei80], ou des logiques modales non-monotones (voir [HC68, Luk90]).

D'autres approches consistent à associer la logique classique à des techniques ou des heuristiques pouvant combler ses lacunes. Ici aussi, de nombreuses idées ont été exposées. L'une d'entre elles revient à exploiter un ordre ou un pré-ordre entre les croyances, qu'il soit exprimé de manière symbolique ou numérique (voir [Poo88, Bre89b, Pea90, Bra91]).

Notons qu'il existe divers travaux sur cette approche concernant, en particulier, la manière d'articuler les formules de la base de croyances et les préférences entre ces formules. Par exemple, certains auteurs utilisent des préférences exogènes entre les formules, donc imposées a priori et n'ayant pas de contrainte spécifique à respecter (voir [Poo88, Bre89b, CRS92]). D'autres travaillent plutôt sur des pré-ordres reflétant la dépendance logique entre les formules de la base ; les préférences ainsi définies doivent alors respecter certaines contraintes, par exemple, la contrainte de dominance (voir [GM94]). On trouve aussi des travaux portant sur la définition de préférences à partir des modèles de la base en utilisant, par exemple, le principe de spécificité (voir [Gef92, Pea90]). Les deux premières approches ont comme point commun l'aspect "syntaxique" de la définition des préférences entre les formules. En effet, les préférences ainsi définies sont issues de la syntaxe de la base et non des modèles de la base, s'opposant ainsi à la troisième approche. Notons que, dans les travaux cités précédemment, les préférences sont exprimées par des pré-ordres ou des ordres totaux. Il existe aussi quelques travaux portant sur la modélisation des préférences par un pré-ordre partiel (voir [Bra91, CRS92, Men94]).

Pour résumer, on trouve de très nombreux travaux sur la modélisation du raisonnement de sens commun par la combinaison de la logique classique et d'un raisonnement préférentiel, et en particulier sur les processus de déduction mis en œuvre : définition de différents ordres ou pré-ordres (voir entre autres [Bre89b, CRS92, DLP91, BCD+93, Leh92]), définitions de nouvelles relations d'inférence (voir par exemple [BDP93, Bra93]), travaux sur la complexité de certaines relations d'inférence (voir [Neb91, Got92]), sur les propriétés de déduction que doivent vérifier les relations d'inférence (voir [KLM90, GM94, Bra93, BCD+93]), entre autres.

L'objectif de cette thèse est de "faire le point" sur ce sujet, en proposant un cadre unificateur pour mener à bien des études comparatives. Il ne s'agit ici pas de créer de nouvelles théories, mais plutôt d'étudier celles qui existent déjà, afin de mieux comprendre les mécanismes mis en œuvre, et ainsi pouvoir identifier les applications auxquelles ces théories sont le plus adaptées.

1.2 Cadre des travaux

Je traiterai des bases de croyances éventuellement inconsistantes sans me préoccuper de savoir d'où provient cette inconsistance. Toutefois, notons que parmi les origines d'une inconsistance, on peut citer :

- un problème de révision : on part d'une base[1] et on rajoute une information considérée comme préférée à toutes celles déjà présentes dans la base ; ce faisant, on se peut se retrouver avec une nouvelle base inconsistante ;
- un problème de fusion de plusieurs bases conflictuelles ;
- la présence dans la base d'assertions conditionnelles avec exception (par exemple, des règles par défauts).

D'autre part, j'ai choisi d'étudier les approches couplant inférence classique et raisonnement préférentiel dans le but de profiter de toutes les connaissances disponibles sur l'inférence classique. De plus, je m'intéresse tout particulièrement à celles s'appuyant sur des préférences exogènes car elles me semblent à la fois plus simples et plus générales :

- il n'y a pas de pré-ordre à créer a posteriori puisque il est donné a priori ; cela me semble une méthode efficace car, du point de vue applicatif, en partant d'une base déjà pré-ordonnée, on économise le temps de calcul de l'ordre entre les croyances ;
- le pré-ordre entre les formules n'a aucune contrainte spécifique à respecter ;
- l'aspect "pré-ordre imposé a priori" correspond bien à l'image d'un pré-ordre donné par un expert[2], alors qu'un pré-ordre déduit de la base (que ce soit syntaxiquement ou sémantiquement) semble moins naturel, le résultat ne correspondant pas forcément à celui qu'aurait donné l'expert.

De plus, je reprends les mêmes hypothèses que [Neb91] en choisissant d'aborder ces approches sous l'aspect "syntaxique", c'est-à-dire en travaillant au niveau de la syntaxe des formules de la base et pas au niveau des modèles (pour des approches "sémantiques" voir [KLM90, BCD+93, GM94]). Je considère chaque croyance de la base comme un tout indivisible. Elle sera donc soit acceptée, soit rejetée en bloc[3]. Cet aspect "tout ou rien" me paraît bien correspondre au fait qu'un expert

[1] Dans le cadre de la révision d'une base, on considère en général que la base est consistante.

[2] Bien sûr, il reste toujours le problème de "récolter" ce pré-ordre auprès des experts, mais ce n'est pas mon propos.

[3] Si on prend l'exemple de la logique propositionnelle, dans une approche syntaxique la formule $a \wedge b$ ne sera pas toujours traitée comme l'ensemble de formules $\{a, b\}$.

a pris la peine d'exprimer une connaissance sous la forme d'une formule plutôt qu'une autre ; cette connaissance doit donc être considérée comme un tout même si elle peut être décomposée en plusieurs éléments de connaissance. Un inconvénient de cette attitude résolument "syntaxique" est que l'analyse d'une base de connaissances doit se faire au niveau des formules et non plus au niveau de la base en tant qu'entité ; cela peut être gênant car deux bases sémantiquement équivalentes (c'est-à-dire ayant les mêmes modèles) ne produiront pas forcément les mêmes conclusions.

L'expression "approches syntaxiques" que j'utiliserai ici recouvre donc plusieurs concepts :

- les préférences étudiées sont exogènes et imposées comme un élément de la syntaxe de la base ; elles ne sont pas déduites de la base ni syntaxiquement, ni sémantiquement ;
- la base étudiée sera traitée au niveau des formules et pas au niveau des modèles.

Et enfin, le but de ma thèse étant de comparer les travaux déjà existants, je me limiterai dans cet ouvrage à l'étude de préférences exprimées sous la forme d'un pré-ordre total, car la grande majorité des travaux portant sur ce sujet concerne des préférences totales entre les formules et non partielles.

1.3 Guide de lecture

Tout ce document repose sur la démarche suivante : présentation de divers formalismes et, *pour chacun d'entre eux*, étude suivant divers points de vue. Le choix de cette méthode de travail, ainsi que les liens existant entre les formalismes analysés, expliquent l'aspect souvent répétitif et parfois redondant de cet ouvrage. Je pense toutefois que cette approche méthodologique s'impose par le fait qu'il s'agit là d'une synthèse de travaux présentant à la fois des points communs et des différences importantes ; il est alors essentiel de ne pas perdre de vue les spécificités de chacun tout en finissant par en tirer des conclusions plus générales. Ainsi, chaque point de vue sera d'abord abordé dans le détail de chaque formalisme, puis examiné de façon plus globale. Dans cette optique, le plan de cette thèse est le suivant.

La première partie (chapitre 2 page 5) constitue une présentation de la construction des relations d'inférence non-monotone que j'ai choisies d'étudier. Cette technique est directement inspirée des travaux de Pinkas et Loui (voir [PL92]) qui présentent les relations d'inférence non-monotone comme la combinaison d'un mécanisme de génération d'ensembles de croyances consistants issus de la base initiale (ordonnés ou pas à l'aide de préférences) et d'un principe d'inférence permettant de déduire de nouvelles croyances à partir de la déduction classique sur les ensembles générés précédemment. Dans cette première partie, je présente donc le mode de création, puis plusieurs mécanismes de génération avec différents types de préférences et enfin quelques principes d'inférence. Cela m'amène à établir la liste précise des relations que je vais étudier, ainsi que quelques résultats globaux importants pour la suite de ma synthèse.

Dans une seconde partie (chapitre 3 page 49), je sélectionne et je présente quelques critères destinés à effectuer la comparaison théorique des relations explicitées et choisies dans la partie précédente.

La comparaison théorique et ses conclusions sont alors exposées dans la troisième partie (chapitre 4 page 73) en procédant d'abord par une analyse au cas par cas puis en concluant de manière plus globale.

La quatrième partie (chapitre 5 page 193) permet de préciser le lien établi par cette modélisation entre diverses problématiques (la révision de croyances et l'inférence plausible) et comment nos propres résultats se retrouvent dans un tel schéma.

La cinquième partie (chapitre 6 page 199) est consacrée aux questions algorithmiques.

La sixième et dernière partie (chapitre 7 page 225) permet de conclure tout en présentant quelques perspectives.

Chapitre 2

Définition des relations d'inférence non-monotone

2.1 Processus de création d'une relation d'inférence non-monotone

Le langage utilisé pour énoncer les formules de la base de croyances dénotée E peut être soit le langage de la logique propositionnelle, soit le langage de la logique des prédicats avec des formules fermées. Toutefois, dans le cadre de mon étude, je me limiterai à des formules de la logique propositionnelle, en particulier à cause de résultats de complexité (voir section 3.2 page 49). Dans toute la suite, E dénotera donc un ensemble fini non vide de formules propositionnelles.

Lorsqu'on disposera d'une relation de préférence sur les éléments de E, elle sera notée $<$ et représentée en général par une relation de pré-ordre (réflexive et transitive) définie sur les formules de la base E. Les propriétés requises pour cette relation seront précisées lorsque ce sera nécessaire (en fonction des approches décrites).

Ayant choisi d'étudier des approches qui couplent inférence classique et préférences, je suis donc amenée naturellement à choisir le processus de création d'une relation d'inférence non-monotone décrit par Pinkas et Loui dans [PL92]. Ce processus est un processus en deux étapes (voir figure 2.1 page suivante) :

- premièrement, sélection de sous-bases préférées (qui se doivent d'être consistantes[1]) de la base E en utilisant les préférences entre les formules de E ; notons que, dans le cas où les préférences sont exprimées sur E (qu'elles soient partielles ou totales), on peut préalablement appliquer un processus d'agrégation afin d'obtenir des préférences entre les sous-bases consistantes et ainsi déterminer les éléments préférés (voir [CRS92]) ;
- deuxièmement, utilisation d'un principe d'inférence afin de définir l'inférence non-monotone à l'aide de l'inférence classique sur certaines des sous-bases préférées, puisqu'elles sont consistantes.

Ce processus est particulièrement bien adapté aux approches étudiées ici, puisqu'il permet l'utilisation de toutes les connaissances dont on dispose en logique classique ; on se ramène à des outils bien connus. On peut ainsi profiter de toutes les améliorations concernant la logique classique.

Remarquons que ce processus correspond à une phase de "restauration de la cohérence" puisqu'on cherche à raisonner avec des informations cohérentes.

Dans la suite, je vais présenter différents mécanismes de sélection et principes d'inférence.

[1] Pour moi, toute sous-base preférée est avant tout consistante. Afin d'alléger l'écriture, j'utiliserai donc l'expression "sous-base preférée" au lieu de "sous-base consistante préférée". Cette remarque ne sera pas répetée.

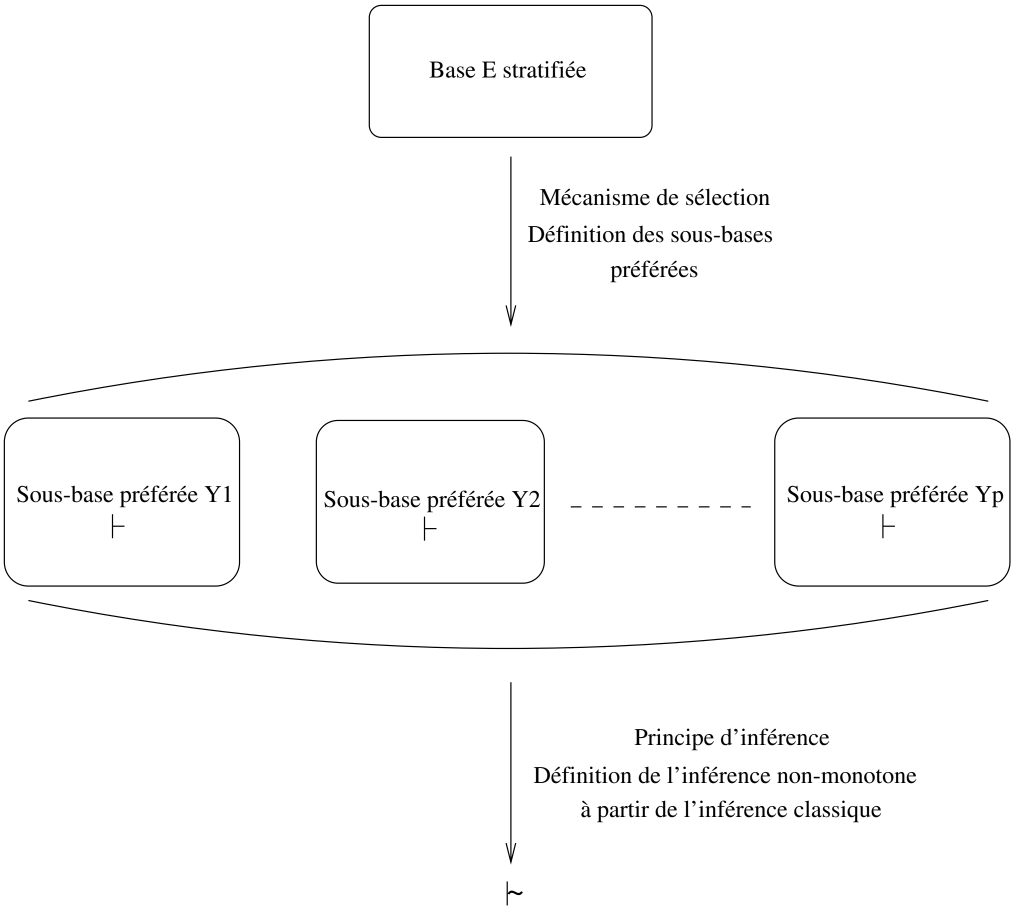


Figure 2.1 : Processus de création d'une relation d'inférence non-monotone

2.2 État de l'art sur les mécanismes de sélection

J'appellerai mécanisme de sélection initial, tout mécanisme permettant d'obtenir un ensemble de sous-bases consistantes ordonné ou pas, à partir d'une base E de formules munie d'une relation de préférence $<$.

Remarquons que, quand $<$ correspond à un pré-ordre, il existe plusieurs techniques pour partitionner E, par exemple la stratification uniforme. On appelle stratification uniforme de E le résultat de l'application d'un tri topologique sur E[2]. Dans ce cas, on dit que la base E est stratifiée et on note : $E = E_1 \cup \ldots \cup E_n$ avec n nombre de strates dans E.

Je présente successivement des formalismes privilégiant la maximalité au sens de l'inclusion (section 2.2.1 page suivante), puis ceux utilisant la maximalité au sens de la cardinalité (section 2.2.2 page 14) qui permettent de raffiner les précédents. En section 2.2.3 page 16, je rappelle quelques notions de la logique des défauts de [Rei80] (il existe des liens étroits entre cette logique et les mécanismes basés sur l'inclusion). Le formalisme présenté ensuite s'appuie sur un principe différent : le principe du minimum de spécificité (voir section 2.2.4 page 19). Et pour finir (voir section 2.2.5 page 24), je cite le formalisme proposé par Gärdenfors et Makinson, qui impose aux préférences entre formules le respect d'un certain nombre de contraintes, ce qui n'était pas le cas pour les mécanismes présentés jusqu'alors. Remarquons que les formalismes donnés en section 2.2.3 page 16 et 2.2.5 page 24 sortent du cadre de mon étude puisque ne faisant pas partie des approches “syntaxiques” ; ils servent ainsi de points de comparaison.

De plus, tout au long de cette section, je vais utiliser un exemple unique, celui des pingouins, représenté par une base propositionnelle stratifiée $(E, <)$. J'ai choisi pour illustrer les différentes définitions trois stratifications (en première colonne le numéro de strate E_i, en seconde colonne les formules de la base, en dernière colonne la signification des formules de la base) :

- stratification numéro 1 de la base des pingouins (une seule formule par strate : stratification stricte) :

E_1	$\rightarrow P$	titi est un pingouin
E_2	$P \rightarrow O$	un pingouin est un oiseau
E_3	$P \rightarrow \neg V$	un pingouin ne sait pas voler
E_4	$O \rightarrow V$	un oiseau sait voler
E_5	$O \rightarrow A$	un oiseau a des ailes
E_6	$A \rightarrow V$	Si on a des ailes, on sait voler

- stratification numéro 2 de la base des pingouins (3 strates, 2 formules par strate) :

E_1	$\rightarrow P$	titi est un pingouin
	$P \rightarrow O$	un pingouin est un oiseau
E_2	$P \rightarrow \neg V$	un pingouin ne sait pas voler
	$O \rightarrow V$	un oiseau sait voler
E_3	$O \rightarrow A$	un oiseau a des ailes
	$A \rightarrow V$	si on a des ailes, on sait voler

[2] Il existe plusieurs possibilités de tris topologiques. Le plus utilisé en raisonnement non-monotone est le suivant. Soit E une base de formules munie d'un pré-ordre $<$, on applique l'algorithme :

```
i ← 1
E1 ← ∅
tant que E non vide faire
        Ei ← Max(E) (*Max(E) = les éléments maximaux de E pour < *)
        E ← E - Ei
        i ← i + 1
fin tant que
```

- stratification numéro 3 de la base des pingouins (une seule strate) :

E_1	$\rightarrow P$	titi est un pingouin
	$P \rightarrow O$	un pingouin est un oiseau
	$P \rightarrow \neg V$	un pingouin ne sait pas voler
	$O \rightarrow V$	un oiseau sait voler
	$O \rightarrow A$	un oiseau a des ailes
	$A \rightarrow V$	Si on a des ailes, on sait voler

2.2.1 Les sous-bases consistantes maximales pour l'inclusion et les ordres associés

2.2.1.1 Définitions principales

Soit E un ensemble fini non vide de formules, on définit :

Définition 2.2.1 *Les sous-bases consistantes A de E sont les sous-ensembles de E consistants au sens de la logique classique.*

Le mécanisme de sélection associé aux sous-bases consistantes sera noté S.

Définition 2.2.2 *Une sous-base A de E est une sous-base consistante maximale pour l'inclusion (sous-base maximale consistante) de E ssi :*

- *A est une sous-base consistante,*
- *il n'existe pas de sous-base consistante de E contenant strictement A.*

Les sous-bases de E maximales consistantes seront aussi appelées thèses de E.

Le mécanisme de sélection associé aux thèses sera noté T. Remarquons alors que nous avons la propriété suivante, qui prendra toute son importance lors de l'analyse des relations issues du mécanisme S (voir les théorèmes 4.1.5 page 81 et 4.1.6 page 82) :

Propriété 2.2.1 *Soit E une base de croyances, $\exists S$ sous-base consistante de E telle que $S \vdash H$ $\Leftrightarrow$ $\exists T$ thèse de E telle que $T \vdash H$.*

> **Preuve :** Il suffit effectivement de constater que si S infère H alors n'importe quelle thèse T contenant S[3] inférera aussi H. De même si T infère H alors il existe une sous-base consistante $S = T$ qui infère H. □

Exemple : Si on calcule les thèses correspondant à l'exemple présenté en tête de la section 2.2 page précédente, on trouve 5 solutions, et ceci quelle que soit la stratification choisie :

Thèse 1	Thèse 2	Thèse 3	Thèse 4	Thèse 5
$\rightarrow P$	$\rightarrow P$	$\rightarrow P$	$\rightarrow P$	$P \rightarrow O$
$P \rightarrow O$	$P \rightarrow O$	$P \rightarrow O$	$P \rightarrow \neg V$	$P \rightarrow \neg V$
$P \rightarrow \neg V$	$P \rightarrow \neg V$	$O \rightarrow V$	$O \rightarrow V$	$O \rightarrow V$
$O \rightarrow A$	$A \rightarrow V$	$O \rightarrow A$	$O \rightarrow A$	$O \rightarrow A$
		$A \rightarrow V$	$A \rightarrow V$	$A \rightarrow V$

[3] Et il en existera nécessairement une.

Il existe de nombreux travaux qui utilisent les thèses de E[4]. On les rassemble sous le terme "Inclusion Based Preference", c'est-à-dire que parmi toutes les sous-bases consistantes, on privilégie les sous-bases maximales consistantes. Ces différentes approches peuvent paraître redondantes. Elles sont en effet intimement liées les unes aux autres (soit parce qu'elles sont équivalentes, soit parce que certaines sont des cas particuliers des autres), et, en définitive, elles correspondent toutes au même mécanisme de sélection que nous noterons Incl (voir section 2.2.1.5 page 14). Dans les paragraphes suivants, et malgré l'aspect répétitif de la tâche, je vais présenter tous ces travaux ainsi que la terminologie utilisée suivant les différents auteurs, afin de mettre en évidence l'historique et les spécificités de chaque possibilité. Ainsi, nous verrons successivement les travaux de Brewka [Bre89b, Bre89a], puis une généralisation de ces travaux [CRS92], et, pour finir, une approche équivalente issue de la logique possibiliste [DLP91].

2.2.1.2 Les sous-théories préférées

Définition 2.2.3 *Soit E une base stratifiée, une sous-théorie préférée est un ensemble $S = S_1 \cup \ldots \cup S_n$ tel que $\forall k \in [1, n]$, $S_1 \cup \ldots \cup S_k$ est un sous-ensemble maximal consistant de $E_1 \cup \ldots \cup E_k$ (voir [Bre89a, Bre89b]).*

Par définition, les sous-théories préférées de Brewka sont des thèses de E.

Exemple 1 : Soit E la base propositionnelle stratifiée correspondant à la stratification numéro 1 de l'exemple présenté en tête de la section 2.2 page 7. En appliquant la méthode de Brewka, on obtient une seule sous-théorie préférée :

$$\left|\begin{array}{l} \rightarrow P \\ P \rightarrow O \\ P \rightarrow \neg V \\ O \rightarrow A \end{array}\right.$$

Exemple 2 : Soit E la base propositionnelle stratifiée correspondant à la stratification numéro 2 de l'exemple présenté en tête de la section 2.2 page 7. Dans ce cas là, on trouve 3 sous-théories préférées :

$$\left|\begin{array}{l} \rightarrow P \\ P \rightarrow O \\ P \rightarrow \neg V \\ O \rightarrow A \\ \end{array}\right. \qquad \left|\begin{array}{l} \rightarrow P \\ P \rightarrow O \\ P \rightarrow \neg V \\ A \rightarrow V \\ \end{array}\right. \qquad \left|\begin{array}{l} \rightarrow P \\ P \rightarrow O \\ O \rightarrow V \\ O \rightarrow A \\ A \rightarrow V \end{array}\right.$$

Exemple 3 : Soit E la base propositionnelle stratifiée correspondant à la stratification numéro 3 de l'exemple présenté en tête de la section 2.2 page 7. En appliquant la méthode de Brewka, on trouve alors que toutes les thèses sont des sous-théories préférées.

Conclusions sur les sous-théories préférées :

1. Brewka calcule directement les sous-théories préférées sans passer par le calcul des thèses.
2. Dans son processus, Brewka ne définit pas de relation de préférence entre ses sous-théories préférées. Donc, dans le cas des exemples 2 et 3 ci-dessus, les solutions obtenues ne sont pas comparables.

[4] L'idée d'utiliser les sous-bases maximales consistantes d'une base éventuellement inconsistante est déjà ancienne (voir [RM70]).

3. Brewka signale dans son article que dans certains cas, on retrouve le même type de résultats qu'avec les extensions de la logique des défauts de Reiter (voir [Rei80])[5].

4. Dans le cas d'une base ne contenant qu'une seule strate, l'ensemble des sous-théories préférées est égal à celui des thèses.

2.2.1.3 Les thèses préférées

Dans [CRS92][6], Cayrol, Royer, Saurel ont défini, à partir d'une relation $<$ (dite de préférabilité) sur les formules de E, plusieurs relations de préférence $\ll$ pour classer les sous-bases consistantes de E.

L'objectif premier était de généraliser les travaux de Brewka en définissant une préférence $\ll$ sur les sous-bases consistantes de E telle que les éléments maximaux pour cet ordre soient les sous-théories préférées définies dans [Bre89b].

Il a été démontré alors que la préférence $\ll$ sur les sous-bases consistantes généralise l'inclusion ensembliste. C'est donc une préférence du type "Inclusion Based Preference".

On a plusieurs résultats suivant que E est ordonné ou pas :

- si E n'est pas ordonné[7], alors $\ll$ correspond exactement à l'ordre basé sur l'inclusion ensembliste,
- si E est ordonné avec un pré-ordre $<$[8], alors on a 2 définitions possibles de l'ordre $\ll$ appelé ordre démocratique :
  - définition 2.2.4 valable dans le cas général, que $<$ soit total ou partiel,
  - définition 2.2.5 si $<$ est total et qui fournit un processus constructif.

L'ordre démocratique : Les définitions de l'ordre démocratique[9] sur les sous-bases consistantes de la base initiale ordonnée $(E, <)$ sont :

Définition 2.2.4 *Soient F et G, 2 sous-bases consistantes de $(E, <)$, F est préférée démocratiquement à G (noté $G \ll^d F$) ssi $\forall g \in G \setminus F$, $\exists f \in F \setminus G$ tel que $g < f$ et que l'on n'ait pas $f < g$.*

Définition 2.2.5 *Soit E la base initiale stratifiée, $E = E_1 \cup \ldots \cup E_n$ (E_1 strate de plus haute priorité, E_n strate de plus faible priorité). $\forall i$ et $\forall F$ un sous-ensemble de E, on note $F_i = F \cap E_i$. Soient F et G, 2 sous-bases consistantes de $(E, <)$, F est préférée démocratiquement à G (noté $G \ll^d F$) ssi $\exists i$ tel que F_i contient strictement G_i et $\forall E_j$ telle que E_j soit une strate de priorité supérieure à E_i[10], on a $F_j = G_j$.*

Dans le cas où $<$ est total sur E, les 2 définitions sont équivalentes et les éléments maximaux de cet ordre sont appelées thèses démo-préférées.

[5] À condition que les défauts exprimés soient des défauts normaux sans pré-requis, que l'ensemble W de Reiter soit égal à la strate la plus prioritaire de E et que l'ordre soit judicieux entre les formules des strates inférieures qui sont traduites sous forme de défaut. Il serait intéressant de formaliser ce que l'on entend par "ordre judicieux", mais ce n'est pas mon propos. Constatons quand même que la méthode de Brewka sur la base de l'exemple 2 fournit exactement les mêmes ensembles que les extensions de Reiter, la fermeture logique mise à part (voir section 2.2.3 page 16). Et si on modifie la base en transformant la strate numéro 1 en 2 strates 1 et 1bis, on obtient les sous-théories préférées $\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V, O \rightarrow A\}$ et $\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V, A \rightarrow V\}$ qui ne correspondent pas aux extensions.

[6] Dans les articles [Cay90] et [Cay92], Cayrol utilise le terme d'interprétation pour parler de sous-bases maximales consistantes. Désormais ce terme est remplacé par celui de thèse.

[7] E est alors composé d'une seule strate.

[8] Ce pré-ordre peut être total, on retrouve alors la stratification comme dans [Bre89b], ou bien seulement partiel.

[9] Le terme "démocratique" provient de travaux sur le choix social (voir bibliographie de [CRS92]).

[10] Ce qui veut dire avec les notations choisies ici que $j < i$.

Exemple 1 : Appliquons cette formule à la base propositionnelle stratifiée correspondant à la stratification numéro 1 de l'exemple présenté en tête de la section 2.2 page 7. On trouve alors l'ordre suivant sur les thèses :

$$\text{thèse 5} \ll^d \text{thèse 4} \ll^d \text{thèse 3} \ll^d \text{thèse 2} \ll^d \text{thèse 1}$$

Exemple 2 : Soit E la base propositionnelle stratifiée correspondant à la stratification numéro 2 de l'exemple présenté en tête de la section 2.2 page 7. En appliquant la définition de l'ordre démocratique, on trouve l'ordre suivant sur les thèses :

$$\text{thèse 4} \ll^d \text{thèse 1,}$$
$$\text{thèse 4} \ll^d \text{thèse 2,}$$
$$\text{thèse 4} \ll^d \text{thèse 3}$$
$$\text{thèse 5} \ll^d \text{thèse 1,}$$
$$\text{thèse 5} \ll^d \text{thèse 2,}$$
$$\text{thèse 5} \ll^d \text{thèse 3}$$

(les thèses 1, 2 et 3 sont incomparables, ainsi que les thèses 4 et 5)

Exemple 3 : Soit E la base propositionnelle stratifiée correspondant à la stratification numéro 3 de l'exemple présenté en tête de la section 2.2 page 7. En appliquant l'ordre démocratique, on trouve alors que toutes les thèses sont préférées et donc incomparables entre elles.

L'ordre élitiste : Dans [CRS92], on trouve aussi une autre relation de préférence : la préférence élitiste[11] qui correspond à la recherche d'explications préférées par rapport à un but à atteindre et pour laquelle sont aussi données 2 définitions :

- définition 2.2.6 valable dans le cas général, que $<$ soit total ou partiel,
- définition 2.2.7 valable uniquement si $<$ est total et qui fournit un processus constructif.

Les définitions de l'ordre élitiste sur les sous-bases consistantes de la base initiale ordonnée $(E, <)$ sont :

Définition 2.2.6 *Soient F et G, 2 sous-bases consistantes de $(E, <)$, F est préférée élitiquement à G (noté $G \ll^e F$) ssi $\forall f \in F \setminus G$, $\exists g \in G \setminus F$ tel que $g < f$ et que l'on n'ait pas $f < g$.*

Définition 2.2.7 *Soit E la base initiale stratifiée, $E = E_1 \cup \ldots \cup E_n$ (E_1 strate de plus haute priorité, E_n strate de plus faible priorité). $\forall i$ et $\forall F$ un sous-ensemble de E, on note $F_i = F \cap E_i$. Soient F et G, 2 sous-bases consistantes de $(E, <)$, F est préférée élitiquement à G (noté $G \ll^e F$) ssi $\exists i$ tel que G_i contient strictement F_i et $\forall E_j$ telle que E_j soit une strate de priorité inférieure à E_i, on a $F_j = G_j$.*

Il n'est pas intéressant d'appliquer cet ordre à l'exemple précédent. En effet, les sous-bases préférées sont ici minimales pour l'inclusion. Cet ordre est surtout destiné à être utilisé dans des problèmes de type abductif (recherche d'explication, diagnostic).

[11] Ici aussi, le terme provient de travaux sur le choix social.

Conclusions sur les thèses préférées :

1. Si le pré-ordre $<$ est partiel sur E alors, l'ordre démocratique issu de $<$ ne permet pas d'ordonner toutes les thèses.
2. Les éléments maximaux obtenus avec l'ordre démocratique sont les thèses démo-préférées et, dans le cas d'un pré-ordre total $<$ sur E, elles correspondent exactement aux sous-théories préférées.
3. L'ordre démocratique généralise les travaux de Brewka car :
   - il permet d'utiliser une base pré-ordonnée partiellement (donc non stratifiée) ;
   - dans le cas d'une base stratifiée, il ne se contente pas de fournir les sous-théories préférées mais il permet d'ordonner toutes les thèses ;
   - dans le cas d'une base avec une strate unique, on retrouve les mêmes résultats qu'avec la méthode de Brewka : toutes les thèses sont préférées.

2.2.1.4 Les sous-bases fortement maximales consistantes

À toute base initiale $E = E_1 \cup \ldots \cup E_n$ stratifiée, on peut associer une base de croyances possibiliste B et vice-versa ; une base possibiliste est une base composée de couples (Φ, α) avec Φ une formule et α appelé le poids de cette formule et représentant la borne inférieure de la mesure de nécessité associée à la formule (voir [DLP91]).

En utilisant les sous-bases maximales consistantes de B, notées $B_1 \ldots B_p$, Dubois, Lang et Prade définissent la notion suivante :

Définition 2.2.8 *Soit K un ensemble de formules possibiliste, on définit $Inc(K)$ par :*

- *soit il existe un j tel que :*
  - $\cup_{i=1\ldots(j-1)}(\phi, \alpha_i)$ *est consistant,*
  - $\cup_{i=1\ldots j}(\phi, \alpha_i)$ *est inconsistant.*

  alors $Inc(K) = \alpha_j$,
- *soit $Inc(K) = 0$.*

On dit que $Inc(K)$ est le degré d'inconsistance de K.

$Inc(K)$ peut être calculé en utilisant le principe de résolution possibiliste (voir [DLP91]). D'autre part, on a une propriété importante : K est consistante ssi $Inc(K) = 0$.

Définition 2.2.9 *B_i est une sous-base fortement maximale consistante de B ssi B_i est consistante et $\forall(\Phi, \alpha) \in B \setminus B_i$, $Inc(B_i \cup \{(\Phi, \alpha)\}) = \alpha$.*

Il a été démontré que les sous-bases fortement maximales consistantes ainsi calculées sur B sont équivalentes aux thèses démo-préférées obtenues sur la base E stratifiée adéquate associée à B (voir dans [BDLP92]).

Exemple 1 : Soit B une base propositionnelle possibiliste correspondant à la stratification numéro 1 de l'exemple présenté en tête de la section 2.2 page 7 :

$$
\begin{array}{l}
B = \{(\rightarrow P, 1) \\
(P \rightarrow O, 0.9) \\
(P \rightarrow \neg V, 0.8) \\
(O \rightarrow V, 0.5) \\
(O \rightarrow A, 0.3) \\
(A \rightarrow V, 0.1)\}
\end{array}
$$

Les 5 sous-bases suivantes sont les versions possibilistes des thèses données en section 2.2.1.1 page 8 correspondant à la base possibiliste précédente :

$B_1 =$	$B_2 =$	$B_3 =$	$B_4 =$	$B_5 =$
$\{(\rightarrow P,\, 1)$	$\{(\rightarrow P,\, 1)$	$\{(\rightarrow P,\, 1)$	$\{(\rightarrow P,\, 1)$	$\{(P \rightarrow O,\, 0.9)$
$(P \rightarrow O,\, 0.9)$	$(P \rightarrow O,\, 0.9)$	$(P \rightarrow O,\, 0.9)$	$(P \rightarrow \neg V,\, 0.8)$	$(P \rightarrow \neg V,\, 0.8)$
$(P \rightarrow \neg V,\, 0.8)$	$(P \rightarrow \neg V,\, 0.8)$	$(O \rightarrow V,\, 0.5)$	$(O \rightarrow V,\, 0.5)$	$(O \rightarrow V,\, 0.5)$
$(O \rightarrow A,\, 0.3)\}$	$(A \rightarrow V,\, 0.1)\}$	$(O \rightarrow A,\, 0.3)$	$(O \rightarrow A,\, 0.3)$	$(O \rightarrow A,\, 0.3)$
		$(A \rightarrow V,\, 0.1)\}$	$(A \rightarrow V,\, 0.1)\}$	$(A \rightarrow V,\, 0.1)\}$

Une seule sous-base est fortement maximale consistante : B_1.

Exemple 2 : Soit B une base propositionnelle possibiliste correspondant à la stratification numéro 2 de l'exemple présenté en tête de la section 2.2 page 7 :

$$
\begin{array}{l}
B = \{(\rightarrow P,\, 1) \\
(P \rightarrow O,\, 1) \\
(P \rightarrow \neg V,\, 0.8) \\
(O \rightarrow V,\, 0.8) \\
(O \rightarrow A,\, 0.3) \\
(A \rightarrow V,\, 0.3)\}
\end{array}
$$

Les versions possibilistes de chaque thèse sont les suivantes :

$B_1 =$	$B_2 =$	$B_3 =$	$B_4 =$	$B_5 =$
$\{(\rightarrow P,\, 1)$	$\{(\rightarrow P,\, 1)$	$\{(\rightarrow P,\, 1)$	$\{(\rightarrow P,\, 1)$	$\{(P \rightarrow O,\, 1)$
$(P \rightarrow O,\, 1)$	$(P \rightarrow O,\, 1)$	$(P \rightarrow O,\, 1)$	$(P \rightarrow \neg V,\, 0.8)$	$(P \rightarrow \neg V,\, 0.8)$
$(P \rightarrow \neg V,\, 0.8)$	$(P \rightarrow \neg V,\, 0.8)$	$(O \rightarrow V,\, 0.8)$	$(O \rightarrow V,\, 0.8)$	$(O \rightarrow V,\, 0.8)$
$(O \rightarrow A,\, 0.3)\}$	$(A \rightarrow V,\, 0.3)\}$	$(O \rightarrow A,\, 0.3)$	$(O \rightarrow A,\, 0.3)$	$(O \rightarrow A,\, 0.3)$
		$(A \rightarrow V,\, 0.3)\}$	$(A \rightarrow V,\, 0.3)\}$	$(A \rightarrow V,\, 0.3)\}$

Trois sous-bases sont fortement maximales consistantes : B_1, B_2 et B_3.

Exemple 3 : Soit B une base propositionnelle possibiliste correspondant à la stratification numéro 3 de l'exemple présenté en tête de la section 2.2 page 7 :

$$
\begin{array}{l}
B = \{(\rightarrow P,\, 0,\, 7) \\
(P \rightarrow O,\, 0.7) \\
(P \rightarrow \neg V,\, 0.7) \\
(O \rightarrow V,\, 0.7) \\
(O \rightarrow A,\, 0.7) \\
(A \rightarrow V,\, 0.7)\}
\end{array}
$$

Les versions possibilistes de chaque thèse sont les suivantes :

$B_1 =$	$B_2 =$	$B_3 =$	$B_4 =$	$B_5 =$
$\{(\rightarrow P,\, 0.7)$	$\{(\rightarrow P,\, 0.7)$	$\{(\rightarrow P,\, 0.7)$	$\{(\rightarrow P,\, 0.7)$	$\{(P \rightarrow O,\, 0.7)$
$(P \rightarrow O,\, 0.7)$	$(P \rightarrow O,\, 0.7)$	$(P \rightarrow O,\, 0.7)$	$(P \rightarrow \neg V,\, 0.7)$	$(P \rightarrow \neg V,\, 0.7)$
$(P \rightarrow \neg V,\, 0.7)$	$(P \rightarrow \neg V,\, 0.7)$	$(O \rightarrow V,\, 0.7)$	$(O \rightarrow V,\, 0.7)$	$(O \rightarrow V,\, 0.7)$
$(O \rightarrow A,\, 0.7)\}$	$(A \rightarrow V,\, 0.7)\}$	$(O \rightarrow A,\, 0.7)$	$(O \rightarrow A,\, 0.7)$	$(O \rightarrow A,\, 0.7)$
		$(A \rightarrow V,\, 0.7)\}$	$(A \rightarrow V,\, 0.7)\}$	$(A \rightarrow V,\, 0.7)\}$

$B_1, \ldots, B_5$ sont toutes fortement maximales consistantes.

Conclusions sur les sous-bases fortement maximales consistantes :

1. Les sous-bases fortement maximales consistantes de Dubois, Lang, Prade correspondent aux thèses démo-préférées.
2. Dubois, Lang, Prade ont aussi défini un ordre entre les sous-bases fortement maximales consistantes appelé ordre lexicographique décrit dans la section 2.2.2.
3. Je peux donc dire, ici aussi, que l'ordre démocratique permet de généraliser le concept de sous-base fortement maximale consistante de [DLP91] par le fait qu'il traite les ensembles partiellement ordonnés.
4. Je retrouve toujours le même type de résultat dans le cas d'une base avec une seule strate.

2.2.1.5 Conclusion sur les méthodes "Inclusion Based Preference"

Toutes les méthodes vues dans cette section sont regroupées sous le terme "Inclusion Based Preference" car elles privilégient parmi les sous-bases consistantes celles qui sont maximales pour l'inclusion, nommées ici des thèses et permettent :

- soit de sélectionner parmi ces thèses la ou les thèses préférées,
- soit d'ordonner partiellement ou totalement les thèses.

Elles fournissent les mêmes résultats, même si les motivations sont au départ différentes.

Le mécanisme de sélection associé aux thèses préférées par les méthodes "Inclusion Based Preference" sera noté Incl.

2.2.2 Les sous-bases consistantes et l'ordre lexicographique

Ici, il ne s'agit pas seulement de travailler à partir de sous-bases consistantes maximales pour l'inclusion, mais plutôt de "raffiner" les mécanismes vus précédemment en ne sélectionnant que certaines de ces sous-bases. Pour cela, on sélectionne des sous-bases consistantes maximales pour la cardinalité. C'est la raison pour laquelle cette méthode, bien que citée dans le cadre des sous-bases fortement maximales consistantes de Dubois, Lang et Prade (voir [DLP91]), n'a pas été présentée avec les méthodes "Inclusion Based Preference".

La définition de base (ordre basé sur la cardinalité) correspondant au cas d'une base de croyances non ordonnée est (voir [DLP91]) :

Définition 2.2.10 *Soit E une base initiale finie, soit A un sous-ensemble de E, on dit que A est une sous-base consistante maximale pour la cardinalité ssi A est consistante et $\forall B$ sous-ensemble de E tel que B soit consistant alors $|B| \leq |A|$*[12].

Lorsqu'on se trouve en présence d'une base de croyances totalement pré-ordonnée on définit :

Définition 2.2.11 *Soit E une base stratifiée finie ($E = E_1 \cup \ldots \cup E_n$), soient A et B deux sous-bases consistantes de E, on définit l'ordre lexicographique de la manière suivante : $A \ll^{lex} B$ ssi $\exists i$ tel que $|A_i| < |B_i|$ et pour tout $j < i$ on a $|A_j| = |B_j|$.*

En fait, comme François Lévy le signale dans une communication personnelle, le terme "lexicographique" est mal choisi. En effet, il désigne ici un ordre particulier basé sur la cardinalité appliqué aux sous-bases issues d'une base stratifiée et prenant en compte l'ordre entre les formules de la base. Or, plus généralement, un ordre lexicographique peut se définir de la façon suivante :

[12] $|A|$ dénote le cardinal de A.

Définition 2.2.12 *Soit E un ensemble muni d'une relation d'ordre strict $<$. Soit S l'ensemble des suites d'éléments de E. L'ordre lexicographique sur S est défini par : $\forall\ s = \{e_0, \ldots, e_{n_s}\}$ et $\forall$ $s' = \{e'_0, \ldots, e'_{n_{s'}}\}$ deux éléments de S, $s <^{lex} s'$ si et seulement si $\exists k$ tel que :*

1. $\forall i < k,\ e_i \equiv e'_i$,
2. $e_k < e'_k$.

Cette définition peut aussi être étendue au cas d'un pré-ordre $<$ sur E.

À la lumière de cette définition, on constate alors que tous les ordres étudiés ici, qui portent sur les sous-bases issues d'une base stratifiée et qui prennent en compte l'ordre entre les formules de la base, sont des ordres lexicographiques. Cette approche est intéressante en particulier parce qu'elle permet d'expliciter mathématiquement les liens existants entre ces différents ordres. On peut ainsi retrouver avec une nouvelle méthode certains des résultats qui vont être présentés ou rappelés ici. Cette voie constitue donc une perspective à explorer, mais sur laquelle je ne m'attarderai pas.

En conséquence, dans toute la suite de ce document, le terme "lexicographique", que je conserve en dépit de son ambiguïté, dénotera l'ordre particulier défini par [DLP91, Leh92]. En effet, il ne me paraît pas nécessaire de multiplier les termes "techniques", et, d'autre part, sachant que ce travail est une synthèse d'autres travaux, il est plus aisé de conserver la terminologie des auteurs.

Définition 2.2.13 *(caractérisation) Soit E une base stratifiée finie ($E = E_1 \cup \ldots \cup E_n$), soit S un sous-ensemble de E, S est une sous-base préférée pour l'ordre lexicographique ssi $\forall k = 1 \ldots n$, $(S_1 \cup \ldots \cup S_k)$ est un sous-ensemble consistant maximal pour la cardinalité de $(E_1 \cup \ldots \cup E_k)$.*

Propriété 2.2.2 *(démonstration dans [BCD$^+$ 93]) L'ordre lexicographique raffine l'"Inclusion Based Preference" : toute sous-base de E se trouvant être élément maximal pour l'ordre $\ll^{lex}$ sera nécessairement élément maximal pour l'ordre "Inclusion Based".*

Voyons ce que cela donne sur un exemple.

Exemple 1 : Soit E la base propositionnelle stratifiée correspondant à la stratification numéro 1 de l'exemple présenté en tête de la section 2.2 page 7. Il n'est pas raisonnable de calculer toutes les sous-bases consistantes. Grâce à la propriété 2.2.2 et sachant que, $\forall S$ une sous-base consistante qui n'est pas une thèse, il existera toujours une thèse T telle que $S \ll^{lex} T$, je peux donc me contenter de partir des sous-bases maximales consistantes, c'est-à-dire des thèses. Ensuite, j'applique l'ordre lexicographique et j'obtiens :

$$\text{thèse } 5 \ll^{lex} \text{thèse } 4 \ll^{lex} \text{thèse } 3 \ll^{lex} \text{thèse } 2 \ll^{lex} \text{thèse } 1$$

Remarque : je trouve ici le même ordre que celui obtenu par la préférence démocratique.

Exemple 2 : Soit E la base propositionnelle stratifiée correspondant à la stratification numéro 2 de l'exemple présenté en tête de la section 2.2 page 7. Dans ce cas là, on trouve les résultats suivants[13]:

$$\text{thèse } 5 \equiv^{lex} \text{thèse } 4 \ll^{lex} \text{thèse } 2 \equiv^{lex} \text{thèse } 1 \ll^{lex} \text{thèse } 3$$

(contrairement aux résultats obtenus avec l'ordre démocratique sur les thèses, on a ici comme élément maximal pour l'ordre lexicographique la thèse 3 tandis que les thèses 1 et 2 sont équivalentes ainsi que les thèses 4 et 5)

[13]Le symbole $\equiv^{lex}$ représente la relation d'équivalence issue de l'ordre $\ll^{lex}$: soient A et B, 2 sous-bases dune base E stratifiée, $A \equiv^{lex} B \Leftrightarrow (\forall i = 1 \ldots n,\ |A_i| = |B_i|$, avec n = nombre de strates de E).

Exemple 3 : Soit E la base propositionnelle stratifiée correspondant à la stratification numéro 3 de l'exemple présenté en tête de la section 2.2 page 7. En appliquant l'ordre lexicographique, j'obtiens que :

$$\text{thèse } 2 \equiv^{lex} \text{thèse } 1 \ll^{lex} \text{thèse } 5 \equiv^{lex} \text{thèse } 4 \equiv^{lex} \text{thèse } 3$$

Conclusions sur les sous-bases consistantes et l'ordre lexicographique :

1. Bien que l'ordre lexicographique soit défini sur des sous-bases consistantes, les éléments maximaux de cet ordre sont pris parmi les thèses (voir propriété 2.2.2 page précédente) ; d'où l'idée d'appliquer cet ordre uniquement sur les thèses afin de limiter les calculs[14].
2. Dans le cas d'une base stratifiée avec des strates ne contenant qu'une seule formule, l'ordre lexicographique appliqué aux thèses fournit des résultats identiques à ceux obtenus par toutes les méthodes précédentes.
3. Dans le cas d'une base stratifiée avec des strates pouvant contenir plusieurs formules et en particulier dans le cas d'une base avec une strate unique, l'ordre lexicographique appliqué aux thèses fournit des résultats différents de ceux obtenus par les méthodes précédentes. C'est son principal avantage, puisqu'il y a *au plus* autant de thèses préférées lexicographiquement que de thèses préférées au sens "Inclusion Based".
4. Quand l'ordre lexicographique est appliqué à une base ne contenant qu'une seule strate, les sous-bases préférées sont celles de cardinal maximal.
5. L'ordre lexicographique permet de pré-ordonner de manière totale les thèses, contrairement à l'ordre démocratique qui ne fournit qu'un ordre partiel sur les thèses.
6. Lehmann dans [Leh92] a défini un ordre semblable.

Le mécanisme de sélection associé aux sous-bases préférées lexicographiquement sera noté Lex, et le cas particulier qui consiste à appliquer l'ordre lexicographique à une base à strate unique sera noté Car.

2.2.3 Les extensions de la logique des défauts

Nous sortons ici du cadre des approches purement "syntaxiques" pour aborder un autre type d'approches : les logiques non-monotones. Les travaux de Reiter (voir par exemple [Rei80]) ont consisté à définir une nouvelle logique avec de nouvelles règles d'inférence prenant en compte des règles générales pouvant comporter des défauts et ainsi permettant de gérer une inconsistance de la base de croyances.

Je vais toutefois citer quelques-uns des résultats obtenus par Reiter puisque ces résultats, dans certains cas très particuliers, sont équivalents à ceux obtenus par l'utilisation de méthodes "Inclusion Based Preference" (voir [Bre89b]).

La base de croyances E est ici partitionnée en :

- W = ensemble de formules fermées propositionnelles ou du premier ordre (prémisses),
- D = ensemble de défauts c'est-à-dire de formules de la forme $a : b/c$ avec a[15], b[16], c[17] des formules propositionnelles ou fermées du premier ordre,

[14] Il est bien sûr évident que cette idée est applicable dans le cadre du petit exemple présenté. Toutefois, dans le cas général, il vaut mieux chercher une méthode constructive ne nécessitant ni le calcul des sous-bases consistantes, ni celui des thèses.

[15] a est appelé le pré-requis.

[16] b est appelé la justification.

[17] c est appelé le conséquent.

Définition 2.2.14 *L'interprétation d'un défaut $a : b/c$ est : "si a est connu et b consistant avec tout ce qui est connu alors inférer c". Le défaut $a : b/c$ est donc une nouvelle règle d'inférence*[18].

Définition 2.2.15 *(définition informelle) Les extensions sont des ensembles de croyances contenant W et fermés pour les 2 types d'inférence : l'inférence classique et l'inférence des défauts.*

Dans [Rei80] on trouve un processus de construction des extensions qui repose sur la propriété suivante :

Propriété 2.2.3 *(voir [Rei80]) Soit $E = (W, D)$ une théorie des défauts. On part de $\mathcal{E}_0 = W$, et on définit récursivement :*

$\forall i \geq 0$, $\mathcal{E}_{i+1} = C_n(\mathcal{E}_i) \cup \{w \text{ tel que } u : v_1 \ldots v_n/w \in D, u \in \mathcal{E}_i, \neg v_1 \notin \mathcal{F}, \ldots, \neg v_n \notin \mathcal{F}\}$[19]

alors $\mathcal{F}$ est une extension de E ssi $\mathcal{F} = \cup_{i \geq 0} \mathcal{E}_i$

La procédure consiste donc à construire $\mathcal{E}_{i+1}$ à partir de $\mathcal{E}_i$ avec un arbre binaire de racine $\mathcal{E}_i$. Chaque étage de cet arbre correspond à l'applicabilité d'un défaut sur $\mathcal{E}_i$ (donc la profondeur de cet arbre est égale au nombre de défauts applicables sur $\mathcal{E}_i$[20]). À chaque nœud de chaque étage, on crée deux branches :

- celle où le défaut correspondant à l'étage est appliqué, donc on prend ici comme hypothèse qu'aucune des négations des justifications n'appartiendra à l'extension ;
- celle où le défaut correspondant à l'étage n'est pas appliqué, donc on prend ici comme hypothèse qu'au moins une des négations des justifications appartiendra à l'extension.

Les feuilles de cet arbre sont des candidats au statut de $\mathcal{E}_{i+1}$. Il suffit alors de vérifier que les hypothèses prises dans la branche menant à cette feuille sont cohérentes avec la feuille. Remarquons qu'il existe des branches menant à des impasses (par exemple, si les hypothèses prises dans la branche consistent à avoir à la fois une formule et sa négation), de telles branches n'étant bien-sûr plus à développer.

Il existe de nombreux cas particuliers de la logique des défauts, dont un qui me sera très utile :

Définition 2.2.16 *Les défauts de la forme $: b_i/b_i$ sont appelés défauts super-normaux, et si $D = \{ : b_i/b_i\}$ on dit que la théorie (W, D) est une théorie des défauts super-normale.*

Dans ce cas-là, on a les propriétés suivantes :

Propriété 2.2.4 *(démonstration dans [Poo88, Bre89b]) Soit la théorie des défauts super-normale (W, D) avec $W = \varnothing$ et $D = \{ : b_i/b_i\}$, alors $\forall \mathcal{F}_j$ une extension de (W, D), $\exists T_j$ une thèse de la base $B = \{bi \text{ tel que } : b_i/b_i \in D\}$ telle que $\mathcal{F}_j = C_n(T_j)$*[21].

Propriété 2.2.5 *(démonstration dans [Poo88, Bre89b]) Soit $B = \{b_1, \ldots, b_n\}$ la base de croyances, alors $\forall T_j$ une thèse de la base, $\exists \mathcal{F}_j$ une extension de la théorie des défauts super-normale (W, D) avec $W = \varnothing$ et $D = \{ : b_i/b_i \text{ tel que } b_i \in B\}$ telle que $\mathcal{F}_j = C_n(T_j)$.*

[18] Les définitions données ici sont volontairement simplifiées. Pour plus de précision, se reporter à [Rei80].
[19] Remarquons la place occupée par $\mathcal{F}$ dans cette définition.
[20] Un défaut est applicable sur un ensemble si et seulement si son pré-requis appartient à l'ensemble.
[21] Soit un ensemble A de formules, $C_n(A)$ = ensemble des conséquences logiques de A.

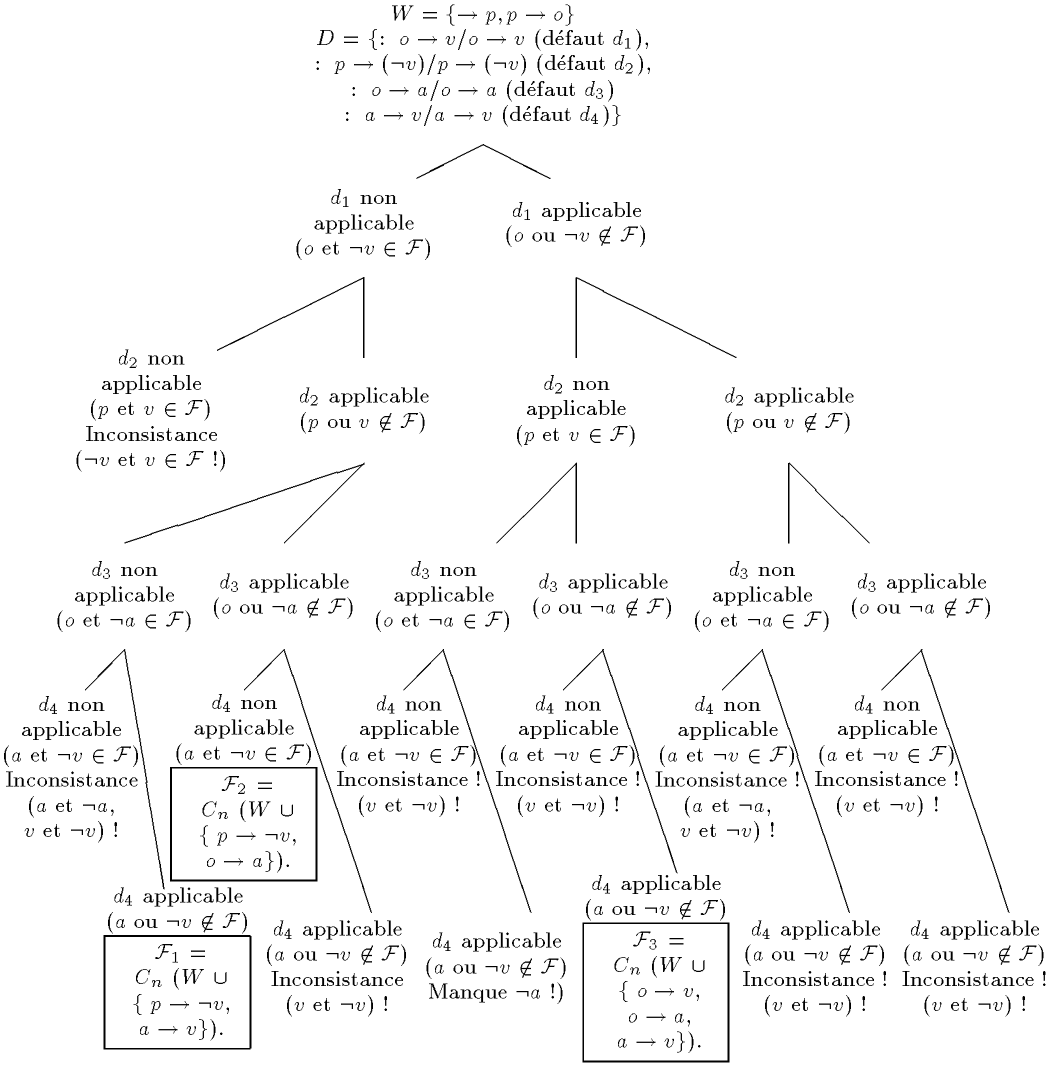


Figure 2.2 : Mécanisme de sélection E : Arbre des extensions

Exemple : Contrairement aux sections précédentes, je n'ai pas 3 exemples à traiter, puisque la base initiale E n'est pas ordonnée. Je choisis de traduire mon exemple des oiseaux avec des défauts normaux sans pré-requis afin de voir les points communs avec les méthodes présentées précédemment (voir [Rei80]) :

$E = (W, D)$ avec :

- $W = \{ \rightarrow P,\ P \rightarrow O\}$,
- $D = \{ :O \rightarrow V/O \rightarrow V,\ :P \rightarrow \neg V/P \rightarrow \neg V,\ :O \rightarrow A/O \rightarrow A,\ :A \rightarrow V/A \rightarrow V\}$

Je construis alors l'arbre de calcul des extensions et trouve 3 extensions possibles $\mathcal{F}_1$, $\mathcal{F}_2$ et $\mathcal{F}_3$ (voir la figure 2.2).

Remarque : ces trois extensions correspondent bien aux thèses préférées obtenues par les méthodes "Inclusion Based Preference" lorsque les formules $(P \rightarrow O)$ et $(\rightarrow P)$ constituent la strate de plus forte priorité dans la base initiale E (voir note sur ce sujet dans la conclusion sur les sous-théories préférées de la section 2.2.1.2 page 9).

Conclusions sur les extensions de la logique des défauts :

1. Chez Reiter et dans le cas d'une théorie (W, D) super-normale les extensions calculées correspondent aux thèses démo-préférées obtenues sur une base initiale comportant 2 strates, la plus prioritaire pour les formules de W et la suivante pour les formules b_i correspondant aux défauts, quelle que soit la méthode utilisée pour les calculer.

2. Reiter n'a pas prévu d'ordre sur ses extensions. Poole, en introduisant les défauts nommés, s'est donné la possibilité d'ordonner la base initiale mais n'a rien défini au niveau extension (voir [Poo88]).

Ce mécanisme de sélection sera noté E pour le cas des défauts généraux et E* pour le cas des défauts super-normaux (attention à ne pas confondre avec E représentant en général la base de croyances). On distingue E de E*, bien qu'il s'agisse initialement du même mécanisme, afin de pouvoir faire la différence dans le chapitre 4 page 73 où seront présentés des résultats concernant soit l'un, soit l'autre.

2.2.4 Les sous-bases consistantes et l'ordre "Best-Out"

L'approche présentée ici, bien qu'appartenant à la famille des approches "syntaxiques", permet d'établir un lien avec des approches plus sémantiques telles que les travaux sur le système Z [Pea90] et sur la fermeture rationnelle [LM92].

Ici, l'idée est de repérer la strate la plus prioritaire amenant une inconsistance, et de considérer alors que les seules formules intéressantes à prendre en compte sont celles de priorité supérieure à la strate fatidique (l'absence ou la présence des autres formules n'ayant dès lors aucune importance tant que la consistance de la sous-base obtenue est vérifiée).

L'ordre "Best-Out" décrit dans [BCD+93] et [DLP91] est défini, comme pour l'ordre lexicographique, à partir de sous-bases consistantes et non à partir des thèses.

Soit une base E initiale stratifiée ($E = E_1 \cup \ldots \cup E_n$), on définit :

Définition 2.2.17 *Soit A un sous-ensemble de E ($A = A_1 \cup \ldots \cup A_n$), on note $a(A)$ la plus haute priorité d'une formule de E qui n'est pas dans A ; $a(A) = min\{i \text{ tel que } \exists \Phi \in E_i \setminus A\}$.*

Définition 2.2.18 *Soient A et B, 2 sous-bases consistantes de E, on définit l'ordre "Best-Out" de la manière suivante : $A \ll^{bo} B$ ssi $a(A) \leq a(B)$ (on dit alors que B est préférée à A pour l'ordre $\ll^{bo}$).*

Attention, contrairement à ce qui se produit pour l'ordre lexicographique, les éléments maximaux de l'ordre "Best-Out" ne sont pas nécessairement des éléments maximaux pour l'inclusion ensembliste. On ne peut donc pas se restreindre au calcul des thèses. Il faut calculer toutes les sous-bases consistantes.

Exemple 1 : Soit E la base propositionnelle stratifiée correspondant à la stratification numéro 1 de l'exemple présenté en tête de la section 2.2 page 7.

Calculons les sous-bases consistantes. Il y a $2^6 - 2$ possibilités à explorer (on ne tient pas compte de la base vide, ni de la base initiale), ce qui fait au plus 62 sous-bases consistantes qu'il faudrait comparer 2 à 2 pour pouvoir les ordonner. En fait, si on cherche à calculer les sous-ensembles consistants A de E en fonction des différents $a(A)$, on trouve 58 solutions classées suivant la valeur $a(A)$ (voir table 2.1 page suivante).

Remarquons que les sous-bases solutions correspondant aux thèses sont les solutions 1, 2, 4, 12, 28.

Avec l'ordre "Best-Out", on trouve donc :

<table>
<tr>
<td>les A consistants tels que $a(A) = 6$: pas de A car l'ensemble $\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V, O \rightarrow V, O \rightarrow A\}$ est inconsistant</td>
<td rowspan="4">les A consistants tels que $a(A) = 2$: 16 possibilités toutes consistantes donc 16 solutions :

$\{\rightarrow P, P \rightarrow \neg V, O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 12
$\{\rightarrow P, P \rightarrow \neg V, O \rightarrow V, O \rightarrow A\}$: sol. 13
$\{\rightarrow P, P \rightarrow \neg V, O \rightarrow V, A \rightarrow V\}$: sol. 14
$\{\rightarrow P, P \rightarrow \neg V, O \rightarrow V\}$: sol. 15
$\{\rightarrow P, P \rightarrow \neg V, O \rightarrow A, A \rightarrow V\}$: sol. 16

$\{\rightarrow P, P \rightarrow \neg V, O \rightarrow A\}$: sol. 17
$\{\rightarrow P, P \rightarrow \neg V, A \rightarrow V\}$: sol. 18
$\{\rightarrow P, P \rightarrow \neg V\}$: sol. 19
$\{\rightarrow P, O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 20
$\{\rightarrow P, O \rightarrow V, O \rightarrow A\}$: sol. 21
$\{\rightarrow P, O \rightarrow V, A \rightarrow V\}$: sol. 22
$\{\rightarrow P, O \rightarrow V\}$: sol. 23
$\{\rightarrow P, O \rightarrow A, A \rightarrow V\}$: sol. 24
$\{\rightarrow P, O \rightarrow A\}$: sol. 25
$\{\rightarrow P, A \rightarrow V\}$: sol. 26
$\{\rightarrow P\}$: sol. 27</td>
<td rowspan="4">les A consistants tels que $a(A) = 1$: 32 possibilités toutes consistantes sauf la base vide donc 31 solutions :

$\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 28
$\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow V, O \rightarrow A\}$: sol. 29
$\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow V, A \rightarrow V\}$: sol. 30
$\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow V\}$: sol. 31
$\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow A, A \rightarrow V\}$: sol. 32
$\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow A\}$: sol. 33
$\{P \rightarrow O, P \rightarrow \neg V, A \rightarrow V\}$: sol. 34
$\{P \rightarrow O, P \rightarrow \neg V\}$: sol. 35
$\{P \rightarrow O, O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 36
$\{P \rightarrow O, O \rightarrow V, O \rightarrow A\}$: sol. 37
$\{P \rightarrow O, O \rightarrow V, A \rightarrow V\}$: sol. 38
$\{P \rightarrow O, O \rightarrow V\}$: sol. 39
$\{P \rightarrow O, O \rightarrow A, A \rightarrow V\}$: sol. 40
$\{P \rightarrow O, O \rightarrow A\}$: sol. 41
$\{P \rightarrow O, A \rightarrow V\}$: sol. 42
$\{P \rightarrow O\}$: sol. 43
$\{P \rightarrow \neg V, O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 44
$\{P \rightarrow \neg V, O \rightarrow V, O \rightarrow A\}$: sol. 45
$\{P \rightarrow \neg V, O \rightarrow V\}$: sol. 46
$\{P \rightarrow \neg V, O \rightarrow V, A \rightarrow V\}$: sol. 47
$\{P \rightarrow \neg V, O \rightarrow A, A \rightarrow V\}$: sol. 48
$\{P \rightarrow \neg V, O \rightarrow A\}$: sol. 49
$\{P \rightarrow \neg V, A \rightarrow V\}$: sol. 50
$\{P \rightarrow \neg V\}$: sol. 51
$\{O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 52
$\{O \rightarrow V, O \rightarrow A, \}$: sol. 53
$\{O \rightarrow V, A \rightarrow V\}$: sol. 54
$\{O \rightarrow V\}$: sol. 55
$\{O \rightarrow A, A \rightarrow V\}$: sol. 56
$\{O \rightarrow A\}$: sol. 57
$\{A \rightarrow V\}$: sol. 58</td>
</tr>
<tr>
<td>les A consistants tels que $a(A) = 5$: pas de A car l'ensemble $\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V, O \rightarrow V\}$ est inconsistant</td>
</tr>
<tr>
<td>les A consistants tels que $a(A) = 4$: 4 possibilités dont une inconsistante donc 3 solutions :
$\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V, O \rightarrow A, A \rightarrow V\}$: inconsistante
$\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V, O \rightarrow A\}$: sol. 1
$\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V, A \rightarrow V\}$: sol. 2
$\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V\}$: sol. 3</td>
</tr>
<tr>
<td>les A consistants tels que $a(A) = 3$: 8 possibilités toutes consistantes donc 8 solutions :
$\{\rightarrow P, P \rightarrow O, O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 4
$\{\rightarrow P, P \rightarrow O, O \rightarrow V, O \rightarrow A\}$: sol. 5
$\{\rightarrow P, P \rightarrow O, O \rightarrow V, A \rightarrow V\}$: sol. 6
$\{\rightarrow P, P \rightarrow O, O \rightarrow V\}$: sol. 7
$\{\rightarrow P, P \rightarrow O, O \rightarrow A, A \rightarrow V\}$: sol. 8
$\{\rightarrow P, P \rightarrow O, O \rightarrow A\}$: sol. 9
$\{\rightarrow P, P \rightarrow O, A \rightarrow V\}$: sol. 10
$\{\rightarrow P, P \rightarrow O\}$: sol. 11</td>
</tr>
</table>

Tableau 2.1 : L'ordre “Best-Out” (exemple 1)

$$\text{solutions 28 à 58} \ll^{bo} \text{solutions 12 à 27} \ll^{bo} \text{solutions 4 à 11} \ll^{bo} \text{solutions 1 à 3}$$

(avec les solutions 1 à 3 équivalentes, de même que les solutions 4 à 11, 12 à 27, 28 à 58)

Si on compare avec les résultats obtenus sur les thèses par d'autres méthodes, on obtient des résultats différents, puisqu'ici les 2 thèses (solution 1 et solution 2) sont équivalentes, alors qu'avec les autres méthodes, la solution 1 était préférée à la solution 2.

Exemple 2 : Soit E la base propositionnelle stratifiée correspondant à la stratification numéro 2 de l'exemple présenté en tête de la section 2.2 page 7.

On trouve bien sûr le même nombre de sous-bases consistantes mais elles ne sont pas classées de la même façon (voir table 2.2 page suivante).

Ici aussi, on se rend compte que les résultats obtenus sont différents de ceux obtenus par les autres méthodes :

$$\text{solutions 12 à 58} \ll^{bo} \text{solutions 1 à 11}$$

(avec les solutions 1 à 11 équivalentes, ainsi que les solutions 12 à 58)

Remarque : Si on regarde l'effet de l'ordre "Best-Out" sur les thèses, on retrouve que les 3 premières thèses sont de même importance ; par contre, c'est aussi le cas pour les 2 dernières, ce qui n'apparaissait pas dans les méthodes "Inclusion Based".

Exemple 3 : Soit E la base propositionnelle stratifiée correspondant à la stratification numéro 3 de l'exemple présenté en tête de la section 2.2 page 7.

On retrouve bien sûr toujours les mêmes 58 sous-bases consistantes, mais cette fois elles sont toutes préférées (voir table 2.3 page 23).

Avec l'ordre "Best-Out", on trouve donc que les solutions 1 à 58 sont équivalentes.

Conclusions sur les sous-bases consistantes et l'ordre "Best-Out" :

1. L'ordre "Best-Out" est défini sur les sous-bases consistantes et pas sur les sous-bases maximales consistantes.
2. Contrairement à l'ordre lexicographique, on ne peut pas se contenter de l'appliquer uniquement sur les thèses, puisque parmi toutes les sous-bases consistantes, les éléments maximaux de l'ordre "Best-Out" ne sont pas nécessairement des thèses.
3. Il s'agit d'une préférence très peu sélective.
4. Les résultats obtenus avec cet ordre sont tout à fait différents de ceux obtenus par les méthodes vues précédemment, excepté dans le cas d'une base avec une strate unique (voir point suivant).
5. Dans le cas d'une base avec une strate unique, toutes les sous-bases consistantes sont préférées, donc en particulier les thèses sont toutes préférées.
6. L'ordre "Best-Out" permet de pré-ordonner totalement l'ensemble des sous-bases consistantes.
7. Les relations d'inférence issues de cet ordre présentent un inconvénient majeur : le "drowning effect" (l'effet de noyade), c'est-à-dire l'incapacité à inférer une conclusion dépendant d'une formule située en dessous du seuil d'inconsistance (voir dans [BCD+93] la manière de calculer ce seuil).

les A consistants tels que $a(A) = 3$: pas de A car l'ensemble $\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V, O \rightarrow V\}$ est inconsistant	les A consistants tels que $a(A) = 2$: 12 possibilités dont une inconsistante donc 11 solutions : $\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V, O \rightarrow A, A \rightarrow V\}$: inconsistante $\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V, O \rightarrow A\}$: sol. 1 $\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V, A \rightarrow V\}$: sol. 2 $\{\rightarrow P, P \rightarrow O, P \rightarrow \neg V\}$: sol. 3 $\{\rightarrow P, P \rightarrow O, O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 4 $\{\rightarrow P, P \rightarrow O, O \rightarrow V, O \rightarrow A\}$: sol. 5 $\{\rightarrow P, P \rightarrow O, O \rightarrow V, A \rightarrow V\}$: sol. 6 $\{\rightarrow P, P \rightarrow O, O \rightarrow V\}$: sol. 7 $\{\rightarrow P, P \rightarrow O, O \rightarrow A, A \rightarrow V\}$: sol. 8 $\{\rightarrow P, P \rightarrow O, O \rightarrow A\}$: sol. 9 $\{\rightarrow P, P \rightarrow O, A \rightarrow V\}$: sol. 10 $\{\rightarrow P, P \rightarrow O\}$: sol. 11	les A consistants tels que $a(A) = 1$: 48 possibilités toutes consistantes sauf la base vide donc 47 solutions : $\{\rightarrow P, P \rightarrow \neg V, O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 12 $\{\rightarrow P, P \rightarrow \neg V, O \rightarrow V,$ O→ A} : sol. 13 $\{\rightarrow P, P \rightarrow \neg V, O \rightarrow V,$ A→ V} : sol. 14 $\{\rightarrow P, P \rightarrow \neg V, O \rightarrow V\}$: sol. 15 $\{\rightarrow P, P \rightarrow \neg V, O \rightarrow A,$ A→ V} : sol. 16 $\{\rightarrow P, P \rightarrow \neg V, O \rightarrow A\}$: sol. 17 $\{\rightarrow P, P \rightarrow \neg V, A \rightarrow V\}$: sol. 18 $\{\rightarrow P, P \rightarrow \neg V\}$: sol. 19 $\{\rightarrow P, O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 20 $\{\rightarrow P, O \rightarrow V, O \rightarrow A\}$: sol. 21 $\{\rightarrow P, O \rightarrow V, A \rightarrow V\}$: sol. 22 $\{\rightarrow P, O \rightarrow V\}$: sol. 23 $\{\rightarrow P, O \rightarrow A, A \rightarrow V\}$: sol. 24 $\{\rightarrow P, O \rightarrow A\}$: sol. 25 $\{\rightarrow P, A \rightarrow V\}$: sol. 26 $\{\rightarrow P\}$: sol. 27 $\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow V,$ O→ A, $A \rightarrow V\}$: sol. 28 $\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow V,$ O→ A} : sol. 29 $\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow V,$ A→ V} : sol. 30 $\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow V\}$: sol. 31 $\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow A,$ A→ V} : sol. 32 $\{P \rightarrow O, P \rightarrow \neg V, O \rightarrow A\}$: sol. 33 $\{P \rightarrow O, P \rightarrow \neg V, A \rightarrow V\}$: sol. 34 $\{P \rightarrow O, P \rightarrow \neg V\}$: sol. 35 $\{P \rightarrow O, O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 36 $\{P \rightarrow O, O \rightarrow V, O \rightarrow A\}$: sol. 37 $\{P \rightarrow O, O \rightarrow V, A \rightarrow V\}$: sol. 38 $\{P \rightarrow O, O \rightarrow V\}$: sol. 39 $\{P \rightarrow O, O \rightarrow A, A \rightarrow V\}$: sol. 40 $\{P \rightarrow O, O \rightarrow A\}$: sol. 41 $\{P \rightarrow O, A \rightarrow V\}$: sol. 42 $\{P \rightarrow O\}$: sol. 43 $\{P \rightarrow \neg V, O \rightarrow V, O \rightarrow A,$ A→ V} : sol. 44 $\{P \rightarrow \neg V, O \rightarrow V, O \rightarrow A\}$: sol. 45 $\{P \rightarrow \neg V, O \rightarrow V\}$: sol. 46 $\{P \rightarrow \neg V, O \rightarrow V, A \rightarrow V\}$: sol. 47 $\{P \rightarrow \neg V, O \rightarrow A, A \rightarrow V\}$: sol. 48 $\{P \rightarrow \neg V, O \rightarrow A\}$: sol. 49 $\{P \rightarrow \neg V, A \rightarrow V\}$: sol. 50 $\{P \rightarrow \neg V\}$: sol. 51 $\{O \rightarrow V, O \rightarrow A, A \rightarrow V\}$: sol. 52 $\{O \rightarrow V, O \rightarrow A, \}$: sol. 53 $\{O \rightarrow V, A \rightarrow V\}$: sol. 54 $\{O \rightarrow V\}$: sol. 55 $\{O \rightarrow A, A \rightarrow V\}$: sol. 56 $\{O \rightarrow A\}$: sol. 57 $\{A \rightarrow V\}$: sol. 58

Tableau 2.2 : L'ordre "Best-Out" (exemple 2)

les A consistants tels que $a(A) = 1$: 58 solutions :	$\{P \rightarrow O,\, P \rightarrow \neg V,\, O \rightarrow V,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 28
$\{\rightarrow P,\, P \rightarrow O,\, P \rightarrow \neg V,\, O \rightarrow A\}$: sol. 1	$\{P \rightarrow O,\, P \rightarrow \neg V,\, O \rightarrow V,\, O \rightarrow A\}$: sol. 29
$\{\rightarrow P,\, P \rightarrow O,\, P \rightarrow \neg V,\, A \rightarrow V\}$: sol. 2	$\{P \rightarrow O,\, P \rightarrow \neg V,\, O \rightarrow V,\, A \rightarrow V\}$: sol. 30
$\{\rightarrow P,\, P \rightarrow O,\, P \rightarrow \neg V\}$: sol. 3	$\{P \rightarrow O,\, P \rightarrow \neg V,\, O \rightarrow V\}$: sol. 31
$\{\rightarrow P,\, P \rightarrow O,\, O \rightarrow V,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 4	$\{P \rightarrow O,\, P \rightarrow \neg V,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 32
$\{\rightarrow P,\, P \rightarrow O,\, O \rightarrow V,\, O \rightarrow A\}$: sol. 5	$\{P \rightarrow O,\, P \rightarrow \neg V,\, O \rightarrow A\}$: sol. 33
$\{\rightarrow P,\, P \rightarrow O,\, O \rightarrow V,\, A \rightarrow V\}$: sol. 6	$\{P \rightarrow O,\, P \rightarrow \neg V,\, A \rightarrow V\}$: sol. 34
$\{\rightarrow P,\, P \rightarrow O,\, O \rightarrow V\}$: sol. 7	$\{P \rightarrow O,\, P \rightarrow \neg V\}$: sol. 35
$\{\rightarrow P,\, P \rightarrow O,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 8	$\{P \rightarrow O,\, O \rightarrow V,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 36
$\{\rightarrow P,\, P \rightarrow O,\, O \rightarrow A\}$: sol. 9	$\{P \rightarrow O,\, O \rightarrow V,\, O \rightarrow A\}$: sol. 37
$\{\rightarrow P,\, P \rightarrow O,\, A \rightarrow V\}$: sol. 10	$\{P \rightarrow O,\, O \rightarrow V,\, A \rightarrow V\}$: sol. 38
$\{\rightarrow P,\, P \rightarrow O\}$: sol. 11	$\{P \rightarrow O,\, O \rightarrow V\}$: sol. 39
$\{\rightarrow P,\, P \rightarrow \neg V,\, O \rightarrow V,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 12	$\{P \rightarrow O,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 40
	$\{P \rightarrow O,\, O \rightarrow A\}$: sol. 41
$\{\rightarrow P,\, P \rightarrow \neg V,\, O \rightarrow V,\, O \rightarrow A\}$: sol. 13	$\{P \rightarrow O,\, A \rightarrow V\}$: sol. 42
$\{\rightarrow P,\, P \rightarrow \neg V,\, O \rightarrow V,\, A \rightarrow V\}$: sol. 14	$\{P \rightarrow O\}$: sol. 43
$\{\rightarrow P,\, P \rightarrow \neg V,\, O \rightarrow V\}$: sol. 15	$\{P \rightarrow \neg V,\, O \rightarrow V,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 44
$\{\rightarrow P,\, P \rightarrow \neg V,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 16	$\{P \rightarrow \neg V,\, O \rightarrow V,\, O \rightarrow A\}$: sol. 45
$\{\rightarrow P,\, P \rightarrow \neg V,\, O \rightarrow A\}$: sol. 17	$\{P \rightarrow \neg V,\, O \rightarrow V\}$: sol. 46
$\{\rightarrow P,\, P \rightarrow \neg V,\, A \rightarrow V\}$: sol. 18	$\{P \rightarrow \neg V,\, O \rightarrow V,\, A \rightarrow V\}$: sol. 47
$\{\rightarrow P,\, P \rightarrow \neg V\}$: sol. 19	$\{P \rightarrow \neg V,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 48
$\{\rightarrow P,\, O \rightarrow V,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 20	$\{P \rightarrow \neg V,\, O \rightarrow A\}$: sol. 49
$\{\rightarrow P,\, O \rightarrow V,\, O \rightarrow A\}$: sol. 21	$\{P \rightarrow \neg V,\, A \rightarrow V\}$: sol. 50
$\{\rightarrow P,\, O \rightarrow V,\, A \rightarrow V\}$: sol. 22	$\{P \rightarrow \neg V\}$: sol. 51
$\{\rightarrow P,\, O \rightarrow V\}$: sol. 23	$\{O \rightarrow V,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 52
$\{\rightarrow P,\, O \rightarrow A,\, A \rightarrow V\}$: sol. 24	$\{O \rightarrow V,\, O \rightarrow A,\, \}$: sol. 53
$\{\rightarrow P,\, O \rightarrow A\}$: sol. 25	$\{O \rightarrow V,\, A \rightarrow V\}$: sol. 54
$\{\rightarrow P,\, A \rightarrow V\}$: sol. 26	$\{O \rightarrow V\}$: sol. 55
$\{\rightarrow P\}$: sol. 27	$\{O \rightarrow A,\, A \rightarrow V\}$: sol. 56
	$\{O \rightarrow A\}$: sol. 57
	$\{A \rightarrow V\}$: sol. 58

Tableau 2.3 : L'ordre “Best-Out” (exemple 3)

8. L'ordre "Best-Out" a une longue histoire. Il est issu du principe du minimum de spécificité (voir [BDP92] et [BDP94a]). Il correspond entre autres au système Z de [Pea90], à la fermeture rationnelle de Lehmann et Magidor ([LM92]), la différence principale résidant dans le fait que le pré-ordre $<$ défini sémantiquement dans [Pea90, LM92] est ici imposé a priori.

Ce mécanisme de sélection sera noté Bo.

2.2.5 Les ensembles de Gärdenfors et Makinson

Nous laissons de côté les approches purement "syntaxiques" pour aborder ici un autre type d'approches : celles qui utilisent le lien sémantique et la dépendance logique existant entre les croyances. Gärdenfors et Makinson ont donc défini une classe de relations d'inférence non-monotone : les relations d'inférence "expectatives" (voir [Gär91] et [GM94])[22]. L'idée est que, lors d'un raisonnement, on utilise des informations dans lesquelles on croit fermement mais aussi des informations que l'on espère, que l'on attend[23]. L'objectif des travaux de Gärdenfors et Makinson est d'obtenir une relation d'inférence non-monotone vérifiant des propriétés choisies parmi celles définies dans [KLM90], [Gär91] et [GM94].

On peut distinguer 2 méthodes :

- l'utilisation d'ensembles d'"expectations" (sous-ensembles de formules choisis dans l'ensemble de formules initial),
- l'utilisation d'un ordre "expectatif" (ordre entre les formules du langage possédant certaines propriétés).

Considérons la base de formules E. Les deux définitions de cette relation d'inférence "expectative" sont :

Définition 2.2.19 *[utilisation d'ensembles d'"expectations"]* $f \mid\!\sim g$ *ssi*
$g \in C_{E,S}(f) = \cap\{C_n(\{f\} \cup D)$ *avec* $D \in S(E\perp\neg f)\}$ *où :*

- $C_n(X)$ = *ensemble des conséquences logiques de* X *au sens de la logique classique,*
- $S(B)$ = *fonction de sélection renvoyant certains éléments de* B*,*
- $E\perp h$ = *sous-ensemble maximal de* E *qui ne peut impliquer logiquement* h*.*

Définition 2.2.20 *[utilisation d'un ordre "expectatif"]* $f \mid\!\sim g$ *ssi*
$g \in C_n(\{f\} \cup \{h \in E$ *tel que* $\neg f < h\})$ *avec l'ordre* $<$ *entre les formules qui est total et qui vérifie les propriétés de dominance (si* $\alpha \vdash \beta$ *alors* $\alpha \leq \beta$*) et de conjonctivité (pour tous* α *et* β*,* $\alpha \leq \alpha \wedge \beta$ *ou* $\beta \leq \alpha \wedge \beta$*) (voir [GM94]).*

Notons que $E\perp\neg f$ est donc l'ensemble des sous-bases de E qui sont maximales (au sens de l'inclusion) consistantes avec f.

Remarque : la définition de la fonction S donnée dans la définition 2.2.19 est tout à fait informelle.

Dans le cas des ensembles d'"expectations", les propriétés de la relation d'inférence ainsi définie dépendent intimement de la construction de la fonction de sélection S.

Dans le cas d'un ordre "expectatif", Gärdenfors et Makinson ont démontré que la relation d'inférence "expectative" ainsi définie vérifie tout un ensemble de propriétés particulièrement intéressantes (la supra-classicité, l'équivalence logique gauche, le ET, la cumulativité, le OU, la monotonie

[22] J'utilise ici les termes "expectations", ordre "expectatif", relations "expectatives" qui sont des néologismes issus directement de l'anglais afin de conserver la terminologie de Gärdenfors et Makinson.
[23] En anglais "to expect" veut dire attendre, espérer.

rationnelle et la préservation de la consistance – voir dans [KLM90, Gär91, GM94] et dans la section 3.4 page 55 de ce document la définition de chacune de ces propriétés).

Ils ont d'autre part démontré l'équivalence des 2 méthodes *à condition que* la fonction S soit construite à partir d'un ordre "expectatif" sur les formules de la base.

Comment situer cette approche par rapport aux approches présentées précédemment ?

Disons tout d'abord que toutes les méthodes précédentes ("Inclusion Based", lexicographique, "Best-Out", etc.) visent à utiliser conjointement des sous-ensembles de formules choisis dans l'ensemble de formules initial et un ordre entre les formules.

L'utilisation de sous-ensembles choisis correspond à la construction d'une fonction S de sélection. Ces approches peuvent donc se situer dans le cadre des ensembles d'"expectations" de Gärdenfors et Makinson. Toutefois, l'ordre entre les formules sous-jacent à la définition de cette fonction S n'est pas un ordre "expectatif" ; en effet, on n'impose rien à l'ordre $<$, à part dans certains cas le fait qu'il soit total. Par conséquence, les méthodes précédentes ne sont pas des relations d'inférence "expectatives" au sens de l'ordre "expectatif".

Conclusions sur les ensembles de Gärdenfors et Makinson :

1. L'approche de Gärdenfors et Makinson différencie[24] la construction de sous-ensembles de formules particuliers issus de la base initiale et l'utilisation d'un ordre entre les formules de la base, alors que les autres approches vues jusqu'à présent ont plutôt tendance à utiliser conjointement ces aspects.

2. Cette approche vise plutôt la définition d'une relation d'inférence possédant certaines propriétés choisies parmi celles présentées dans [KLM90], [Gär91] et [GM94]. En conséquence, elle impose à l'ordre utilisé (soit directement par la méthode de l'ordre "expectatif", soit indirectement par la définition de la fonction de sélection S) un ensemble de propriétés (par exemple, la dominance).

3. Les autres approches vues jusqu'alors correspondent à la définition d'une fonction S de sélection. Ce sont donc des approches "expectatives" dans le sens des ensembles d'"expectations" (voir définition 2.2.19 page précédente).

4. Par contre, ces autres approches n'imposent rien à l'ordre $<$. Ce ne sont donc pas des approches "expectatives" dans le sens d'un ordre "expectatif" (voir définition 2.2.20 page ci-contre).

2.3 Choix des mécanismes et motivations de ces choix

Parmi tous les mécanismes de sélection présentés dans la section 2.2 page 7, je vais en privilégier certains. J'ai toutefois tenu à les présenter tous pour les raisons suivantes :

- S, qui sélectionne les sous-bases consistantes, sert de point de repère pour mesurer l'impact d'un mode de sélection dit évolué, c'est-à-dire un mode de sélection plus restrictif que celui reposant seulement sur la consistance des ensembles choisis ; à part cela, ce mécanisme n'a pas grand intérêt ;
- T, qui sélectionne les thèses, est le plus classique et sert d'idée de base à beaucoup d'autres approches ; il est connu depuis bien longtemps déjà (voir [RM70]) ;
- Incl, qui sélectionne les thèses "Inclusion Based" préférées, est un raffinement de T ;
- Lex, qui sélectionne les thèses préférées pour l'ordre lexicographique, est un raffinement de Incl ;

[24] Même s'ils montrent ensuite les liens étroits entre ces deux aspects, en particulier du point de vue des propriétés recherchées pour la relation d'inférence "expectative".

- CAR, qui sélectionne les thèses préférées pour la cardinalité, est un raffinement de T et un cas particulier de LEX ;
- BO, qui sélectionne les sous-bases consistantes préférées pour l'ordre "Best Out", fait le lien avec l'inférence possibiliste, le système Z et d'autres formulations qui appartiennent plutôt aux approches sémantiques ;
- E et E*, qui sélectionnent les extensions (dans le cas général et dans le cas particulier d'une théorie super-normale), font le lien avec la logique des défauts qui est un des mécanismes de sélection[25] les plus connus appartenant à la famille des logiques non-monotones ;

Ces mécanismes sont à répartir en deux groupes :

- ceux qui utilisent les préférences entre formules (mécanismes INCL, LEX, BO) ;
- ceux qui n'utilisent pas les préférences entre formules (mécanismes S, T, CAR).

Notons que les mécanismes E et E* sont des mécanismes hybrides qui ne savent gérer qu'une base possédant au maximum deux strates :

- la strate qui contient les prémisses (informations considérées comme certaines, qui sont donc les "plus préférées") et qui doit être consistante pour que le calcul des extensions présente un intérêt ;
- la strate qui correspond aux défauts qui sont donc "moins préférés" que les prémisses.

Les mécanismes sur lesquels je vais centrer mon étude sont les mécanismes utilisant la stratification des connaissances (INCL, LEX, BO). Les autres mécanismes serviront essentiellement de point de repère et feront l'objet d'une moindre attention.

2.4 État de l'art sur les principes d'inférence

Grâce aux mécanismes de sélection présentés dans la section 2.2 page 7, on peut définir $(P^E, \ll)$ avec $P^E = \{P_1^E, \ldots, P_n^E\}$, un ensemble de sous-bases consistantes, et $\ll$, une relation de préférence (au moins un pré-ordre) sur cet ensemble.

On veut maintenant étudier différents moyens d'inférer de façon non-monotone une formule Φ à partir de $(P^E, \ll)$; par exemple, on peut choisir d'inférer la formule Φ à partir de $(P^E, \ll)$ si et seulement si toutes les sous-bases P_i^E de P^E infèrent classiquement Φ. Pour ce faire, l'objectif est d'utiliser au maximum les techniques d'inférence classique.

Remarques importantes :

- Cette section s'inspire très largement des travaux de Pinkas et Loui (voir [PL92]). J'ai donc gardé leur point de départ, c'est-à-dire le fait que l'ensemble P^E soit un ensemble de sous-bases consistantes associé à un pré-ordre. Ce qui va m'amener à présenter 6 principes d'inférence.
- Ici, je ne m'attache pas à la construction des sous-bases et à l'élaboration de l'ordre entre les sous-bases. En effet, le mécanisme de définition de l'ensemble P^E et de l'ordre $\ll$ est indépendant du principe d'inférence choisi[26].

Je décris ici les principaux mécanismes d'inférence présentés dans [PL92].

[25] Bien qu'elle n'ait pas été introduite en tant que telle !

[26] Même si, historiquement, les auteurs d'un mécanisme de définition de P^E avaient prévu de l'utiliser avec un principe d'inférence particulier.

2.4.1 La conséquence forte

Il s'agit de la mise en œuvre d'un raisonnement circonspect (prudent)[27] :

Définition 2.4.1 $(P^E, \ll) \mid\!\sim^{\forall} \Phi$ *ssi quelle que soit la sous-base* P_i^E *élément de* P^E*, on a* $P_i^E \vdash \Phi$.

Ce principe d'inférence dit fort ou universel sera noté avec le symbole $\mid\!\sim^{\forall}$, et on appelle conséquence forte le type de conséquence ainsi défini.
Remarques :

- Ce principe a été introduit par [RM70].
- Dans ce principe d'inférence, l'ordre $\ll$ n'intervient pas.
- J'appellerai ce principe : le principe UNI (UNI pour universel).
- Ce principe présente la particularité suivante : parmi toutes les sous-bases P_i^E de P^E, on peut avoir la sous-base vide ; la définition devient alors :
  Si $\exists i$ tel que $P_i^E = \varnothing$ alors $((P^E, \ll) \mid\!\sim^{\forall} \Phi \Leftrightarrow \Phi$ est une tautologie).
  Ce cas se présente rarement. Lorsqu'il se produira, je présenterai 2 résultats différents, un pour le cas où $\varnothing$ est élément de P^E, et un pour le cas où $\varnothing$ est exclu de P^E.

2.4.2 La conséquence faible

Par opposition au principe précédent, j'ai ici la mise en œuvre d'un raisonnement crédule :

Définition 2.4.2 $(P^E, \ll) \mid\!\sim^{\exists} \Phi$ *ssi il existe au moins une sous-base* P_i^E *élément de* P^E *telle que* $P_i^E \vdash \Phi$.

Ce principe d'inférence dit faible ou existentiel sera noté avec le symbole $\mid\!\sim^{\exists}$, et on appelle conséquence faible le type de conséquence ainsi défini.
Remarques :

- Ce principe a été introduit par [RM70].
- Dans ce principe d'inférence, l'ordre $\ll$ n'intervient pas.
- J'appellerai ce principe : le principe EXI (EXI pour existentiel).

2.4.3 La conséquence argumentative

L'idée exploitée ici est la suivante : Φ est conséquence faible mais $\neg\Phi$ n'est même pas conséquence faible.

Définition 2.4.3 $(P^E, \ll) \mid\!\sim^{A} \Phi$ *ssi il existe au moins une sous-base* P_i^E *élément de* P^E *telle que* $P_i^E \vdash \Phi$ *et quelle que soit* P_j^E *élément de* P^E*, on a* $P_j^E \nvdash \neg\Phi$.

Ce principe d'inférence dit argumentatif sera noté avec le symbole $\mid\!\sim^{A}$, et on appelle conséquence argumentative le type de conséquence ainsi défini.
Remarques :

- Dans ce principe d'inférence, l'ordre $\ll$ n'intervient pas.
- J'appellerai ce principe : le principe ARG (ARG pour argumentatif).
- Ce principe est défini aussi par [BDP93] dans le cadre de l'argumentation, car il existe des liens étroits entre ce domaine et les relations d'inférence non-monotone (voir [Cay95a, Cay95b] pour une étude sur ce sujet).

[27] On prend le plus possible de précautions avant d'inférer une nouvelle information. Par opposition, on parlera de raisonnement crédule. Cette notion de prudence est présentée dans la section 3.3 page 54.

2.4.4 La conséquence forte avec préférences

Prenons maintenant en compte des préférences entre les sous-bases. En les associant à un raisonnement circonspect, on obtient :

Définition 2.4.4 $(P^E, \ll) \mid\!\sim^{\forall,\mathrm{pref}} \Phi$ *ssi quelle que soit la sous-base* P_i^E *élément de* P^E *préférée selon l'ordre* $\ll$*, on a* $P_i^E \vdash \Phi$.

Ce principe d'inférence dit universel avec préférences sera noté avec le symbole $\mid\!\sim^{\forall,\mathrm{pref}}$ avec "pref" fonction de l'ordre $\ll$, et on appelle conséquence forte avec préférences le type de conséquence ainsi défini.

Remarques :

- Dans ce principe d'inférence, l'ordre $\ll$ intervient ; il sert à définir les sous-bases préférées de P^E.
- J'appellerai ce principe : le principe UNI-PREF (UNI pour universel et PREF pour l'ordre $\ll$).
- Même remarque que pour UNI sur l'inclusion de l'ensemble vide dans P^E.

2.4.5 La conséquence faible avec préférences

Associons des préférences et un raisonnement crédule :

Définition 2.4.5 $(P^E, \ll) \mid\!\sim^{\exists,\mathrm{pref}} \Phi$ *ssi il existe au moins une sous-base* P_i^E *élément de* P^E *préférée selon l'ordre* $\ll$ *telle que* $P_i^E \vdash \Phi$.

Ce principe d'inférence dit existentiel avec préférences sera noté avec le symbole $\mid\!\sim^{\exists,\mathrm{pref}}$ avec "pref" fonction de l'ordre $\ll$, et on appelle conséquence faible avec préférences le type de conséquence ainsi défini.

Remarques :

- Dans ce principe d'inférence, l'ordre $\ll$ intervient ; il sert à définir les sous-bases préférées de P^E.
- J'appellerai ce principe : le principe EXI-PREF (EXI pour existentiel et PREF pour l'ordre $\ll$).

2.4.6 La conséquence argumentative avec préférences

Associons des préférences et l'idée de la conséquence argumentative :

Définition 2.4.6 $(P^E, \ll) \mid\!\sim^{A,\mathrm{pref}} \Phi$ *ssi :*

- *il existe au moins une sous-base* P_i^E *élément de* P^E *préférée selon l'ordre* $\ll$ *telle que* $P_i^E \vdash \Phi$,
- *et quelle que soit* P_j^E *élément de* P^E *préférée selon l'ordre* $\ll$*, on a* $P_j^E \nvdash \neg\Phi$.

Ce principe d'inférence dit argumentatif avec préférences sera noté avec le symbole $\mid\!\sim^{A,\mathrm{pref}}$ avec "pref" fonction de l'ordre $\ll$, et on appelle conséquence argumentative avec préférences le type de conséquence ainsi défini.

Remarques :

- Dans ce principe d'inférence, l'ordre $\ll$ intervient ; il sert à définir les sous-bases préférées de P^E.
- J'appellerai ce principe : le principe ARG-PREF (ARG pour argumentatif et PREF pour l'ordre $\ll$).
- Ce principe est aussi invoqué par [BDP93], mais dans le cadre des systèmes d'argumentation (voir [Cay95a, Cay95b]).

2.4.7 Les autres principes

Dans l'article de Pinkas et Loui, sont cités de nombreux autres principes d'inférence qui proviennent essentiellement d'une combinaison entre les principes vus ci-dessus et que je me contenterai de citer sans les étudier plus en détail. On a par exemple :

- $(P^E, \ll) \mid\!\sim \Phi$ ssi $\forall P_i^E$ préférée selon l'ordre $\ll$, on a $P_i^E \vdash \Phi$ ou $\exists P_j^E$ (pas forcément préférée pour l'ordre $\ll$) telle que $P_j^E \not\vdash \neg\Phi$.
- $(P^E, \ll) \mid\!\sim \Phi$ ssi $\exists P_i^E$ préférée selon l'ordre $\ll$ telle que $P_i^E \vdash \Phi$ et $\forall P_j^E$ (pas forcément préférée pour l'ordre $\ll$) on a $P_j^E \not\vdash \neg\Phi$.
- $(P^E, \ll) \mid\!\sim \Phi$ ssi $\exists P_i^E$ (préférée ou pas pour l'ordre $\ll$) telle que $P_i^E \vdash \Phi$ et $\forall P_j^E$ (préférée ou pas pour l'ordre $\ll$) telle que $P_i^E \ll P_j^E$, on a $P_j^E \not\vdash \neg\Phi$ (avec 2 cas possibles : $P_i^E \ll P_j^E$ vu de manière stricte ou non).

2.5 Choix des principes d'inférence et motivations de ces choix

Il existe beaucoup d'autres principes d'inférence (voir [PL92] pour un panorama plus complet) et je n'ai choisi de présenter et d'étudier que les trois principes d'inférence UNI, EXI et ARG (et leur version avec préférences) pour les raisons suivantes :

- UNI et EXI étaient quasiment imposés par le fait que ce sont des principes d'inférence très connus dans la littérature (voir [RM70]) et qu'une classification se devait de les prendre en compte ; d'autre part, ils sont simples à exprimer et à appréhender ;
- ARG représente un moyen terme entre UNI et EXI qui semble beaucoup plus réaliste que UNI (trop pessimiste) et EXI (trop optimiste) ; cela semble un bon compromis du point de vue de l'expressivité.

2.6 Différentes classes de relations d'inférence

Dans les sections précédentes, j'ai présenté divers mécanismes de sélection m et divers principes d'inférence p. À l'aide de ces "outils", on peut alors définir des relations d'inférence en utilisant les notations suivantes :

- E : la base initiale (ensemble fini de formules propositionnelles),
- $<$: ordre[28] entre les formules de E,
- p : principe d'inférence (voir section 2.4 page 26),
- m : mécanisme de définition de sous-ensembles consistants de E avec un ordre entre ces sous-ensembles induit de $<$ (voir section 2.2 page 7),

[28] Désormais, le mot "ordre" est employé dans un sens générique. Il recouvre indifféremment les notions mathématiques d'ordre et de pré-ordre.

- Ψ et Φ : formules de la logique propositionnelle.

En fait, on peut définir au moins deux classes de relations d'inférence :

- les relations d'inférence du type $(E,<) \mid\!\sim^{p,m} \Phi$;
- les relations d'inférence du type $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$.

La première classe correspond à des relations définissant les "théorèmes non-monotones" pour un contexte donné (la base E).

La seconde classe correspond à des relations de déductibilité entre formules pour un contexte donné (la base E).

Implicitement, en présentant les principes d'inférence dans la section 2.4 page 26, je me suis placée dans le cadre de la première classe. Or, la seconde classe va être utile pour la partie de mon étude concernant les propriétés de déduction (voir section 4.3 page 153). Il faut donc étudier la définition des relations d'inférence non-monotone de la classe 2, et c'est l'objet de la section présente. Notons que les définitions proposées pour les relations de classe 2 s'appuient toutes sur des relations de classe 1, ce qui paraît le plus approprié étant donné que les relations de classe 2 sont des généralisations de celles de classe 1.

Afin de bien distinguer les deux classes de relations, j'utiliserai les notations suivantes :

Notation 2.6.1 *Les deux classes de relations d'inférence non-monotone sont notées de la façon suivante :*

- *la classe des relations d'inférence du type $(E,<) \mid\!\sim^{p,m} \Phi$ est notée p-m ;*
- *la classe des relations d'inférence du type $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$ est notée G-p-m.*

Ces notations, qui peuvent paraître arbitraires, sont dues à mes choix de définition pour la classe numéro 2 et seront explicitées dans la section 2.6.6 page 46.

2.6.1 La première définition pour les relations de classe 2

Définition 2.6.1 $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$ *ssi* $(\Psi \oplus (E,<)) \mid\!\sim^{p,m} \Phi$

L'opérateur $\oplus$ peut être défini de différentes manières.

Définition 2.6.2 *Soit $(E,<)$ une base de croyances propositionnelle stratifiée ou pas et Ψ une formule propositionnelle, $(\Psi \oplus (E,<)) = (E \cup \{\Psi\}, <')$, avec l'ordre $<'$ étendant l'ordre $<$ en rajoutant une première strate à E ne contenant que Ψ*[29].

Cette définition 2.6.2 représente le point de vue de la révision de connaissances selon lequel la formule rajoutée est systématiquement conservée. Je m'inspire ici des travaux sur la révision menés entre autres par Nebel (cf. [Neb91]). Cette définition est la plus connue et la plus utilisée. C'est celle que j'exploiterai principalement dans cette étude.

Rappelons que le cadre syntaxique, dans lequel je définis les relations d'inférence non-monotone, oblige à considérer la formule Ψ rajoutée comme un tout indivisible (elle est acceptée ou rejetée dans son intégralité) et ceci même si Ψ est une conjonction de formules[30].

Par contre, en s'éloignant du point de vue de la révision, on peut définir de nouveaux opérateurs $\oplus$.

[29] C'est-à-dire que Ψ est strictement préférée à toute autre formule de E.

[30] Cette remarque aura son importance dans un certain nombre de démonstrations présentées dans cette thèse.

Définition 2.6.3 *Soit $(E, <)$ une base de croyances propositionnelle, stratifiée ou pas, et Ψ une formule propositionnelle, $(\Psi \oplus (E, <)) = E \cup \{\Psi\}$, ensemble dans lequel toutes les formules sont de même priorité.*

La sémantique de cette définition 2.6.3 repose sur le fait suivant : ce n'est pas parce qu'on rajoute une nouvelle information Ψ à l'ensemble des croyances, que Ψ est plus fiable que les autres croyances de ma base. Cette définition pourra être utilisée pour les relations d'inférence non-monotone n'exploitant pas l'ordre existant entre les formules de la base E.

On peut aussi considérer que l'opérateur $\oplus$ consiste à rajouter une information en précisant comment classer cette nouvelle croyance par rapport au reste de la base. Cela revient à insérer Ψ dans E soit en créant une strate spécifique pour Ψ, soit en plaçant Ψ dans une strate déjà existante, ce qui signifie qu'il faut mettre l'ordre $<$ à jour. Cette possibilité est complexe à mettre en œuvre et ne sera pas exploitée ici.

2.6.2 La seconde définition pour les relations de classe 2

Définition 2.6.4 $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$ *ssi* $(E, <) \mid\!\sim^{p,m} (\Psi \rightarrow \Phi)$

Cette définition présente l'avantage d'éviter de calculer de nouveaux sous-ensembles préférés. Travailler avec une nouvelle formule Ψ peut alors se faire à partir des sous-ensembles préférés de E. Ce n'est pas le cas avec la définition 2.6.1 page précédente dans laquelle on définit un nouvel ensemble $\Psi \oplus E$.

2.6.3 Caractérisation des sous-ensembles préférés dans la première définition

Dans cette section, je vais reprendre chacun des mécanismes de génération m (présentés dans le détail en section 2.2 page 7 et récapitulés en section 2.3 page 25), afin de caractériser les sous-ensembles m-préférés de $(\Psi \oplus (E, <))$ par rapport aux sous-ensembles m-préférés de $(E, <)$. Ainsi, de l'étude des particularités de chaque mécanisme (un théorème de caractérisation pour chaque m), je pourrai extraire des règles de caractérisation plus générales (voir sections 2.6.4 page 43, 2.6.5 page 45 et 2.6.6 page 46).

2.6.3.1 Mécanisme de génération T

L'ordre $<$, s'il existe, n'intervient pas dans le mécanisme de génération T ; en conséquence, la base $(E, <)$ sera simplement notée E.

L'opérateur $\oplus$ peut correspondre soit à la définition 2.6.3, soit à la définition 2.6.2 page ci-contre. Or, l'application de la définition 2.6.2 page précédente au mécanisme T constitue un cas particulier du mécanisme INCL ; elle sera donc prise en compte au travers du théorème de caractérisation 2.6.3 page 33. Dans la section courante, nous allons nous contenter d'étudier l'impact de la définition 2.6.3 sur le mécanisme T ; on obtient ainsi le théorème de caractérisation 2.6.1.

Rappelons que les sous-ensembles préférés de E par le mécanisme T sont les thèses de E.

Théorème de caractérisation 2.6.1 *Les thèses de $E \cup \{\Psi\}$ sont :*

- *soit des thèses Y de E ssi Y est inconsistante avec Ψ,*
- *soit de la forme $\{\Psi\} \cup Y$ avec :*
  - *$Y \subseteq E$,*
  - *Y consistante avec Ψ,*

- *Y est maximale pour l'inclusion parmi les sous-bases Ψ-consistantes*[31] *de E.*

Preuve : Une thèse est un sous-ensemble consistant maximal pour l'inclusion. D'autre part, l'opérateur $\oplus$ correspond à l'union. Par conséquent, un sous-ensemble Y' consistant maximal pour l'inclusion de $E \cup \{\Psi\}$ peut être :

- soit un sous-ensemble Y consistant maximal pour l'inclusion de E à condition que Y soit inconsistant avec Ψ (sinon Y n'est pas maximal dans $\Psi \oplus E$),
- soit l'union de Ψ avec une sous-base Y ; par construction, cette Y est incluse dans E et consistante avec Ψ ; Y est-elle maximale pour l'inclusion parmi les sous-bases de E consistantes avec Ψ ? la démonstration se fait en deux étapes :
  - condition suffisante : soit l'ensemble $Y' = \{\Psi\} \cup Y$ avec $Y \subseteq E$, Y consistante avec Ψ et Y maximale pour l'inclusion parmi les sous-bases de E consistantes avec Ψ alors Y' est maximale pour l'inclusion pour $\Psi \oplus E$, c'est donc une thèse de $\Psi \oplus E$;
  - condition nécessaire : soit Y' une thèse de $\Psi \oplus E$ telle que Y' soit de la forme $\{\Psi\} \cup Y$ avec $Y \subseteq E$, Y Ψ-consistante ; raisonnons par l'absurde et supposons que Y ne soit pas maximale pour l'inclusion parmi les sous-bases de E consistantes avec Ψ ; il existe alors $Z \subseteq E$ maximale pour l'inclusion parmi les sous-bases de E Ψ-consistantes telle que $Y \subset Z \Rightarrow \{\Psi\} \cup Y \subseteq \{\Psi\} \cup Z$ $\Rightarrow \{\Psi\} \cup Y$ n'est pas une thèse de $E' \Rightarrow$ contradiction avec l'hypothèse Y' thèse de $\Psi \oplus E$.

□

Proposition 2.6.1 *Il y a au moins autant de thèses dans $E \cup \{\Psi\}$ que dans E.*

Preuve : D'après le théorème 2.6.1 page précédente, chaque thèse Y de E sert à construire une thèse de $E \cup \{\Psi\}$ et les thèses ainsi construites sont différentes les unes des autres puisque les Y de E sont distinctes deux à deux. On obtient donc au moins autant de thèses préférées dans $E \cup \{\Psi\}$ que dans E. □

Exemple : Soit la base $E = \{\rightarrow p, p \rightarrow o, p \rightarrow (\neg v), o \rightarrow v, o \rightarrow a\}$ et la formule $\Psi = (a \rightarrow v)$. On a 4 thèses dans E :

$Y_1 =$ $\{ \rightarrow p,$ $p \rightarrow o,$ $p \rightarrow (\neg v),$ $o \rightarrow a\}$ Y_1 Ψ-inconsistante.	$Y_2 =$ $\{ \rightarrow p,$ $p \rightarrow o,$ $o \rightarrow v,$ $o \rightarrow a\}$ Y_2 Ψ-consistante.	$Y_3 =$ $\{ \rightarrow p,$ $p \rightarrow (\neg v),$ $o \rightarrow v,$ $o \rightarrow a\}$ Y_3 Ψ-consistante.	$Y_4 =$ $\{ p \rightarrow o,$ $p \rightarrow (\neg v),$ $o \rightarrow v,$ $o \rightarrow a\}$ Y_4 Ψ-consistante.

Dans $E' = E \cup \{\Psi\}$, on a 5 thèses :

$Y'_1 = Y_1 =$ $\{ \rightarrow p,$ $p \rightarrow o,$ $p \rightarrow (\neg v),$ $o \rightarrow a\}$	$Y'_2 = Y_2 \cup \{\Psi\} =$ $\{ \rightarrow p,$ $p \rightarrow o,$ $o \rightarrow v,$ $o \rightarrow a,$ $a \rightarrow v\}$	$Y'_3 = Y_3 \cup \{\Psi\} =$ $\{ \rightarrow p,$ $p \rightarrow (\neg v),$ $o \rightarrow v,$ $o \rightarrow a,$ $a \rightarrow v\}$	$Y'_4 = Y_4 \cup \{\Psi\} =$ $\{ p \rightarrow o,$ $p \rightarrow (\neg v),$ $o \rightarrow v,$ $o \rightarrow a,$ $a \rightarrow v\}$	$Y'_5 = S \cup \{\Psi\} =$ $\{ \rightarrow p,$ $p \rightarrow o,$ $p \rightarrow (\neg v),$ $a \rightarrow v\}$ (S partie de Y_1)

[31] C'est-à-dire, les sous-bases qui sont consistantes avec Ψ.

2.6.3.2 Mécanisme de génération S

Ici aussi, comme pour le mécanisme T, l'opérateur $\oplus$ peut correspondre soit à la définition 2.6.3 page 31, soit à la définition 2.6.2 page 30. Or, l'application de la définition 2.6.2 page 30 au mécanisme S ne correspond à aucun des mécanismes étudiés ici ; elle ne sera donc pas prise en compte. Dans la section courante, nous allons nous contenter d'étudier l'impact de la définition 2.6.3 page 31 sur le mécanisme S ; on obtient ainsi le théorème de caractérisation 2.6.2.

Théorème de caractérisation 2.6.2 *Les sous-bases consistantes de $E \cup \{\Psi\}$ sont :*

- *soit Y, quelle que soit Y sous-base consistante de E,*
- *soit de la forme $\{\Psi\} \cup Y$ ssi Y est une sous-base consistante de E consistante avec Ψ.*

Exemple : Soit la base E suivante : $E = \{\rightarrow a, ab \rightarrow, a \rightarrow b\}$. Posons $\Psi = \neg a$. Il y a 7 sous-bases consistantes dans E, dont 4 consistantes avec Ψ.

$Y_1 = \varnothing$ Y_1 Ψ-consistante	$Y_2 = \{\rightarrow a\}$ Y_2 Ψ-inconsistante	$Y_3 = \{\rightarrow a, ab \rightarrow\}$ Y_3 Ψ-inconsistante	$Y_4 = \{\rightarrow a, a \rightarrow b\}$ Y_4 Ψ-inconsistante
$Y_5 = \{ab \rightarrow, a \rightarrow b\}$ Y_5 Ψ-consistante	$Y_6 = \{ab \rightarrow\}$ Y_6 Ψ-consistante	$Y_7 = \{a \rightarrow b\}$ Y_7 Ψ-consistante	

Dans $E' = \{\Psi\} \cup E$, on a 11 sous-bases consistantes (7 + 4).

$Y'_1 = Y_1 = \varnothing$	$Y'_2 = Y_2 = \{\rightarrow a\}$	$Y'_3 = Y_3 =$ $\{\rightarrow a, ab \rightarrow\}$	$Y'_4 = Y_4 =$ $\{\rightarrow a, a \rightarrow b\}$
$Y'_5 = Y_5 =$ $\{ab \rightarrow, a \rightarrow b\}$	$Y'_6 = Y_6 = \{ab \rightarrow\}$	$Y'_7 = Y_7 = \{a \rightarrow b\}$	$Y'_8 = \{a \rightarrow\}$
$Y'_9 =$ $\{a \rightarrow, ab \rightarrow, a \rightarrow b\}$	$Y'_{10} = \{a \rightarrow, ab \rightarrow\}$	$Y'_{11} = \{a \rightarrow, a \rightarrow b\}$	

2.6.3.3 Mécanisme de génération INCL

La base de croyances étant stratifiée, seule la définition 2.6.2 page 30 de l'opérateur $\oplus$ présente un intérêt.
Rappelons que les sous-ensembles préférés de E pour le mécanisme INCL sont les thèses de E préférées pour l'inclusion (appelées thèses incl-préférées). On obtient ainsi le théorème 2.6.3.

Théorème de caractérisation 2.6.3 *Les thèses de $\Psi \oplus E$ incl-préférées sont de la forme $\{\Psi\} \cup S$ avec :*

- *$S \subseteq E$,*
- *S consistante avec Ψ,*
- *S incl-préférée parmi les sous-bases de E consistantes avec Ψ.*

Preuve : La démonstration se fait en deux étapes (c'est une généralisation d'une partie de la démonstration du théorème 2.6.1 page 31). La base E est stratifiée de la manière suivante : $E = E_1 \cup \ldots \cup E_n$ avec E_i la strate numéro i de E et $E_0 = \{\Psi\}$.

- condition suffisante : soit l'ensemble $Y = \{\Psi\} \cup S$ avec $S \subseteq E$, S consistante avec Ψ et S incl-préférée parmi les sous-bases de E consistantes avec Ψ, montrons par l'absurde que Y est incl-préférée dans $\Psi \oplus E$; si ce n'est pas le cas alors il existe $Z \subseteq \Psi \oplus E$ telle que Z est incl-préférée à Y ; c'est-à-dire, si on en revient à la définition (voir dans [CLS93]) : $\exists i$, $0 \leq i \leq n$, tel que $Y_i \subset Z_i$, et $\forall j < i$, $Y_j = Z_j$; or $Y = \{\Psi\} \cup S \Rightarrow Y_0 \neq \varnothing \Rightarrow i \geq 1$ et $Z_0 = \{\Psi\} \Rightarrow \exists i$, $1 \leq i \leq n$, tel que $S_i = Y_i \subset Z_i$, et $\forall j < i$, $Y_j = Z_j$ ce qui voudrait dire que S n'est pas incl-préférée parmi les sous-bases Ψ-consistantes, donc contradiction.

- condition nécessaire : soit Y une thèse incl-préférée de $\Psi \oplus E$, alors Y est de la forme $\{\Psi\} \cup S$ avec $S \subseteq E$, S Ψ-consistante ; raisonnons par l'absurde et supposons que S ne soit pas incl-préférée parmi les sous-bases de E consistantes avec Ψ ; il existe alors $Z \subseteq E$ incl-préférée parmi les sous-bases de E Ψ-consistantes telle que Z est incl-préférée à S ; c'est-à-dire, si on en revient à la définition (voir dans [CLS93]) : $\exists i$, $1 \leq i \leq n$, tel que $S_i \subset Z_i$, et $\forall j < i$, $S_j = Z_j$;
ceci implique que $Z \cup \{\Psi\}$ est incl-préférée à $S \cup \{\Psi\} = Y$, ce qui est en contradiction avec l'hypothèse de départ.

□

Proposition 2.6.2 *Il y a au moins autant de thèses incl-préférées dans $\Psi \oplus E$ que de thèses incl-préférées consistantes avec Ψ dans E.*

> **Preuve :** D'après le théorème 2.6.3 page précédente, le nombre de thèses incl-préférées de $\Psi \oplus E$ est égal au nombre de S incl-préférées parmi les sous-bases de E Ψ-consistantes. Parmi ces S, on retrouve toutes les thèses incl-préférées dans E consistantes avec Ψ. Donc il y a au moins autant de thèses incl-préférées dans $\Psi \oplus E$ que de thèses incl-préférées consistantes avec Ψ dans E. □

Remarque : Chaque thèse incl-préférée de E inconsistante avec Ψ permet de trouver au moins un sous-ensemble S satisfaisant les conditions du théorème 2.6.3 page précédente. En effet, soit Y une thèse incl-préférée de E inconsistante avec Ψ, il suffit d'appliquer l'algorithme 2.1.

Algorithme 2.1 :
début
- $X \leftarrow \{\Psi\}$
- **pour** chaque strate i de Y **faire**
  - déterminer $X_i \subseteq Y_i$ maximal pour l'inclusion et X-consistant
  - $X \leftarrow X \cup X_i$
- $S \leftarrow X \setminus \{\Psi\}$

fin

Cependant, chaque thèse incl-préférée de E inconsistante avec Ψ peut permettre de trouver plusieurs S. Par exemple, la thèse incl-préférée Y inconsistante avec Ψ suivante permet de trouver deux sous-ensembles S_1 et S_2.

$$Y \text{ est : } \left| \begin{array}{c} b \\ \hline c \quad d \quad c \rightarrow (\neg\Psi) \\ \hline e \end{array} \right.$$

$$S_1 \text{ est : } \left| \begin{array}{c} b \\ \hline c \qquad d \\ \hline e \end{array} \right.$$

$$S_2 \text{ est : } \left| \begin{array}{c} b \\ \hline d \qquad c \rightarrow (\neg\Psi) \\ \hline e \end{array} \right.$$

De même, un sous-ensemble S peut être issu de plusieurs thèses incl-préférées de E inconsistantes avec Ψ. Par exemple, la base E suivante admet deux thèses incl-préférées toutes inconsistantes avec $\Psi = a$ et il n'existe qu'un seul S issu de ces deux thèses :

$$E \text{ est : } \left| \begin{array}{c} (\neg a) \wedge b \qquad (\neg a) \wedge (\neg b) \\ \hline c \end{array} \right.$$

$$S \text{ est : } \left| \; c \right.$$

Exemple : Soient la formule $\Psi = (a \rightarrow v)$ et la base E suivante :

$$\begin{array}{c} \rightarrow p \\ p \rightarrow o \\ \hline p \rightarrow (\neg v) \\ o \rightarrow v \\ \hline o \rightarrow a \end{array}$$

Les 2 thèses incl-préférées de E sont :

$$\begin{array}{c|c} Y_1 = & Y_2 = \\ \{ \rightarrow p, & \{ \rightarrow p, \\ p \rightarrow o, & p \rightarrow o, \\ p \rightarrow (\neg v), & o \rightarrow v, \\ o \rightarrow a\} & o \rightarrow a\} \\ Y_1\ \Psi\text{-inconsistante.} & Y_2\ \Psi\text{-consistante.} \end{array}$$

Dans $E' = \Psi \oplus E$, on a 2 thèses incl-préférées :

$$\begin{array}{c|c} Y_1' = \{\Psi\} \cup S = & Y_2' = Y_2 \cup \{\Psi\} = \\ \{ a \rightarrow v, & \{ a \rightarrow v, \\ \rightarrow p, & \rightarrow p, \\ p \rightarrow o, & p \rightarrow o, \\ p \rightarrow (\neg v)\} & o \rightarrow v, \\ \text{avec } S \text{ partie de } E & o \rightarrow a, \\ & a \rightarrow v\} \end{array}$$

2.6.3.4 Mécanisme de génération Bo

Dans ce mécanisme de génération issu de l'ordre "Best-Out", on repère le numéro de la strate la plus prioritaire à partir de laquelle se produit une inconsistance ; dans E, ce numéro sera noté *amax* et dans $E' = \Psi \oplus E$, il sera noté *amax2*.

D'autre part, comme pour le mécanisme Incl et pour les mêmes raisons, la seule définition de $\oplus$ utilisée sera la définition 2.6.2 page 30.

On a alors les propriétés suivantes :

Propriété 2.6.1 *L'ensemble* $N(E) = \cup_{i=1,\ldots,\mathrm{amax}-1} E_i$ *appelé le* noyau *de E est consistant.*

Propriété 2.6.2 $\forall Y$*, sous-base bo-préférée de E,* $Y = N(E) \cup S$ *avec* $S \subset E_{\mathrm{amax}} \cup \ldots \cup E_n$ *et S noyau-consistant.*

Quelle que soit la formule Ψ que l'on rajoute en première strate à E pour constituer E', on se retrouve dans un des 4 cas suivants :

1. soit l'apparition de Ψ provoque une inconsistance dans une strate i de E' plus prioritaire que celle pointée par le *amax* de E, on en conclut que *amax2* est égal à i (voir figure 2.3 page suivante).
2. soit l'apparition de Ψ provoque une inconsistance dans une strate i de E' moins prioritaire que celle pointée par le *amax* de E, on en conclut que *amax2* est égal à *amax* (voir figure 2.3 page suivante).
3. soit l'apparition de Ψ provoque une inconsistance dans l'ancienne strate *amax* de E, on en conclut que *amax2* est égal à *amax* (voir figure 2.3 page suivante).

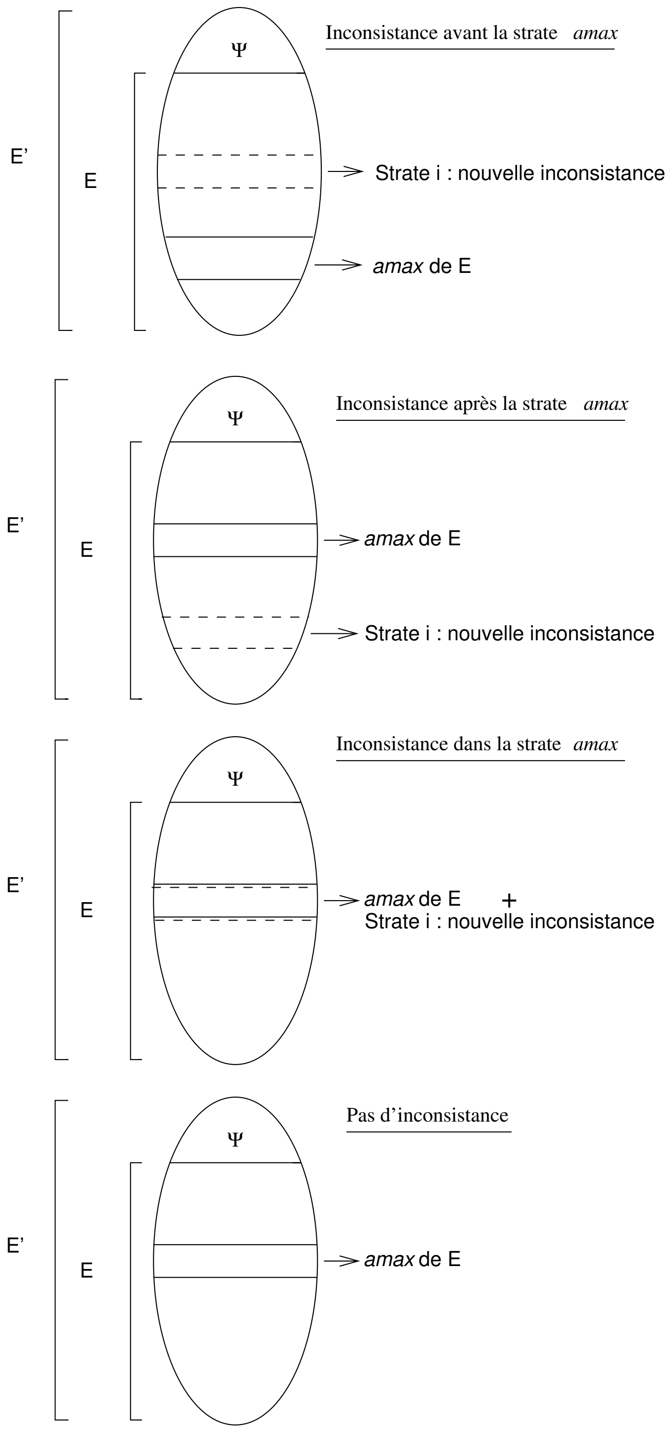


Figure 2.3 : Mécanisme de sélection Bo

4. soit l'apparition de Ψ ne provoque aucune nouvelle inconsistance, on en conclut que *amax2* est égal à *amax* (voir figure 2.3 page précédente).

On constate donc que, dans tous les cas, $amax2 \leq amax$. On obtient le théorème de caractérisation suivant :

Théorème de caractérisation 2.6.4 *Les sous-bases bo-préférées de* $E' = \Psi \oplus E$ *sont définies par :*

- *soit* $amax2 = amax$ *: les seules sous-bases bo-préférées de* E' *sont de la forme* $Y' = \{\Psi\} \cup Y$ *avec* Y *sous-base bo-préférée de* E Ψ*-consistante*[32] *; on a alors autant de* Y' *que de* Y Ψ*-consistantes ;*
- *soit* $amax2 < amax$ *: les seules sous-bases bo-préférées de* E' *sont de la forme* $Y' = \{\Psi\} \cup Z$ *où* Z *est un sous-ensemble de* E Ψ*-consistant et tel que* $N(E') \subseteq Z$*.*

On a alors la propriété suivante :

Propriété 2.6.3 $amax2 < amax \Rightarrow \forall Y$ *sous-base bo-préférée de* E*,* Y *est* Ψ*-inconsistante.*

Preuve : $\forall Y$, sous-base bo-préférée de E, $Y = N(E) \cup S$ avec $S \subset E_{\mathrm{amax}} \cup \ldots \cup E_n$ et S noyau-consistant ; or, si $amax2 < amax$, alors $\{\Psi\} \cup N(E)$ inconsistant ; donc $N(E)$ Ψ-inconsistant ; donc $\forall$ Y, Y est Ψ-inconsistante. $\square$

Rappelons aussi quelques propriétés données dans [BCD+93] :

Propriété 2.6.4 *Parmi les sous-bases bo-préférées de* E*, on trouve les sous-bases incl-préférées de* E*.*

Propriété 2.6.5 *(Cas particulier de la propriété 2.6.6) Les sous-bases bo-préférées de* E *sont les sous-bases consistantes de* E *contenant* $N(E)$ $(N(E) = \cup_{i=1,\ldots,\mathrm{amax}-1} E_i)$*.*

Propriété 2.6.6 *(Généralisation de la propriété 2.6.5) Soit* $amax(\Psi)$ *l'indice maximal tel que* $\cup_{i=1,\ldots,\mathrm{amax}(\Psi)-1} E_i$ *soit* Ψ*-consistant, alors les sous-bases de* E *qui sont bo-préférées parmi les sous-bases* Ψ*-consistantes de* E *sont les sous-bases consistantes contenant* $\cup_{i=1,\ldots,\mathrm{amax}(\Psi)-1} E_i$*.*

On en déduit un nouveau théorème de caractérisation :

Théorème de caractérisation 2.6.5 *Les sous-bases bo-préférées de* $\Psi \oplus E$ *sont exactement de la forme* $\{\Psi\} \cup S$ *avec* $S \subseteq E$ *sous-base bo-préférée parmi les sous-bases* Ψ*-consistantes de* E*.*

Preuve : On détermine $N(\Psi \oplus E) = \{\Psi\} \bigcup (\cup_{i=1,\ldots,\mathrm{amax2}-1} E_i)$; or, par définition, *amax2* correspond exactement à $amax(\Psi)$ l'indice maximal tel que $\cup_{i=1,\ldots,\mathrm{amax}(\Psi)-1} E_i$ soit Ψ-consistant ; donc en utilisant les propriétés 2.6.5 et 2.6.6, on obtient que $N(\Psi \oplus E) = \{\Psi\} \bigcup (\cup_{i=1,\ldots,\mathrm{amax}(\Psi)-1} E_i)$, d'où le théorème annoncé. $\square$

[32] En effet, $amax2 = amax \Rightarrow N(E') = \{\Psi\} \cup N(E) \Rightarrow Y' = N(E') \cup S' = \{\Psi\} \cup N(E) \cup S' = \{\Psi\} \cup Y$ avec Y Ψ-consistante.

Exemple : Je retrouve ici les deux cas présentés ci-dessus.

- Soit la base E suivante composée de 3 strates :

$$\begin{array}{c} E = \\ \{\ \rightarrow p, \\ p \rightarrow o, \\ \hline p \rightarrow (\neg v), \\ o \rightarrow v, \\ \hline o \rightarrow a\} \end{array}$$

Ici, $amax = 2$. Posons $\Psi = a \rightarrow v$. Les 6 sous-bases bo-préférées de E sont :

$Y_1 =$ $\{\ \rightarrow p,$ $p \rightarrow o\}$	$Y_2 =$ $\{\ \rightarrow p,$ $p \rightarrow o,$ $p \rightarrow (\neg v)\}$	$Y_3 =$ $\{\ \rightarrow p,$ $p \rightarrow o,$ $o \rightarrow v\}$	$Y_4 =$ $\{\ \rightarrow p,$ $p \rightarrow o,$ $o \rightarrow v,$ $o \rightarrow a\}$	$Y_5 =$ $\{\ \rightarrow p,$ $p \rightarrow o,$ $p \rightarrow (\neg v),$ $o \rightarrow a\}$	$Y_6 =$ $\{\ \rightarrow p,$ $p \rightarrow o,$ $o \rightarrow a\}$

avec Y_5 qui est Ψ-inconsistante et les autres qui sont Ψ-consistantes.
On a alors $amax2 = amax$ et les 5 sous-bases bo-préférées de E' sont :

$Y'_1 = \{\Psi\} \cup Y_1$ $=$ $\{\ a \rightarrow v,$ $\rightarrow p,$ $p \rightarrow o\}$	$Y'_2 = \{\Psi\} \cup Y_2$ $=$ $\{\ a \rightarrow v,$ $\rightarrow p,$ $p \rightarrow o,$ $p \rightarrow (\neg v)\}$	$Y'_3 = \{\Psi\} \cup Y_3$ $=$ $\{\ a \rightarrow v,$ $\rightarrow p,$ $p \rightarrow o,$ $o \rightarrow v\}$	$Y'_4 = \{\Psi\} \cup Y_4$ $=$ $\{\ a \rightarrow v,$ $\rightarrow p,$ $p \rightarrow o,$ $o \rightarrow v,$ $o \rightarrow a\}$	$Y'_5 = \{\Psi\} \cup Y_6$ $=$ $\{\ a \rightarrow v,$ $\rightarrow p,$ $p \rightarrow o,$ $o \rightarrow a\}$

- Soit la base E suivante composée de deux strates :

$$\begin{array}{c} a \\ \neg v \\ \hline a \rightarrow o \\ o \rightarrow v \end{array}$$

Ici, $amax = 2$. Posons $\Psi = a \rightarrow v$. Les 3 sous-bases bo-préférées de E sont :

$Y_1 =$ $\{\ a,$ $\neg v\}$	$Y_2 =$ $\{\ a,$ $\neg v,$ $a \rightarrow o\}$	$Y_3 =$ $\{\ a,$ $\neg v,$ $o \rightarrow v\}$

On a alors $amax2 = 1 < amax$ et les 11 sous-bases bo-préférées de E' sont :

$Y'_1 =$ $\{\ a \rightarrow v\}$	$Y'_2 =$ $\{\ a \rightarrow v,$ $a\}$	$Y'_3 =$ $\{\ a \rightarrow v,$ $a,$ $a \rightarrow o\}$	$Y'_4 =$ $\{\ a \rightarrow v,$ $a,$ $o \rightarrow v\}$	$Y'_5 =$ $\{\ a \rightarrow v,$ $a,$ $a \rightarrow o,$ $o \rightarrow v\}$	$Y'_6 =$ $\{\ a \rightarrow v,$ $\neg v\}$
$Y'_7 =$ $\{\ a \rightarrow v,$ $\neg v,$ $a \rightarrow o\}$	$Y'_8 =$ $\{\ a \rightarrow v,$ $\neg v,$ $o \rightarrow v\}$	$Y'_9 =$ $\{\ a \rightarrow v,$ $a \rightarrow o\}$	$Y'_{10} =$ $\{\ a \rightarrow v,$ $a \rightarrow o,$ $o \rightarrow v\}$	$Y'_{11} =$ $\{\ a \rightarrow v,$ $o \rightarrow v\}$	

2.6.3.5 Mécanisme de génération Car

Les sous-ensembles préférés de E par le mécanisme Car sont les thèses de cardinalité maximale. Notons n le cardinal des thèses car-préférées de E.

L'opérateur $\oplus$ peut correspondre soit à la définition 2.6.3 page 31, soit à la définition 2.6.2 page 30. Or, l'application de la définition 2.6.2 page 30 au mécanisme CAR constitue un cas particulier du mécanisme LEX ; elle sera donc prise en compte au travers du théorème de caractérisation 2.6.7 page suivante. Dans la section courante, nous allons donc nous contenter d'étudier l'impact de la définition 2.6.3 page 31 sur le mécanisme CAR ; on obtient ainsi le théorème de caractérisation 2.6.6.

Théorème de caractérisation 2.6.6 *Les thèses car-préférées Y' de $E \cup \{\Psi\}$ sont caractérisées par :*

Si $\exists Y$ telle que Y soit une thèse car-préférée de E et Y soit Ψ-consistante

- *alors $Y' = \{\Psi\} \cup Y$*
- *sinon*
  1. *soit $Y' = Y$ avec Y thèse car-préférée de E (donc inconsistante avec Ψ) ;*
  2. *soit $Y' = \{\Psi\} \cup S$ avec S est une sous-base de E consistante avec Ψ telle que $|S| = n - 1$.*

Preuve : Une sous-base car-préférée est un sous-ensemble consistant maximal pour la cardinalité. Remarquons que les sous-bases car-préférées de E ont toutes le même cardinal noté n. D'autre part, l'opérateur $\oplus$ correspond à l'union. On a deux cas possibles :

- soit il existe au moins une thèse car-préférée de E qui est Ψ-consistante ; posons alors $Y' = \{\Psi\} \cup Y$ avec Y thèse car-préférée Ψ-consistante ; sachant que $|Y| = n$, on a alors $|Y'| = n+1$; démontrons que Y' est car-préférée de $E \cup \{\Psi\}$; raisonnons par l'absurde en supposant que Y' n'est pas car-préférée, ceci implique qu'il existe Z' telle $|Z'| > |Y'|$; on a alors deux cas possibles :
  - soit Z' contient Ψ et alors il existe un sous-ensemble Z de E tel que $|Z| > n$ ce qui est en contradiction avec l'hypothèse que n est le cardinal maximal des sous-ensembles de E ;
  - soit Z' ne contient pas Ψ et alors Z' est un sous-ensemble de E tel que $|Z'| > n + 1$, donc $|Z'| > n$ ce qui est en contradiction avec l'hypothèse que n est le cardinal maximal des sous-ensembles de E ;

  donc $Y' = Y \cup \{\Psi\}$ est bien car-préférée de $E \cup \{\Psi\}$;
- soit aucune des thèses car-préférées de E n'est Ψ-consistante ; donc la cardinalité maximale pour les sous-bases de $E \cup \{\Psi\}$ est n ; donc toutes les thèses car-préférées de E sont aussi car-préférées pour $E \cup \{\Psi\}$; de plus, posons $Y' = \{\Psi\} \cup S$, on a alors deux cas possibles :
  - soit $|S| < n - 1$: $|S \cup \{\Psi\}| < n$ et alors $\forall Y$ sous-base car-préférée de E, $|S \cup \{\Psi\}| < |Y|$; alors $S \cup \{\Psi\}$ ne peut être une sous-base car-préférée de $\Psi \oplus E$;
  - soit $|S| = n - 1$: $|S \cup \{\Psi\}| = n$ et alors l'ensemble $Y' = S \cup \{\Psi\}$ a une cardinalité maximale de n, et les Y' ainsi définies sont aussi des car-préférées de $\Psi \oplus E$.

□

Propriété 2.6.7 *S'il existe des Y thèses car-préférées de E qui sont Ψ-consistantes :*

- *alors il y a autant de thèses car-préférées dans $E \cup \{\Psi\}$ que de thèses car-préférées de E Ψ-consistantes ;*
- *sinon il y a au moins autant de thèses car-préférées dans $E \cup \{\Psi\}$ que de thèses car-préférées de E.*

Exemple : Soit la base $E = \{\rightarrow p, p \rightarrow o, p \rightarrow (\neg v), o \rightarrow v, o \rightarrow a\}$ et la formule $\Psi = (a \rightarrow v)$. On a 4 thèses car-préférées dans E :

$Y_1 =$	$Y_2 =$	$Y_3 =$	$Y_4 =$
$\{ \rightarrow p,$	$\{ \rightarrow p,$	$\{ \rightarrow p,$	$\{ p \rightarrow o,$
$p \rightarrow o,$	$p \rightarrow o,$	$p \rightarrow (\neg v),$	$p \rightarrow (\neg v),$
$p \rightarrow (\neg v),$	$o \rightarrow v,$	$o \rightarrow v,$	$o \rightarrow v,$
$o \rightarrow a\}$	$o \rightarrow a\}$	$o \rightarrow a\}$	$o \rightarrow a\}$
Y_1 Ψ-inconsistante.	Y_2 Ψ-consistante.	Y_3 Ψ-consistante.	Y_4 Ψ-consistante.

Dans $E' = E \cup \{\Psi\}$, on a 3 thèses car-préférées :

$Y'_1 = Y_2 \cup \{\Psi\} =$	$Y'_2 = Y_3 \cup \{\Psi\} =$	$Y'_3 = Y_4 \cup \{\Psi\} =$
$\{ \rightarrow p,$	$\{ \rightarrow p,$	$\{ p \rightarrow o,$
$p \rightarrow o,$	$p \rightarrow (\neg v),$	$p \rightarrow (\neg v),$
$o \rightarrow v,$	$o \rightarrow v,$	$o \rightarrow v,$
$o \rightarrow a,$	$o \rightarrow a,$	$o \rightarrow a,$
$a \rightarrow v\}$	$a \rightarrow v\}$	$a \rightarrow v\}$

Autre exemple : La base E est la même que celle de l'exemple ci-dessus, on a donc les mêmes thèses car-préférées pour E. Par contre, on choisit $\Psi = ((a \rightarrow v) \wedge o \wedge (\neg v))$. Les 4 thèses car-préférées de E sont toutes Ψ-inconsistantes. On obtient 5 thèses car-préférées pour E' :

- $Y'_1 = Y_1$,
- $Y'_2 = Y_2$,
- $Y'_3 = Y_3$,
- $Y'_4 = Y_4$,
- $Y'_5 = \Psi \cup \{\rightarrow p, p \rightarrow o, p \rightarrow (\neg v)\}$.

Remarque : Je travaille dans un cadre syntaxique donc, bien que la formule Ψ soit un ensemble de trois clauses, elle est considérée comme une seule croyance.

Encore un exemple : La base $E = \{a, b\}$. Elle est consistante donc elle admet une seule thèse car-préférée : $Y = \{a, b\}$. On choisit $\Psi = ((\neg a) \wedge (\neg b))$. Y est Ψ-inconsistante. On obtient 1 thèse car-préférée pour E' : $Y' = Y$ (toujours à cause du traitement syntaxique des croyances).

2.6.3.6 Mécanisme de génération Lex

Les sous-ensembles préférés de E par le mécanisme Lex sont les thèses de E préférées lexicographiquement.

Pour les mêmes raisons que celles évoquées pour le mécanisme Incl, la seule définition utilisée pour l'opérateur $\oplus$ est la définition 2.6.2 page 30. On obtient ainsi le théorème 2.6.7.

Théorème de caractérisation 2.6.7 *Les thèses de $\Psi \oplus E$ lex-préférées sont de la forme $\{\Psi\} \cup S$ avec S sous-base de E consistante avec Ψ telle que S est lex-préférée parmi les sous-bases Ψ-consistantes de $(E, <)$.*

> **Preuve :** La démonstration se fait en deux étapes. La base E est stratifiée de la manière suivante : $E = E_1 \cup \ldots \cup E_n$ avec E_i la strate numéro i de E et $E_0 = \{\Psi\}$.

- condition suffisante : soit l'ensemble $Y = \{\Psi\} \cup S$ avec $S \subseteq E$, S consistante avec Ψ et S lex-préférée parmi les sous-bases de E consistantes avec Ψ, montrons par l'absurde que Y est lex-préférée dans $\Psi \oplus E$; si ce n'est pas le cas alors il existe $Z \subseteq \Psi \oplus E$ telle que Z est lex-préférée à Y ; c'est-à-dire, si on en revient à la définition (voir dans [CLS93]) : $\exists i$, $0 \leq i \leq n$, tel que $|Y_i| < |Z_i|$, et $\forall j < i$, $|Y_j| = |Z_j|$;
or $Y = \{\Psi\} \cup S \Rightarrow Y_0 \neq \varnothing \Rightarrow i \geq 1$ et $Z_0 = \{\Psi\} \Rightarrow \exists i$, $1 \leq i \leq n$, tel que $|S_i| = |Y_i| < |Z_i|$, et $\forall j < i$, $|Y_j| = |Z_j|$ ce qui voudrait dire que S n'est pas lex-préférée parmi les sous-bases Ψ-consistantes, donc contradiction.
- condition nécessaire : soit Y une thèse lex-préférée de $\Psi \oplus E$, alors Y est de la forme $\{\Psi\} \cup S$ avec $S \subseteq E$, S Ψ-consistante ; raisonnons par l'absurde et supposons que S ne soit pas lex-préférée parmi les sous-bases de E consistantes avec Ψ ; il existe alors $Z \subseteq E$ lex-préférée parmi les sous-bases de E Ψ-consistantes telle que Z est lex-préférée à S ; c'est-à-dire, si on en revient à la définition (voir dans [CLS93]) : $\exists i$, $1 \leq i \leq n$, tel que $|S_i| < |Z_i|$, et $\forall j < i$, $|S_j| = |Z_j|$;
ceci implique que $Z \cup \{\Psi\}$ est lex-préférée à $S \cup \{\Psi\} = Y$, ce qui est en contradiction avec l'hypothèse de départ.

$\square$

Propriété 2.6.8 *S'il existe des Y thèses lex-préférées de E qui sont Ψ-consistantes alors le nombre de thèses lex-préférées dans $\Psi \oplus E$ est égal au nombre de thèses lex-préférées de E Ψ-consistantes.*

Exemple : Soient la formule $\Psi = (a \rightarrow v)$ et la base E suivante :

$$\begin{array}{c} \rightarrow p \\ p \rightarrow o \\ \hline p \rightarrow (\neg v) \\ \hline o \rightarrow v \\ \hline o \rightarrow a \end{array}$$

Les 2 thèses lex-préférées de E sont :

$$\begin{array}{c|c} Y_1 = & Y_2 = \\ \{ \rightarrow p, & \{ \rightarrow p, \\ p \rightarrow o, & p \rightarrow o, \\ p \rightarrow (\neg v), & o \rightarrow v, \\ o \rightarrow a\} & o \rightarrow a\} \\ Y_1\ \Psi\text{-inconsistante.} & Y_2\ \Psi\text{-consistante.} \end{array}$$

Dans $E' = \Psi \oplus E$, on a 1 thèse lex-préférée :

$$\begin{array}{c} Y_1' = Y_2 \cup \{\Psi\} = \\ \{ a \rightarrow v, \\ \rightarrow p, \\ p \rightarrow o, \\ o \rightarrow v, \\ o \rightarrow a\} \end{array}$$

Autre exemple : La base E est inchangée. On prend pour $\Psi = \neg o$. Les deux thèses lex-préférées de E sont désormais toutes les deux Ψ-inconsistantes. On obtient ainsi 2 thèses lex-préférées pour E' :

$$
\begin{array}{c|c}
Y_1' = & Y_2' = \\
\{ \neg o, & \{ \neg o, \\
\rightarrow p, & p \rightarrow o, \\
p \rightarrow (\neg v), & p \rightarrow (\neg v), \\
o \rightarrow v, & o \rightarrow v, \\
o \rightarrow a\} & o \rightarrow a\}
\end{array}
$$

2.6.3.7 Mécanisme de génération E*

Je ne m'intéresse ici qu'au cas des défauts super-normaux (donc au mécanisme E* – voir section 2.2.3 page 16) puisque, dans ce cas là, il existe un lien entre ce mécanisme et celui de Brewka (voir [Bre89b]) qui va nous permettre d'obtenir un théorème de caractérisation.

L'opérateur $\oplus$ consiste à rajouter la formule Ψ à l'ensemble des prémisses W sans modifier l'ensemble des défauts D. Donc $E' = (W', D)$ avec $W' = \{\Psi\} \cup W$. On obtient ainsi le théorème de caractérisation 2.6.8.

Théorème de caractérisation 2.6.8 *Les extensions de $\Psi \oplus E$ sont caractérisées par une seule des situations suivantes :*

1. *soit W' est inconsistant, la seule extension possible est alors inconsistante (ce cas ne présente bien-sûr aucun intérêt) ;*
2. *soit W' est consistant et on peut alors garder l'arbre de recherche des extensions de E et recalculer les conclusions de chaque branche :*

État de la branche de l'arbre de E	*Résultat sur la construction de l'arbre pour E'*
inconsistance	*inconsistance*
extension Y	*extension Y' de E' = $C_n(Y \cup \{\Psi\})$ lorsque Y est Ψ-consistante*
échec car certaines hypothèses sur l'extension n'ont pas été vérifiées	*peut-être une solution si la seule hypothèse manquante est satisfaite par le fait que "$\Psi \in$ l'extension"*

On a ainsi deux cas possibles pour Y' :

- *soit $Y' = C_n(Y \cup \{\Psi\})$ lorsque Y est Ψ-consistante,*
- *soit $Y' = C_n(S \cup \{\Psi\})$ lorsque S est le résultat d'une branche dont l'hypothèse manquante était "$\Psi \in$ l'extension".*

Preuve : Il suffit de se ramener au mécanisme de construction des extensions décrit par Reiter dans [Rei80] et rappelé en section 2.2.3 page 16 pour se rendre compte que le rajout de la prémisse Ψ à l'ensemble W ne peut se traduire que par les 3 cas donnés dans le théorème 2.6.8. □

Propriété 2.6.9 *Si W' est consistant alors le nombre d'extensions de (W', D) est supérieur ou égal au nombre d'extensions Ψ-consistantes de (W, D).*

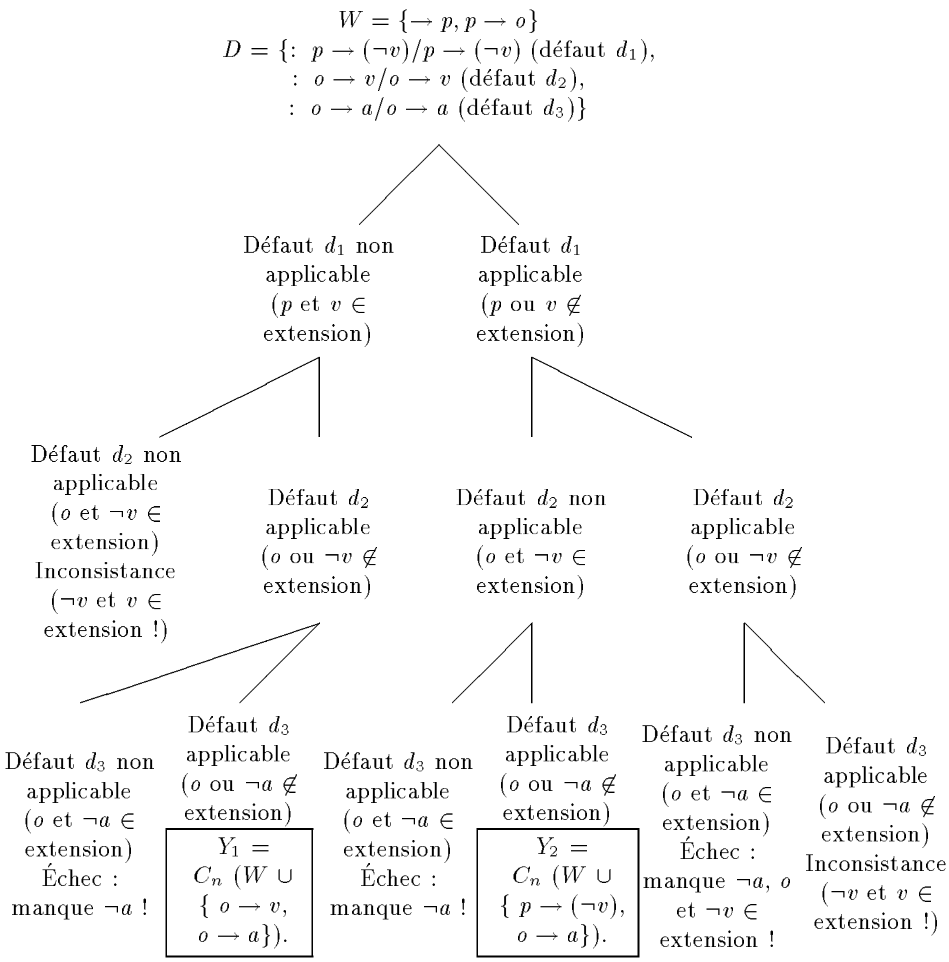


Figure 2.4 : Mécanisme de sélection E* : Arbre de recherche des extensions de E

Exemple : Soit la base $E = (W, D)$ avec :

- $W = \{\rightarrow p, p \rightarrow o\}$,
- $D = \{: \ p \rightarrow (\neg v)/p \rightarrow (\neg v), : \ o \rightarrow v/o \rightarrow v, : \ o \rightarrow a/o \rightarrow a\}$

et la formule $\Psi = (a \rightarrow v)$. On obtient 2 extensions pour E à l'aide de l'arbre de recherche de la figure 2.4. Parmi ces extensions, seule Y_1 est Ψ-consistante.

Puis, pour E' on prolonge l'arbre de recherche pour obtenir le résultat présenté dans la figure 2.5 page suivante, c'est-à-dire les deux extensions Y_1' et Y_2'. On retrouve alors Y_1 dans Y_1' (c'est le cas où $Y' = C_n(Y \cup \{\Psi\})$). Par contre, l'extension Y_2 de E qui était Ψ-inconsistante n'apparaît plus parmi les extensions de E'. Quant à l'extension Y_2', elle correspond au cas où $Y' = C_n(S \cup \{\Psi\})$.

2.6.4 Comparaison des deux définitions

Dans cette section, je cherche à établir des liens entre les deux définitions proposées pour une relation de classe 2 (donc du type $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$) : définitions 2.6.1 page 30 et 2.6.4 page 31 avec l'opérateur $\oplus$ correspondant à la définition 2.6.2 page 30.

Ces liens dépendent étroitement du principe p et du mécanisme de génération m.

Théorème 2.6.1 *Si* $p =$ UNI *alors* $\forall m \in \{$INCL, LEX, BO, E*$\}$,

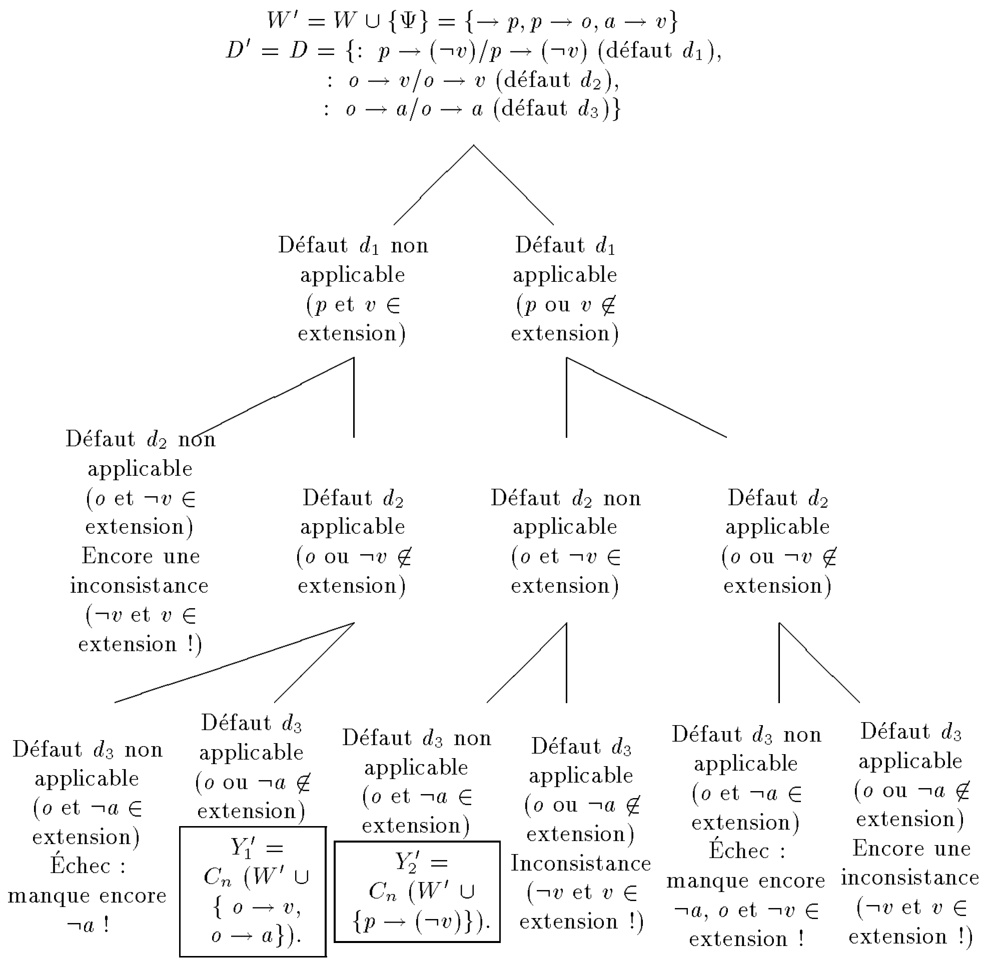


Figure 2.5 : Mécanisme de sélection E* : Arbre de recherche des extensions de $\Psi \oplus E$

$$(\Psi \oplus (E,<)) \mathrel{|\!\sim}^{p,m} \Phi \Rightarrow (E,<) \mathrel{|\!\sim}^{p,m} (\Psi \rightarrow \Phi).$$

La démonstration de ce théorème repose sur la propriété suivante :

Propriété 2.6.10 *Pour tout mécanisme $m \in \{$BO, INCL, LEX, E$^*\}$, $\forall Y$ sous-base m-préférée de E, si Y est Ψ-consistante alors $Y \cup \{\Psi\}$ est une sous-base m-préférée de $\Psi \oplus E$.*

> **Preuve de la propriété :** La preuve est donnée pour chaque mécanisme m par le théorème de caractérisation correspondant (voir les théorèmes 2.6.3 page 33, 2.6.4 page 37, 2.6.7 page 40 et 2.6.8 page 42). □
>
> **Preuve du théorème :**
>
> Prenons comme hypothèse que $p =$ UNI et que $(\Psi \oplus (E,<)) \mathrel{|\!\sim}^{p,m} \Phi$. Cela implique que $\forall Y'$ sous-base m-préférée de $\Psi \oplus E$, $Y' \vdash \Phi$.
>
> Étudions maintenant les sous-bases m-préférées de E. Soit Y une telle sous-base. On a alors deux cas possibles :
>
> - soit Y est Ψ-consistante et alors la propriété 2.6.10 s'applique et on sait que $Y' = Y \cup \{\Psi\}$ est une sous-base m-préférée de $\Psi \oplus E$, donc d'après l'hypothèse $Y \cup \{\Psi\} \vdash \Phi$, donc avec le théorème de la déduction $Y \vdash (\Psi \rightarrow \Phi)$;
> - soit Y est Ψ-inconsistante et alors $Y \vdash (\neg\Psi)$, donc trivialement $Y \vdash (\Psi \rightarrow \Phi)$.
>
> Ainsi, toute sous-base m-préférée de E infère $(\Psi \rightarrow \Phi)$, donc $(E,<) \mathrel{|\!\sim}^{\forall,m} (\Psi \rightarrow \Phi)$ □

Ce théorème correspond au fait que les relations UNI-m vérifient la propriété de conditionnalisation faible (voir les articles [Gär91, GM94] et la section 3.4.2 page 67 de ce document).

La réciproque de ce théorème est en général fausse (voir tous les exemples fournis dans la section 2.6.3 page 31 dans lesquels on constate que toutes les sous-bases m-préférées de $\Psi \oplus E$ ne sont pas forcément issues des sous-bases m-préférées de E).

Toutefois, j'ai identifié un cas où les deux définitions sont équivalentes ;

Théorème 2.6.2 $\forall m \in \{$INCL, LEX, BO, E$^*\}$, *soit une base E dont toutes les sous-bases m-préférées sont Ψ-consistantes, alors si $p =$ UNI on a*

$$(E,<) \mathrel{|\!\sim}^{p,m} (\Psi \rightarrow \Phi) \Rightarrow (\Psi \oplus (E,<)) \mathrel{|\!\sim}^{p,m} \Phi.$$

> **Preuve :** L'opérateur $\oplus$ correspondant à la définition 2.6.2 page 30, par construction, $\forall m \in \{$INCL, LEX, BO$\}$, toutes les sous-bases m-préférées de $\Psi \oplus E$ sont de la forme $Y' = Y \cup \{\Psi\}$ avec Y sous-base m-préférée de E et dans le cas où $m =$ E*, on sait que $Y' = C_n(Y \cup \{\Psi\})$ avec Y extension de E. D'autre part, par hypothèse, on sait que pour un m donné, $\forall Y$ sous-base m-préférée de E, $Y \vdash (\Psi \rightarrow \Phi)$, donc avec le théorème de déduction $Y \cup \{\Psi\} \vdash \Phi$. On en conclut que $\forall Y'$, $Y' \vdash \Phi$ □

2.6.5 Définition utilisée pour les relations de classe 2

J'ai choisi d'utiliser la définition 2.6.1 page 30, avec l'opérateur $\oplus$ donné par la définition 2.6.2 page 30, pour les raisons suivantes :

- c'est une définition bien connue dans la littérature et qui reflète particulièrement bien le point de vue de la révision ;

- et, dans certains cas, elle est plus générale que la définition 2.6.4 page 31 (voir le théorème 2.6.1 page 43).

Définition 2.6.5 *Soit une base stratifiée* $(E, <)$*, soit* p *un principe d'inférence, soit* m *un mécanisme de génération, la relation d'inférence G-p-m est définie par l'ensemble des couples* (Ψ, Φ) *tels que* $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$ *avec* $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$ *correspondant aux définitions 2.6.1 page 30 et 2.6.2 page 30, c'est-à-dire :*

$\Psi \mid\!\sim^{p,m}_{E,<} \Phi$ *ssi* $(\Psi \oplus (E, <)) \mid\!\sim^{p,m} \Phi$ *avec* $(\Psi \oplus (E, <)) = (E \cup \{\Psi\}, <')$*,*

l'ordre $<'$ *étendant l'ordre* $<$ *en rajoutant une première strate à* E *ne contenant que* Ψ*.*

Remarque : Dans le cadre des relations d'inférence de classe 2, je cesse donc de m'intéresser aux mécanismes de sélection T, S et Car puisqu'ils ne prennent pas du tout en compte ni la stratification de E, ni la position particulière de Ψ.

Avec la définition 2.6.1 page 30, on a la propriété suivante, qui récapitule ainsi une partie de la section 2.6.3 page 31 :

Propriété 2.6.11 $\forall m \in \{\text{Bo}, \text{Incl}, \text{Lex}\}$*, les sous-bases m-préférées de* $\Psi \oplus E$ *sont exactement de la forme* $\{\Psi\} \cup S$ *avec* $S \subseteq E$ *sous-base m-préférée parmi les sous-bases* Ψ*-consistantes de* E*.*

2.6.6 Lien entre les deux classes de relations d'inférence non-monotone

Les deux classes de relations d'inférence non-monotone *p-m* (classe 1) et *G-p-m* (classe 2) sont doublement liées. Tout d'abord, on peut constater que la première classe est un cas particulier de la seconde classe.

Théorème 2.6.3 *Si* Ψ *est une tautologie alors on a l'équivalence suivante :*

$\Psi \mid\!\sim^{p,m}_{E,<} \Phi \Leftrightarrow E \mid\!\sim^{p,m} \Phi$

Preuve : Cela est dû à la définition choisie pour $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$ (voir définition 2.6.5). □

Et ensuite, la seconde classe peut s'exprimer dans les mêmes termes que la première puisque $\Psi \mid\!\sim^{p,m}_{E,<} \Phi \Leftrightarrow \Psi \oplus E \mid\!\sim^{p,m} \Phi$, et que $\Psi \oplus E$ peut être vue comme une nouvelle base E'.

Ces remarques sont importantes car elles justifient le fait qu'une partie de mon étude porte sur les relations de la classe 1 et une autre partie sur celles de la classe 2, et que malgré tout je vais quand même pouvoir faire une synthèse des différents résultats.

2.7 Récapitulatif des relations à étudier

Il s'agit ici de synthétiser les sections précédentes (voir les tableaux 2.4 page ci-contre et 2.5 page suivante).

Légende des tableaux : Nous trouvons en colonne, les différents mécanismes de définition d'un ensemble P^E et d'un ordre $\ll$ sur P^E et nous avons en ligne, les différents principes d'inférence considérés (uniquement les principaux).

	Sous-bases consistantes sans ordre	Sous-bases consistantes avec ordre "Best-Out"	Thèses sans ordre	Thèses avec ordre basé sur l'inclusion	Thèses avec ordre basé sur la cardinalité	Thèses avec ordre lexicographique	Extensions Logique des défauts sans ordre (cas général ou cas des défauts super-normaux)
Uni	Uni-S		Uni-T				Uni-E ou Uni-E*
Exi	Exi-S		Exi-T				Exi-E ou Exi-E*
Arg	Arg-S		Arg-T				Arg-E ou Arg-E*
Uni-Pref		Uni-Bo		Uni-Incl	Uni-Car	Uni-Lex	
Exi-Pref		Exi-Bo		Exi-Incl	Exi-Car	Exi-Lex	
Arg-Pref		Arg-Bo		Arg-Incl	Arg-Car	Arg-Lex	

Tableau 2.4 : Les relations d'inférence non-monotone de la classe 1 notées *p-m*

	Sous-bases consistantes avec ordre "Best-Out"	Thèses avec ordre basé sur l'inclusion	Thèses avec ordre lexicographique	Extensions Logique des défauts sans ordre (cas général ou cas des défauts super-normaux)
Uni				*G*-Uni-E ou *G*-Uni-E*
Exi				*G*-Exi-E ou *G*-Exi-E*
Arg				*G*-Arg-E ou *G*-Arg-E*
Uni-Pref	*G*-Uni-Bo	*G*-Uni-Incl	*G*-Uni-Lex	
Exi-Pref	*G*-Exi-Bo	*G*-Exi-Incl	*G*-Exi-Lex	
Arg-Pref	*G*-Arg-Bo	*G*-Arg-Incl	*G*-Arg-Lex	

Tableau 2.5 : Les relations d'inférence non-monotone de la classe 2 notées *G-p-m*

Pourquoi un tel découpage ? Tout d'abord, l'ensemble P^E peut être constitué par 3 types de sous-bases consistantes (voir sections précédentes) :

- soit de sous-bases consistantes quelconques,
- soit uniquement de thèses (sous-bases maximales consistantes),
- soit d'extensions au sens de la logique des défauts ; les défauts super-normaux de la logique des défauts constituent un cas particulier qui me sera très utile tout le long de cette étude et c'est la raison pour laquelle ce mécanisme est mis en évidence ici au même titre que le cas général de la logique des défauts.

De plus, dans les sections précédentes, j'ai aussi répertorié 4 types de préférences $\ll$ sur des sous-ensembles consistants de formules :

- soit un ordre de type "Inclusion Based" ([Bre89b, CRS92, DLP91]), prévu pour ordonner des thèses[33], et qui est une généralisation de l'utilisation de thèses ;
- soit l'ordre basé sur la cardinalité ([DLP91]), prévu pour ordonner initialement des sous-bases consistantes issues d'une base non ordonnée, mais pour lequel les éléments préférés sont par construction des thèses ;
- soit l'ordre lexicographique ([DLP91]), qui est un raffinement de l'ordre basé sur la cardinalité, tout comme l'ordre "Inclusion Based" est un raffinement de l'utilisation de thèses ; cet ordre est lui aussi prévu pour ordonner initialement des sous-bases consistantes mais ses éléments préférés sont par construction des thèses ;
- soit l'ordre "Best-Out" ([BCD$^+$93, DLP91]), prévu pour ordonner des sous-bases consistantes.

Partout où les principes Uni, Exi, Arg s'appliquent, on obtient une relation d'inférence notée respectivement Uni-X, Exi-X, Arg-X (ou G-Uni-X, G-Exi-X, G-Arg-X) avec X remplacé par une lettre explicitant le type de l'ensemble P^E utilisé (S = sous-bases consistantes, T = thèses, E = extensions).

Partout où les principes Uni-Pref, Exi-Pref, Arg-Pref s'appliquent, c'est-à-dire quand on prend en compte les préférences, on obtient une relation d'inférence notée respectivement Uni-Pref, Exi-Pref, Arg-Pref (ou G-Uni-Pref, G-Exi-Pref, G-Arg-Pref) avec Pref remplacé par un mot explicitant l'ordre utilisé (Bo = ordre "Best-Out", Incl = ordre basé sur l'inclusion, Car = ordre basé sur la cardinalité, Lex = ordre lexicographique) et donc, par là même, le type de l'ensemble P^E (sous-bases consistantes pour Bo et thèses pour Incl, Car et Lex).

Je me trouve donc devant 21 relations d'inférence non-monotone de classe 1 et 12 relations d'inférence non-monotone de classe 2, qu'il va falloir étudier[34].

On pose donc les notations suivantes :

> Soient la base $(E,<)$, les formules Φ et Ψ, le principe d'inférence $p \in \{$Uni, Exi, Arg$\}$, et le mécanisme de sélection $m \in \{$S, T, Incl, Bo, Car, Lex, E, E$^*\}$:
>
> - $(E,<) \mid\!\sim^{p,m} \Phi$ (relation de classe 1) signifie que la formule Φ est conséquence non-monotone de $(E,<)$ selon m et p ;
> - $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$ (relation de classe 2) signifie que Ψ est conséquence non-monotone de la formule Φ selon m et p dans le contexte $(E,<)$.

[33] En fait, suivant les auteurs, l'ordre "Inclusion Based" est prévu pour ordonner des thèses ou simplement des sous-bases consistantes. Par exemple, Brewka l'a défini sur des thèses, alors que l'ordre démocratique de Cayrol, Royer et Saurel est applicable sur des sous-bases consistantes. Toutefois, toute sous-base préférée pour l'ordre "Inclusion Based" est une thèse. C'est la raison pour laquelle cet ordre ne sera étudié que sur des thèses.

[34] On peut rajouter des lignes à mon tableau en définissant de nouveaux principes d'inférence (par exemple ceux cités par Pinkas et Loui), ou des colonnes en définissant de nouveaux couples $(P^E, \ll)$.

Chapitre 3

État de l'art sur quelques critères de comparaison théoriques

3.1 Liste de quelques critères possibles, choix et motivations de ces choix

J'ai élu la complexité, la prudence et les propriétés de déduction comme les trois critères de comparaison à utiliser pour les raisons suivantes :

- la complexité est un critère capital quand on pense efficacité ; que va coûter la réponse à la question "la base E infère-t-elle la formule Φ" ? le coût de cette réponse peut être soit d'ordre temporel, soit d'ordre spatial et l'expérience nous apprend que c'est le coût en temps qui pose problème en premier (voir [GJ79, Neb91, Got92]) ; je m'attacherai donc à étudier la complexité temporelle des processus de déduction non-monotone ;
- la prudence est un critère subjectif et dépendant du contexte ; malgré cela, elle reste un critère important car elle cherche à exprimer le réalisme d'une relation du point de vue du nombre de conclusions à obtenir (plus une relation est prudente et moins elle produit de conclusions, donc moins elle risque d'inférer des conclusions erronées tout en risquant d'oublier d'en inférer certaines) (voir [PL92]) ;
- les propriétés de déduction sont un critère important car elles définissent une approche axiomatique de l'inférence non-monotone et ce faisant elles représentent un "idéal"[1] vers lequel ces relations devraient tendre ; cet idéal a été défini par quelques spécialistes du domaine sous diverses formes qui se recoupent à peu près toutes ; ce sont les propriétés de déduction qui prouvent l'adéquation de la relation au raisonnement que l'on cherche à modéliser au travers des dites relations (voir [KLM90, GM94]).

Notons qu'il existe d'autres critères, que l'on peut étudier, mais qui ne seront pas traités ici : insensibilité à la redondance, à la forme clausale ...(voir par exemple [BDP94b]).

3.2 Complexité

Les rappels fournis ici sont volontairement succincts et les lecteurs intéressés pourront se reporter aux ouvrages de référence dans le domaine (par exemple, [GJ79]).

La complexité à laquelle on s'intéresse est la complexité temporelle[2] de résolution d'un problème

[1] Toutefois, mon étude montre que cet "idéal" n'est pas encore parfait, car il existe des relations d'inférence le vérifiant et présentant par ailleurs des problèmes importants, tout comme il existe des relations d'inférence ne le vérifiant pas et qui demeurent pourtant des relations intéressantes par d'autres aspects.

[2] Il existe aussi une complexité spatiale (voir [GJ79]).

dans le *pire des cas.* Il s'agit de caractériser la façon dont évolue le temps de résolution d'un problème suivant la taille des données et ceci dans le pire cas, afin d'obtenir une borne supérieure de la complexité réelle. Naturellement, les notions de temps, de résolution et de problème devront être formalisées.

Une distinction est habituellement faite entre les problèmes dits "faciles", dont la complexité dans le pire des cas est bornée par une fonction polynomiale de la taille des données (exemple : tri d'une liste d'entiers) et les autres problèmes (exponentiels, super exponentiels...). Ce premier classement voit son intérêt fortement limité par la difficulté qu'il peut y avoir à montrer qu'un problème *ne peut pas être résolu en temps polynomial* : pour de très nombreux problèmes, d'utilité pratique importante, on ne connaît pas d'algorithme polynomial et on n'a pas pu démontrer qu'il n'en existait pas.

La théorie de la complexité offre un palliatif à cette situation en introduisant la notion d'algorithmes non déterministes polynomiaux (NP, contenant la classe des algorithmes polynomiaux) et en distinguant parmi ceux-ci la classe particulièrement riche des algorithmes "les plus difficiles de NP" (ou NP-complets : elle contient de très nombreux problèmes classiques de logique, de recherche opérationnelle...) telle que si l'un de ces problèmes se résout en temps polynomial (personne n'a encore réussi cet exploit), alors tous les problèmes de NP se résolvent en temps polynomial.

Dire d'un problème qu'il est NP-complet montre alors que le problème est aussi difficile que bien d'autres problèmes pour lesquels une gigantesque communauté de chercheurs n'a pas trouvé d'algorithme polynomial depuis plus de 30 ans.

Cette simple distinction P/NP a été ensuite enrichie via la définition d'une hiérarchie de classes de problèmes, de plus en plus difficiles. Dans le cas qui m'occupe, je cherche à situer dans cette hiérarchie, les différents mécanismes d'inférence afin de pouvoir les comparer.

Avant toute précision supplémentaire, voyons comment l'on peut formaliser les notions de problème et d'algorithme.

3.2.1 Qu'est-ce-qu'un problème ? Comment le formaliser ?

Définition 3.2.1 *Un problème est une question générique (qui s'applique à un ensemble d'éléments).*

Exemple de problème : "Déterminer si un entier naturel est pair ou impair".

Définition 3.2.2 *Une instance du problème est la question posée pour un élément particulier de l'ensemble. C'est donc une version du problème dans laquelle les paramètres non spécifiés dans le problème initial ont été instanciés.*

Exemple d'un problème et d'une instance de problème : le problème "Soit A un nombre entier naturel, A est-il pair ou impair ?" admet comme instances "3 est-il pair ou impair ?", "15624 est-il pair ou impair ?" etc.

Afin de simplifier les notations, une instance d'un problème donné Q ne sera notée qu'avec l'élément particulier qui la définit. En effet, pour un problème donné et quelle que soit son instance, la question posée reste toujours la même. Ainsi, au lieu de dire que "3 est-il pair ou impair ?" est une instance du problème Q, on dira que "3" est une instance de Q.

À chaque instance de problème est associée une réponse.

Définition 3.2.3 *Un algorithme résout un problème s'il peut être appliqué à toute instance du problème pour fournir une réponse en un temps fini.*

Définition 3.2.4 *Un problème de décision est un problème dont la réponse est oui ou non, quelle que soit l'instance du problème.*

Exemple : “Soit A un nombre entier naturel, A est-il premier ?”.

On parlera donc d'instances positives (celles dont la réponse est oui), et d'instances négatives (celles dont la réponse est non). “3” est une instance positive et “15624” est une instance négative du problème défini par “Soit A un entier naturel, A est-il premier ?”.

Définition 3.2.5 *Un problème est décidable s'il existe un algorithme pour résoudre toutes ses instances, sinon il est dit indécidable.*

En complexité, on n'étudie que les problèmes décidables.

À chaque instance I d'un problème Q, on associe par une fonction de codage e un mot d'un langage formel L (voir [GJ79]). La fonction d'encodage n'est pas quelconque et doit être raisonnable (non redondante et décodable). Désormais, chaque problème Q sera vu à travers son langage L. On identifiera Q et L. On appelle taille de l'instance I, la taille du mot $e(I)$.

Définition 3.2.6 *Dans le cas des problèmes de décision, on définit la notion de problème complémentaire : soit un problème Q, le problème complémentaire de Q, noté co-Q, est le problème dont la réponse est oui quand celle de Q est non et vice-versa.*

Remarque : La notation “co-” s'applique aussi aux langages : L et co-L.

Exemple : Le co-problème de “Soit A un nombre entier naturel, A est-il premier ?” sera : “Soit A un nombre entier naturel, A est-il le produit de 2 nombres entiers naturels strictement inférieurs à A ?”.

3.2.2 Les machines de Turing et les classes de problèmes

Nous avons vu à la section précédente que l'on peut coder une instance d'un problème donné. Il faut maintenant définir une machine sur laquelle résoudre le problème encodé. Cette machine est la machine de Turing. C'est un automate auquel on prête la propriété suivante : toute fonction intuitivement calculable peut l'être par une machine de Turing[3].

Définition 3.2.7 *On dit qu'une machine de Turing résout un problème Q, quand, pour toute instance I de Q et pour un encodage donné, la machine s'arrête en renvoyant la réponse oui quand I est une instance positive (on dit que le mot représentant I est accepté par la machine) et la réponse non quand I est une instance négative (on dit que le mot représentant I est rejeté par la machine).*

Définition 3.2.8 *La complexité temporelle d'un algorithme est une fonction qui associe à la taille n d'une instance (la longueur du mot représentant l'instance) le nombre de pas d'exécution effectués par la machine jusqu'à trouver la solution et ce pour le pire cas (sur tous les mots de longueur n).*

Il existe plusieurs types de machines de Turing, chacune permettant de définir une classe de problèmes.

Définition 3.2.9 *Les machines déterministes*[4] *induisent la classe* P *des problèmes dits polynomiaux :*

[3] C'est la thèse de Church-Turing.

[4] Informellement, une machine de Turing déterministe est une machine de Turing qui, à partir d'un état donné, ne peut passer que dans un seul autre état. Rappelons que les machines de Turing sont des automates passant d'un état à un autre en fonction des entrées et produisant des sorties.

un problème est polynomial, s'il existe un algorithme pour le résoudre sur une machine de Turing déterministe dont la complexité est bornée par une fonction polynomiale de la taille du problème.

Définition 3.2.10 *Les machines non déterministes*[5] *induisent la classe* NP *des problèmes dits non déterministes polynomiaux :*

un problème est non déterministe polynomial, s'il existe un algorithme pour le résoudre sur une machine de Turing non déterministe dont la complexité est bornée par une fonction polynomiale de la taille du problème[6]*.*

Définition 3.2.11 *Les machines déterministes à oracle de complexité* X[7] *induisent la classe* P^X *:*

un problème est de complexité P^X*, s'il existe un algorithme pour le résoudre sur une machine de Turing déterministe à oracle de complexité* X *dont la complexité est bornée par une fonction polynomiale de la taille du problème.*

Définition 3.2.12 *Les machines non déterministes à oracle de complexité* X *induisent la classe* NP^X *:*

un problème est de complexité NP^X*, s'il existe un algorithme pour le résoudre sur une machine de Turing non déterministe à oracle de complexité* X *dont la complexité est bornée par une fonction polynomiale de la taille du problème.*

On peut définir une hiérarchie de classes de problèmes. C'est la hiérarchie polynomiale (notée PH) :

- $\Sigma_0^p = \Delta_0^p = \Pi_0^p = \mathsf{P}$ (machines déterministes)
- $\Delta_{k+1}^p = \mathsf{P}^{\Sigma_k^p}$ (machines déterministes à oracle de complexité Σ_k^p)
- $\Sigma_{k+1}^p = \mathsf{NP}^{\Sigma_k^p}$ (machines non déterministes à oracle de complexité Σ_k^p)
- $\Pi_{k+1}^p = \text{co-}\Sigma_{k+1}^p$ (complémentaire)

Remarque : $\Delta_1^p = \mathsf{P}$, $\Sigma_1^p = \mathsf{NP}$, $\Pi_1^p = \text{co-}\mathsf{NP}$.

Quand un problème Q est de complexité X, alors son co-problème sera de complexité co-X.

Introduisons trois dernières notions : la transformation polynomiale, la complétude et la notion de problème X-difficile.

Définition 3.2.13 *Une transformation polynomiale est une fonction* f *calculable en temps polynomial permettant de "passer" d'un langage* L_1 *à un langage* L_2 *(donc d'un problème* Q_1 *à un problème* Q_2*) – voir définition précise dans l'annexe A page 235. On note :* $L_1 \propto L_2$*.*

Définition 3.2.14 *Un problème* Q *est X-complet avec X représentant une classe de complexité, ssi* Q *est de complexité X et* $\forall Q'$ *un problème de complexité X,* Q' *se ramène à* Q *par une transformation polynomiale.*

Définition 3.2.15 *Un problème* Q *est X-difficile avec X représentant une classe de complexité, ssi* $\forall Q'$ *un problème de complexité X,* Q' *se ramène à* Q *par une transformation polynomiale.*

[5] Par opposition aux machines déterministes, dans une machine non déterministe, à partir d'un état donné, on peut passer dans un état *choisi* parmi plusieurs états différents. Un problème NP est donc un problème qui peut se résoudre efficacement si on sait toujours faire les bons choix.

[6] La complexité temporelle d'un algorithme dans le cas d'une machine de Turing non déterministe = le maximum sur toutes les instances positives, donc pour tous les mots $e(I)$ correspondants, du temps de reconnaissance de ces mots ; sachant que ce temps de reconnaissance pour un mot $e(I)$ donné = le minimum sur tous les choix possibles du nombre de pas d'exécution effectués jusqu'à trouver la solution.

[7] Informellement, une machine de Turing à oracle de complexité X est une machine de Turing faisant appel à un sous-programme exécuté sur une autre machine sachant résoudre les problèmes de la classe X. Chaque appel compte alors pour un pas d'exécution de la machine appelante.

3.2.3 Les exemples de problèmes

Exemple d'un problème de classe P : "Soient A et B, 2 ensembles finis, A est-il inclus dans B ?".

Exemple d'un problème NP-complet : le problème de satisfiabilité en logique propositionnelle "Soit Φ une formule propositionnelle, Φ est-elle satisfiable ?". Ici je m'intéresse au problème de satisfiabilité d'une formule propositionnelle quelconque, appelé "Satisfiability of boolean expressions" dans [GJ79] et noté le plus souvent GSAT. Il existe aussi un problème SAT défini dans [GJ79] qui ne concerne que la satisfiabilité d'une formule propositionnelle exprimée sous forme normale conjonctive. Ces deux problèmes sont de même complexité. Remarquons que dans le cas de la logique du premier ordre le problème de la satisfiabilité d'une formule est indécidable.

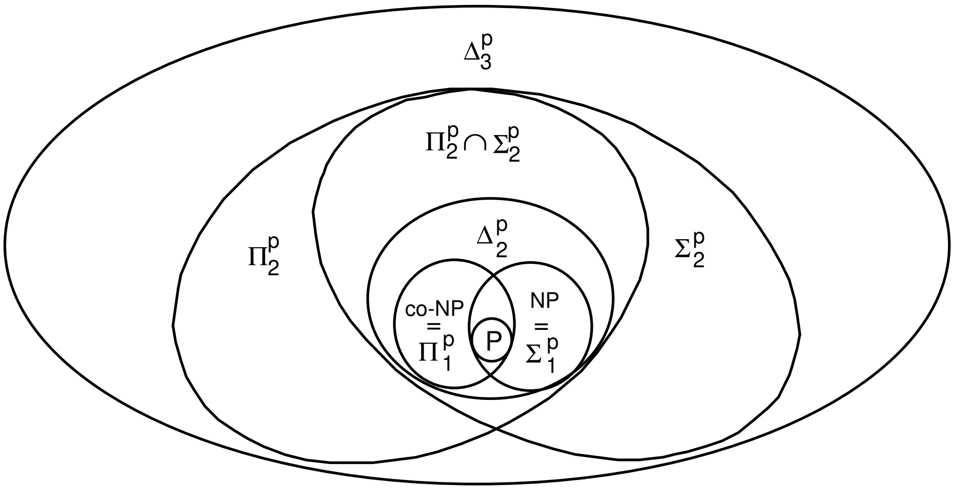
Exemple d'un problème co-NP-complet : le complémentaire du problème de satisfiabilité en logique propositionnelle "Soit Φ une formule propositionnelle, Φ est-elle non satisfiable ?". Ce problème est appelé UNGSAT[8] (ou co-GSAT ou GVALID[9]).

Exemple d'un problème Σ_2^p-complet : le problème 2-QBF défini par : "La formule $\exists a \forall b H(a,b)$ est-elle satisfiable ?". $\exists a \forall b H(a,b)$ est une formule d'une méta-logique dans laquelle a et b sont 2 vecteurs représentant les assignations des ensembles de variables propositionnelles $(a_1, \ldots, a_n)$ et $(b_1, \ldots, b_m)$. "$\exists a \forall b H(a,b)$ satisfiable" signifie donc : "il existe une assignation a des variables propositionnelles $a_1, \ldots, a_n$, telle que, quelque soit l'assignation b des variables propositionnelles $b_1, \ldots, b_m$, la formule $H(a,b)$ est vraie". (voir dans [Neb91] et [Joh90]). On note parfois : 2-QBF-$\exists$.

Exemple d'un problème Π_2^p-complet : le complémentaire du problème 2-QBF "La formule $\exists a \forall b H(a,b)$ est-elle non satisfiable ?". Ce qui est équivalent au problème "La formule $\forall a \exists b \neg H(a,b)$ est-elle satisfiable ?", qui est parfois noté : 2-QBF-$\forall$.

3.2.4 Représentation de la hiérarchie polynomiale

etc.



P est inclus dans NP et dans co-NP eux-mêmes inclus dans Δ_2^p, etc.
On ne sait pas à l'heure actuelle si ces inclusions sont strictes. Si P = NP alors toute la hiérarchie polynomiale s'effondre sur P (voir section 3.2.5 page suivante).

[8] Même remarque que pour GSAT.
[9] La non satisfiabilité d'une formule est équivalente à la validité de sa négation.

3.2.5 Les conjectures de la théorie de la complexité

Il existe tout un ensemble de conjectures en théorie de la complexité, entre autres :

Conjecture 3.2.1 $\mathsf{P} \neq \mathsf{NP}$.

Conjecture 3.2.2 $\mathsf{NP} \neq co\text{-}\mathsf{NP}$.

Conjecture 3.2.3 $\Sigma_k^p \neq \Pi_k^p \ \forall k$.

Ces conjectures ont fait l'objet de recherches intensives de la part des mathématiciens et des théoriciens de l'informatique depuis plusieurs dizaines d'années. Et personne n'a encore réussi à prouver leur véracité ou leur fausseté.

Remarquons qu'il "suffirait" de trouver un algorithme de complexité polynomiale pour un problème connu comme étant NP-complet pour que les classes NP et co-NP soient ramenées à la classe P. On obtiendrait ainsi un "tassement" de la hiérarchie polynomiale.

Ces conjectures sont sous-entendues dans tout ce qui suit.

3.3 Prudence

Dans cette section, je rappelle la notion de prudence telle qu'on la rencontre dans la littérature (voir [PL92, Bra93]) :

Définition 3.3.1 *On dira que R est une relation de conséquence ssi $R = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ avec $\forall i$, X_i et Y_i sont des formules et $X_i \mid\!\sim^R Y_i$.*

Définition 3.3.2 *Soient R_1 et R_2, deux relations de conséquence telles que définies par la définition 3.3.1. On dit que R_1 est plus prudente que R_2 (noté R_1 +prec R_2) ssi R_1 est incluse dans R_2 (l'ordre +prec est un ordre partiel).*

Les définitions 3.3.1 et 3.3.2 issues de [PL92] conviennent parfaitement à l'étude des relations d'inférence que je note *G-p-m*. Dans [PL92], on trouve un schéma représentant l'ordre partiel de prudence entre certaines relations de conséquence. C'est ce schéma de base que je vais raffiner dans la section 4.2 page 151 en introduisant différents types de préférence.

Dans la littérature, on trouve aussi le cas particulier suivant :

Définition 3.3.3 *On dira que R est une relation de conséquence ssi $R = \{(E, X_1), \ldots, (E, X_n)\}$ avec $\forall i$, X_i est une formule et $E \mid\!\sim^R X_i$.*

Définition 3.3.4 *Soit une base de croyances E, soient R_1 et R_2 deux relations d'inférence non-monotone telles que définies par la définition 3.3.3, on dit que R_1 est plus prudente que R_2 ssi l'ensemble des formules déduites de E par R_1 est inclus dans l'ensemble des formules déduites de E par R_2.*

Les définitions 3.3.3 et 3.3.4 sont celles utilisées dans [Bra93]. Elles conviennent à l'étude des relations d'inférence que je note *p-m*.

3.4 Propriétés de déduction

Quelle que soit la relation d'inférence non-monotone que nous définissons, il reste une question essentielle : "Cette relation est-elle acceptable ?". Donc, pour être plus précis : "Quelles sont les propriétés que doit vérifier une relation d'inférence non-monotone afin de définir un raisonnement non-monotone "acceptable" et que recouvre cette notion d'"acceptabilité" ?".

Les principaux papiers abordant ce sujet sont [KLM90, LM92, Gär91, GM94] et je vais présenter un ensemble de propriétés de déduction issues de ces articles avec la sémantique associée.

3.4.1 Les propriétés de Kraus, Lehmann et Magidor et leur sémantique

Dans [KLM90], Kraus, Lehmann et Magidor proposent une classification des relations d'inférence non-monotone, qu'ils appellent *relations de conséquence*. Le langage L utilisé est un ensemble de formules propositionnelles et est fermé pour les connecteurs propositionnels classiques ($\wedge$, $\vee$, $\rightarrow$, $\neg$). On suppose aussi que l'*hypothèse de compacité* est vérifiée.

Définition 3.4.1 *Un langage L est compact ssi $\forall a$ conséquence logique d'un ensemble $A \subseteq L$, alors $\exists A'$ sous-ensemble fini de A tel que a soit conséquence logique de A'.*

On a ainsi la propriété suivante :

Propriété 3.4.1 *(***Propriété de compacité***) Un ensemble de formules d'un langage compact L est consistant ssi tous ses sous-ensembles finis le sont.*

Définition 3.4.2 *Soit le langage L, soient Ψ et Φ, 2 formules de L, on a alors :*

- *$\Psi \mid\sim \Phi$ est appelée assertion conditionnelle (ou plus simplement assertion), cela signifie "si on a Ψ alors on a normalement Φ" ou "Φ est une conséquence plausible de la formule Ψ" ; on dira que Φ est un fait plausible ;*
- *Ψ est alors appelée l'antécédent de l'assertion ;*
- *Φ est appelée le conséquent de l'assertion.*

Définition 3.4.3 *Les relations de conséquence sont des ensembles d'assertions conditionnelles.*

Avec le langage L, on définit une sémantique à l'aide d'un ensemble de mondes U et une relation binaire de satisfaction entre les mondes et les formules. L'ensemble U est l'univers de référence. On peut considérer que les mondes sont des interprétations au sens de la logique classique.

Remarquons que la sémantique des différentes relations d'inférence étudiées s'appuie sur la notion de modèle, elle-même utilisant la notion d'état et que Kraus, Lehmann et Magidor ne donnent pas de définition formelle de cette dernière.

Définition 3.4.4 *Un modèle est un triplet $(S, l, <)$ dans lequel S est un ensemble d'états, l est une fonction de S dans 2^U qui permet à chaque état de pointer sur un ensemble non vide de mondes et $<$ est une relation binaire sur S.*

Le terme "modèle" est ambigu puisqu'il possède un sens bien précis en logique classique qui n'est pas celui utilisé ici. J'ai toutefois gardé cette terminologie parce qu'elle est celle de Kraus, Lehmann et Magidor, et donc par là-même facilite la lecture de cette section et sa mise en relation avec l'article cité.

Remarquons aussi que la fonction l permet de définir un état comme étant un ensemble de mondes, donc un ensemble d'interprétations.

Définition 3.4.5 *Soit le modèle $(S, l, <)$, soit la formule A, on dira que l'état $s \in S$ satisfait A ssi $\forall m \in l(s)$, $m \models A$.*

La relation de conséquence définie par un modèle $W = (S, l, <)$ est notée $\models_w$ et est définie par :

Définition 3.4.6 *$A \models_w B$ ssi pour tout état $s \in S$ minimal pour $<$ satisfaisant A, s satisfait aussi B.*

Je vais maintenant passer en revue les différents systèmes proposés par Kraus, Lehmann et Magidor ainsi que les sémantiques correspondantes. Un "système" est défini par un ensemble donné de propriétés et correspond donc à l'ensemble des relations de conséquence vérifiant les dites propriétés. La construction de ces systèmes se fait par ajout successif de propriétés les rendant de plus en plus puissants.

Intuitivement, un système sera de plus en plus puissant ssi il est défini par un ensemble de propriétés de déduction de plus en plus important, donc ssi les relations de conséquence associées contiennent de plus en plus d'assertions conditionnelles. Remarquons que cette notion de puissance peut donc se définir de manière tout à fait formelle.

3.4.1.1 Le système C

Présentons d'abord le plus faible : le *système C*.

Définition 3.4.7 *Une relation de conséquence sera dite relation de conséquence cumulative ssi elle est fermée pour les règles de déduction suivantes :*

- *propriété de réflexivité :* $\alpha \mid\!\sim \alpha$,
- *équivalence logique gauche :* $\dfrac{\models \alpha \leftrightarrow \beta \;;\; \alpha \mid\!\sim \gamma}{\beta \mid\!\sim \gamma}$,
- *affaiblissement droit ("right weakening") :* $\dfrac{\models \alpha \rightarrow \beta \;;\; \gamma \mid\!\sim \alpha}{\gamma \mid\!\sim \beta}$,
- *coupure :* $\dfrac{\alpha \mid\!\sim \beta \;;\; \alpha \wedge \beta \mid\!\sim \gamma}{\alpha \mid\!\sim \gamma}$,
- *monotonie prudente :* $\dfrac{\alpha \mid\!\sim \beta \;;\; \alpha \mid\!\sim \gamma}{\alpha \wedge \beta \mid\!\sim \gamma}$.

Une telle relation de conséquence est notée : $\mid\!\sim_c$.

La signification intuitive de ces propriétés est la suivante :

- la réflexivité signifie que toute formule peut être déduite d'elle-même ;
- l'équivalence logique gauche indique que des formules équivalentes doivent avoir les mêmes conséquences ;
- l'affaiblissement droit stipule que les conséquences logiques de faits plausibles sont aussi des faits plausibles ;
- la coupure exprime les conditions dans lesquelles on peut se passer de fait plausible pour déduire de nouveaux faits plausibles ;
- la monotonie prudente est une forme restrictive de la monotonie.

Définition 3.4.8 *Le système logique correspondant à la classe des relations de conséquence cumulatives est appelé système C.*

4 nouvelles propriétés sont dérivées des règles de déduction précédentes :

- *propriété de l'équivalence :* $\dfrac{\alpha \mid\!\sim \beta\ ; \beta \mid\!\sim \alpha\ ; \alpha \mid\!\sim \gamma}{\beta \mid\!\sim \gamma}$,
- *propriété du ET :* $\dfrac{\alpha \mid\!\sim \beta\ ; \alpha \mid\!\sim \gamma}{\alpha \mid\!\sim \beta \wedge \gamma}$,
- *propriété du MPC (modus ponens) :* $\dfrac{\alpha \mid\!\sim \beta \rightarrow \gamma\ ; \alpha \mid\!\sim \beta}{\alpha \mid\!\sim \gamma}$,
- *propriété numéro 9 de Kraus, Lehmann, Magidor :* $\dfrac{\alpha \vee \beta \mid\!\sim \alpha\ ; \alpha \mid\!\sim \gamma}{\alpha \vee \beta \mid\!\sim \gamma}$.

Théorème 3.4.1 *L'utilisation des règles de déduction du système C rend équivalentes les 3 règles suivantes :*

- *propriété de la monotonie :* $\dfrac{\models \alpha \rightarrow \beta\ ; \beta \mid\!\sim \gamma}{\alpha \mid\!\sim \gamma}$
- *propriété EHD ("easy half deduction") :* $\dfrac{\alpha \mid\!\sim \beta \rightarrow \gamma}{\alpha \wedge \beta \mid\!\sim \gamma}$
- *propriété de la transitivité :* $\dfrac{\alpha \mid\!\sim \beta\ ; \beta \mid\!\sim \gamma}{\alpha \mid\!\sim \gamma}$

À partir du moment où l'on renonce à la monotonie, il faut aussi renoncer aux 2 autres.

Kraus, Lehmann et Magidor présentent aussi la *propriété de la contraposition* :

$$\frac{\alpha \mid\!\sim \beta}{\neg\beta \mid\!\sim \neg\alpha}$$

Théorème 3.4.2 *La propriété de contraposition implique la propriété de monotonie.*

Je vais maintenant présenter la sémantique correspondant au raisonnement cumulatif (système C).

Définition 3.4.9 *Un modèle cumulatif est un triplet $(S, l, <)$ dans lequel S est un ensemble d'états, l est une fonction de S dans 2^U qui permet à chaque état de pointer sur un ensemble de mondes non vide et $<$ est une relation binaire sur S satisfaisant la condition de régularité (voir les définitions 3.4.10 et 3.4.11). La relation de conséquence induite par un tel modèle est notée : $\models_c$.*

Un exemple de modèle cumulatif est présenté par la figure 3.1 page suivante.

Définition 3.4.10 *Soient $P \subseteq V$ et $<$ une relation binaire sur V, on dira que P est un ensemble régulier*[10] *ssi $\forall t \in P$, soit t est minimal pour P, soit $\exists s$ minimal pour P tel que $s < t$.*

Définition 3.4.11 *La relation $<$ du modèle $(S, l, <)$ vérifie la condition de régularité ssi $\forall A \in L$ (A est une formule), l'ensemble des états satisfaisant A est régulier (donc borné inférieurement pour la relation $<$).*

Remarquons que ces définitions sont très générales et délicates à mettre en œuvre. En particulier, il est difficile de vérifier la condition de régularité pour une relation binaire quelconque, même quand l'ensemble des états est fini. Pour pouvoir effectuer cette vérification sans trop de difficulté, il faut que la relation $<$ soit définie comme un ordre partiel.

Kraus, Lehmann et Magidor obtiennent ainsi le théorème de représentation 3.4.1 page suivante.

[10] "smooth" en anglais.

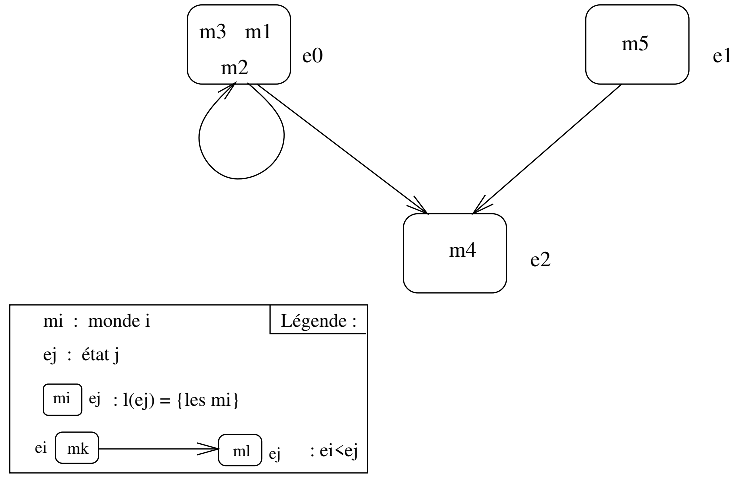


Figure 3.1 : Exemple d'un modèle cumulatif

Théorème de représentation 3.4.1 *Pour toute relation* $\mid\!\sim_c$, *il existe un modèle cumulatif définissant une relation* $\models_c$ *et vérifiant : étant données* Ψ *et* Φ *deux formules quelconques du langage* L, $\Psi \mid\!\sim_c \Phi \Leftrightarrow \Psi \models_c \Phi$

Ce théorème signifie que les relations de conséquence définies à l'aide de modèles cumulatifs respectent toutes les propriétés du système C. De même, il existe un modèle cumulatif tel qu'une relation cumulative (définie à l'aide des propriétés du système C) soit équivalente à la relation de conséquence définie avec ce modèle cumulatif.

Le système C reste toutefois un système encore trop faible pour être représentatif d'un raisonnement non-monotone puissant.

3.4.1.2 Le système CL

En rajoutant des propriétés au système C, on obtient le *système CL* (système C avec boucles - "loop" en anglais) :

Définition 3.4.12 *Le système CL est bâti à partir du système C et de la propriété de la boucle :*

$$\frac{\alpha_0 \mid\!\sim \alpha_1 \ ; \alpha_1 \mid\!\sim \alpha_2 \ ; \ldots \ ; \alpha_{k-1} \mid\!\sim \alpha_k \ ; \alpha_k \mid\!\sim \alpha_0}{\alpha_0 \mid\!\sim \alpha_k}$$

Ce système définit une relation de conséquence notée : $\mid\!\sim_{cl}$.

À partir de la propriété ci-dessus, on peut en déduire une autre appelée *propriété numéro 15 de Kraus, Lehmann, Magidor* :

$$\frac{\alpha_0 \mid\!\sim \alpha_1 \ ; \alpha_1 \mid\!\sim \alpha_2 \ ; \ldots \ ; \alpha_{k-1} \mid\!\sim \alpha_k \ ; \alpha_k \mid\!\sim \alpha_0}{\alpha_i \mid\!\sim \alpha_j, \forall i, j = 0, \ldots, k}$$

La sémantique associée à ce nouveau système CL est identique à celle décrite pour le système C, en utilisant cette fois des *modèles cumulatifs ordonnés* (voir exemple figure 3.2 page suivante).

Définition 3.4.13 *Un modèle cumulatif ordonné est un modèle cumulatif pour lequel la relation* $<$ *est un ordre strict partiel. Il définit une relation de conséquence notée :* $\models_{cl}$.

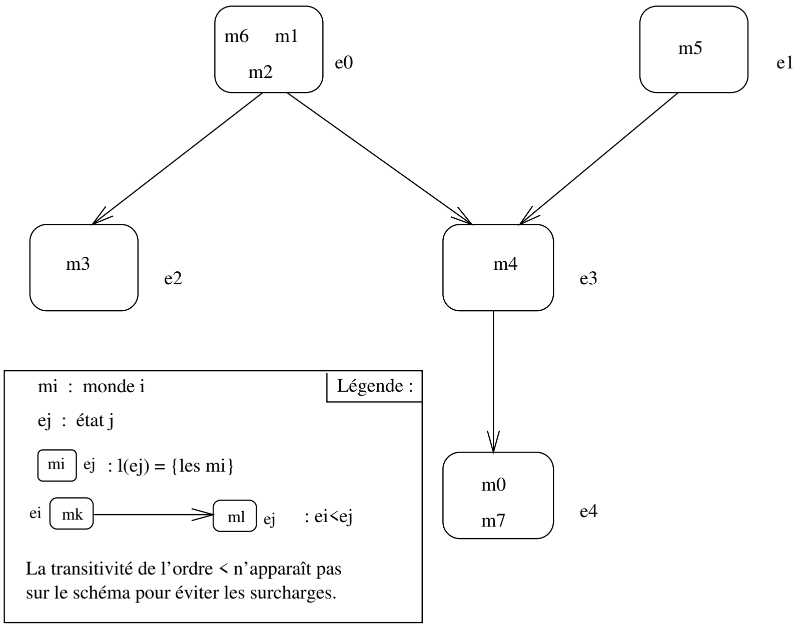


Figure 3.2 : Exemple d'un modèle cumulatif ordonné

Comme dans le cas cumulatif, Kraus, Lehmann, Magidor obtiennent alors le théorème de représentation 3.4.2.

Théorème de représentation 3.4.2 *Pour toute relation* $\mid\!\sim_{cl}$*, il existe un modèle cumulatif ordonné définissant une relation* $\models_{cl}$ *et vérifiant : étant données* Ψ *et* Φ *deux formules quelconques du langage L,* $\Psi \mid\!\sim_{cl} \Phi \Leftrightarrow \Psi \models_{cl} \Phi$

3.4.1.3 Le système P

De la même façon, Kraus, Lehmann et Magidor ont défini le *système P* (P pour préférence) :

Définition 3.4.14 *Le système P est bâti à partir du système C et de la propriété du OU :*

$$\frac{\alpha \mid\!\sim \gamma \;;\; \beta \mid\!\sim \gamma}{\alpha \vee \beta \mid\!\sim \gamma}$$

Ce système définit une relation de conséquence notée : $\mid\!\sim_p$.

Il est plus puissant que le système CL et surtout il permet de traiter les disjonctions de formules. On dira alors qu'une relation de conséquence respectant le système P est une *relation de conséquence préférentielle*. À partir de la propriété ci-dessus et de certaines des propriétés du système C, on peut en déduire de nouvelles propriétés :

- *propriété S* (appelée *conditionnalisation* dans [GM94]) : $\dfrac{\alpha \wedge \beta \mid\!\sim \gamma}{\alpha \mid\!\sim \beta \rightarrow \gamma}$
- *propriété D* : $\dfrac{\alpha \wedge \neg\beta \mid\!\sim \gamma \;;\; \alpha \wedge \beta \mid\!\sim \gamma}{\alpha \mid\!\sim \gamma}$
- *propriété numéro 19 de Kraus, Lehmann, Magidor* : $\dfrac{\alpha \mid\!\sim \gamma \;;\; \beta \mid\!\sim \mu}{\alpha \vee \beta \mid\!\sim \gamma \vee \mu}$

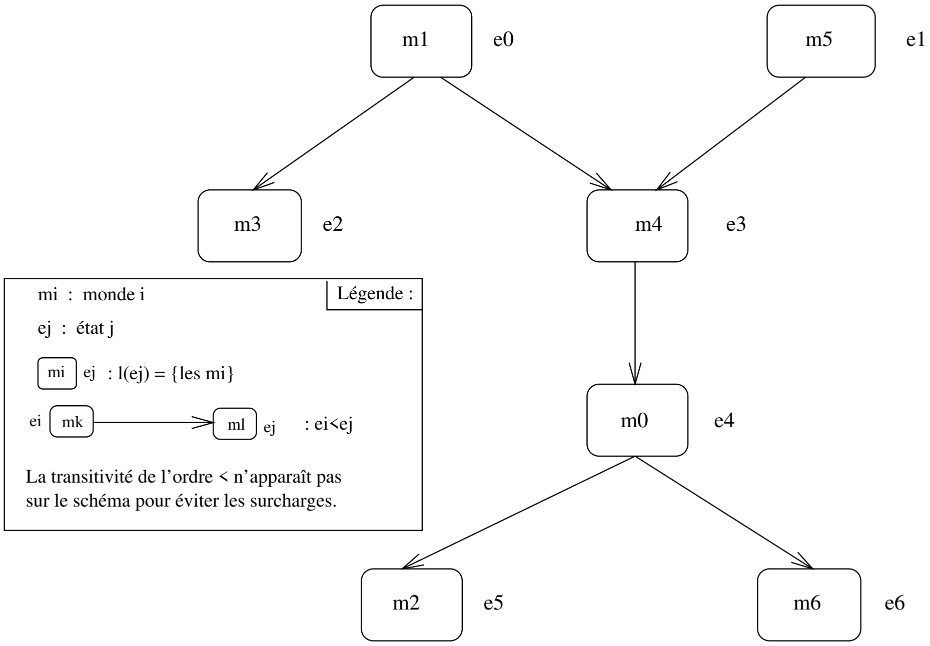


Figure 3.3 : Exemple d'un modèle préférentiel

- *propriété numéro 20 de Kraus, Lehmann, Magidor* : $\dfrac{\alpha \vee \gamma \mid\!\sim \gamma \ ; \alpha \mid\!\sim \beta}{\gamma \mid\!\sim \alpha \rightarrow \beta}$
- *propriété numéro 21 de Kraus, Lehmann, Magidor* : $\dfrac{\alpha \vee \beta \mid\!\sim \alpha \ ; \beta \vee \gamma \mid\!\sim \beta}{\alpha \vee \gamma \mid\!\sim \alpha}$
- *propriété numéro 22 de Kraus, Lehmann, Magidor* : $\dfrac{\alpha \vee \beta \mid\!\sim \alpha \ ; \beta \vee \gamma \mid\!\sim \beta}{\alpha \mid\!\sim \gamma \rightarrow \beta}$

La sémantique de ce système repose sur des *modèles préférentiels* (voir exemple figure 3.3).

Définition 3.4.15 *Un modèle préférentiel est un modèle cumulatif ordonné dans lequel la fonction l fait pointer chaque état vers un monde unique (et non plus vers un ensemble de mondes). Il définit une relation de conséquence notée :* $\models_p$.

Kraus, Lehmann, Magidor obtiennent alors le théorème de représentation 3.4.3.

Théorème de représentation 3.4.3 *Pour toute relation* $\mid\!\sim_p$*, il existe un modèle préférentiel définissant une relation* $\models_p$ *et vérifiant : étant données* Ψ *et* Φ *deux formules quelconques du langage L,* $\Psi \mid\!\sim_p \Phi \Leftrightarrow \Psi \models_p \Phi$

P peut-il être considéré comme un système convenable pour faire du raisonnement non-monotone ? Il semblerait que l'on ne puisse ajouter aucune autre règle de type déductif (permettant de déduire de nouvelles assertions). On peut par contre rajouter des règles d'un autre type, celles qui déduisent de l'absence d'une assertion l'absence d'une autre assertion :

- *propriété de la rationalité de la négation* : $\dfrac{\alpha \wedge \gamma \not\mid\!\sim \beta \ ; \alpha \wedge \neg\gamma \not\mid\!\sim \beta}{\alpha \not\mid\!\sim \beta}$
- *propriété de la rationalité disjonctive* : $\dfrac{\alpha \not\mid\!\sim \gamma \ ; \beta \not\mid\!\sim \gamma}{\alpha \vee \beta \not\mid\!\sim \gamma}$

- *propriété de la monotonie rationnelle :* $\dfrac{\alpha \wedge \beta \not\mid\!\sim \gamma \;;\; \alpha \not\mid\!\sim \neg\beta}{\alpha \not\mid\!\sim \gamma}$

  Cette propriété de la monotonie rationnelle est présentée aussi par Gärdenfors et Makinson dans [Gär91, GM94] mais sous une autre forme.

L'utilisation de la monotonie rationnelle était initialement destinée à résoudre le problème de non-pertinence qui apparaissait dans le système P : si une formule δ est une conséquence non-monotone de la formule α et si la formule β n'a "rien à voir" avec δ (par exemple, parce que β ne figure pas dans la base), alors on voudrait que δ soit une conséquence non-monotone de $\alpha \wedge \beta$. L'exemple suivant, emprunté à [Ben94] illustre bien ce problème :

> soit la base $E = \{O \rightarrow V, P \rightarrow O, P \rightarrow (\neg V)\}$ avec $O =$ oiseau, $P =$ pingouin, $V =$ vole ; posons $R =$ rouge, si on sait que $O \mid\!\sim V$, on voudrait alors que $(O \wedge R) \mid\!\sim V$, ce qui n'est pas le cas.

Lehmann et Magidor ont donc poursuivi leurs investigations sur les relations d'inférence préférentielle en définissant la notion de *relation d'inférence rationnelle* (voir [LM92]). Remarquons que Gärdenfors et Makinson dans [GM94] ont introduit un type semblable de relation d'inférence, qui utilise lui-aussi la notion de monotonie rationnelle (voir définition 3.4.22 page 68 et théorème 3.4.9 page 70).

Définition 3.4.16 *Une relation d'inférence rationnelle est une relation vérifiant les propriétés du système P et la propriété de monotonie rationnelle. Ce système, appelé système R, définit une relation de conséquence notée :* $\mid\!\sim_r$.

La sémantique correspondante s'appuie sur la notion de *modèle rangé* (voir exemple figure 3.4 page suivante). Ces modèles sont des modèles préférentiels particuliers.

Définition 3.4.17 *Un modèle préférentiel* $(S, l, <)$ *est appelé modèle rangé ssi la relation* $<$ *est définie de la manière suivante : il existe un ensemble totalement ordonné* Ω *(représenté par la relation* $<_\Omega$*) et une fonction* R *de* S *dans* Ω *telle que :* $\forall (s, t)$ *un couple d'états de* S, $s < t$ *ssi* $R(s) <_\Omega R(t)$. *Il définit une relation de conséquence notée :* $\models_r$.

L'ordre sur Ω étant total, on a une notion d'équivalence entre états de même rang, ce qui n'existe pas dans les types de modèles présentés jusqu'alors. Lehmann et Magidor obtiennent alors le théorème de représentation 3.4.4.

Théorème de représentation 3.4.4 *Pour toute relation* $\mid\!\sim_r$*, il existe un modèle rangé définissant une relation* $\models_r$ *et vérifiant : étant données* Ψ *et* Φ *deux formules quelconques du langage* L, $\Psi \mid\!\sim_r \Phi \Leftrightarrow \Psi \models_r \Phi$

Or, quand on calcule l'ensemble des formules issu d'une relation d'inférence rationnelle pour un ensemble des formules initial donné, on constate que cet ensemble n'est pas unique. Lehmann et Magidor résolvent ce problème en proposant un mode de sélection d'un ensemble particulier, qu'ils appellent la *fermeture rationnelle*. La construction de cet ensemble se fait à l'aide d'une fonction qui associe à chaque formule du langage un nombre entier positif symbolisant l'"exceptionnalité" de la formule. La base de croyances est ainsi stratifiée (voir l'algorithme de définition de cette fonction dans [LM92]).

La sémantique de cette relation d'inférence rationnelle particulière repose alors sur un modèle rangé particulier dont la définition est donnée dans [LM92].

En fait, cette propriété se révélera insuffisante pour résoudre la totalité du problème de non-pertinence (voir [Ben94] et la section 4.4.4 page 182 de ce document).

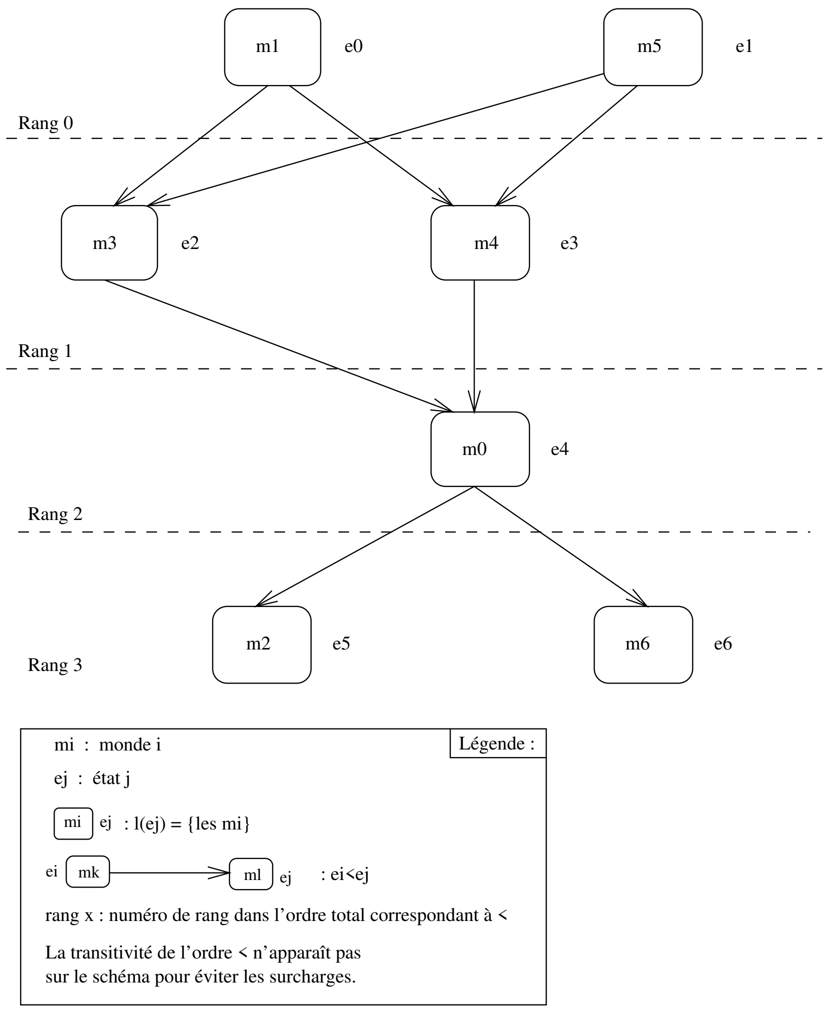


Figure 3.4 : Exemple d'un modèle rangé

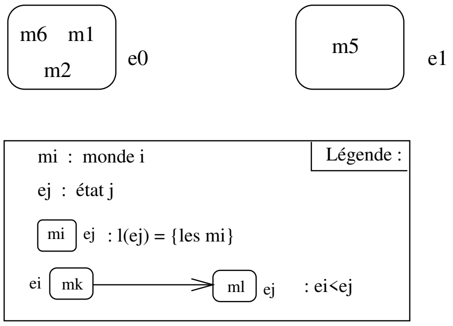


Figure 3.5 : Exemple d'un modèle simplement cumulatif

3.4.1.4 Le système CM

Puis, Kraus, Lehmann et Magidor introduisent le *système CM* issu du système C en rajoutant les règles de monotonie, EHD, transitivité déjà présentées dans le théorème 3.4.1 page 57. Le système obtenu est plus puissant que le système C mais incomparable avec le système P et induit une relation de conséquence notée : $\mid\!\sim_{cm}$. Cette relation est appelée *relation de conséquence cumulative monotone*. La sémantique de ce système repose sur des *modèles simplement cumulatifs* (voir exemple figure 3.5).

Définition 3.4.18 *Un modèle simplement cumulatif est un modèle cumulatif dont la relation binaire $<$ est vide. Il définit une relation de conséquence notée : $\models_{cm}$.*

Kraus, Lehmann, Magidor obtiennent ainsi le théorème de représentation 3.4.5.

Théorème de représentation 3.4.5 *Pour toute relation $\mid\!\sim_{cm}$, il existe un modèle simplement cumulatif définissant une relation $\models_{cm}$ et vérifiant : étant données Ψ et Φ deux formules quelconques du langage L, $\Psi \mid\!\sim_{cm} \Phi \Leftrightarrow \Psi \models_{cm} \Phi$*

3.4.1.5 Le système M

Pour finir, le *système M* est issu du système C en rajoutant la règle de contraposition déjà présentée dans le théorème 3.4.1 page 57. Le système obtenu est plus puissant que les systèmes CM et P et induit une relation de conséquence notée : $\mid\!\sim_{m}$. Cette relation est appelée *relation de conséquence monotone*. La sémantique de ce système repose sur des *modèles simplement préférentiels* (voir exemple figure 3.6 page suivante).

Définition 3.4.19 *Un modèle simplement préférentiel est un modèle préférentiel dont la relation binaire $<$ est vide. Il définit une relation de conséquence notée : $\models_{m}$.*

Kraus, Lehmann, Magidor obtiennent ainsi le théorème de représentation 3.4.6.

Théorème de représentation 3.4.6 *Pour toute relation $\mid\!\sim_{m}$, il existe un modèle simplement préférentiel définissant une relation $\models_{m}$ et vérifiant : étant données Ψ et Φ deux formules quelconques du langage L, $\Psi \mid\!\sim_{m} \Phi \Leftrightarrow \Psi \models_{m} \Phi$*

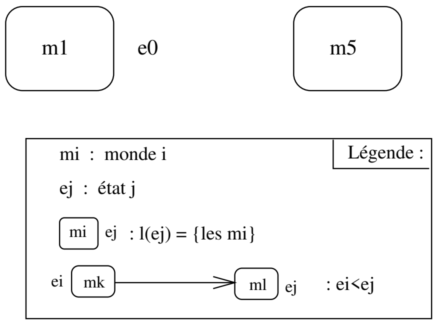


Figure 3.6 : Exemple d'un modèle simplement préférentiel

Propriétés de déduction	Système	Relation de conséquence issue du système	Modèle	Relation de conséquence issue du modèle
Les propriétés de base : ▪ réflexivité, ▪ équivalence logique gauche, ▪ affaiblissement droit, ▪ coupure, ▪ monotonie prudente.	C	$\mid\sim_c$	cumulatif	$\models_c$
Les propriétés de base et la boucle.	CL	$\mid\sim_{cl}$	cumulatif ordonné	$\models_{cl}$
Les propriétés de base et le OU.	P	$\mid\sim_p$	préférentiel	$\models_p$
Les propriétés de base plus le OU et la monotonie rationnelle.	R	$\mid\sim_r$	rangé	$\models_r$
Les propriétés de base plus la monotonie, l'EHD et la transitivité.	CM	$\mid\sim_{cm}$	simplement cumulatif	$\models_{cm}$
Les propriétés de base et la contraposition.	M	$\mid\sim_m$	simplement préférentiel	$\models_m$

Tableau 3.1 : Les systèmes de Kraus, Lehmann et Magidor – Récapitulation et sémantique.

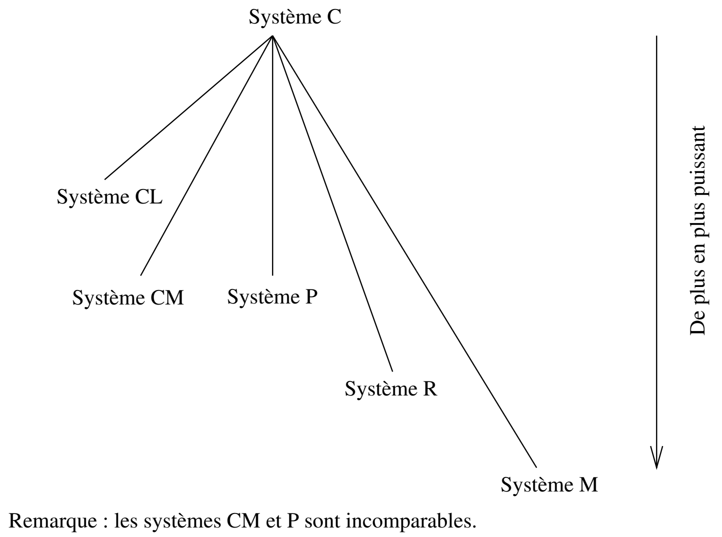


Figure 3.7 : Les systèmes de Kraus, Lehmann et Magidor – Classement par ordre de puissance.

3.4.1.6 Conclusion sur les travaux de Kraus, Lehmann et Magidor

Kraus, Lehmann et Magidor ont ainsi défini 6 familles de modèles et les relations de conséquence associées. Chaque famille est caractérisée par un système logique. Ces systèmes ne sont pas équivalents. Ils sont récapitulés dans le tableau 3.1 page ci-contre et représentés par ordre de puissance sur la figure 3.7.

Signalons d'autre part qu'il existe plusieurs variantes du système P et des relations d'inférence rationnelle proposées par les mêmes auteurs. Ces variantes sont présentées dans le tableau 3.2 page suivante.

Les passages d'une variante à l'autre sont donnés par les propriétés suivantes :

Propriété 3.4.2 *(démonstration dans [KLM90]) Les propriétés d'affaiblissement droit, de conditionnalisation et du ET impliquent la propriété de la coupure.*

Propriété 3.4.3 *(démonstration dans [KLM90]) Les propriétés de réflexivité, d'équivalence logique gauche, d'affaiblissement droit, de coupure et de monotonie prudente impliquent la propriété du ET.*

Propriété 3.4.4 *(démonstration dans [KLM90]) Les propriétés de réflexivité, d'équivalence logique gauche, d'affaiblissement droit, de coupure, de monotonie prudente et du OU impliquent la propriété de conditionnalisation.*

Propriété 3.4.5 *Les propriétés de réflexivité, d'équivalence logique gauche, d'affaiblissement droit, de la conditionnalisation et du ET impliquent la propriété du OU.*

> **Preuve :** (suggérée par Lehmann lors d'une communication personnelle)
>
> Supposons que $p \mathrel{|\!\sim} r$ et $q \mathrel{|\!\sim} r$;
>
> avec l'équivalence logique gauche, on a $\models p \leftrightarrow (p \vee q) \wedge p$ et $p \mathrel{|\!\sim} r$ qui impliquent $((p \vee q) \wedge p) \mathrel{|\!\sim} r$;
>
> de même, $\models q \leftrightarrow (p \vee q) \wedge q$ et $q \mathrel{|\!\sim} r$ impliquent $((p \vee q) \wedge q) \mathrel{|\!\sim} r$;
>
> en appliquant la conditionnalisation sur ces deux formules, on obtient $(p \vee q) \mathrel{|\!\sim} (p \rightarrow r)$ et $(p \vee q) \mathrel{|\!\sim} (q \rightarrow r)$;

Propriétés de déduction	[KLM90]		[LM92]		[FL94]
	sys. P	sys. R	sys. P	sys. R	sys. R qui préservent la consistance
réflexivité	X	X	X	X	X
équivalence logique gauche	X	X	X	X	X
affaiblissement droit	X	X	X	X	X
coupure	X	X			
monotonie prudente	X	X	X	X	
OU	X	X	X	X	
monotonie rationnelle		X		X	X
ET			X	X	X
conditionnalisation (propriété S)					X
préservation de la consistance					X

Tableau 3.2 : Les variantes du système P et des relations rationnelles

puis, avec le ET, on a $(p \vee q) \mid\!\sim (p \rightarrow r) \wedge (q \rightarrow r)$;

or, on sait que $\models ((p \rightarrow r) \wedge (q \rightarrow r)) \rightarrow ((p \vee q) \rightarrow r)$ et $(p \vee q) \mid\!\sim (p \rightarrow r) \wedge (q \rightarrow r)$ impliquent $(p \vee q) \mid\!\sim ((p \vee q) \rightarrow r)$;

de plus, la réflexivité implique que $(p \vee q) \mid\!\sim (p \vee q)$, ce qui nous donne avec la formule précédente en appliquant le ET : $(p \vee q) \mid\!\sim ((p \vee q) \rightarrow r)$;

sachant que $\models ((p \vee q) \wedge r) \rightarrow r$, un dernier passage par l'affaiblissement droit nous suffit alors pour conclure que $(p \vee q) \mid\!\sim r$. □

Propriété 3.4.6 *Les propriétés de réflexivité et d'affaiblissement droit impliquent la propriété de supra-classicité (voir définition 3.4.20 page suivante issue de [GM94]).*

Preuve : Supposons que $p \vdash q$; on sait alors que $\vdash (p \rightarrow q)$; de plus, avec la réflexivité, on a $p \mid\!\sim p$; donc en appliquant l'affaiblissement droit avec $\vdash (p \rightarrow q)$ et $p \mid\!\sim p$, on obtient $p \mid\!\sim q$. □

Propriété 3.4.7 *Les propriétés de monotonie rationnelle, de préservation de la consistance, de supra-classicité et du ET impliquent la propriété de monotonie prudente.*

Preuve : (suggérée par Lehmann lors d'une communication personnelle)

Supposons que $p \mid\!\sim q$ et $p \mid\!\sim r$;

nous avons ici deux cas possibles :

- si $p \mid\!\not\sim (\neg q)$, alors en appliquant la monotonie rationnelle avec $p \mid\!\sim r$ et $p \mid\!\not\sim (\neg q)$, on obtient $(p \wedge q) \mid\!\sim r$;
- si $p \mid\!\sim (\neg q)$, alors en appliquant le ET avec $p \mid\!\sim q$ et $p \mid\!\sim (\neg q)$, on obtient $p \mid\!\sim \perp$; donc avec la préservation de la consistance $p \vdash \perp$; donc $(p \wedge q) \vdash \perp$; donc $(p \wedge q) \vdash x, \forall x$ en particulier pour $x = r$; donc, avec la supra-classicité $(p \wedge q) \mid\!\sim r$.

□

3.4.2 Les propriétés de Gärdenfors et Makinson et leur sémantique

Gärdenfors et Makinson ont exploré les différentes voies dans lesquelles le concept d'"expectation" peut être utilisé pour faire un raisonnement non-monotone (voir [GM94]). L'idée est que, lors d'un raisonnement, on utilise des informations dans lesquelles on croit fermement mais aussi des informations attendues ou espérées (ce que l'on nomme en anglais : *"expectations"*). Un tel concept peut prendre plusieurs formes : les *ensembles d'"expectations"*, et les *ordres d'"expectations"*. D'un côté, les "expectations" peuvent donc être considérées comme des propositions exprimées avec le même langage que les croyances certaines. Ainsi on peut parler d'ensembles d'"expectations" et utiliser ces ensembles pour définir une nouvelle relation d'inférence non-monotone : $\alpha \mid\!\sim \beta$ ssi β est impliqué logiquement par α avec les ensembles d'"expectations" (le plus possible d'"expectations" qui soient consistantes avec α). De l'autre côté, les "expectations" peuvent être considérées comme un ordre entre des propositions, en particulier celles en lesquelles on ne croit pas fermement : $\alpha \mid\!\sim \beta$ ssi β est impliqué logiquement par α avec toutes les propositions qui sont suffisamment bien attendues à la lumière de α. Le problème technique est bien-sûr de savoir définir le "suffisamment bien attendues". Ces 2 idées sont liées et reviennent à peu-près au même (voir l'étude présentée en section 2.2.5 page 24).

Remarquons avant toute chose que Gärdenfors et Makinson se placent systématiquement dans une optique de révision, ce qui signifie que leur ensemble de départ est en général consistant et que c'est l'ajout d'une nouvelle croyance qui amène l'inconsistance.

Gärdenfors et Makinson se basent sur les hypothèses suivantes. Le langage utilisé L est propositionnel et fermé pour les 4 connecteurs logiques. Il n'y aura aucun rajout de règle par défaut, d'opérateur modal, etc. Toutes les "expectations" seront exprimées avec L. Le langage ainsi utilisé est supposé compact, comme chez Kraus, Lehmann et Magidor.

3.4.2.1 Les propriétés de Gärdenfors et Makinson

Gärdenfors et Makinson proposent les propriétés suivantes.

Définition 3.4.20 *Une relation $\mid\!\sim$ est une relation d'inférence non-monotone ssi elle possède les 4 propriétés suivantes :*

- *la supra-classicité :* $\dfrac{\alpha \vdash \beta}{\alpha \mid\!\sim \beta}$,
- *l'équivalence logique gauche :* $\dfrac{\models \alpha \leftrightarrow \beta \ ; \alpha \mid\!\sim \gamma}{\beta \mid\!\sim \gamma}$,
- *l'affaiblissement droit ("right weakening") :* $\dfrac{\models \alpha \rightarrow \beta \ ; \gamma \mid\!\sim \alpha}{\gamma \mid\!\sim \beta}$,
- *le ET :* $\dfrac{\alpha \mid\!\sim \beta \ ; \alpha \mid\!\sim \gamma}{\alpha \mid\!\sim \beta \wedge \gamma}$.

Remarques :

- La supra-classicité implique la propriété de réflexivité.
- L'affaiblissement droit avec le ET et la compacité de $\vdash$ impliquent la *propriété de fermeture* :

$$\frac{\alpha \mid\!\sim \beta_i \text{ pour tout } \beta_i \in B \ ; B \vdash \gamma}{\alpha \mid\!\sim \gamma}$$

.

Définition 3.4.21 *On appelle ensemble des postulats de base l'ensemble suivant de propriétés :*

- *les 4 propriétés servant à la définition d'une relation d'inférence non-monotone,*

- *la conditionnalisation faible :* $\dfrac{\alpha \mid\!\sim \beta}{\mid\!\sim \alpha \rightarrow \beta}$,
- *la monotonie rationnelle faible :* $\dfrac{\not\mid\!\sim \neg\alpha \ ; \mid\!\sim \alpha \rightarrow \beta}{\alpha \mid\!\sim \beta}$,
- *la préservation de la consistance :* $\dfrac{\alpha \mid\!\sim \bot}{\alpha \vdash \bot}$*(avec* $\bot$ *représentant la contradiction).*

Définition 3.4.22 *On appelle ensemble étendu des postulats l'ensemble suivant de propriétés :*

- *les postulats de base,*
- *la cumulativité :* $\dfrac{\alpha \mid\!\sim \beta \ ; \beta \vdash \alpha}{\alpha \mid\!\sim \gamma \Leftrightarrow \beta \mid\!\sim \gamma}$,
- *le OU :* $\dfrac{\alpha \mid\!\sim \gamma \ ; \beta \mid\!\sim \gamma}{\alpha \vee \beta \mid\!\sim \gamma}$,
- *la monotonie rationnelle :* $\dfrac{\alpha \mid\!\sim \gamma \ ; \alpha \not\mid\!\sim \neg\beta}{\alpha \wedge \beta \mid\!\sim \gamma}$.

Remarques :

- La rationalité disjonctive est une conséquence de l'ensemble étendu de postulats.
- La cumulativité, avec les postulats de base, est équivalente à la coupure plus la monotonie prudente (voir la propriété 4.3.1 page 170 explicitant le lien entre toutes ces propriétés).
- La cumulativité, avec les postulats de base, est équivalente à la *réciprocité* :

$$\frac{\alpha \mid\!\sim \beta \ ; \beta \mid\!\sim \alpha}{\alpha \mid\!\sim \gamma \Leftrightarrow \beta \mid\!\sim \gamma}$$

- Le OU, avec les postulats de base, est équivalent à la *conditionnalisation* (appelée propriété S dans [KLM90]) :

$$\frac{\alpha \wedge \beta \mid\!\sim \gamma}{\alpha \mid\!\sim \beta \rightarrow \gamma}$$

- La monotonie rationnelle s'exprime ici de manière différente mais elle est équivalente à celle de Kraus, Lehmann, Magidor.

3.4.2.2 Le cas des ensembles d'"expectations"

Notons Δ un ensemble de formules de L jouant le rôle d'un ensemble d'"expectations"[11], et rappelons les définitions suivantes données dans [Gär91, GM94] :

Définition 3.4.23 *Un ensemble D est un sous-ensemble maximal de Δ qui ne peut impliquer logiquement α ssi :*

1. $D \subseteq \Delta$,
2. $\alpha \notin C_n(D)$,
3. $\alpha \in C_n(D')$ *pour tout* D' *tel que* $D \subset D' \subseteq \Delta$.

L'ensemble de tous les sous-ensembles maximaux de Δ ne pouvant impliquer α est noté $\Delta\bot\alpha$.

[11] Ici, les "expectations" correspondent exactement à la notion de croyances.

Pour vérifier si $\alpha \mid\!\sim \beta$, on utilise $\Delta \bot \neg\alpha$. Toutefois, il subsiste un problème important : en général l'ensemble $\Delta \bot \neg\alpha$ n'est pas un singleton. Comment choisir entre tous les ensembles maximaux ? Pour cela, on rajoute une fonction de sélection S_Δ qui dépend de Δ. Cette fonction prend en argument l'ensemble de tous les ensembles maximaux et elle renvoie les ensembles préférés.

Remarquons que $\Delta \bot \neg\alpha$ est l'ensemble des sous-ensembles maximaux α-consistants de Δ.

Définition 3.4.24 *Une opération d'inférence "expectative" $C_{\Delta,S}$ est définie pour tout $\alpha \in L$ par l'équation*[12] *:*

- $C_{\Delta,S}(\alpha) = \cap\{C_n(\{\alpha\} \cup D)$ *tel que* $D \in S(\Delta \bot \neg\alpha)\}$ *avec Δ un ensemble non vide d'"expectations" et S une fonction de sélection ;*
- $C_{\Delta,S}$ *est dite fermée quand $\Delta = C_n(\Delta)$;*
- $C_{\Delta,S}$ *est dite générée avec consistance quand Δ est consistant.*

On définit donc $\alpha \mid\!\sim_{\Delta,S} \beta$ par $\beta \in C_{\Delta,S}(\alpha)$ (c'est-à-dire β est impliqué logiquement par toutes les unions de α avec un sous-ensemble de Δ maximal α-consistant préféré).

Remarques :

- L'approche par ensembles d'"expectations" est une approche prudente, donc correspond à ce que j'appelle le principe Uni.
- Dans le cas où Δ est fermé pour la déduction ($\Delta = C_n(\Delta)$), pour tout α et pour tout $D \in S(\Delta \bot \neg\alpha)$ les ensembles $C_n(\{\alpha\} \cup D)$ peuvent être vus comme des mondes dans lesquels α est vrai. On rejoint ainsi la terminologie de Shoham et ses modèles préférés (voir [Sho87]).

Gärdenfors et Makinson obtiennent alors le théorème de représentation 3.4.7.

Théorème de représentation 3.4.7 *Une relation d'inférence non-monotone $\mid\!\sim$ satisfait les postulats de base ssi il existe une relation d'inférence "expectative" fermée et générée avec consistance $\mid\!\sim_{\Delta,S}$ telle que $\alpha \mid\!\sim \beta$ ssi $\alpha \mid\!\sim_{\Delta,S} \beta$ pour tous α et β de L.*

Si on impose des restrictions à la fonction de sélection S, on obtient une relation d'inférence vérifiant d'autres propriétés que celles apparaissant dans les postulats de base.

Prenons, par exemple, la condition (SC) suivante :

Définition 3.4.25 *La condition (SC) est :*

$$\text{si } S(\Delta \bot \neg\alpha) \subseteq \Delta \bot \neg\beta \subseteq \Delta \bot \neg\alpha \text{ alors } S(\Delta \bot \neg\beta) = S(\Delta \bot \neg\alpha).$$

Cette condition signifie intuitivement : si les sous-bases α-consistantes préférées sont des sous-bases β-consistantes, elles-même étant des sous-bases α-consistantes, alors les sous-bases β-consistantes préférées sont les sous-bases α-consistantes préférées.

Gärdenfors et Makinson obtiennent ainsi le théorème de représentation 3.4.8.

Théorème de représentation 3.4.8 *Une relation d'inférence non-monotone $\mid\!\sim$ satisfait les postulats de base et la propriété de cumulativité ssi il existe une relation d'inférence "expectative" fermée et générée avec consistance $\mid\!\sim_{\Delta,S}$ vérifiant la condition (SC) telle que $\alpha \mid\!\sim \beta$ ssi $\alpha \mid\!\sim_{\Delta,S} \beta$ pour tous α et β de L.*

Un autre cas possible pour restreindre la fonction S est l'utilisation de préférences dans la définition de S.

[12] Dans une optique de révision, Δ est l'ensemble initial et α est la formule avec laquelle on révise.

Définition 3.4.26 *Une fonction de sélection S est relationnelle sur Δ ssi il existe une relation $\preceq$ sur les sous-ensembles de Δ telle que, pour toute formule α consistante, on a :*

$$S(\Delta\bot\neg\alpha) = \{D \in \Delta\bot\neg\alpha \text{ tel que } D \preceq D' \text{ pour tout } D' \in \Delta\bot\neg\alpha\}.$$

S est dite relationnelle transitivement ssi S est relationnelle en utilisant une relation transitive $\preceq$.

Gärdenfors et Makinson obtiennent ainsi le théorème de représentation 3.4.9.

Théorème de représentation 3.4.9 *Une relation d'inférence non-monotone $\mid\!\sim$ satisfait l'ensemble étendu des postulats de base ssi il existe une relation d'inférence "expectative" fermée, générée avec consistance et relationnelle transitivement $\mid\!\sim_{\Delta,S}$ telle que $\alpha \mid\!\sim \beta$ ssi $\alpha \mid\!\sim_{\Delta,S} \beta$ pour tous α et β de L.*

3.4.2.3 Le cas des ordres d'"expectations"

Bien qu'il soit possible de trouver des théorèmes de représentation, l'utilisation des fonctions de sélection comme mécanisme pour générer des inférences non-monotones n'est pas satisfaisant dans le cadre du calcul de ces inférences, la définition fonctionnelle de la fonction S n'étant en général pas disponible.

Gärdenfors et Makinson ont donc proposé d'utiliser des ordres sur les formules de Δ. Ces ordres sont destinés à représenter les divers degrés de certitude des formules. L'ordre "expectatif" va concerner toutes les formules, on n'a plus besoin d'isoler un ensemble d'"expectations". On travaille donc à partir de $(L, \leq)$ et $\alpha \leq \beta$ signifiera que β est au moins autant attendu, espéré que α.

Cette relation de pré-ordre possède les propriétés suivantes :

- la transitivité : si $\alpha \leq \beta$ et $\beta \leq \gamma$ alors $\alpha \leq \gamma$ (propriété $\mathcal{P}_1$),
- la dominance : si $\alpha \vdash \beta$ alors $\alpha \leq \beta$ (propriété $\mathcal{P}_2$),
- la conjonctivité : pour tous α et β, $\alpha \leq \alpha \wedge \beta$ ou $\beta \leq \alpha \wedge \beta$ (propriété $\mathcal{P}_3$).

Remarque : ces trois conditions impliquent que l'ordre est total.

Définition 3.4.27 *$\mid\!\sim$ est une relation d'inférence "expectative" comparative ssi il existe un ordre $\leq$ satisfaisant les 3 propriétés $\mathcal{P}_1$ à $\mathcal{P}_3$ et tel que la condition (C$\mid\!\sim$) soit vérifiée :*

$$\alpha \mid\!\sim \gamma \text{ ssi } \gamma \in C_n(\{\alpha\} \cup \{\beta \text{ tel que } \neg\alpha < \beta\}).$$

Remarque la condition (C$\mid\!\sim$) peut aussi s'exprimer de la façon suivante : $\alpha \mid\!\sim \gamma$ ssi, soit $\alpha \vdash \gamma$, soit il existe $\beta \in L$ tel que $\alpha \wedge \beta \vdash \gamma$ et $\neg\alpha < \beta$.

Gärdenfors et Makinson obtiennent alors le théorème de représentation 3.4.10.

Théorème de représentation 3.4.10 *$\mid\!\sim$ est une relation d'inférence non-monotone sur L satisfaisant l'ensemble étendu des postulats ssi $\mid\!\sim$ est une relation d'inférence "expectative" comparative.*

Le lien entre cette approche utilisant les ordres "expectatifs" et les modèles préférés de Shoham (cf [Sho87]) est le suivant :

Théorème 3.4.3 *Pour tout ordre "expectatif", on peut trouver un modèle préférentiel adéquat tel que l'inférence "expectative" soit identique à l'inférence issue de ce modèle, et vice-versa.*

Avec la définition suivante d'un modèle préférentiel adéquat :

Définition 3.4.28 *Soit le triplet* $M = (S, \models, \sqsubset)$ *avec* S *un ensemble non vide d'états,* $\models$ *une relation de satisfaction entre les éléments de* S *et les formules de* L*, et* $\sqsubset$ *une relation entre états.* M *est un modèle préférentiel adéquat ssi il vérifie les 4 propriétés suivantes :*

1. *la classicité :* $\forall s \in S$ *et* $\alpha, \beta \in L$,
   - $s \models \neg\alpha$ *ssi* $s \not\models \alpha$, *et*
   - $s \models \alpha \wedge \beta$ *ssi* $s \models \alpha$ *et* $s \models \beta$ *;*
2. *l'extension : pour tout* α *consistant au sens de* C_n*, il existe un* $s \in S$ *tel que* $s \models \alpha$ *;*
3. *basé sur une relation de préférence* $\sqsubset$ *qui est irréflexive, transitive et rangée ;*
4. *limité de manière finie*[13] *: c'est-à-dire que pour toute formule* α *qui est satisfaite par un état non préféré, il existe un état préféré qui la satisfait.*

Gärdenfors et Makinson donnent alors un théorème récapitulatif :

Théorème 3.4.4 *Soit* $\mid\!\sim$ *une relation entre des formules de* L*, les 4 conditions suivantes sont équivalentes :*

1. $\mid\!\sim$ *est déterminée par* $\mid\!\sim_{\Delta,S}$ *une relation d'inférence "expectative", transitivement relationnelle, générée avec consistance et fermée ;*
2. $\mid\!\sim$ *est déterminée par un ordre "expectatif"* $\leq$ *sur* L *en utilisant la condition* $(C\mid\!\sim)$ *;*
3. $\mid\!\sim$ *est déterminée par un modèle préférentiel adéquat ;*
4. $\mid\!\sim$ *satisfait les propriétés de supra-classicité, équivalence logique gauche, ET, cumulativité, OU, monotonie rationnelle et préservation de la consistance.*

3.4.3 Quelle est la classe de relations concernée par l'étude des propriétés de déduction ?

Comme nous avons pu le constater dans toute cette section, les propriétés de déduction sont définies pour des relations d'inférence non-monotone du type $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$, et non pour celles du type $(E, <) \mid\!\sim^{p,m} \Phi$. Cela signifie donc que toute mon étude portant sur les propriétés de déduction se fera sur des relations d'inférence du type *G-p-m* (donc de la classe 2).

[13] On retrouve ici la condition de régularité de [KLM90].

Chapitre 4

Résultats de la comparaison théorique

4.1 Complexité

4.1.1 Champ de l'étude de complexité

J'ai choisi d'étudier ici la complexité des 21 relations d'inférence $(E,<) \mid\!\sim^{p,m} H$ de la classe 1 avec les notations suivantes :

- $E =$ la base initiale (ensemble de formules propositionnelles) supposée finie,
- $< =$ ordre entre les formules de E,
- $p =$ principe d'inférence,
- $m =$ mécanisme de définition de sous-ensembles consistants de E avec un ordre entre ces sous-ensembles induit de $<$,
- $H =$ formule de la logique propositionnelle.

Rappelons, qu'en fait, les 12 autres relations d'inférence $(\Psi \mid\!\sim^{p,m}_{E,<} \Phi)$ de la classe 2 vont elles aussi être traitées par cette étude (voir sections 2.6.6 page 46 et 4.1.4 page 144).

Pourquoi se limiter au cas de la logique propositionnelle ? Tout simplement parce que le mécanisme d'inférence classique auquel on va se ramener, qui est décidable en logique des propositions, est indécidable en logique du premier ordre. Ce qui signifie qu'il n'existe pas d'algorithme pouvant résoudre le problème dans le cas du premier ordre sur toutes ses instances. Or les calculs de complexité s'appuient sur l'existence de tels algorithmes de résolution.

Je traiterai d'abord le cas général où $(E,<)$ est un ensemble pré-ordonné de formules propositionnelles quelconques et où H est une formule propositionnelle quelconque.

Nous verrons dans la section 4.1.3 page 110 quatre cas particuliers pour $(E,<)$ et H (une formule par strate, la forme conjonctive normale, les conjonctions de clauses de Horn, une formule par strate avec des conjonctions de clauses de Horn) qui, pour deux d'entre eux, réduisent considérablement la complexité des problèmes posés.

Notons que l'étude de complexité porte sur des problèmes et non sur des relations. Les problèmes que j'étudie sont donc les suivants :

Définition 4.1.1 *Soient une base ordonnée $(E,<)$, une formule H, un mécanisme de sélection m et un principe d'inférence p, les problèmes étudiés sont : $(E,<) \mid\!\sim^{p,m} H$?*

Ces problèmes seront notés p-m, c'est-à-dire de la même façon que les classes de relations d'inférence associées. Ce choix de notation est destiné à faciliter la lecture de ce document en évitant la multiplication des sigles.

Remarques importantes :

- Dans toute cette section, ainsi que dans la section 4.1.3 page 110, j'utiliserai sans les rappeler divers théorèmes et définitions de la théorie de la complexité qui sont répertoriés dans l'annexe A page 235[1].
- D'autre part, les démonstrations données dans cette section pourront paraître répétitives, voire redondantes. En effet, la méthode suivie est toujours la même quel que soit le problème traité ; elle consiste en deux étapes :
  - premièrement, exhiber un algorithme permettant de résoudre ce problème ; on obtient ainsi une borne supérieure pour la complexité du dit-problème (on parle de preuve d'appartenance à une classe de complexité) ;
  - deuxièmement, affiner le résultat obtenu précédemment soit en trouvant une borne inférieure (par preuve de complétude), soit en prouvant l'impossibilité d'appartenance à des sous-classes de la classe de complexité concernée ; dans tous les cas, cela revient à trouver une transformation polynomiale permettant de passer d'un problème "connu" (au sens de la complexité) au problème étudié.

  De plus, les problèmes traités ont de nombreux points communs, ce qui permet d'utiliser souvent le même type d'algorithmes pour la preuve d'appartenance ou les mêmes problèmes pour les preuves de complétude. Ainsi, il n'est pas surprenant de constater que certaines preuves sont *presque* identiques à d'autres. On pourrait être tenté de "factoriser" ces démonstrations s'il n'y avait justement ce "presque" qui représente les spécificités de chaque problème et impose par là-même la multiplication des démonstrations. Aux lecteurs pressés, je conseillerai donc de passer directement à la section 4.1.4 page 144 dans laquelle je récapitule tous les résultats de complexité obtenus. Quant à ceux intéressés par les preuves, voici quelques fils d'Ariane pour les guider dans le labyrinthe des démonstrations :

  - les preuves pour les problèmes EXI-m, qui sont tous de même complexité (par exemple Σ_2^p-complets dans le cas général), sont *presque* toutes identiques (algorithmes très ressemblants, utilisation du même problème pour la partie complétude) ;
  - les preuves pour les problèmes ARG-m, qui sont tous de même complexité (par exemple, dans le cas général, Δ_3^p - $(\Sigma_2^p \cup \Pi_2^p)$, si $\Sigma_2^p \neq \Pi_2^p$), sont *presque* toutes identiques (algorithmes très ressemblants, pas de preuve de complétude mais utilisation des mêmes problèmes pour affiner la borne supérieure de complexité) ;
  - le cas des UNI-m est un peu plus complexe car certains appartiennent à une classe de complexité et d'autres à une autre classe ; on a donc deux techniques distinctes :
    - les problèmes UNI-m qui sont Π_2^p-complets sont traités à partir de leurs co-problèmes (qui sont Σ_2^p-complets) de la même façon que les EXI-m ;
    - les autres sont autant de cas particuliers à traiter ; en particulier ceux faisant intervenir la notion de cardinalité qui induisent un phénomène de "masquage" constituant le cœur des démonstrations de complexité de ces problèmes ; de même, le cas de UNI-S est tout à fait spécial, puisqu'on constate que ce problème est équivalent à l'inférence classique et qu'il n'apporte donc rien dans le cadre d'un raisonnement non-monotone ;

[1] Aux lecteurs intéressés par le contenu des preuves, je conseille une lecture rapide de l'annexe citée, dans laquelle se trouvent les théorèmes et définitions utilisés, mais surtout les démonstrations de la complexité de quelques processus de révision de croyances. Ce sont les mécanismes mis en œuvre dans ces démonstrations qui m'ont largement inspirée pour la suite.

- on cherche à exploiter au maximum les cas particuliers pour traiter les cas généraux[2] ; c'est par exemple les cas entre les mécanismes T et Incl, ou entre Car et Lex ;
- de même, on utilise au mieux les liens, les équivalences entre certains problèmes (par exemple, ceux induits par les mécanismes T et S, ou ceux issus de Incl et E*).

Tous les résultats de cette section ont fait l'objet de publications (voir [CLS93, LS94, CLS94b]).

4.1.2 Étude de complexité dans le cas général

4.1.2.1 Étude de la complexité de Uni-T

Rappel : Le problème Uni-T est le suivant : "vérifier que H est une conséquence forte de E[3] en utilisant les thèses définies à partir de E".
Notation : $E \mid\!\sim^{\forall,T} H$.

Théorème 4.1.1 Uni-T *est* Π_2^p*-complet.*

Preuve : 1ère partie : l'appartenance à une classe.

Notons que ce mécanisme d'inférence non-monotone correspond exactement au mécanisme de révision Sbr étudié par Nebel dans [Neb91] (voir l'annexe A page 235). La démonstration proposée par Nebel s'applique donc parfaitement au mécanisme Uni-T. Je vais quand même la reprendre dans son intégralité[4].

Raisonner sur Uni-T nécessite de passer en revue toutes les thèses de E pour savoir si H est conséquence forte. On va donc faire la démonstration plutôt sur le problème co-Uni-T ($E \not\mid\!\sim^{\forall,T} H$) car dans ce cas-là il suffira de trouver une seule thèse qui n'infère pas H[5]. On propose l'algorithme 4.1 pour co-Uni-T.

Algorithme 4.1 : co-Uni-T (E, H)
début
    deviner un sous-ensemble Y de E
1   vérifier que Y est une thèse de E
    vérifier que Y n'infère pas H
fin

Pour résoudre le point 1 de l'algorithme 4.1, on peut vérifier la consistance de Y puis prendre chaque élément z de $E \setminus Y$ et vérifier l'inconsistance de $Y \cup \{z\}$.

On obtient ainsi pour co-Uni-T le nouvel algorithme 4.2 page suivante[6].

Quant à la résolution des points 1, 2, et 3 de l'algorithme ci-dessus, elle peut se faire en utilisant un oracle non déterministe polynomial, puisque le problème "Soit un ensemble de formules BC et une formule H, $BC \not\vdash H$?" est NP et que le problème "Soit un ensemble de formules BC et une formule H, $BC \vdash H$?" est co-NP. L'algorithme donné pour co-Uni-T est donc non déterministe (à cause du "deviner") polynomial et il fait

[2] En effet, la classe de complexité dans le cas général constitue une borne supérieure à la complexité dans le cas particulier. De même, la complétude dans le cas particulier implique la complétude dans le cas général si les deux cas correspondent à une même classe de complexité.

[3] Ici, l'ordre $<$ ne sert à rien, j'utiliserai donc la notation E au lieu de $(E, <)$. Cette remarque est valable pour Uni (Exi, Arg)-T (S, Car, E).

[4] Pourquoi la reprendre au lieu de se contenter de citer le résultat de Nebel ? Parce que les mécanismes mis en œuvre ici vont être rediscutés lors de l'étude des cas particuliers (voir section 4.1.3 page 110).

[5] Cette remarque est valable pour toutes les démonstrations portant sur une relation d'inférence utilisant la conséquence forte. Elle ne sera donc pas répétée.

[6] Cette technique de raffinage de l'algorithme initial sera systématiquement utilisée pour toute étude d'appartenance concernant les relations d'inférence utilisant des thèses. Elle ne sera donc pas répétée.

Algorithme 4.2 : co-Uni-T (E, H) (version raffinée)

début

deviner un sous-ensemble Y de E

1 vérifier que Y est consistant

pour chaque élément z de $E \setminus Y$ **faire**

2 ⌊ vérifier que $Y \cup \{z\}$ est inconsistant

3 vérifier que Y n'infère pas H

fin

appel à un oracle lui aussi non déterministe polynomial. Ainsi, le problème co-Uni-T appartient à la classe de complexité $\mathsf{NP}^{\mathrm{NP}} = \Sigma_2^p$. Le problème Uni-T appartient alors à la classe de complexité Π_2^p.

Attention, il s'agit là d'une limite maximale[7].

2ème **partie : la complétude ?**

On espère montrer que Uni-T est Π_2^p-complet, donc que co-Uni-T est Σ_2^p-complet. On connaît des problèmes Σ_2^p-complets, et si on arrive à ramener un de ces problèmes à co-Uni-T par une transformation polynomiale, on aura démontré la complétude[8]. Prenons par exemple le problème 2-Qbf (voir entre autres [Neb91], [Joh90] et la démonstration de Sbr dans l'annexe A page 235). La question est de savoir si on peut passer de 2-Qbf à co-Uni-T grâce à une transformation polynomiale.

Soit "$\exists a \forall b G(a,b)$" une instance de 2-Qbf, posons :
$E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, \neg G(a,b)\}$, et $H = \neg G(a,b)$.

Remarques :

- les thèses de E sont de la forme :
  - soit $\{l_1, \ldots, l_n\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i$, et $\{l_1, \ldots, l_n\}$ inconsistant avec $\neg G(a,b)$,
  - soit $\{l_1, \ldots, l_n, \neg G(a,b)\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i$, et $\{l_1, \ldots, l_n\}$ consistant avec $\neg G(a,b)$,
- on a dans E toutes les "valeurs" possibles pour les variables propositionnelles $a_1, \ldots, a_n$ dont on cherche s'il existe un modèle,
- on n'impose aucune contrainte sur les variables propositionnelles $b_1, \ldots, b_m$, dont tous les modèles doivent être pris en compte.

On a alors :

$\exists a \forall b G(a,b)$ satisfiable

$\Leftrightarrow$

il existe une assignation des variables propositionnelles $a_1, \ldots, a_n$ telle que sans la moindre contrainte sur les variables propositionnelles $b_1, \ldots, b_m$ (donc pour toute assignation de $b_1, \ldots, b_m$) on a $G(a,b)$ vraie[9]

$\Leftrightarrow$

$\exists \{l_1, \ldots, l_n\}$ telle que $\{l_1, \ldots, l_n\} \models G(a,b)$

$\Leftrightarrow$

[7] Cette remarque est valable pour toutes les démonstrations d'appartenance à une classe de complexité. Elle ne sera donc pas répétée.

[8] Ce mécanisme de raisonnement est systématiquement utilisé dès que l'on cherche à prouver la complétude.

[9] Cette étape du raisonnement sur 2-Qbf permettant le passage de "$\exists a \forall b G(a,b)$ satisfiable" à "$\exists \{l_1, \ldots, l_n\}$ telle que $\{l_1, \ldots, l_n\} \vdash G(a,b)$" est systématique et ne sera donc pas répétée.

$$\exists\{l_1,\ldots,l_n\} \text{ telle que } \{l_1,\ldots,l_n\} \text{ inconsistante avec } \neg G(a,b)$$
$$\Leftrightarrow$$
$$\exists \text{ une thèse } \{l_1,\ldots,l_n\} \not\vdash \neg G(a,b)$$
$$\Leftrightarrow$$
$$E \not\mid\!\sim^{\forall,T} H$$

On a donc montré que 2-QBF $\propto$ co-UNI-T. Or 2-QBF est Σ_2^p-complet et on a vu que co-UNI-T $\in \Sigma_2^p$. On en déduit que co-UNI-T est Σ_2^p-complet et donc que UNI-T est Π_2^p-complet. □

4.1.2.2 Étude de la complexité de EXI-T

Rappel : Le problème EXI-T est le suivant : "vérifier que H est une conséquence faible de E en utilisant les thèses définies à partir de E".
Notation : $E \mid\!\sim^{\exists,T} H$.

Théorème 4.1.2 EXI-T *est* Σ_2^p*-complet.*

Preuve : 1ère partie : l'appartenance à une classe.

On propose l'algorithme 4.3 pour EXI-T.

Algorithme 4.3 : EXI-T (E, H)
début
 deviner un sous-ensemble Y de E
 vérifier que Y est une thèse de E
 vérifier que Y infère H
fin

Cet algorithme est non déterministe (à cause du "deviner") polynomial et il fait appel à un oracle lui aussi non déterministe polynomial. Le problème EXI-T appartient donc à la classe de complexité $\mathsf{NP}^{\mathrm{NP}} = \Sigma_2^p$.

2ème partie : la complétude ?

Reprenons le problème 2-QBF et essayons de passer de 2-QBF à EXI-T grâce à une transformation polynomiale.

Soit "$\exists a \forall b G(a,b)$", une instance de 2-QBF, posons :
$H = G(a,b)$ et $E = \{a_1,\ldots,a_n,\neg a_1,\ldots,\neg a_n\}$.

Les remarques faites lors de la démonstration de UNI-T sont toujours valables, exception faite de la première qui devient :

- la base E est égale à $\{a_1,\ldots,a_n,\neg a_1,\ldots,\neg a_n\}$, ce qui signifie que quelle que soit la thèse Y, c'est-à-dire une sous-base maximale consistante, on a $Y = \{l_1,\ldots,l_n\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i = 1\ldots n$.

Montrons alors que : "$\{a_1,\ldots,a_n,\neg a_1,\ldots,\neg a_n\}$ infère $G(a,b)$ avec la méthode EXI-T" équivaut à "$\exists a \forall b G(a,b)$ satisfiable" :

$$\{a_1,\ldots,a_n,\neg a_1,\ldots,\neg a_n\} \text{ infère } G(a,b) \text{ avec la méthode EXI-T}$$
$$\Leftrightarrow$$
$$\text{il existe un sous-ensemble } Y \text{ de } E \text{ tel que } Y \text{ soit une thèse et infère } G(a,b)$$

$$\Leftrightarrow$$

$$\text{il existe un sous-ensemble } Y = \{l_1, \ldots, l_n\} \text{ avec } l_i = a_i \text{ ou } \neg a_i,\ \forall i = 1, \ldots, n$$
$$\text{tel que } Y \vdash G(a,b)$$
$$\Leftrightarrow$$
$$\text{il existe } \{l_1, \ldots, l_n\} \text{ qui infère } G(a,b)$$
$$\Leftrightarrow$$
$$\exists a \forall b G(a,b) \text{ est satisfiable}$$

On a donc montré que 2-QBF $\propto$ EXI-T. Or 2-QBF est Σ_2^p-complet et EXI-T $\in \Sigma_2^p$. On en déduit que EXI-T est Σ_2^p-complet[10]. □

4.1.2.3 Étude de la complexité de ARG-T

Rappel : Le problème ARG-T est le suivant : "vérifier que H est une conséquence argumentative de E en utilisant les thèses définies à partir de E".
Notation : $E \mid\!\sim^{A,T} H$.

Théorème 4.1.3 *Si* $\Sigma_2^p \neq \Pi_2^p$, ARG-T $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$.

J'ai choisi d'exprimer ce résultat sous cette forme afin de bien "visualiser" la place occupée par le problème étudié dans la classe Δ_3^p, plutôt que d'utiliser la forme "ARG-T $\in \Delta_3^p$ et est Σ_2^p-difficile et Π_2^p-difficile".

Preuve : 1ère partie : l'appartenance à une classe.

On propose l'algorithme 4.4 pour ARG-T.

Algorithme 4.4 : ARG-T (E, H)
début
 vérifier que $E \not\mid\!\sim^{\exists,T} \neg H$
 vérifier que $E \mid\!\sim^{\exists,T} H$
fin

Cet algorithme est polynomial et il fait appel à un oracle de complexité Σ_2^p (EXI-T)[11].
La complexité de ARG-T est donc au plus $\mathsf{P}^{\Sigma_2^p} = \Delta_3^p$.

2ème **partie : la complétude ?**

L'étude de la complétude est plus difficile. En effet, on connaît quelques problèmes Δ_3^p-complets, par exemple le problème du double sac à dos de [Joh90], ou quelques problèmes d'abduction de [EG93]. Toutefois, je n'ai pas réussi à ramener polynomialement un de ces problèmes à ARG-T[12]. On ne peut donc pas prouver simplement la complétude, si complétude il y a.

---

[10] Ce résultat n'est pas surprenant. Dans [EG93], Eiter et Gottlob ont défini le problème abductif suivant :

instance : $P = (V, H, M, T)$ un PAP (problème d'abduction propositionnel) avec V un ensemble de variables propositionnelles, H un ensemble d'hypothèses (atomes propositionnels), M un ensemble de manifestations (formules propositionnelles), T une théorie consistante (formules propositionnelles) ;
question : Existe-t-il une explication pour P ? (une explication est un sous-ensemble H' tel que $H' \subseteq H$, $T \cup H'$ est consistant, $(T \cup H') \models M$).

Ce problème peut être transformé en EXI-T de la manière suivante : $E = T \cup H$ et $G = M$.

[11] Voir la démonstration de la complexité du mécanisme de révision FMR (théorème A.2.1 page 238 de l'annexe A page 235) pour l'utilisation dans un algorithme de deux oracles, l'un de complexité X et l'autre de complexité co-X.

[12] Le problème du double sac à dos de [Joh90] se ramène sans difficulté majeure à un problème de déduction en logique des prédicats. Malheureusement, il faut que la logique utilisée dans le problème ARG-T étudié ici soit la logique propositionnelle, sinon les problèmes de satisfiabilité deviennent indécidables et donc n'appartiennent plus à NP.

Par contre, on peut peut-être raffiner le résultat d'appartenance à une classe comme Nebel l'a fait dans le calcul des complexités de FMR et UBR (voir dans [Neb91] et dans l'annexe A page 235), c'est-à-dire montrer que ARG-T $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$.

Pour cela on compare le problème ARG-T à des problèmes de classe inférieure dans la hiérarchie polynomiale.

On constate alors que :

- le problème EXI-T se ramène polynomialement à ARG-T :

  Soit "E, H", une instance de EXI-T, posons :
  - $f(E) = E \cup \{H \rightarrow G\}$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans E),
  - $f(H) = G$.

  Remarques préliminaires :

  - $H \rightarrow G$ est consistante avec n'importe quelle thèse Y' de E : soit Y' thèse. Y' est consistante, donc admet un modèle M'. Or G n'apparaît pas dans E, donc n'apparaît pas dans Y'. On peut ainsi étendre le modèle M' de Y' à un modèle M en rajoutant la valeur *vrai* pour la variable propositionnelle G. On a ainsi M modèle de Y' et modèle de $H \rightarrow G$. $H \rightarrow G$ est donc consistante avec toute thèse de E.
  - Aucune des thèses de $f(E)$ n'infère $\neg G$.
  - Quelle que soit Y thèse de $f(E)$, Y sera de la forme $Y = Y' \cup \{H \rightarrow G\}$ avec Y' thèse de E. Ceci vient du fait que la formule $H \rightarrow G$ est consistante avec toutes les thèses de E.
  - Quelle que soit Y thèse de $f(E)$, $Y \vdash G$ ssi $Y' \vdash H$, car :
    - $Y' \vdash H \Rightarrow \forall M'$ modèle de Y', $M'(H) = vrai$ (soit X une formule, $M(X)$ est la valeur de vérité de X dans l'interprétation M) ; on cherche à montrer que $Y \vdash G$, donc que $\forall M$ modèle de Y, M est un modèle de G ; M étant un modèle de Y et $Y = Y' \cup \{H \rightarrow G\}$, M est donc un modèle de Y' et de $H \rightarrow G$; ainsi, on a $M(H) = vrai$ et $M(H \rightarrow G) = vrai$, donc $M(G) = vrai$, donc $Y \vdash G$.
    - $Y \vdash G$ et supposons que $Y' \not\vdash H \Rightarrow \exists M'$ modèle de Y' tel que $M'(H) = faux \Rightarrow \exists M$ modèle de Y défini par M' étendu avec $M(G) = faux$ ce qui contredit l'hypothèse $Y \vdash G$.

  Montrons alors que : $f(E) \mid\!\sim^{A,T} f(H) \Leftrightarrow E \mid\!\sim^{\exists,T} H$:

  $$f(E) \mid\!\sim^{A,T} G$$
  $$\Leftrightarrow$$
  il existe Y thèse de $f(E)$ telle que $Y \vdash G$ et aucune des autres thèses n'infère $\neg G$
  $$\Leftrightarrow$$
  il existe Y thèse de $f(E)$ telle que $Y \vdash G$ (voir remarques préliminaires)
  $$\Leftrightarrow$$
  il existe Y' thèse de E telle que $Y' \vdash H$ (voir remarques préliminaires)
  $$\Leftrightarrow$$
  $$E \mid\!\sim^{\exists,T} H$$

- le problème co-EXI-T se ramène polynomialement à ARG-T :

  Soit "E, H", une instance de co-EXI-T, posons :
  - $f(E) = E \cup \{\neg H\}$,
  - $f(H) = \neg H$.

  Remarque préliminaire : Soit Y une thèse de $f(E) = E \cup \{\neg H\}$:

- soit Y contient $\neg H$:
  - soit $Y = Y' \cup \{\neg H\}$ avec Y' thèse de E consistante avec $\neg H$ et $\neg H \notin Y'$ donc $Y' \not\vdash H$ et Y infère $\neg H$,
  - soit $Y = Y'$ avec Y' thèse de E consistante avec $\neg H$ et $\neg H \in Y'$ donc $Y' \not\vdash H$ et Y infère $\neg H$,
  - soit il n'existe pas de thèse Y' de E consistante avec $\neg H$ et alors il existe un sous-ensemble S de E consistant, qui peut être vide, tel que $Y = S \cup \{\neg H\}$,
- soit Y ne contient pas $\neg H$ et alors $Y = Y'$ avec Y' thèse de E inconsistante avec $\neg H$ donc inférant H.

On a alors les propriétés suivantes :

Propriété 4.1.1 *Si toutes les thèses Y' de E sont consistantes avec $\neg H$ (donc n'infèrent pas H) alors toutes les thèses de $f(E)$ sont :*

- *soit de la forme $Y = Y' \cup \{\neg H\}$ si $\neg H \notin Y'$,*
- *soit de la forme $Y = Y'$ si $\neg H \in Y'$,*

et, bien sûr, elles infèrent $\neg H$ et n'infèrent pas H ;

Propriété 4.1.2 *S'il existe une thèse Y' de E qui infère H (donc inconsistante avec $\neg H$) alors il existe une thèse Y de $f(E)$ de la forme $Y = Y'$ qui infère H ;*

Propriété 4.1.3 *(contraposée de la propriété 4.1.2) si aucune thèse Y de $f(E)$ n'infère H alors aucune thèse Y' de E n'infère H.*

Montrons alors que : $f(E) \mid\!\sim^{A,T} f(H) \Leftrightarrow E \mid\!\not\sim^{\exists,T} H$.
Tout d'abord :

$$\begin{gathered} f(E) \mid\!\sim^{A,T} \neg H \\ \Leftrightarrow \\ \exists Y \text{ thèse de } f(E) \text{ telle que } Y \vdash \neg H \\ \text{et } \forall Y \text{ thèse de } f(E)\ Y \not\vdash H \\ \Rightarrow \\ \text{(à cause de la propriété 4.1.3)} \\ \forall Y' \text{ thèse de } E,\ Y' \not\vdash H \\ \Leftrightarrow \\ E \mid\!\not\sim^{\exists,T} H \end{gathered}$$

D'autre part :

$$\begin{gathered} E \mid\!\not\sim^{\exists,T} H \\ \Leftrightarrow \\ \forall Y' \text{ thèse de } E,\ Y' \not\vdash H \\ \Rightarrow \\ \text{(à cause de la propriété 4.1.1)} \\ \forall Y \text{ thèse de } f(E),\ Y \not\vdash H \text{ et } Y \vdash \neg H \\ \Rightarrow \\ \exists Y \text{ thèse de } f(E) \text{ telle que } Y \vdash \neg H \\ \text{et } \forall Y \text{ thèse de } f(E)\ Y \not\vdash H \\ \Leftrightarrow \\ f(E) \mid\!\sim^{A,T} \neg H \end{gathered}$$

On a donc EXI-T $\propto$ ARG-T et co-EXI-T $\propto$ ARG-T (avec EXI-T qui est Σ_2^p-complet et co-EXI-T qui est Π_2^p-complet). Donc ARG-T $\in \Delta_3^p$ et est Σ_2^p-difficile et Π_2^p-difficile. En utilisant le théorème A.1.4 page 237 donné dans l'annexe A page 235[13], on prouve donc que, si $\Sigma_2^p \neq \Pi_2^p$, ARG-T $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$. □

[13] Ce théorème A.1.4 page 237 sera utilisé de manière systématique dans les démonstrations du type "le problème $\in \Delta_k^p$ - $(\Sigma_{k-1}^p \cup \Pi_{k-1}^p)$". Il ne sera donc pas cité chaque fois.

4.1.2.4 Étude de la complexité de Uni-S

Rappel : Le problème Uni-S est le suivant : "vérifier que H est une conséquence forte de E en utilisant les sous-bases consistantes définies à partir de E".
Notation : $E \mid\!\sim^{\forall,S} H$.

Théorème 4.1.4 Uni-S *est co-*NP*-complet.*

> **Preuve : 1$^{\text{ère}}$ partie : l'appartenance à une classe.**
>
> On propose l'algorithme 4.5 pour Uni-S.
>
> **Algorithme 4.5 :** Uni-S (E, H)
> **début**
>   **pour** tout élément G_i de la base E **faire**
>   vérifier que $G_i \vdash H$
> **fin**
>
> Or, on sait que : $(\forall G_i \in E, i = 1 \ldots n, G_i \vdash H) \Leftrightarrow ((\bigvee_i G_i) \vdash H)$. Donc l'algorithme 4.5 se simplifie et devient l'algorithme 4.6.
>
> **Algorithme 4.6 :** Uni-S (E, H) (version simplifiée)
> **début**
>   vérifier que $(\bigvee_i G_i) \vdash H$ avec $i = 1 \ldots n$
> **fin**
>
> Le problème Uni-S se ramène donc exactement au problème de l'inférence classique et est donc co-NP-complet. □

D'autre part, se posent les problèmes de la prise en compte des tautologies et de l'ensemble vide en tant que sous-bases consistantes. Pour cette raison, cette relation d'inférence non-monotone ne présente aucun intérêt, et ne sera donc pas étudiée plus en détail.

4.1.2.5 Étude de la complexité de Exi-S

Rappel : Le problème Exi-S est le suivant : "vérifier que H est une conséquence faible de E en utilisant les sous-bases consistantes définies à partir de E".
Notation : $E \mid\!\sim^{\exists,S} H$.

Théorème 4.1.5 Exi-S *est Σ_2^p-complet.*

> **Preuve : 1$^{\text{ère}}$ partie : l'appartenance à une classe.**
>
> Rappelons d'abord que nous avons la propriété 2.2.1 page 8 qui revient à dire que toute instance positive pour Exi-T est aussi une instance positive pour Exi-S, et vice-versa. Donc Exi-T = Exi-S. Ainsi, Exi-S a la même complexité que Exi-T. □

4.1.2.6 Étude de la complexité de Arg-S

Rappel : Le problème Arg-S est le suivant : "vérifier que H est une conséquence argumentative de E en utilisant les sous-bases consistantes définies à partir de E".
Notation : $E \mid\!\sim^{A,S} H$.

Théorème 4.1.6 *Si* $\Sigma_2^p \neq \Pi_2^p$, ARG-S $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$.

Preuve :

J'ai toujours la propriété 2.2.1 page 8 déjà utilisée pour la démonstration de EXI-S.

Or, l'algorithme 4.7 que je propose pour le problème ARG-S utilise les oracles EXI-S et co-EXI-S. Cet algorithme est donc équivalent à l'algorithme 4.8.

Algorithme 4.7 : ARG-S (E, H)
début
 vérifier que $E \not\mid\!\sim^{\exists,S} \neg H$
 vérifier que $E \mid\!\sim^{\exists,S} H$
fin

Algorithme 4.8 : ARG-S (E, H) (nouvelle version)
début
 vérifier que $E \not\mid\!\sim^{\exists,T} \neg H$
 vérifier que $E \mid\!\sim^{\exists,T} H$
fin

Donc ARG-T = ARG-S. Ainsi, ARG-S a la même complexité que ARG-T. □

4.1.2.7 Étude de la complexité de UNI-BO

Rappel : Le problème UNI-BO est le suivant : "vérifier que H est une conséquence forte avec préférences de $(E,<)$[14] en utilisant l'ordre "Best-Out" sur les sous-bases consistantes".
Notation : $(E,<) \mid\!\sim^{\forall,Bo} H$.

Théorème 4.1.7 *Si* NP $\neq$ *co*-NP, UNI-BO $\in \Delta_2^p$ - (NP $\cup$ *co*-NP).

Preuve : 1^ère^ partie : l'appartenance à une classe.

Il faut d'abord remarquer, vu le mécanisme de construction des sous-bases consistantes préférées pour l'ordre "Best-Out" (voir section 2.2.4 page 19), que :

- les sous-bases consistantes préférées pour l'ordre "Best-Out" sont les sous-ensembles Y qui ont tous le même $a(Y) = amax$ et tels que $\cap Y = \cup_{i=1\ldots amax-1} E_i$ (cette intersection étant consistante),
- pour que toutes les sous-bases consistantes préférées pour l'ordre "Best-Out" infèrent une même formule, il faut que cette formule soit inférée par leur intersection : $\forall Y_i,\ Y_i \vdash H \Leftrightarrow \cap Y_i \vdash H$.

On propose l'algorithme 4.9 pour UNI-BO.

Algorithme 4.9 : UNI-BO ($(E,<)$, H)
début
1 calculer le $amax$ de $(E,<)$
 $Y = \cup_{i=1\ldots amax-1} E_i$ *(* Y est consistant *)*
 vérifier que Y infère H
fin

La décomposition de l'étape 1 *"calculer le amax de* $(E,<)$*"* se fait à l'aide de l'algorithme 4.10 page ci-contre.

[14] À partir de maintenant, les problèmes considérés prennent en compte l'ordre $<$ sur E. On utilisera donc la notation $(E,<)$ au lieu de E. Cette remarque est valable pour UNI(EXI, ARG)-BO(INCL, LEX).

Algorithme 4.10 : CALCUL AMAX $((E, <))$
début

$nb \leftarrow 1$
$Z \leftarrow E_1$
tant que Z est consistant et $nb \leq$ nombre total de strates dans E **faire**
  $nb \leftarrow nb + 1$
  $Z \leftarrow Z \cup E_{nb}$
$amax \leftarrow nb$

fin

L'algorithme complet est donc polynomial et il fait appel à un oracle non déterministe polynomial. Le problème UNI-BO appartient donc à la classe de complexité $\mathsf{P}^{\mathrm{NP}} = \Delta_2^p$.

2ème **partie : la complétude ?**

L'étude de la complétude est plus difficile. Essayons de comparer le problème UNI-BO avec des problèmes de classe inférieure dans la hiérarchie polynomiale, c'est-à-dire des problèmes NP et co-NP, par exemple les problèmes GSAT et UNGSAT. On constate alors que :

- le problème GSAT se ramène au problème UNI-BO par une transformation polynomiale :

  Soit "G" une instance de GSAT, posons :
  $E = \{G\}$ et $H = G$.

  Remarquons que $(E, <)$ est réduit à une seule strate, il est donc noté E[15]. Deux cas se présentent :
  - si G est consistante (donc satisfiable) alors $amax = 2$ et la seule sous-base consistante préférée pour l'ordre "Best-Out" est $Y = \{G\}$; le problème UNI-BO se ramène donc à montrer que $G \vdash G$, ce qui est toujours vrai ;
  - sinon (G est inconsistante donc non satisfiable) $amax = 1$ et la seule sous-base consistante préférée pour l'ordre "Best-Out" est $Y = \varnothing$; le problème UNI-BO se ramène donc à montrer que $\vdash G$, ce qui est faux quand G est non satisfiable.

  Ainsi on aura bien : G est satisfiable $\Leftrightarrow E \mid\!\sim^{\forall, Bo} H$.
  En conclusion, GSAT se transforme polynomialement en UNI-BO (noté GSAT $\propto$ UNI-BO).
- le problème UNGSAT se ramène au problème UNI-BO de manière polynomiale :

  Soit "G" une instance de UNGSAT, posons :
  $E = \varnothing$ et $H = \neg G$.

  Remarquons alors que $amax = 1$ et que la seule sous-base consistante préférée pour l'ordre "Best-Out" est $Y = \varnothing$; le problème UNI-BO se ramène donc à montrer que $\vdash \neg G$, ce qui est faux quand G est satisfiable et vrai quand G est non satisfiable.
  On a donc : UNGSAT $\Leftrightarrow G$ non satisfiable $\Leftrightarrow E \mid\!\sim^{\forall, Bo} H$.
  En conclusion, on a UNGSAT $\propto$ UNI-BO.

En suivant le même raisonnement que pour ARG-T (voir la preuve du théorème 4.1.3 page 78), on prouve que, si NP $\neq$ co-NP, UNI-BO $\in \Delta_2^p$ - (NP $\cup$ co-NP). □

4.1.2.8 Étude de la complexité de EXI-BO

Rappel : Le problème EXI-BO est le suivant : "vérifier que H est une conséquence faible avec préférences de $(E, <)$ en utilisant l'ordre "Best-Out" sur les sous-bases consistantes".

[15] Ce cas se produit souvent. Cette remarque ne sera donc pas répétée.

Notation : $(E,<) \mid\!\sim^{\exists, Bo} H$.

Théorème 4.1.8 Exi-Bo *est* Σ_2^p*-complet.*

Preuve : 1$^{\text{ère}}$ partie : l'appartenance à une classe.

On propose l'algorithme 4.11 pour Exi-Bo.

Algorithme 4.11 : Exi-Bo $((E,<),\ H)$
début
deviner un sous-ensemble Y de $(E,<)$
1 vérifier que Y est une sous-base consistante bo-préférée de $(E,<)$
vérifier que Y infère H
fin

La décomposition de l'étape 1 *"vérifier que Y est une sous-base consistante préférée pour l'ordre "Best-Out" de $(E,<)$"* se fait à l'aide de l'algorithme 4.12.

Algorithme 4.12 : Vérifier que Y est une sous-base consistante préférée pour l'ordre "Best-Out" de $(E,<)$
début
vérifier que Y est consistant
calculer $a(Y)$
$nb \leftarrow 1$
$Z \leftarrow E_1$
tant que Z est consistant et $nb \leq$ nombre total de strates dans E **faire**
  $nb \leftarrow nb + 1$
  $Z \leftarrow Z \cup E_{nb}$
$amax \leftarrow nb$
vérifier que $a(Y) = amax$
fin

L'algorithme complet est non déterministe (à cause du "deviner") polynomial et il fait appel à un oracle lui aussi non déterministe polynomial. Le problème Exi-Bo appartient donc à la classe de complexité $\mathsf{NP}^{\mathrm{NP}} = \Sigma_2^p$.

2$^{\text{ème}}$ **partie : la complétude ?**

On connaît des problèmes Σ_2^p-complets. La question est donc de savoir si on peut passer de l'un de ces problèmes à Exi-Bo grâce à une transformation polynomiale. Choisissons par exemple le problème 2-Qbf.

Soit "$\exists a \forall b G(a,b)$", une instance de 2-Qbf, posons :
$H = G(a,b)$, $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n\}$.

Ainsi :

- on a dans E toutes les "valeurs" possibles pour les variables propositionnelles $a_1, \ldots, a_n$ dont on cherche s'il existe une assignation,
- on n'impose aucune contrainte sur les variables propositionnelles $b_1, \ldots, b_m$, dont toutes les assignations doivent être prises en compte.

Remarques préliminaires :

- La base E est égale à $\{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n\}$ avec une seule strate, ce qui signifie que quelle que soit Y sous-base consistante, $a(Y) = 1$ donc Y est préférée pour l'ordre "Best-Out" et Y est un sous-ensemble de $\{l_1, \ldots, l_n\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i = 1 \ldots n$.

- Soit une formule H, s'il existe une sous-base consistante Y qui infère H, alors pour toute $Y' = Y \cup \{l_{k_1}, \dots, l_{k_m}\}$ avec les l_{k_j} représentant les variables propositionnelles a_k non affectées par Y, Y' sera consistant et inférera H puisque la relation d'inférence classique est monotone.
- Soit une formule H, s'il existe un ensemble $Y' = \{l_1, \dots, l_n\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i = 1 \dots n$ tel que Y' infère H, alors Y' est une sous-base préférée pour l'ordre "Best-Out".

Montrons alors que : "$\{a_1, \dots, a_n, \neg a_1, \dots, \neg a_n\}$ infère $G(a,b)$ avec la méthode Exi-Bo" équivaut à "$\exists a \forall b G(a,b)$ est satisfiable" :

$\{a_1, \dots, a_n, \neg a_1, \dots, \neg a_n\}$ infère $G(a,b)$ avec la méthode Exi-Bo

$\Leftrightarrow$

il existe un sous-ensemble Y de E tel que Y soit une sous-base consistante préférée pour l'ordre "Best-Out" et infère $G(a,b)$

$\Leftrightarrow$

il existe une sous-base consistante Y, sous-ensemble de $\{l_1, \dots, l_n\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i = 1 \dots n$, telle que $Y \vdash G(a,b)$ et $a(Y) = 1$

$\Leftrightarrow$

il existe $\{l_1, \dots, l_n\}$ qui infère $G(a,b)$

$\Leftrightarrow$

$\exists a \forall b G(a,b)$ est satisfiable

On a donc montré que 2-Qbf $\propto$ Exi-Bo. Or 2-Qbf est Σ_2^p-complet et Exi-Bo $\in \Sigma_2^p$. On en déduit que Exi-Bo est Σ_2^p-complet. □

4.1.2.9 Étude de la complexité de Arg-Bo

Rappel : Le problème Arg-Bo est le suivant : "vérifier que H est une conséquence argumentative avec préférences de $(E,<)$ en utilisant l'ordre "Best-Out" sur les sous-bases consistantes".
Notation : $(E,<) \mid\!\sim^{A,Bo} H$.

Théorème 4.1.9 *Si* $\Sigma_2^p \neq \Pi_2^p$, Arg-Bo $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$.

Preuve : 1ère partie : l'appartenance à une classe.

On propose l'algorithme 4.13 pour Arg-Bo.

Algorithme 4.13 : Arg-Bo $((E,<), H)$
début
  vérifier que $(E,<) \not\mid\!\sim^{\exists,Bo} \neg H$
  vérifier que $(E,<) \mid\!\sim^{\exists,Bo} H$
fin

Cet algorithme est polynomial et il fait appel à un oracle de complexité Σ_2^p (Exi-Bo).

La complexité de Arg-Bo est donc au plus $\mathsf{P}^{\Sigma_2^p} = \Delta_3^p$.

2ème partie : la complétude ?

Essayons de comparer le problème Arg-Bo avec des problèmes de classe inférieure dans la hiérarchie polynomiale.

On constate alors que :

- le problème EXI-BO se ramène polynomialement à ARG-BO :

> Soit "$(E, <), H$", une instance de EXI-BO, posons :
> - $f(H) = G$,
> - $f((E, <)) = (E \cup \{H \rightarrow G\}, <)$[a]
>
> avec G nouvelle variable propositionnelle (G n'apparaît pas dans $(E, <)$) et la formule $H \rightarrow G$ placée seule en *première* strate.
>
> [a] En fait, il ne s'agit pas du pré-ordre $<$ (défini uniquement sur E) mais du pré-ordre $<'$ correspondant à l'extension de $<$ en rajoutant $\Phi < (H \rightarrow G)$, $\forall \Phi \in E$. Toutefois, afin de ne pas alourdir les notations, je continuerai à utiliser $<$ au lieu de $<'$. Cette remarque est valable pour tous les cas où l'on modifie la base E initiale stratifiée en rajoutant un nouvel élément.

Remarques préliminaires :

- $H \rightarrow G$ est consistante avec n'importe quelle sous-base consistante préférée pour l'ordre "Best-Out" Y' de E : soit Y' sous-base consistante préférée pour l'ordre "Best-Out". Y' est consistante, donc admet un modèle M'. Or G n'apparaît pas dans E, donc n'apparaît pas dans Y'. On peut ainsi étendre le modèle M' de Y' à un modèle M en rajoutant la valeur *vrai* pour la variable propositionnelle G. On a ainsi M modèle de Y' et modèle de $H \rightarrow G$. $H \rightarrow G$ est donc consistante avec toute sous-base consistante préférée pour l'ordre "Best-Out" de E.
- Quelle que soit Y sous-base consistante préférée pour l'ordre "Best-Out" de $f((E, <))$, Y sera de la forme $Y = Y' \cup \{H \rightarrow G\}$ avec Y' sous-base consistante préférée pour l'ordre "Best-Out" de $(E, <)$. Ceci vient du fait que la formule $H \rightarrow G$ est la plus prioritaire de $f((E, <))$ et qu'elle est consistante avec toutes les sous-bases consistantes préférées pour l'ordre "Best-Out" de $(E, <)$.
- Quelle que soit Y sous-base consistante préférée pour l'ordre "Best-Out" de $f((E, <))$, $Y \vdash G$ ssi $Y' \vdash H$ (voir argumentation dans la démonstration du théorème 4.1.3 page 78).

Montrons alors que : "$f((E, <))$ infère $f(H)$ avec la méthode ARG-BO" équivaut à "$(E, <)$ infère H avec la méthode EXI-BO" :

$f((E, <))$ infère G avec la méthode ARG-BO

$\Leftrightarrow$

il existe Y sous-base consistante préférée pour l'ordre "Best-Out" de $f((E, <))$ telle que $Y \vdash G$ et aucune des autres sous-bases consistantes préférées pour l'ordre "Best-Out" n'infère $\neg G$.

$\Leftrightarrow$

il existe Y sous-base consistante préférée pour l'ordre "Best-Out" de $f((E, <))$ telle que $Y \vdash G$ (voir remarques préliminaires)

$\Leftrightarrow$

il existe Y' sous-base consistante préférée pour l'ordre "Best-Out" de $(E, <)$ telle que $Y' \vdash H$ (voir remarques préliminaires)

$\Leftrightarrow$

$(E, <)$ infère H avec la méthode EXI-BO

- le problème co-EXI-BO se ramène polynomialement à ARG-BO :

> Soit "$(E, <), H$", une instance de co-EXI-BO, posons :
> - $f(H) = \neg H$,
> - $f((E, <)) = (E \cup \{\neg H\}, <)$
>
> avec $\neg H$ constituant seule la *dernière* strate.

Remarques préliminaires :

Soit Y une sous-base consistante préférée pour l'ordre "Best-Out" de $f((E,<))$, on a 2 cas possibles :

- soit Y ne contient pas $\neg H$ et alors Y est une sous-base consistante préférée pour l'ordre "Best-Out" de $(E,<)$ qui peut être soit consistante, soit inconsistante avec $\neg H$,
- soit Y contient $\neg H$ et :
  - soit $Y = Y' \cup \{\neg H\}$ où Y' est une sous-base consistante préférée pour l'ordre "Best-Out" de $(E,<)$ qui est consistante avec $\neg H$ et qui ne contient pas $\neg H$,
  - soit $Y = Y'$ où Y' est une sous-base consistante préférée pour l'ordre "Best-Out" de $(E,<)$ qui contient $\neg H$.

On constate alors que, si aucune des sous-bases consistantes préférées pour l'ordre "Best-Out" de $(E,<)$ n'infère H, alors aucune des sous-bases consistantes préférées pour l'ordre "Best-Out" de $f((E,<))$ n'inférera H, et réciproquement. D'autre part, s'il existe au moins une sous-base consistante préférée pour l'ordre "Best-Out" de $(E,<)$ consistante avec $\neg H$, alors on aura au moins une sous-base consistante préférée pour l'ordre "Best-Out" de $f((E,<))$ qui contiendra $\neg H$ et donc inférera $\neg H$. On a donc :

$$(E,<) \not\mid\!\sim^{\exists,Bo} H$$
$$\Leftrightarrow$$
il n'existe pas de sous-base consistante Y' préférée pour
l'ordre "Best-Out" de $(E,<)$ qui infère H
(donc toutes les sous-bases Y' sont consistantes avec $\neg H$)
$$\Leftrightarrow$$
il n'existe pas de sous-base consistante Y préférée pour
l'ordre "Best-Out" de $f((E,<))$ qui infère H et
il existe au moins une sous-base Y préférée pour l'ordre "Best-Out"
de $f((E,<))$ qui contient $\neg H$ et donc infère $\neg H$
$$\Leftrightarrow$$
$$f((E,<)) \mid\!\sim^{A,Bo} \neg H$$

On a donc Exi-Bo $\propto$ Arg-Bo et co-Exi-Bo $\propto$ Arg-Bo (avec Exi-Bo qui est Σ_2^p-complet et co-Exi-Bo qui est Π_2^p-complet). Donc, comme pour Arg-T, on prouve que, si $\Sigma_2^p \neq \Pi_2^p$, Arg-Bo $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$. □

4.1.2.10 Étude de la complexité de Uni-Incl

Rappel : Le problème Uni-Incl est le suivant : "vérifier que H est une conséquence forte avec préférences de $(E,<)$ en utilisant l'ordre "Inclusion Based" sur les thèses de E".

Notation : $(E,<) \mid\!\sim^{\forall,In} H$.

Théorème 4.1.10 Uni-Incl *est* Π_2^p*-complet.*

Preuve : 1ère partie : l'appartenance à une classe.

Notons que ce mécanisme d'inférence non-monotone correspond exactement au mécanisme de révision Pbr étudié par Nebel dans [Neb91] (voir l'annexe A page 235). Comme pour Uni-T, la démonstration proposée par Nebel s'applique parfaitement au mécanisme Uni-Incl. Pour les mêmes raisons que celles présentées pour Uni-T (voir théorème 4.1.1 page 75), cette démonstration est donc reprise dans son intégralité en utilisant Uni-T à la place de Sbr.

On propose l'algorithme 4.14 page suivante pour co-Uni-Incl.

Algorithme 4.14 : co-Uni-Incl $((E,<),\ H)$
début
deviner un sous-ensemble Y de $(E,<)$
vérifier que Y est une thèse préférée pour l'ordre "Inclusion Based" de $(E,<)$
vérifier que Y n'infère pas H
fin

Cet algorithme est non déterministe polynomial (à cause du "deviner") et il fait appel à un oracle lui aussi non déterministe polynomial. Le problème co-Uni-Incl appartient à la classe de complexité $\mathsf{NP}^{\mathrm{NP}} = \Sigma_2^p$.

Le problème Uni-Incl appartient alors à la classe de complexité Π_2^p.

$2^{\text{ème}}$ **partie : la complétude ?**

On connaît des problèmes Π_2^p-complets. Par exemple, le problème Uni-T. Or, il se trouve que Uni-Incl est une généralisation de Uni-T.

Soit "E, H", une instance de Uni-T, considérons toutes les formules de E en une seule strate.

Propriété 4.1.4 *Quand E est constitué d'une seule strate, les thèses de E sont exactement les thèses préférées pour l'ordre "Inclusion Based".*

On a alors :

$$E \mathrel{|\!\!\sim}^{\forall,T} H$$
$$\Leftrightarrow$$
Quelle que soit Y thèse de E, Y infère H
$$\Leftrightarrow$$
Quelle que soit Y thèse préférée pour l'ordre "Inclusion Based" de E, Y infère H
$$\Leftrightarrow$$
$$E \mathrel{|\!\!\sim}^{\forall,In} H$$

On a donc montré que Uni-T $\propto$ Uni-Incl. Or Uni-T est Π_2^p-complet et Uni-Incl $\in \Pi_2^p$. On en déduit que Uni-Incl est Π_2^p-complet. □

4.1.2.11 Étude de la complexité de Exi-Incl

Rappel : Le problème Exi-Incl est le suivant : "vérifier que H est une conséquence faible avec préférences de $(E,<)$ en utilisant l'ordre "Inclusion Based" sur les thèses de E".

Notation : $(E,<) \mathrel{|\!\!\sim}^{\exists,In} H$.

Théorème 4.1.11 Exi-Incl *est Σ_2^p-complet.*

Preuve : $1^{\text{ère}}$ **partie : l'appartenance à une classe.**

On propose l'algorithme 4.15 page suivante pour Exi-Incl.

Cet algorithme est non déterministe polynomial (à cause du "deviner") et il fait appel à un oracle lui aussi non déterministe polynomial. Le problème Exi-Incl appartient donc à la classe de complexité $\mathsf{NP}^{\mathrm{NP}} = \Sigma_2^p$.

$2^{\text{ème}}$ **partie : la complétude ?**

De la même manière que pour Uni-Incl, il est possible de passer de Exi-T à Exi-Incl grâce à une transformation polynomiale.

Algorithme 4.15 : EXI-INCL $((E,<),\ H)$
début
deviner un sous-ensemble Y de $(E,<)$
vérifier que Y est une thèse préférée pour l'ordre "Inclusion Based" de $(E,<)$
vérifier que Y infère H
fin

Soit "E, H", une instance de EXI-T, considérons toutes les formules de E en une seule strate.

La propriété 4.1.4 page ci-contre reste valable. On a alors :

$$E \mid\!\sim^{\exists,T} H$$
$$\Leftrightarrow$$
$$\text{il existe } Y \text{ thèse de } E \text{ qui infère } H$$
$$\Leftrightarrow$$
$$\text{il existe } Y \text{ thèse préférée pour l'ordre "Inclusion Based" de } E \text{ qui infère } H$$
$$\Leftrightarrow$$
$$E \mid\!\sim^{\exists,In} H$$

On a donc montré que EXI-T $\propto$ EXI-INCL. Or EXI-T est Σ_2^p-complet et EXI-INCL $\in \Sigma_2^p$. On en déduit que EXI-INCL est Σ_2^p-complet. □

4.1.2.12 Étude de la complexité de ARG-INCL

Rappel : Le problème ARG-INCL est le suivant : "vérifier que H est une conséquence argumentative avec préférences de $(E,<)$ en utilisant l'ordre "Inclusion Based" sur les thèses de E".
Notation : $(E,<) \mid\!\sim^{A,In} H$.

Théorème 4.1.12 *Si* $\Sigma_2^p \neq \Pi_2^p$, ARG-INCL $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$.

Preuve : 1$^{\text{ère}}$ partie : l'appartenance à une classe.

On propose l'algorithme 4.16 pour ARG-INCL.

Algorithme 4.16 : ARG-INCL $((E,<),\ H)$
début
vérifier que $(E,<) \mid\!\not\sim^{\exists,In} \neg H$
vérifier que $(E,<) \mid\!\sim^{\exists,In} H$
fin

Cet algorithme est polynomial et il fait appel à un oracle de complexité Σ_2^p (EXI-INCL).
La complexité de ARG-INCL est donc au plus $\mathsf{P}^{\Sigma_2^p} = \Delta_3^p$.

2$^{\text{ème}}$ **partie : la complétude ?**

Je vais comparer le problème ARG-INCL avec des problèmes de classe inférieure dans la hiérarchie polynomiale.

Pour cela, en utilisant encore la propriété 4.1.4 page ci-contre, on constate d'abord que ARG-T se ramène polynomialement à ARG-INCL.

Soit "E, H", une instance de ARG-T, considérons toutes les formules de E en une seule strate.

On a alors :

$$E \mid\!\sim^{A,T} H$$

$$\Leftrightarrow$$

il existe Y thèse de E qui infère H et aucune thèse de E n'infère $\neg H$

$$\Leftrightarrow$$

il existe Y thèse préférée pour l'ordre "Inclusion Based" de E qui infère H et aucune des thèses préférées pour l'ordre "Inclusion Based" de E n'infère $\neg H$

$$\Leftrightarrow$$

$$E \mid\!\sim^{A,In} H$$

On a donc bien ARG-T $\propto$ ARG-INCL. Or on a démontré que EXI-T $\propto$ ARG-T et que co-EXI-T $\propto$ ARG-T, donc par la transitivité de $\propto$, on a EXI-T $\propto$ ARG-INCL et co-EXI-T $\propto$ ARG-INCL (avec EXI-T qui est Σ_2^p-complet et co-EXI-T qui est Π_2^p-complet). Comme pour ARG-T, on prouve donc que, si $\Sigma_2^p \neq \Pi_2^p$, ARG-INCL $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$.
□

4.1.2.13 Étude de la complexité de UNI-CAR

Rappel : Le problème UNI-CAR est le suivant : "vérifier que H est une conséquence forte avec préférences de E en utilisant un ordre basé sur la cardinalité sur les thèses de E".
Notation : $E \mid\!\sim^{\forall,Car} H$.

Théorème 4.1.13 UNI-CAR *est* Δ_2^p*-complet.*

Preuve : 1ère partie : l'appartenance à une classe.

Remarque : d'après la définition donnée à la section 2.2.2 page 14, on a la propriété suivante :

Propriété 4.1.5 *Soit E un ensemble ordonné de formules, toutes les thèses préférées pour la cardinalité dans E ont le même cardinal.*

Cette remarque permet de proposer l'algorithme 4.17 page ci-contre pour UNI-CAR.

Cet algorithme est le résultat de mes recherches sur UNI-LEX dont l'étude, bien que présentée après, a été effectuée avant celle de UNI-CAR (voir le théorème 4.1.16 page 97) et de suggestions proposées par Daniel Lehmann lors d'une communication personnelle.

On y utilise deux oracles distincts : MAX-GSAT et NGSAT-CAR dont il faut déterminer la complexité.

- Le problème MAX-GSAT est défini par :
  instance : E un ensemble de formules et k un entier,
  question : existe-t-il une interprétation satisfaisant au moins k formules de E ?

  L'algorithme 4.18 page ci-contre permet de résoudre ce problème.
  Il s'agit d'un algorithme non déterministe, donc MAX-GSAT appartient à la classe NP. Montrons maintenant la complétude. C'est immédiat. En effet, il suffit de constater que GSAT se ramène polynomialement à MAX-GSAT.

  Soit "G" une instance de GSAT, posons :
  $E = \{G\}$, $k = 1$.

Algorithme 4.17 : UNI-CAR (E, H)
début

$k \leftarrow 0$

$n_f \leftarrow$ nombre de formules dans E

fini $\leftarrow$ *faux*

tant que ($n_f \geq 0$) et (non *fini*) **faire**

  $k \leftarrow n_f$

  si MAX-GSAT (E,k) **alors**

    fini $\leftarrow$ *vrai*

  sinon

    $n_f \leftarrow n\_1$

si NGSAT-CAR (E,H,k) **alors**

  échec $\leftarrow$ *vrai*

sinon

  échec $\leftarrow$ *faux*

vérifier que l'on n'a pas échec (*échec* = *faux*)

fin

Algorithme 4.18 : MAX-GSAT (E, k)
début

deviner une interprétation M

$p \leftarrow 0$

pour chaque formule G de E **faire**

  si M satisfait G **alors**

    $p \leftarrow p + 1$

vérifier que $p \geq k$

fin

On a alors : G est satisfiable $\Leftrightarrow$ il existe un modèle M de G $\Leftrightarrow$ il existe une interprétation M telle que M satisfait $k = 1$ formule de E $\Leftrightarrow$ MAX-GSAT ($\{G\}$, 1). On a donc MAX-GSAT NP-complet (remarque : MAX-GSAT reste NP-complet, même pour des clauses).

- Le problème NGSAT-CAR est défini par :

  instance : E un ensemble de formules, H une formule propositionnelle, k un entier avec k = nombre maximum de formules pouvant être satisfaites dans E,

  question : existe-t-il une interprétation M qui satisfait k formules de E telle que M ne satisfait pas H ?

  On propose l'algorithme 4.19 pour NGSAT-CAR.

Algorithme 4.19 : NGSAT-CAR (E, H, k)
début
deviner une interprétation M
(vérifier qu'elle satisfait k formules de E *)*
$k' \leftarrow 0$
pour chaque formule G de E **faire**
  si M satisfait G **alors**
    $k' \leftarrow k' + 1$
vérifier que $k' = k$

vérifier que M ne satisfait pas H
fin

Remarquons que, quelle que soit l'interprétation M, vérifier que M satisfait ou pas un ensemble de formules propositionnelles se fait en temps polynomial. L'algorithme ci-dessus pour NGSAT-CAR est non déterministe polynomial, donc NGSAT-CAR appartient à la classe de complexité NP. Quant à la preuve de complétude, elle se fait en ramenant polynomialement GSAT à NGSAT-CAR.

Soit "G" une instance de GSAT, posons :
$E = \{G\}$, $H = \neg G$ et $k = 1$.

On a alors : G satisfiable $\Leftrightarrow$ il existe une interprétation M qui satisfait G $\Leftrightarrow$ Dans E, le nombre maximal de formules pouvant être satisfaites est 1 $\Leftrightarrow$ il existe une interprétation M satisfaisant k formules qui ne satisfait pas $\neg G$. NGSAT-CAR est donc NP-complet (y compris pour des clauses).

Ce qui me permet de dire que l'algorithme proposé pour UNI-CAR est déterministe polynomial et qu'il fait appel à un oracle non déterministe polynomial[16]. Le problème UNI-CAR appartient donc à la classe de complexité $\mathsf{P}^{\mathrm{NP}} = \Delta_2^p$.

$2^{\text{ème}}$ **partie : la complétude ?**

Dans l'article [EG93] de Eiter et Gottlob, plusieurs problèmes Δ_2^p-complets sont cités. Parmi ceux-ci, j'ai choisi d'utiliser, pour prouver la complétude de UNI-CAR, le problème suivant :

> **Instance :** Soit $C = \{C_1, \ldots, C_m\}$ ensemble de clauses, soit $k \in \{1, \ldots, m\}$ un entier,
>
> **Question :** toutes les assignations card-maximales V_c des variables de C vérifient-elles : $V_c(C_k) = vrai$?

[16] En fait, cet algorithme fait appel à deux oracles $O1$ (MAX-GSAT) et $O2$ (NGSAT-CAR). Mais ces 2 oracles sont NP-complets. Il est donc possible de transformer de manière polynomiale un des oracles en l'autre et ainsi de ne plus faire appel qu'à un seul oracle. Cette remarque est valable pour plusieurs problèmes et ne sera donc pas répétée.

Je note ce problème ACM (pour Assignation de Cardinalité Maximum) et rappelle la définition d'une assignation card-maximale :

Définition 4.1.2 *Soit un ensemble de clauses $C = \{C_1, \ldots, C_m\}$ et $X = \{x_1, \ldots, x_n\}$ l'ensemble des variables de C, soient deux assignations V et W de X, on dit que $V >^{card} W$ ssi V satisfait strictement plus de clauses de C que W (en nombre de clauses). De même, on dit que $V <^{card} W$ ssi V satisfait strictement moins de clauses de C que W.*

Définition 4.1.3 *Soit C un ensemble de clauses, V est une assignation card-maximale de C ssi V est maximale pour l'ordre $>^{card}$ parmi toutes les assignations des variables de C.*

Remarque préliminaire : Soit $C = \{C_1, \ldots, C_m\}$ ensemble de clauses. Soit V une assignation des variables de C, posons $Y(V) = \{C_i$ tel que $V(C_i) =$ *vrai* $, i = 1 \ldots m\}$. On a alors les propriétés suivantes :

Propriété 4.1.6 *$Y(V)$ est un sous-ensemble de C consistant.*

> **Preuve :** D'après la définition de $Y(V)$, l'assignation V satisfait toutes les clauses de $Y(V)$, donc $Y(V)$ est consistant. □

Propriété 4.1.7 *Pour tout sous-ensemble Y de C maximal (pour l'inclusion) consistant, il existe une assignation V des variables de C telle que $Y = Y(V)$.*

> **Preuve :** Y est consistant donc il existe une assignation V telle que V satisfait toutes les clauses de Y.
> On peut donc définir $Y(V) = \{C_i$ tel que $V(C_i) =$ *vrai* $, i = 1 \ldots m\}$. On a alors $Y \subseteq Y(V)$. D'autre part, Y est maximal pour l'inclusion donc il n'existe pas de Y' sous-ensemble de C différent de Y tel que $Y \subseteq Y'$. Donc $Y = Y(V)$. □

Propriété 4.1.8 *Soient V et W deux assignations des variables de C. $V >^{card} W$ ssi $Y(V) \gg^{card} Y(W)$ (l'ordre $\gg^{card}$ est l'ordre basé sur la cardinalité défini par [BCD+ 93] dans le cas d'une base non stratifiée : $Y \gg^{card} Z$ ssi $|Y| > |Z|$; de même, $Y \ll^{card} Z$ ssi $|Y| < |Z|$).*

> **Preuve :** $V >^{card} W \Leftrightarrow V$ satisfait plus de clauses de C que $W \Leftrightarrow |Y(V)| > |Y(W)| \Leftrightarrow$ par définition $Y(V) \gg^{card} Y(W)$ □

Propriété 4.1.9 *Soit $C = \{C_1, \ldots, C_m\}$ un ensemble de clauses, V est une assignation card-maximale de C ssi $Y(V)$ est un sous-ensemble consistant de $\{C_1, \ldots, C_m\}$ maximal pour la cardinalité.*

> **Preuve :** V est une assignation card-maximale $\Leftrightarrow$ il n'existe pas d'assignation W de C telle que $V <^{card} W \Leftrightarrow$ il n'existe pas d'assignation W de C, $Y(V) \ll^{card} Y(W) \Leftrightarrow Y(V)$ est un sous-ensemble consistant de $\{C_1, \ldots, C_m\}$ maximal pour la cardinalité. □

Nous allons utiliser ce problème ACM pour prouver la complétude de UNI-CAR.

Soit "$C = \{C_1, \ldots, C_m\}$, $k \in \{1, \ldots, m\}$" une instance de ACM,   posons : $H = C_k$ et $E = \{C_1, \ldots, C_m\}$

Démontrons alors que ACM $\propto$ UNI-CAR :

- ACM implique UNI-CAR :

Toute assignation V_c card-maximale pour C vérifie : $V_c(C_k) = vrai$
$\Rightarrow$ (à cause de la propriété 4.1.7 page précédente)
$\forall Y$ thèse préférée pour la cardinalité de E, il existe une assignation V telle que $Y = Y(V)$
$\Rightarrow$ (à cause de la propriété 4.1.9 page précédente)
$\forall Y$ thèse préférée pour la cardinalité de E, l'assignation V telle que $Y = Y(V)$ est card-maximale et vérifie donc $V_c(C_k) = vrai$
$\Rightarrow$
$\forall Y$ thèse préférée pour la cardinalité de E, Y contient C_k
$\Leftrightarrow$
$\forall Y$ thèse préférée pour la cardinalité de E, Y infère C_k

- UNI-CAR implique ACM :

$\forall Y$ thèse préférée pour la cardinalité de E, Y infère C_k
$\Leftrightarrow$
$\forall Y$ thèse préférée pour la cardinalité de E, Y contient C_k
$\Rightarrow$ (à cause de la propriété 4.1.9 page précédente)
$\forall V_c$ assignation card-maximale de C, $Y(V_c)$ est un sous-ensemble maximal pour la cardinalité de C
$\Rightarrow$
$\forall V_c$ assignation card-maximale de C, $Y(V_c)$ est une des thèses préférées pour la cardinalité de E
$\Rightarrow$
$\forall V_c$ assignation card-maximale de C, $Y(V_c)$ contient C_k
$\Rightarrow$
Toute assignation V_c card-maximale pour C vérifie : $V_c(C_k) = vrai$

J'en conclus que ACM $\propto$ UNI-CAR, et ainsi que UNI-CAR est Δ_2^p-complet. □

4.1.2.14 Étude de la complexité de EXI-CAR

Rappel : Le problème EXI-CAR est le suivant : "vérifier que H est une conséquence faible avec préférences de E en utilisant un ordre basé sur la cardinalité sur les thèses de E".
Notation : $E \mid\!\sim^{\exists, Car} H$.

Théorème 4.1.14 EXI-CAR *est Σ_2^p-complet.*

Preuve : 1ère partie : l'appartenance à une classe.

On propose l'algorithme 4.20 pour EXI-CAR.

Algorithme 4.20 : EXI-CAR (E, H)
début
deviner un sous-ensemble Y de E
1 vérifier que Y est une thèse préférée pour la cardinalité de E
vérifier que Y infère H
fin

La propriété 4.1.5 page 90 déjà citée pour UNI-CAR permet un raffinage pour EXI-CAR dans lequel le point 1 de l'algorithme 4.20 est détaillé *"vérifier que Y est une thèse préférée pour la cardinalité de E"*. On obtient ainsi l'algorithme 4.21 page ci-contre.

Algorithme 4.21 : EXI-CAR (E, H) (nouvelle version)
début

deviner un sous-ensemble Y de E
(vérifier que Y est une thèse préférée pour la cardinalité de E *)*
$échec \leftarrow faux$
$k \leftarrow |Y|$
si il existe une sous-base consistante Z de E telle que $|Z| > k$ **alors**
 $échec \leftarrow vrai$

si non *échec* **alors**
 vérifier que Y infère H

fin

Donc, on obtient pour EXI-CAR un algorithme non déterministe polynomial faisant appel à deux oracles : un de complexité NP ("vérifier que Y infère H") et l'autre de complexité inconnue ("existe-t-il une sous-base consistante Z de E telle que $|Z| > k$").

Il me faut donc détailler cet oracle-là défini par :

instance : E une base et k un entier,
question : existe-t-il une interprétation satisfaisant strictement plus de k formules de E ?

Remarquons que ce problème est très proche du problème MAX-GSAT étudié pour UNI-CAR, sa seule différence venant du type d'inégalité utilisée qui est ici une inégalité stricte. Je vais donc noter ce problème MAX-GSAT-STRICT et étudier sa complexité.

On propose l'algorithme 4.22 pour MAX-GSAT-STRICT est le suivant :

Algorithme 4.22 : MAX-GSAT-STRICT (E, k)
début

deviner une interprétation M
$p \leftarrow 0$
pour chaque formule G de E **faire**
 si M satisfait G **alors**
  $p \leftarrow p + 1$

vérifier que $p > k$
fin

Le problème MAX-GSAT-STRICT appartient donc à la classe de complexité NP (à cause du "deviner"). Montrons maintenant la complétude. Il suffit de constater que GSAT se ramène polynomialement à MAX-GSAT-STRICT.

Soit "G" une instance de GSAT, posons :
$E = \{G\}$, $k = 0$.

On a alors : G est satisfiable $\Leftrightarrow$ il existe un modèle M de G $\Leftrightarrow$ il existe une sous-base consistante Z de E telle que $|Z| > 0$ $\Leftrightarrow$ MAX-GSAT-STRICT ($\{G\}$, 0). On a donc MAX-GSAT-STRICT NP-complet (y compris dans le cas de clauses). Remarquons que j'aurais aussi pu démontrer la complétude à partir du problème MAX-GSAT.

L'algorithme proposé pour EXI-CAR est donc non déterministe (à cause du "deviner") polynomial et il fait appel à un oracle lui aussi non déterministe polynomial. La complexité de EXI-CAR est donc au plus $\mathsf{NP}^{\mathsf{NP}} = \Sigma_2^p$.

$2^{\text{ème}}$ partie : la complétude ?

Essayons de passer de 2-QBF à EXI-CAR grâce à une transformation polynomiale.

Soit "$\exists a \forall b G(a,b)$", une instance de 2-QBF, posons : $H = G(a,b)$; $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n\}$

Ainsi :

- on a dans E toutes les "valeurs" possibles pour les variables propositionnelles $a_1, \ldots, a_n$ dont on cherche s'il existe une assignation,
- on n'impose aucune contrainte sur les variables propositionnelles $b_1, \ldots, b_m$, dont toutes les assignations doivent être prises en compte.

Remarque préliminaire : Y est une thèse préférée pour la cardinalité de $E \Leftrightarrow Y$ est de la forme $\{l_1, \ldots, l_n\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i = 1 \ldots n$.

Montrons alors que : "$\{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n\}$ infère $G(a,b)$ avec la méthode EXI-CAR" équivaut à "$\exists a \forall b G(a,b)$ est satisfiable" :

$\{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n\}$ infère $G(a,b)$ avec la méthode EXI-CAR

$\Leftrightarrow$

il existe un sous-ensemble Y de E tel que Y soit une thèse préférée pour la cardinalité et infère $G(a,b)$

$\Leftrightarrow$

il existe un sous-ensemble $Y = \{l_1, \ldots, l_n\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i = 1 \ldots n$, tel que $Y \vdash G(a,b)$

$\Leftrightarrow$

il existe $\{l_1, \ldots, l_n\}$ qui infère $G(a,b)$

$\Leftrightarrow$

$\exists a \forall b G(a,b)$ est satisfiable

Le problème 2-QBF se ramène donc polynomialement à EXI-CAR. Et on en déduit que le problème EXI-CAR est Σ_2^p-complet. Remarque : c'est la même démonstration que pour EXI-T. □

4.1.2.15 Étude de la complexité de ARG-CAR

Rappel : Le problème ARG-CAR est le suivant : "vérifier que H est une conséquence argumentative avec préférences de E en utilisant un ordre basé sur la cardinalité sur les thèses de E".
Notation : $E \mid\!\sim^{A,Car} H$.

Théorème 4.1.15 ARG-CAR $\in \Delta_3^p$ *et est* Σ_2^p*-difficile.*

Preuve : $1^{\text{ère}}$ partie : l'appartenance à une classe.

On propose l'algorithme 4.23 page ci-contre pour ARG-CAR.

Cet algorithme est polynomial et il fait appel à un oracle de complexité Σ_2^p (EXI-CAR).

La complexité de ARG-CAR est donc au plus $\mathsf{P}^{\Sigma_2^p} = \Delta_3^p$.

$2^{\text{ème}}$ partie : la complétude ?

Essayons de comparer le problème ARG-CAR avec des problèmes de classe inférieure dans la hiérarchie polynomiale. On constate alors que :

Algorithme 4.23 : ARG-CAR (E, H)
début
vérifier que $E \not\mid\!\sim^{\exists,Car} \neg H$
vérifier que $E \mid\!\sim^{\exists,Car} H$
fin

- le problème EXI-CAR se ramène polynomialement à ARG-CAR :

  Soit "E, H" une instance de EXI-CAR, posons :
  - $f(H) = G$,
  - $f(E) = E \cup \{H \rightarrow G\}$

  avec G nouvelle variable propositionnelle (G n'apparaît pas dans E).

  Remarques préliminaires :
  - $H \rightarrow G$ est consistante avec n'importe quelle thèse préférée pour la cardinalité Y' de E (voir argumentation donnée pour la démonstration de ARG-T, en sachant que toute thèse préférée pour la cardinalité est aussi tout simplement une thèse !).
  - Quelle que soit Y thèse préférée pour la cardinalité de $f(E)$, Y sera de la forme $Y' \cup \{H \rightarrow G\}$ avec Y' une thèse préférée pour la cardinalité de E. Ceci vient du fait que la formule $H \rightarrow G$ est consistante avec toutes les thèses préférées pour la cardinalité de E.
  - Quelle que soit Y thèse préférée pour la cardinalité de $f(E)$, $Y \vdash G$ ssi $Y' \vdash H$ (voir argumentation dans la démonstration de ARG-T).

  Montrons alors que : "$f(E)$ infère $f(H)$ avec la méthode ARG-CAR" équivaut à "E infère H avec la méthode EXI-CAR" :

  $f(E)$ infère G avec la méthode ARG-CAR

  $\Leftrightarrow$

  il existe Y thèse préférée pour la cardinalité de $f(E)$ telle que $Y \vdash G$ et aucune des autres thèses préférées pour la cardinalité n'infère $\neg G$.

  $\Leftrightarrow$

  il existe Y thèse préférée pour la cardinalité de $f(E)$ telle que $Y \vdash G$ (voir remarques préliminaires)

  $\Leftrightarrow$

  il existe Y' thèse préférée pour la cardinalité de E telle que $Y' \vdash H$ (voir remarques préliminaires)

  $\Leftrightarrow$

  E infère H avec la méthode EXI-CAR

- par contre, je n'ai pas réussi à montrer qu'un problème Π_2^p-complet (par exemple : co-EXI-CAR) se ramène polynomialement à ARG-CAR.

Je ne peux donc pas refaire le raisonnement effectué pour les autres relations d'inférence utilisant la conséquence argumentative (voir par exemple théorème 4.1.3 page 78). Seul le résultat d'appartenance demeure valable. □

4.1.2.16 Étude de la complexité de UNI-LEX

Rappel : Le problème UNI-LEX est le suivant : "vérifier que H est une conséquence forte avec préférences de $(E, <)$ en utilisant un ordre lexicographique sur les thèses de $(E, <)$".
Notation : $(E, <) \mid\!\sim^{\forall,Lex} H$.

Théorème 4.1.16 UNI-LEX *est* Δ_2^p*-complet.*

On peut donner 2 preuves pour ce théorème, pour la partie "appartenance à une classe" comme pour la partie "complétude".

Preuve numéro 1 : 1$^{\text{ère}}$ partie : l'appartenance à une classe.

Remarque : Cette relation d'inférence est une généralisation de la relation Uni-Car à une base stratifiée. Et, comme pour Uni-Car, on a la propriété :

Propriété 4.1.10 *Soit $(E,<)$ un ensemble ordonné de formules, toutes les thèses préférées lexicographiquement dans $(E,<)$ ont le même cardinal strate par strate.*

La réflexion que j'ai menée me conduit alors aux faits suivants.

Soient $(E,<)$ une base stratifiée et H une formule, posons :
$f((E,<)) = (E \cup \{H \to G, \neg G\}, <)$ et $f(H) = \neg G$ avec :
- G nouvelle variable propositionnelle n'apparaissant pas dans E,
- $H \to G$ constituant seule la première strate,
- $\neg G$ constituant seule la dernière strate.

Puis, calculons les thèses Y préférées lexicographiquement de $f((E,<))$ en utilisant la propriété 4.1.10 :

<table>
<tr><th>Y' thèse préférée lexicographiquement de $(E,<)$ $(|Y'| = n)$</th><th>Y'' thèse préférée lexicographiquement de $(E \cup \{H \to G\}, <)$ avec 1$^{\text{ère}}$ strate = $\{H \to G\}$</th><th>Y thèse préférée lexicographiquement de $f((E,<)) =$ $(E \cup \{H \to G, \neg G\}, <)$ avec 1$^{\text{ère}}$ strate = $\{H \to G\}$ et dernière strate = $\{\neg G\}$</th></tr>
<tr><td>Y' infère H</td><td>$Y'' = Y' \cup \{H \to G\}$
$(|Y''| = n+1)$
avec $Y'' \vdash H$, $Y'' \vdash G$
donc inconsistante avec $\neg G$</td><td>Y (noté $Y_{1^{\text{er}}\ \text{cas}}$) $= Y'' =$
$Y' \cup \{H \to G\}$
$(|Y| = n+1)$
avec $Y \vdash H$, $Y \vdash G$ et $Y \nvdash \neg G$
et il existe une telle Y seulement si le cas donné dans la case ci-dessous ne se produit jamais
(puisque $|Y_{1^{\text{er}}\ \text{cas}}| < |Y_{2^{\text{ème}}\ \text{cas}}|$
et que l'on a la propriété 4.1.10)</td></tr>
<tr><td>Y' n'infère pas H</td><td>$Y'' = Y' \cup \{H \to G\}$
$(|Y''| = n+1)$
avec $Y'' \nvdash H$, $Y'' \nvdash G$
donc consistante avec $\neg G$</td><td>Sachant que $\neg G \notin Y'$ donc $\neg G \notin Y''$,
Y (noté $Y_{2^{\text{ème}}\ \text{cas}}$) $= Y'' \cup \{\neg G\}$
$= Y' \cup \{H \to G, \neg G\}$
$(|Y| = n+2)$
avec $Y \nvdash H$, $Y \nvdash G$ et $Y \vdash \neg G$
et c'est la seule forme possible pour toute thèse préférée lexicographiquement de $f((E,<))$ (pour la même raison que celle fournie dans le 1$^{\text{er}}$ cas). Ainsi on masque les Y' qui inféraient H.</td></tr>
</table>

On a alors les propriétés suivantes :

Propriété 4.1.11 *Il existe une thèse préférée lexicographiquement de $f((E,<))$ qui infère G ssi toute thèse préférée lexicographiquement de $(E,<)$ infère H.*

Propriété 4.1.12 *Il existe une thèse préférée lexicographiquement de $(E,<)$ qui n'infère pas H ssi toute thèse préférée lexicographiquement de $f((E,<))$ infère $\neg G$.*

De ces deux propriétés, on déduit facilement que :

- UNI-LEX $\propto$ EXI-LEX,
- co-UNI-LEX $\propto$ UNI-LEX.

Ce qui donne pour UNI-LEX, l'algorithme 4.24[17] :

Algorithme 4.24 : UNI-LEX $((E,<), H)$
début
  $E' \leftarrow \{H \rightarrow G\} \cup E \cup \{\neg G\}$
  $k \leftarrow <0,0,\ldots,0>$ *(* vecteur de dimension n' = nombre de strates dans E' *)*
  pour $n_s \leftarrow 1$ à n' **faire**
    $n_f \leftarrow$ nombre de formules dans la strate E'_{n_s}
    fini $\leftarrow$ *faux*
    tant que $(n_f \geq 0)$ et (non *fini*) **faire**
      $k[n_s] \leftarrow n_f$
      si MAX-GSAT-ARRAY (E',k) **alors**
        fini $\leftarrow$ *vrai*
      sinon
        $n_f \leftarrow n_f - 1$
  vérifier que $k[n']$ est différent de 1
fin

Remarquons que, en fait, ce n'est pas la négation de la formule H elle-même que l'on rajoute en dernière strate, mais la négation de la formule G qui est une nouvelle variable propositionnelle n'apparaissant pas dans E et dans le même temps, on rajoute en tant que première strate la formule $H \rightarrow G$; ainsi, on ne risque pas de rajouter une formule qui existerait déjà dans la base, ce qui transformerait la base en un multi-ensemble.

Je retrouve ici le problème MAX-GSAT présenté pour UNI-CAR mais généralisé pour les bases stratifiées et noté alors MAX-GSAT-ARRAY :

> **instance :** $(E,<)$ une base et k un vecteur d'entiers de dimension n (n = nb de strates dans E),
> **question :** existe-t-il une interprétation satisfaisant pour chaque strate E_i de E au moins $k[i]$ formules) ?

Ce problème peut être résolu avec l'algorithme 4.25 page suivante.

Il s'agit d'un algorithme non déterministe, donc MAX-GSAT-ARRAY appartient à la classe NP. La preuve de complétude est immédiate. En effet, MAX-GSAT qui est NP-complet (voir démonstration pour UNI-CAR) est une restriction de MAX-GSAT-ARRAY.

Soit "E,k" une instance de MAX-GSAT, posons :
- $f(E) = E$,
- $f(k) = <k>$

avec $f(E)$ une base stratifiée constituée d'une seule strate.

[17] Cet algorithme repose sur le fait que la base est stratifiée et que la formule $\neg G$ est placée seule dans la dernière strate. Il aurait pu être utilisé pour UNI-CAR en plaçant toutes les formules de E dans la même strate (la strate intermédiaire).

Algorithme 4.25 : MAX-GSAT-ARRAY $((E,<),\ k)$
début
  deviner une interprétation M
  $i \leftarrow 1$ *(* strate la plus prioritaire *)*
  $\acute{e}chec \leftarrow faux$
  tant que $i \leq n$ (nombre de strates de E) et (non $\acute{e}chec$) **faire**
    $p \leftarrow 0$
    pour chaque formule G de $(E,<)$ **faire**
      si M satisfait G **alors**
        $p \leftarrow p+1$
    si $p \geq k[i]$ **alors**
      $i \leftarrow i+1$
    sinon
      $\acute{e}chec \leftarrow vrai$
  vérifier que l'on n'a pas échec ($\acute{e}chec = faux$)
fin

On a alors : MAX-GSAT $(E,k) \Leftrightarrow$ MAX-GSAT-ARRAY $(f(E),f(k))$.

On a donc MAX-GSAT-ARRAY NP-complet. Ce qui me permet de dire que l'algorithme proposé pour UNI-LEX est déterministe polynomial et qu'il fait appel à un oracle non déterministe polynomial. Le problème UNI-LEX appartient donc à la classe de complexité $\mathsf{P}^{\mathrm{NP}} = \Delta_2^p$.

$2^{\text{ème}}$ **partie : la complétude ?**

On sait que UNI-CAR est Δ_2^p-complet. Or UNI-CAR est une restriction de UNI-LEX, donc on peut prouver que UNI-CAR $\propto$ UNI-LEX en utilisant la transformation polynomiale suivante :

> Soit "E, H" une instance de UNI-CAR, posons :
> - $f(H) = H$,
> - $f(E) = E$,
>
> $f(E)$ étant alors une base stratifiée ne contenant qu'une seule strate.

Donc UNI-LEX est Δ_2^p-complet. □

Preuve numéro 2 : $1^{\text{ère}}$ **partie : l'appartenance à une classe.**

Un autre algorithme possible pour UNI-LEX s'inspire de celui proposé pour UNI-CAR. On y utilise le problème NGSAT-LEX :

> **instance :** $(E,<)$ une base stratifiée, H une formule propositionnelle, k un vecteur de dimension n (n = nb de strates dans E) avec $k[i]$ = nombre maximum de formules pouvant être satisfaites dans la strate i (même vecteur k que celui décrit dans l'algorithme précédent),
>
> **question :** existe-t-il une interprétation M qui satisfait $k[i]$ formules de chaque strate i de E telle que M ne satisfait pas H ?

Ce problème NGSAT-LEX est une généralisation du problème NGSAT-CAR défini lors de la démonstration pour UNI-CAR de la même manière que MAX-GSAT-ARRAY est une généralisation de MAX-GSAT. NGSAT-LEX est NP-complet. Cela se démontre en utilisant l'algorithme 4.26 page ci-contre.

Algorithme 4.26 : NGSAT-LEX $((E,<), H, k)$
début

deviner une interprétation M
(vérifier qu'elle satisfait $k[i]$ formules de chaque strate i de E *)*
$k' \leftarrow < 0, 0, \ldots, 0 >$ *(* k' de même dimension n que k *)*
pour chaque strate i de 1 à n **faire**
    pour chaque formule G de la strate i **faire**
        si M satisfait G **alors**
            $k'[i] \leftarrow k'[i] + 1$

vérifier que $k' = k$

vérifier que M ne satisfait pas H

fin

L'algorithme pour NGSAT-LEX est non déterministe polynomial, donc NGSAT-LEX appartient à la classe de complexité NP. Quant à la preuve de complétude, elle se fait en ramenant polynomialement NGSAT-CAR à NGSAT-LEX. Soit "E,H,k" une instance de NGSAT-CAR, posons : $f(E) = E$, $f(H) = H$ et $f(k)$ vecteur de dimension 1 égal à $< k >$. On a alors : NSAT-CAR $\propto$ NGSAT-LEX. On peut ainsi utiliser l'algorithme 4.27 pour UNI-LEX.

Algorithme 4.27 : UNI-LEX $((E,<), H)$ (nouvelle version)
début

(calcul du vecteur k *)*
$k \leftarrow < 0, 0, \ldots, 0 >$ *(* vecteur de dimension n (nombre de strates dans E) *)*
pour $n_s \leftarrow 1$ à n **faire**
    $n_f \leftarrow$ nombre de formules dans la strate E_{n_s}
    fini $\leftarrow$ *faux*
    tant que $(n_f \geq 0)$ et (non *fini*) **faire**
        $k[n_s] \leftarrow n_f$
        si MAX-GSAT-ARRAY (E,k) **alors**
            fini $\leftarrow$ *vrai*
        sinon
            $n_f \leftarrow n_f - 1$

si NGSAT-LEX (E,H,k) **alors**
    échec $\leftarrow$ *vrai*
sinon
    échec $\leftarrow$ *faux*
vérifier que l'on n'a pas échec (*échec* = *faux*)

fin

Cet algorithme montre lui-aussi que UNI-LEX appartient à la classe Δ_2^p. Remarquons que l'on utilise ici aussi l'oracle MAX-GSAT-ARRAY qui a été présenté avec le premier algorithme pour UNI-LEX. Signalons toutefois que le premier algorithme proposé pour UNI-LEX est plus efficace puisqu'il ne fait appel qu'à un seul oracle NP alors que le second algorithme en utilise 2.

2ème partie : la complétude ?

On peut démontrer directement la complétude pour UNI-LEX à partir du même type de problème que celui utilisé pour UNI-CAR. Je vais présenter cette preuve ici car elle sera réutilisée lors de l'étude de la complétude du problème 1/STRATE (cf. section 4.1.26 page 113). Le problème choisi est (cf. [EG93]) :

> **Instance :** Soit $C = \{C_1, \ldots, C_m\}$ ensemble de clauses satisfiable, soit $X = \{x_1, \ldots, x_n\}$ ensemble des variables propositionnelles apparaissant dans C, soit $O(X) =< x_1, \ldots, x_n >$ une prioritisation de X (une "prioritisation" de X est une suite strictement ordonnée $< x_1, \ldots, x_n >$ des variables de X),
>
> **Question :** L'assignation lex-maximale V_M pour la prioritisation $O(X)$ satisfaisant C vérifie-t-elle : $V_M(x_n) = vrai$?

Je note ce problème ALM (pour Assignation Lexicographiquement Maximale) et rappelle la définition d'une assignation lex-maximale :

Définition 4.1.4 *Soient un ensemble de variables $X = \{x_1, \ldots, x_n\}$ et $O(X) = < x_1, \ldots, x_n >$ une prioritisation de X, avec x_1 la plus prioritaire et x_n la moins prioritaire, soient deux assignations V et W sur cet ensemble de variables, on dit que $V >^{lex} W$ pour $O(X)$ ssi $\exists i \in [1 \ldots n]$ tel que $\forall j < i$, $V(x_j) = W(x_j)$, $V(x_i) = vrai$, $W(x_i) = faux$ (de même, $V <^{lex} W$ pour $O(X)$ ssi $\exists i \in [1 \ldots n]$ tel que $\forall j < i$, $V(x_j) = W(x_j)$, $V(x_i) = faux$, $W(x_i) = vrai$).*

Définition 4.1.5 *Pour une prioritisation de X donnée, V_M est une assignation lex-maximale ssi V_M est maximale pour l'ordre $>^{lex}$.*

Remarques préliminaires : Soit $C = \{C_1, \ldots, C_m\}$ ensemble de clauses satisfiable, soient $X = \{x_1, \ldots, x_n\}$ ensemble des variables propositionnelles apparaissant dans C et $O(X) =< x_1, \ldots, x_n >$ une prioritisation de X.

1. L'ordre $>^{lex}$ est strict et total, donc il n'existe qu'une seule assignation lex-maximale satisfaisant C pour une prioritisation de X donnée.
2. Soit V une assignation de X satisfaisant C, posons $Y(V) = \{x_i$ tel que $V(x_i) = vrai, i = 1 \ldots n\}$. On a alors les propriétés suivantes :

   Propriété 4.1.13 *$Y(V)$ est une sous-base de X consistante avec C.*

   Propriété 4.1.14 *Soient V et W assignations de X, soit $O(X)$ une prioritisation de X. Si $V >^{lex} W$ pour $O(X)$ alors $Y(V) \gg^{lex} Y(W)$ (l'ordre $\gg^{lex}$ est l'ordre lexicographique tel que défini par [BCD+93], voir section 2.2.2 page 14 : $Y \gg^{lex} Z$ ssi $\exists i$ tel que $|Y_i| > |Z_i|$ et $\forall j, j < i$, $|Y_j| = |Z_j|$; de même, $Y \ll^{lex} Z$ ssi $\exists i$ tel que $|Y_i| < |Z_i|$ et $\forall j, j < i$, $|Y_j| = |Z_j|$).*

   Preuve : Avec $O(X) =< x_1, \ldots, x_n >$, $V >^{lex} W$ pour $O(X) \Leftrightarrow \exists i \in [1 \ldots n]$ tel que $\forall j < i$, $V(x_j) = W(x_j)$, $V(x_i) = vrai$, $W(x_i) = faux \Rightarrow$ $\exists i \in [1 \ldots n]$ tel que $\forall j < i$ on a :
   - soit $(x_j \in Y(V)$ et $x_j \in Y(W))$, soit $(x_j \notin Y(V)$ et $x_j \notin Y(W))$
   - et $x_i \in Y(V)$ et $x_i \notin Y(W)$

   $\Rightarrow$ par définition $Y(V) \gg^{lex} Y(W)$ □

On utilise ce problème ALM pour prouver la complétude de UNI-LEX.

> Soit "$C = \{C_1, \ldots, C_m\}$ satisfiable, $X = \{x_1, \ldots, x_n\}$, $O(X) =< x_1, \ldots, x_n >$" une instance de ALM, posons :
> $H = x_n$ et
> $(E, <) = \{C_1 \wedge \ldots \wedge C_m, x_1, \ldots, x_n\}$,
> l'ordre $<$ étant le suivant : la formule $CC = C_1 \wedge \ldots \wedge C_m$ est plus prioritaire que la formule x_1 elle-même plus prioritaire que la formule x_2 elle-même plus prioritaire que ... plus prioritaire que x_n.

Remarques préliminaires :

- La première strate E_1 de E est consistante puisque C est satisfiable.
- Toute thèse préférée lexicographiquement contient la formule de la première strate.
- Comme il n'y a qu'une formule par strate dans E, il n'existe alors qu'une seule et unique thèse préférée lexicographiquement qui est aussi la seule préférée pour l'inclusion (voir propriété 4.1.22 page 110). Notons Y_0 cette thèse. $Y_0 = E_1 \cup (\{\alpha_1, \ldots, \alpha_n\} \setminus \{\top\}$ avec :
  - $\top$ = tautologie,
  - $\alpha_1 = x_1$ s'il existe un modèle de CC avec x_1 à *vrai*, sinon $\alpha_1 = \top$,
  - $\alpha_2 = x_2$ s'il existe un modèle de CC avec α_1, x_2 à *vrai*, sinon $\alpha_2 = \top$,
  - $\vdots$
  - $\alpha_n = x_n$ s'il existe un modèle de CC avec $\alpha_1, \ldots, \alpha_{n-1}$, x_n à *vrai*, sinon $\alpha_n = \top$.

  Définition 4.1.6 *Soit V_0 l'assignation définie de la façon suivante : $V_0(x_i) =$ vrai si $x_i \in Y_0$, faux sinon. On a alors $Y(V_0) \cup E_1 = Y_0$ et V_0 satisfait C.*

  On a alors la propriété suivante :

  Propriété 4.1.15 *V_0 ainsi construite est exactement l'assignation lex-maximale pour $O(X)$.*

  Preuve : Raisonnons par l'absurde. Si V_0 n'est pas l'assignation lex-maximale pour $O(X)$ satisfaisant C alors il existe une autre assignation notée V_M qui est l'assignation lex-maximale pour $O(X)$ satisfaisant C. Donc $V_M >^{lex} V_0$ pour $O(X)$ alors $Y(V_M) \gg^{lex} Y(V_0)$. Cela implique que $Y(V_0) \cup E_1$ n'est pas lex-préférée pour E. On a donc une contradiction avec la condition imposée à la construction de V_0 (cf. définition 4.1.6). □
- $V_M(x_n) =$ *vrai* ssi Y_0 contient x_n

Démontrons alors que ALM $\propto$ UNI-LEX :

$$\text{Soient } C, X \text{ et } O(X), \text{ on a } V_M(x_n) = vrai$$
$$\Leftrightarrow$$
$$\forall Y \text{ thèse préférée lexicographiquement de } E, Y \text{ contient } x_n$$
$$\Leftrightarrow$$
$$\forall Y \text{ thèse préférée lexicographiquement de } E, Y \text{ infère } x_n$$

J'en conclus que ALM $\propto$ UNI-LEX, et ainsi que UNI-LEX est Δ_2^p-complet. □

4.1.2.17 Étude de la complexité de EXI-LEX

Rappel : Le problème EXI-LEX est le suivant : "vérifier que H est une conséquence faible avec préférences de $(E, <)$ en utilisant un ordre lexicographique sur les thèses de $(E, <)$".
Notation : $(E, <) \mid\!\sim^{\exists, Lex} H$.

Théorème 4.1.17 EXI-LEX *est Σ_2^p-complet.*

Preuve : 1$^{\text{ère}}$ partie : l'appartenance à une classe.

Remarquons que EXI-LEX est une généralisation de EXI-CAR pour des bases stratifiées.

Algorithme 4.28 : EXI-LEX $((E,<),\ H)$
début
  deviner un sous-ensemble Y de $(E,<)$
  vérifier que Y est une thèse préférée lexicographiquement de $(E,<)$
  vérifier que Y infère H
fin

Algorithme 4.29 : EXI-LEX $((E,<),\ H)$ (nouvelle version)
début
  deviner un sous-ensemble Y de $(E,<)$
  (vérifier que Y est une thèse préférée lexicographiquement de $(E,<)$ *)*
  échec $\leftarrow$ *faux*
  $i \leftarrow 1$ *(* strate la plus prioritaire *)*
  tant que non *échec* et $i < n+1$ **faire**
    $k \leftarrow |Y_1| + |Y_2| + \ldots + |Y_i|$
    si il existe une sous-base consistante Z de $E_1 \cup E_2 \cup \ldots \cup E_i$ telle que $|Z| > k$ **alors**
      échec $\leftarrow$ *vrai*
    sinon
      $i \leftarrow i+1$
  si non *échec* **alors**
    vérifier que Y infère H
fin

On propose l'algorithme 4.28 pour EXI-LEX.

Comme pour EXI-CAR, on peut raffiner cet algorithme et ainsi obtenir l'algorithme 4.29.

Donc j'obtiens pour EXI-LEX un algorithme non déterministe polynomial faisant appel à deux oracles de complexité NP (GSAT et MAX-GSAT-STRICT déjà présenté pour EXI-CAR). Ce qui me permet de dire que l'algorithme proposé pour EXI-LEX est non déterministe (à cause du "deviner") polynomial et qu'il fait appel à un oracle lui aussi non déterministe polynomial. La complexité de EXI-LEX est donc au plus $\mathsf{NP}^{\mathrm{NP}} = \Sigma_2^p$.

2ème **partie : la complétude ?**

EXI-CAR étant une restriction de EXI-LEX, on peut définir la transformation polynomiale suivante :

> Soit "E, H" une instance de EXI-CAR, posons :
> - $f(H) = H$,
> - $f(E) = E$,
>
> $f(E)$ étant alors une base stratifiée ne contenant qu'une seule strate.

On a donc : EXI-CAR $\propto$ EXI-LEX. Or EXI-CAR est Σ_2^p-complet. Et on en déduit que le problème EXI-LEX est Σ_2^p-complet. Remarque : c'est le même type de démonstration que pour EXI-INCL. □

4.1.2.18 Étude de la complexité de ARG-LEX

Rappel : Le problème ARG-LEX est le suivant : "vérifier que H est une conséquence argumentative avec préférences de $(E,<)$ en utilisant un ordre lexicographique sur les thèses de $(E,<)$".

Notation : $(E,<) \mid\!\sim^{A,Lex} H$.

Théorème 4.1.18 ARG-LEX $\in \Delta_3^p$ *et est* Σ_2^p*-difficile.*

Preuve : $1^{\text{ère}}$ partie : l'appartenance à une classe.

Remarquons que ARG-LEX est une généralisation de ARG-CAR aux bases stratifiées.

On propose l'algorithme 4.30 pour ARG-LEX.

Algorithme 4.30 : ARG-LEX $((E, <), H)$
début
  vérifier que $(E, <) \not\mid\!\sim^{\exists, Lex} \neg H$
  vérifier que $(E, <) \mid\!\sim^{\exists, Lex} H$
fin

Cet algorithme est polynomial et il fait appel à un oracle de complexité Σ_2^p (EXI-LEX). La complexité de ARG-LEX est donc au plus $\mathsf{P}^{\Sigma_2^p} = \Delta_3^p$.

$2^{\text{ème}}$ **partie : la complétude ?**

ARG-CAR étant une restriction de ARG-LEX, on peut définir la transformation polynomiale suivante :

> Soit "E, H" une instance de ARG-CAR, posons :
> - $f(H) = H$,
> - $f(E) = E$,
>
> $f(E)$ étant alors une base stratifiée ne contenant qu'une seule strate.

On obtient ainsi que ARG-CAR $\propto$ ARG-LEX, or on sait que EXI-CAR $\propto$ ARG-CAR, donc EXI-CAR $\propto$ ARG-LEX. Par contre, comme pour ARG-CAR, je n'ai pas réussi à montrer qu'un problème Π_2^p-complet (par exemple : co-EXI-LEX ou co-EXI-CAR) se ramène polynomialement à ARG-LEX.

Je ne peux donc pas refaire le raisonnement effectué pour les autres relations d'inférence utilisant la conséquence argumentative (voir par exemple le théorème 4.1.3 page 78).

Seul le résultat d'appartenance demeure valable. □

4.1.2.19 Étude de la complexité de UNI-E

Rappel : Le problème UNI-E est le suivant : "vérifier que H est une conséquence forte de $E = (W, D)$ théorie des défauts propositionnelle en utilisant les extensions de E".

Notation : $E \mid\!\sim^{\forall, \mathrm{E}} H$.

Ce problème a été étudié par Gottlob dans le cas général (voir [Got92]). Ses conclusions sont les suivantes.

Théorème 4.1.19 UNI-E *est* Π_2^p*-complet.*

Remarques : Gottlob a montré que UNI-E reste Π_2^p-complet même si $E = (W, D)$ est une théorie des défauts propositionnelle normale (c'est-à-dire dont les défauts sont de la forme $a : b/b$). Et, dans le cas particulier correspondant au mécanisme E^*, on obtient un résultat similaire.

Théorème 4.1.20 UNI-E^* *est* Π_2^p*-complet.*

Preuve : Le théorème 4.1.19 fournit la preuve d'appartenance, et pour la complétude, la propriété 2.2.5 page 17 fournit une transformation polynomiale de UNI-T en UNI-E^*.
□

4.1.2.20 Étude de la complexité de Exi-E

Rappel : Le problème Exi-E est le suivant : "vérifier que H est une conséquence faible de $E = (W, D)$ théorie des défauts propositionnelle en utilisant les extensions de E".

Notation : $E \mid\!\sim^{\exists,\mathrm{E}} H$.

Ce problème a été étudié par Gottlob dans le cas général (voir [Got92]). Ses conclusions sont les suivantes.

Théorème 4.1.21 Exi-E *est* Σ_2^p*-complet.*

Remarques : Gottlob a montré que Exi-E reste Σ_2^p-complet même si $E = (W, D)$ est une théorie des défauts propositionnelle normale.

Et, dans le cas particulier correspondant au mécanisme E*, on obtient un résultat similaire.

Théorème 4.1.22 Exi-E* *est* Σ_2^p*-complet.*

> **Preuve :** Même principe que pour le théorème 4.1.20 page précédente. □

4.1.2.21 Étude de la complexité de Arg-E

Rappel : Le problème Arg-E est le suivant : "vérifier que H est une conséquence argumentative de $E = (W, D)$ théorie des défauts propositionnelle en utilisant les extensions de E".

Notation : $E \mid\!\sim^{\mathrm{A},\mathrm{E}} H$.

À ma connaissance, Gottlob ne s'est pas encore intéressé à ce type de relation d'inférence sur une théorie des défauts.

Théorème 4.1.23 *Si* $\Sigma_2^p \neq \Pi_2^p$*,* Arg-E $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$*.*

Preuve : 1$^{\text{ère}}$ partie : l'appartenance à une classe.

On propose l'algorithme 4.31 pour Arg-E.

Algorithme 4.31 : Arg-E $((E, <), H)$
début
  vérifier que $(E, <) \mid\!\not\sim^{\exists,E} \neg H$
  vérifier que $(E, <) \mid\!\sim^{\exists,E} H$
fin

Cet algorithme est polynomial et il fait appel à un oracle de complexité Σ_2^p (Exi-E).

La complexité de Arg-E est donc au plus $\mathsf{P}^{\Sigma_2^p} = \Delta_3^p$.

2$^{\text{ème}}$ partie : la complétude ?

J'ai toujours les mêmes difficultés à étudier la complétude, donc j'essaye de montrer que Arg-E $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$.

Pour cela on compare le problème Arg-E avec des problèmes de classe inférieure dans la hiérarchie polynomiale.

On constate alors que :

- le problème Exi-E se ramène polynomialement à Arg-E :

État de la branche de l'arbre de E	Résultat sur la construction de l'arbre pour $f(E)$
inconsistance	inconsistance
extension Y'	extension Y de $f(E) = C_n(Y' \cup \{H \rightarrow G\})$
échec car certaines hypothèses sur l'extension n'ont pas été vérifiées	peut-être une solution si la seule hypothèse manquante est satisfaite par le fait : "$H \rightarrow G \in$ l'extension" ; or ce cas est impossible puisque G n'apparaît pas dans E ; on a donc un échec

Tableau 4.1 : Démonstration de complexité pour ARG-E : EXI-E $\propto$ ARG-E

Soit "$E = (W, D), H$", une instance de EXI-E, posons :
- $f(H) = G$,
- $f(E) = (W \cup \{H \rightarrow G\}, D)$

avec G nouvelle variable propositionnelle (G n'apparaît pas dans W, ni dans D).

Remarques préliminaires :

- $H \rightarrow G$ est consistante avec n'importe quelle extension Y' de E : si Y' est une extension, elle est consistante et elle admet donc un modèle M'. Or G n'apparaît pas dans E, donc n'apparaît pas dans Y'. On peut ainsi étendre le modèle M' de Y' à un modèle M en rajoutant la valeur *vrai* pour la variable propositionnelle G. On a ainsi M modèle de Y' et modèle de $H \rightarrow G$. $H \rightarrow G$ est donc consistante avec toute extension Y' de E.
- $H \rightarrow G$ ne permet pas l'activation d'autres défauts que ceux activés pour le calcul des extensions de E, puisque G n'apparaît pas dans D (et en particulier, dans les pré-requis des défauts de D).
- Le fait de rajouter cette prémisse à E permet de reprendre en intégralité l'arbre de recherche des extensions de E et de construire l'arbre de recherche des extensions de $f(E)$ à partir de l'étude sur chaque branche de l'arbre de E de l'ajout de la prémisse $H \rightarrow G$ (voir la méthode de construction de l'arbre de recherche en section 2.2.3 page 16).

Cela donne les résultats présentés dans le tableau 4.1 et on obtient les propriétés suivantes :

Propriété 4.1.16 *Quelle que soit Y extension de $f(E)$, Y est de la forme $Y' \cup \{H \rightarrow G\}$ avec Y' extension de E.*

Propriété 4.1.17 *Quelle que soit Y extension de $f(E)$, $Y \vdash G$ ssi $Y' \vdash H$ (voir argumentation dans la démonstration du théorème 4.1.3 page 78).*

Propriété 4.1.18 *Aucune extension de E ou de $f(E)$ ne peut inférer $\neg G$.*

Montrons alors que : "$f(E)$ infère $f(H)$ avec la méthode ARG-E" équivaut à "E infère H avec la méthode EXI-E" :

$f(E)$ infère G avec la méthode ARG-E
$\Leftrightarrow$
il existe Y extension de $f(E)$ telle que $Y \vdash G$
et aucune des autres extensions n'infère $\neg G$.
$\Leftrightarrow$
(à cause de la propriété 4.1.18 page précédente)
il existe Y extension de $f(E)$ telle que $Y \vdash G$
$\Leftrightarrow$
(à cause des propriétés 4.1.16 page précédente et 4.1.17 page précédente)
il existe Y' extension de E telle que $Y' \vdash H$
$\Leftrightarrow$
E infère H avec la méthode EXI-E

- le problème co-EXI-E se ramène polynomialement à ARG-E :

Soit "$E = (W, D), H$", une instance de co-EXI-E, posons : - $f(E) = (W, D \cup \{: \neg H / \neg H\}$, - $f(H) = \neg H$.

Remarque préliminaire :
Le fait de rajouter un défaut à E permet de reprendre dans son intégralité l'arbre de recherche des extensions de E et de construire l'arbre de recherche des extensions de $f(E)$ à partir de l'arbre pour E et de l'étude sur chaque branche de l'arbre de E de l'activation ou non du défaut $:\neg H/\neg H$. Cela donne les résultats présentés dans le tableau 4.2 page ci-contre et on obtient les propriétés suivantes :

Propriété 4.1.19 *Si toutes les extensions Y' de E sont consistantes avec $\neg H$ (donc n'infèrent pas H) alors toutes les extensions de $f(E)$ infèrent $\neg H$ et n'infèrent pas H ;*

Propriété 4.1.20 *S'il existe une extension Y' de E qui infère H (donc inconsistante avec $\neg H$) alors il existe une extension Y de $f(E)$ qui infère H ($Y = Y'$) ;*

Propriété 4.1.21 *(contraposée de la propriété 4.1.20) Si aucune extension Y de $f(E)$ n'infère H alors aucune extension Y' de E n'infère H.*

Montrons alors que : "$f(E)$ infère $f(H)$ avec la méthode ARG-E" équivaut à "E n'infère pas H avec la méthode EXI-E".
Tout d'abord :

$f(E)$ infère $\neg H$ avec la méthode ARG-E
$\Leftrightarrow$
$\exists Y$ extension de $f(E)$ telle que $Y \vdash \neg H$
et $\forall Y$ extension de $f(E)$, $Y \not\vdash H$
$\Rightarrow$
(à cause de la propriété 4.1.21)
$\forall Y'$ extension de E, $Y' \not\vdash H$
$\Leftrightarrow$
E n'infère pas H avec la méthode EXI-E

Et réciproquement :

E n'infère pas H avec la méthode EXI-E
$\Leftrightarrow$
$\forall Y'$ extension de E, $Y' \not\vdash H$
$\Rightarrow$
(à cause de la propriété 4.1.19)
$\forall Y$ extension de $f(E)$, $Y \not\vdash H$ et $Y \vdash \neg H$

État de la branche de l'arbre de E	Statut du défaut :$\neg H / \neg H$ (appliqué ou non)	Résultat sur la construction de l'arbre pour $f(E)$
inconsistance	dans les deux cas	inconsistance
extension Y' consistante avec $\neg H$ donc n'inférant pas H	application du défaut : H ne doit pas $\in$ à l'extension finale	extension Y de $f(E) = C_n(Y' \cup \{\neg H\})$ si $\neg H \notin Y'$ ou $C_n(Y')$ sinon Y contient $\neg H$ et n'infère pas H
	non application du défaut : H doit $\in$ à l'extension finale	échec : H ne peut pas $\in$ à l'extension finale
extension Y' inconsistante avec $\neg H$ donc inférant H	application du défaut : H ne doit pas $\in$ à l'extension finale	échec : $H \in$ à l'extension finale
	non application du défaut : H doit $\in$ à l'extension finale	extension Y de $f(E) = Y'$ Y ne contient pas $\neg H$ et infère H
échec car certaines hypothèses sur l'extension n'ont pas été vérifiées	application du défaut : H ne doit pas $\in$ à l'extension finale	peut-être une solution si la seule hypothèse manquante est satisfaite par le fait : "$\neg H \in$ l'extension" ; dans ce cas là, on a : extension Y de $f(E) = C_n(S \cup \{\neg H\})$ avec $S = W \cup$ {les conséquents des défauts de E activés pour cette branche} Y contient $\neg H$ et n'infère pas H
	non application du défaut : H doit $\in$ à l'extension finale	échec : la non application du défaut ne permet pas de vérifier les hypothèses manquant déjà pour l'extension de E

Tableau 4.2 : Démonstration de complexité pour Arg-E : co-Exi-E $\propto$ Arg-E

$$\Rightarrow$$
$$\exists Y \text{ extension de } f(E) \text{ telle que } Y \vdash \neg H$$
$$\text{et } \forall Y \text{ extension de } f(E),\ Y \not\vdash H$$
$$\Leftrightarrow$$
$$f(E) \text{ infère } \neg H \text{ avec la méthode ARG-E}$$

On a donc EXI-E $\propto$ ARG-E et co-EXI-E $\propto$ ARG-E (avec EXI-E qui est Σ_2^p-complet et co-EXI-E qui est Π_2^p-complet). Ainsi, comme pour ARG-T, on prouve que, si $\Sigma_2^p \neq \Pi_2^p$, ARG-E $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$. □

Et, dans le cas particulier correspondant au mécanisme E*, on obtient un résultat similaire.

Théorème 4.1.24 *Si $\Sigma_2^p \neq \Pi_2^p$, ARG-E* $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$*

Preuve : Même principe que pour le théorème 4.1.20 page 105. □

4.1.3 Étude de complexité des différentes relations d'inférence dans des cas particuliers

L'étude de complexité des relations d'inférence présentées dans la section précédente est reprise ici en traitant quatre cas particuliers :

- celui où $(E, <)$ est un ensemble totalement et strictement ordonné ; ce qui revient à dire que l'on a une base initiale E stratifiée avec une seule formule par strate ;
- celui où $(E, <)$ est un ensemble totalement ordonné de formules sous forme conjonctive normale (formules CNF) ;
- celui où $(E, <)$ est un ensemble totalement ordonné de conjonctions de clauses de Horn ;
- celui où $(E, <)$ est un ensemble totalement et strictement ordonné de conjonctions de clauses de Horn.

Les relations d'inférence UNI (EXI, ARG)-T (S, CAR, E) ne sont concernées ni par le premier, ni par le dernier de ces quatre cas particuliers. En effet, ces relations d'inférence ne tiennent pas compte de l'ordre sur E. On peut aussi montrer que, dans ces cas là, les problèmes UNI (EXI, ARG)-INCL (LEX) deviennent équivalents (voir démonstration dans la section suivante).

L'étude du second cas particulier permet de traiter une sous-famille de la logique propositionnelle très souvent utilisée (les clauses) mais ne permet cependant aucune simplification de la complexité des problèmes étudiés.

Dans le troisième cas particulier, le problème GSAT devient polynomial et sera noté SAT-HORN, ce qui induit bien sûr des complexités plus faibles pour les problèmes posés.

Le quatrième cas permet de combiner les premier et troisième cas et d'obtenir ainsi de meilleurs résultats mais sur une classe de problèmes très restreinte.

4.1.3.1 Cas d'une base stratifiée avec une seule formule par strate

Dans toute cette section, $(E, <)$ dénotera une base stratifiée avec une seule formule par strate.

Dans le cas particulier d'une base strictement stratifiée, on a la propriété suivante :

Propriété 4.1.22 *Soit $<$ ordre total et strict sur E, il y a une seule thèse préférée pour l'ordre "Inclusion Based" qui est aussi la seule thèse préférée lexicographiquement.*

Preuve :

Prenons les notations suivantes : $E = E_1 \cup \ldots \cup E_n$ (E_i strate composée d'une seule formule), on note $F_i = E_i \cap F$, $\forall i$ et $\forall F$ sous-ensemble de E.

Rappelons les deux définitions suivantes :

> **Définition 4.1.7** *Soient A et B deux sous-ensembles consistants de E, $A = A_1 \cup \ldots \cup A_n$ et $B = B_1 \cup \ldots \cup B_n$, A est préféré à B pour l'ordre "Inclusion Based" (noté $B \ll^{incl} A$) ssi il existe i tel que $B_i \subset A_i$[18] et $\forall j < i$, $A_j = B_j$[19].*

> **Définition 4.1.8** *Soient A et B deux sous-ensembles consistants de E, $A = A_1 \cup \ldots \cup A_n$ et $B = B_1 \cup \ldots \cup B_n$, A est préféré à B pour l'ordre lexicographique (noté $B \ll^{lex} A$) ssi il existe i tel que $|B_i| < |A_i|$ et $\forall j < i$, $|A_j| = |B_j|$.*

Remarquons en premier lieu que, dans le cas d'une base dont les strates sont composées d'une seule formule, on a : $\forall i$, $A_i = \varnothing$ ou $A_i = \{f_i\}$ avec f_i la formule de la strate numéro i. De même pour B.

Montrons alors que les deux ordres donnent le même résultat : $\forall B$ et $\forall A$, $B \ll^{incl} A \Leftrightarrow B \ll^{lex} A$.

$$B \ll^{incl} A$$
$$\Leftrightarrow$$
$$\exists i \text{ tel que } B_i \subset A_i \text{ et } \forall j < i,\ A_j = B_j$$
$$\Leftrightarrow$$
$$\exists i \text{ tel que } B_i = \varnothing \text{ et } A_i = \{f_i\} \text{ et } \forall j < i, \text{ soit } A_j = B_j = \varnothing, \text{ soit } A_j = B_j = \{f_j\}$$
$$\Leftrightarrow$$
$$\exists i \text{ tel que } |B_i| = 0 \text{ et } |A_i| = 1 \text{ et } \forall j < i, \text{ soit } |A_j| = |B_j| = 0, \text{ soit } |A_j| = |B_j| = 1$$
$$\Leftrightarrow$$
$$\exists i \text{ tel que } |B_i| < |A_i| \text{ et } \forall j < i,\ |A_j| = |B_j|$$
$$\Leftrightarrow$$
$$B \ll^{lex} A.$$

Montrons ensuite l'unicité des thèses préférées :

> **Définition 4.1.9** *Une thèse préférée pour l'ordre "Inclusion Based"[20] de E est un ensemble $S = S_1 \cup \ldots \cup S_n$ tel que $\forall k \in [1 \ldots n]$ $S_1 \cup \ldots \cup S_k$ est un sous-ensemble maximal consistant de $E_1 \cup \ldots \cup E_k$.*

Cette définition induit le processus constructif suivant : pour chaque strate k en partant de la plus prioritaire, soit la formule f_k (de la strate k) est consistante avec l'ensemble solution déjà construit, alors le nouvel ensemble solution est égal à l'union de $\{f_k\}$ avec l'ancien ensemble solution, soit f_k est inconsistante avec l'ensemble solution déjà construit, alors l'ensemble solution reste inchangé. Dans tous les cas, à la fin de chaque étape du processus, on a qu'un seul ensemble solution. □

[18] Le symbole $\subset$ dénote l'inclusion stricte.
[19] Rappelons que cette définition correspond à l'ordre démocratique.
[20] Cette fois, nous utilisons la définition de Brewka pour son aspect constructif.

Théorème 4.1.25 (Complexité de UNI (EXI, ARG)**-**BO**)** *Dans le cas d'une base stratifiée strictement,* UNI-BO *(respectivement* EXI-BO, ARG-BO*) est noté 1/*STRATE-UNI-BO *(respectivement 1/*STRATE-EXI-BO*, 1/*STRATE-ARG-BO*). Les complexités sont :*

- *si* $\mathsf{NP} \neq co\text{-}\mathsf{NP}$*, 1/*STRATE-UNI-BO $\in \Delta_2^p$ - $(\mathsf{NP} \cup co\text{-}\mathsf{NP})$
- *1/*STRATE-EXI-BO *est* Σ_2^p*-complet*
- *si* $\Sigma_2^p \neq \Pi_2^p$*, 1/*STRATE-ARG-BO $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$

Preuve : Étude d'appartenance.

Contrairement à ce qui se passe pour les méthodes "Inclusion Based" et lexicographique (voir propriété 4.1.22 page 110), je n'ai pas ici de simplification des algorithmes. En effet, le fait qu'il n'y ait qu'une formule par strate intervient dans le calcul des $a(Y)$ mais ne permet pas de se ramener à une seule et unique sous-base préférée pour l'ordre "Best-Out" (voir l'exemple donné à la section 2.2.4 page 19). On garde donc les mêmes résultats d'appartenance que ceux vus précédemment :

- 1/STRATE-UNI-BO appartient au plus à la classe de complexité Δ_2^p,
- 1/STRATE-EXI-BO appartient au plus à la classe de complexité Σ_2^p,
- 1/STRATE-ARG-BO appartient au plus à la classe de complexité Δ_3^p.

Étude de complétude.

- Pour 1/STRATE-UNI-BO, on peut conserver exactement la même démonstration que celle du cas général, puisque les transformations utilisées définissent des bases avec une seule formule par strate.
  On prouve donc que : 1/STRATE-UNI-BO $\in \Delta_2^p$ - $(\mathsf{NP} \cup \text{co-}\mathsf{NP})$.
- Par contre, pour 1/STRATE-EXI-BO, la démonstration du cas général ne s'applique plus puisque la base $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n\}$ n'est pas strictement stratifiée. Je propose alors une nouvelle transformation polynomiale :

  Soit "E, H", une instance de EXI-S, posons :
  - $f(H) = H$,
  - $(f(E), <) = \{\bot\} \cup E_{strat}$
  avec $\bot$ symbolisant la contradiction et constituant seul la première strate et E_{strat} l'ensemble constitué des formules de E ordonnées strictement (ordre arbitraire).

  D'après la définition de l'ordre "Best-Out", le *amax* correspondant à $f(E)$ est 1. Ainsi, les sous-bases préférées pour l'ordre "Best-Out" de $f(E)$ sont exactement les sous-bases consistantes de E_{strat}, elles-même étant les sous-bases consistantes de E. On a donc :

  $\exists Y$ sous base consistante de E telle que $Y \vdash H$

  $\Leftrightarrow$

  $\exists Y$ sous base consistante préférée pour l'ordre "Best-Out" de $f(E)$ telle que $Y \vdash H$

  On a ainsi EXI-S $\propto$ 1/STRATE-EXI-BO. Le problème 1/STRATE-EXI-BO est donc Σ_2^p-complet.
- Pour 1/STRATE-ARG-BO, je peux garder la même démonstration que celle du cas général en partant des problèmes 1/STRATE-EXI-BO et co-1/STRATE-EXI-BO, puisque les deux transformations polynomiales utilisées génèrent des sous-bases stratifiées strictement. J'en conclus que 1/STRATE-ARG-BO $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$ si $\Sigma_2^p \neq \Pi_2^p$.

□

Propriété 4.1.23 (Uni(Exi, Arg)-Incl **et** Uni(Exi, Arg)-Lex) *Comme il existe une unique thèse préférée à la fois pour l'ordre "Inclusion Based" et pour l'ordre lexicographique, on constate alors que les problèmes* Uni-Incl*,* Exi-Incl *et* Arg-Incl *deviennent équivalents ainsi que les problèmes* Uni-Lex*,* Exi-Lex *et* Arg-Lex*. J'obtiens ainsi un unique problème noté 1/*Strate*.*

Théorème 4.1.26 (Complexité de Uni(Exi, Arg)-Incl et Uni(Exi, Arg)-Lex)
*1/*Strate *est* Δ_2^p*-complet.*

Preuve : Étude d'appartenance.

On propose l'algorithme 4.32 pour 1/Strate (ce problème est noté $(E,<) \mid\!\sim^{1-s} H$).

Algorithme 4.32 : 1/Strate $((E,<),\ H)$
début
$X \leftarrow \varnothing$
$n_s \leftarrow 1$
1 **si** $X \cup E_{n_s}$ est consistant **alors**
  $X \leftarrow X \cup E_{n_s}$
$n_s \leftarrow n_s + 1$
si $n_s =$ nb total de strates dans E **alors**
  vérifier que $X \vdash H$
sinon
  aller étape 1
fin

Cet algorithme est déterministe polynomial et il fait appel à un oracle de complexité NP. Le problème 1/Strate appartient donc à la classe de complexité Δ_2^p.

Étude de complétude.

Pour le théorème 4.1.16 page 97, j'ai présenté une démonstration de complétude pour Uni-Lex utilisant le problème Alm donné par Eiter et Gottlob dans [EG93]. Or cette preuve s'applique en intégralité pour le problème 1/Strate puisque la transformation polynomiale utilisée génère une base E strictement ordonnée :

Soit "$C = \{C_1, \ldots, C_m\}$ satisfiable, $X = \{x_1, \ldots, x_n\}$,
$O(X) =< x_1, \ldots, x_n >$" une instance de Alm, posons :
$H = x_n$ et
$(E,<) = \{C_1 \wedge \ldots \wedge C_m, x_1, \ldots, x_n\}$,
l'ordre $<$ étant le suivant : la formule $CC = C_1 \wedge \ldots \wedge C_m$ est plus prioritaire que la formule x_1 elle-même plus prioritaire que la formule x_2 elle-même plus prioritaire que ... plus prioritaire que x_n.

Donc Alm $\propto$ 1/Strate et on en conclut que 1/Strate est Δ_2^p-complet. □

4.1.3.2 Cas d'une base de formules Cnf

Dans ce paragraphe, les formules composant la base étudiée $(E,<)$ ainsi que la formule H sont sous forme normale conjonctive. Ce sont donc des conjonctions de disjonctions, ce qui, en logique propositionnelle, correspond à des ensembles de clauses. Dans ce cas précis, le problème de satisfiabilité utilisé n'est plus Gsat mais Sat qui est lui-aussi NP-complet (voir [GJ79])[21]. Par

[21] Même remarque pour Ungsat qui devient Unsat.

contre, le problème 2-QBF n'est plus Σ_2^p-complet dans le cas de formules CNF alors qu'il l'est toujours pour des formules DNF[22] (voir [Sto77]). Cela remet donc en cause quelques-unes de mes démonstrations. D'autre part, de nombreuses transformations polynomiales utilisées lors des démonstrations de complétude dans le cas général ne conservent pas la forme conjonctive normale (voir en particulier les démonstrations pour les relations argumentatives).

Les problèmes *p-m* seront notés dans le cas d'une base de formules CNF : *p-m*-CNF.

Théorème 4.1.27 (Complexité de UNI (EXI, ARG)-T-CNF)

- UNI-T-CNF *est* Π_2^p*-complet*
- EXI-T-CNF *est* Σ_2^p*-complet*
- ARG-T-CNF $\in \Delta_3^p$ - *(*$\Sigma_2^p \cup \Pi_2^p$*) si* $\Sigma_2^p \neq \Pi_2^p$

Preuve : Étude d'appartenance.

Tous les algorithmes étant conservés, on obtient les mêmes résultats d'appartenance que dans le cas général :

- UNI-T-CNF appartient au plus à la classe de complexité Π_2^p,
- EXI-T-CNF appartient au plus à la classe de complexité Σ_2^p,
- ARG-T-CNF appartient au plus à la classe de complexité Δ_3^p.

Étude de complétude.

- pour UNI-T-CNF, au lieu d'utiliser le problème 2-QBF général comme dans la démonstration de UNI-T, on choisit d'utiliser le problème 2QBF-DNF qui est Σ_2^p-complet (voir [Sto77]) : "$\exists a \forall b G(a,b)$ satisfiable ? avec $G(a,b)$ sous forme DNF". On peut garder exactement la même transformation polynomiale puisque l'ensemble E et la formule H sont alors en forme CNF (rappel : $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, \neg G(a,b)\}$ et $H = \neg G(a,b)$). Donc UNI-T-CNF est Π_2^p-complet.
- pour EXI-T-CNF, on va utiliser le problème co-2QBF-DNF qui est Π_2^p-complet.

Soit "$\forall a \exists b G(a,b)$ avec $G(a,b)$ sous-forme CNF", une instance de co-2-QBF-DNF, posons : - $H = s$, - $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, G(a,b) \vee s\}$ avec s une nouvelle variable propositionnelle.

  Les thèses de E sont alors de la forme $Y = \{l_1, \ldots, l_n, G(a,b) \vee s\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i$. En effet, il suffit de remarquer que la formule $G(a,b) \vee s$ est toujours consistante avec n'importe quel ensemble $\{l_1, \ldots, l_n\}$ puisque s est un nouveau symbole et qu'il suffit d'étendre les interprétations satisfaisant les l_i en rajoutant la valeur *vrai* pour s. D'autre part, on remarque aussi que :

  - $\{l_1, \ldots, l_n\}$ consistant avec $G(a,b) \Leftrightarrow$
    il existe une extension M de l'assignation $\{l_1, \ldots, l_n\}$ pour les autres symboles de $G(a,b)$ (les b_i) telle que M satisfait $G(a,b)$, et M satisfait $G(a,b) \vee s$ quelle que soit la valeur de $s \Leftrightarrow$
    $\{l_1, \ldots, l_n, G(a,b) \vee s\}$ n'infère pas s.
  - $\{l_1, \ldots, l_n\}$ inconsistant avec $G(a,b) \Leftrightarrow$
    $\{l_1, \ldots, l_n\}$ infère $\neg G(a,b) \Leftrightarrow$
    $\{l_1, \ldots, l_n, G(a,b) \vee s\}$ infère s.

  On a alors :

[22] Les formules DNF sont sous forme normale disjonctive, c'est-à-dire sous la forme de disjonctions de conjonctions.

$$\forall a \exists b G(a,b) \text{ satisfiable}$$
$$\Rightarrow$$
$$\forall \{l_1, \ldots, l_n\}, \{l_1, \ldots, l_n\} \text{ consistant avec } G(a,b)$$
$$\Rightarrow$$
$$\forall Y \text{ thèse de } E,\ Y \text{ n'infère pas } s$$

et réciproquement :

$$\forall Y \text{ thèse de } E,\ Y \text{ n'infère pas } s$$
$$\Rightarrow$$
$$\forall \{l_1, \ldots, l_n\},\ \{l_1, \ldots, l_n\} \text{ n'infère pas } \neg G(a,b)$$
$$\Rightarrow$$
$$\forall \{l_1, \ldots, l_n\},\ \{l_1, \ldots, l_n\} \text{ consistant avec } G(a,b)$$
$$\Rightarrow$$
$$\forall a \exists b G(a,b) \text{ satisfiable}$$

On en conclut donc que co-2QBF-DNF $\propto$ co-EXI-T-CNF et donc que EXI-T-CNF est Σ_2^p-complet.

- pour ARG-T-CNF, les transformations utilisées dans le cas général ne conservent pas la forme conjonctive normale : si E et H sont sous forme CNF, alors $f(E) = E \cup \{H \rightarrow G\}$ ou $f(E) = E \cup \{\neg H\}$, et $f(H) = \neg H$ ne sont pas sous forme CNF mais sous forme quelconque (certaines formules sont en CNF, d'autres en DNF). Par contre, on peut étudier un nouveau problème EXI-T-MIXTE défini par la restriction de EXI-T à une base de formules E CNF et à une formule H DNF. L'algorithme de EXI-T s'applique à EXI-T-MIXTE qui appartient ainsi à la classe Σ_2^p. D'autre part la démonstration de complétude donnée pour EXI-T-CNF s'applique parfaitement à EXI-T-MIXTE, puisque on a la transformation suivante :

Soit "$\forall a \exists b G(a,b)$ avec $G(a,b)$ sous-forme CNF", une instance de co-2-QBF-DNF, posons : $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, G(a,b) \vee s\}$ avec s une nouvelle variable propositionnelle (E est une base de formules CNF), et $H = s$ (H est alors une formule DNF).

  EXI-T-MIXTE est donc un problème Σ_2^p-complet. On peut alors réutiliser les transformations du cas général :

Soit "E, H", une instance de EXI-T-MIXTE (E en CNF et H en DNF), posons : $f(E) = E \cup \{H \rightarrow G\}$ avec G une nouvelle variable propositionnelle (E est une base de formules CNF), et $f(H) = G$ ($f(H)$ est alors une formule CNF).

  et :

Soit "E, H", une instance de co-EXI-T-MIXTE (E en CNF et H en DNF), posons : $f(E) = E \cup \{\neg H\}$ (E est une base de formules CNF), et $f(H) = \neg H$ ($f(H)$ est alors une formule CNF).

  On en déduit ainsi que : ARG-T-CNF $\in \Delta_3^p$ - ($\Sigma_2^p \cup \Pi_2^p$) si $\Sigma_2^p \neq \Pi_2^p$.

$\square$

Théorème 4.1.28 (Complexité de UNI (EXI, ARG)-S-CNF)

- UNI-S-CNF *est co-*NP*-complet*

- EXI-S-CNF *est* Σ_2^p*-complet*
- ARG-S-CNF $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$ *si* $\Sigma_2^p \neq \Pi_2^p$

Preuve : Étude d'appartenance et de complétude.

On garde exactement les mêmes démonstrations que celles de UNI-S, EXI-S et ARG-S (voir les théorèmes 4.1.4 page 81, 4.1.5 page 81 et 4.1.6 page 82) en se ramenant à l'inférence classique sur des formules CNF (pour UNI-S-CNF), à EXI-T-CNF (pour EXI-S-CNF) et à ARG-T-CNF (pour ARG-S-CNF) :

- UNI-S-CNF est co-NP-complet,
- EXI-S-CNF est Σ_2^p-complet.
- ARG-S-CNF $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$ si $\Sigma_2^p \neq \Pi_2^p$.

□

Théorème 4.1.29 (Complexité de UNI (EXI, ARG)-BO-CNF)

- UNI-BO-CNF $\in \Delta_2^p$ - (NP $\cup$ *co*-NP) *si* NP $\neq$ *co*-NP
- EXI-BO-CNF *est* Σ_2^p*-complet*
- ARG-BO-CNF $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$ *si* $\Sigma_2^p \neq \Pi_2^p$

Preuve : Étude d'appartenance.

Tous les algorithmes étant conservés, on obtient les mêmes résultats d'appartenance que dans le cas général :

- UNI-BO-CNF appartient au plus à la classe de complexité Δ_2^p,
- EXI-BO-CNF appartient au plus à la classe de complexité Σ_2^p,
- ARG-BO-CNF appartient au plus à la classe de complexité Δ_3^p.

Étude de complétude.

- pour UNI-BO-CNF, on peut conserver la même transformation polynomiale que pour UNI-BO (voir théorème 4.1.7 page 82), pour prouver que SAT $\propto$ UNI-BO-CNF. Par contre, pour la seconde partie de la démonstration, on va utiliser une autre transformation.

  | Soit "G" une instance de UNSAT, posons : $E = \{\neg G \to s\}$ et $H = s$ avec s un nouveau symbole propositionnel.

  Remarquons que G et H sont sous forme CNF, et que passer E sous forme CNF se fait en temps polynomial. De plus, la seule sous-base préférée pour l'ordre "Best-Out" est $\{\neg G \to s\}$.
  On a alors :

  $$G \text{ non satisfiable} \Leftrightarrow s \text{ est UNI-BO-CNF-inférée par } E.$$

  Donc UNSAT $\propto$ UNI-BO-CNF et UNI-BO-CNF appartient à Δ_2^p - (NP $\cup$ co-NP) si NP $\neq$ co-NP.
- pour EXI-BO-CNF, on fait le même raisonnement que pour EXI-T-CNF. On utilise le problème co-2QBF-DNF qui est Π_2^p-complet.

Soit "$\forall a \exists b G(a,b)$ avec $G(a,b)$ sous-forme CNF", une instance de co-2-QBF-DNF, posons :
- $H = s$,
- $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, G(a,b) \vee s\}$
avec s une nouvelle variable propositionnelle.

Les sous-bases préférées pour l'ordre "Best-Out" de E sont de la forme $Y = \{l_1, \ldots, l_n, G(a,b) \vee s\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i$. Toutes les remarques faites pour EXI-T-CNF restent valables ici dans le cadre des sous-bases préferées pour l'ordre "Best-Out".
On a alors :

$$\forall a \exists b G(a,b) \text{ satisfiable}$$
$$\Rightarrow$$
$$\forall \{l_1, \ldots, l_n\}, \{l_1, \ldots, l_n\} \text{ consistant avec } G(a,b)$$
$$\Rightarrow$$
$$\forall Y \text{ sous-base préférée pour l'ordre "Best-Out" de } E,\ Y \text{ n'infère pas } s$$

et réciproquement :

$$\forall Y \text{ sous-base préférée pour l'ordre "Best-Out" de } E,\ Y \text{ n'infère pas } s$$
$$\Rightarrow$$
$$\forall \{l_1, \ldots, l_n\},\ \{l_1, \ldots, l_n\} \text{ n'infère pas } \neg G(a,b)$$
$$\Rightarrow$$
$$\forall \{l_1, \ldots, l_n\},\ \{l_1, \ldots, l_n\} \text{ consistant avec } G(a,b)$$
$$\Rightarrow$$
$$\forall a \exists b G(a,b) \text{ satisfiable}$$

On en conclut donc que co-2QBF-DNF $\propto$ co-EXI-BO-CNF et donc que EXI-BO-CNF est Σ_2^p-complet.

- pour ARG-BO-CNF, je retrouve les mêmes problèmes que ceux rencontrés pour ARG-T-CNF (voir théorème 4.1.27 page 114). Ici aussi, il faut étudier le problème EXI-BO-MIXTE, qui pour les mêmes raisons que celles évoquées pour EXI-T-MIXTE, sera Σ_2^p-complet. On garde les transformations données pour le cas général en utilisant les problèmes EXI-BO-MIXTE et co-EXI-BO-MIXTE. On prouve ainsi que ARG-BO-CNF $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$ si $\Sigma_2^p \neq \Pi_2^p$.

□

Théorème 4.1.30 (Complexité de UNI (EXI, ARG)-INCL-CNF)

- UNI-INCL-CNF *est* Π_2^p*-complet*
- EXI-INCL-CNF *est* Σ_2^p*-complet*
- ARG-INCL-CNF $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$ *si* $\Sigma_2^p \neq \Pi_2^p$

Preuve : Étude d'appartenance.

Tous les algorithmes étant conservés, on obtient les mêmes résultats d'appartenance que dans le cas général :

- UNI-INCL-CNF appartient au plus à la classe de complexité Π_2^p,
- EXI-INCL-CNF appartient au plus à la classe de complexité Σ_2^p,
- ARG-INCL-CNF appartient au plus à la classe de complexité Δ_3^p.

Étude de complétude.

- pour UNI-INCL-CNF, on garde la même démonstration que pour UNI-INCL (voir théorème 4.1.10 page 87) en utilisant UNI-T-CNF et on prouve que UNI-INCL-CNF est Π_2^p-complet.

- pour Exi-Incl-Cnf, on garde la même démonstration que pour Exi-Incl (voir théorème 4.1.11 page 88) en utilisant Exi-T-Cnf et on prouve que Exi-Incl-Cnf est Σ_2^p-complet.
- pour Arg-Incl-Cnf, on garde la même démonstration que pour Arg-Incl (voir théorème 4.1.12 page 89) en utilisant Arg-T-Cnf (donc Exi-T-Mixte et co-Exi-T-Mixte) et on prouve que Arg-Incl-Cnf $\in \Delta_3^p$ - $(\Sigma_2^p \cup \Pi_2^p)$ si $\Sigma_2^p \neq \Pi_2^p$.

□

Théorème 4.1.31 (Complexité de Uni (Exi, Arg)-Car-Cnf)

- Uni-Car-Cnf *est* Δ_2^p*-complet*
- Exi-Car-Cnf *est* Σ_2^p*-complet*
- Arg-Car-Cnf $\in \Delta_3^p$ *et est* Σ_2^p *-difficile*

Preuve : Étude d'appartenance.

Tous les algorithmes étant conservés, on obtient les mêmes résultats d'appartenance que dans le cas général, puisque les oracles utilisés (Sat, Max-Gsat, Ngsat-Car) restent NP-complets dans le cas de formules Cnf :

- Uni-Car-Cnf appartient au plus à la classe de complexité Δ_2^p,
- Exi-Car-Cnf appartient au plus à la classe de complexité Σ_2^p,
- Arg-Car-Cnf appartient au plus à la classe de complexité Δ_3^p.

Étude de complétude.

- pour Uni-Car-Cnf, on peut garder la même transformation que celle du cas général (voir théorème 4.1.13 page 90) puisque la base E et la formule H ainsi générées sont sous forme Cnf.
  Donc Uni-Car-Cnf est Δ_2^p-complet.
- pour Exi-Car-Cnf, on fait le même raisonnement que pour Exi-T-Cnf. On utilise le problème co-2Qbf-Dnf qui est Π_2^p-complet.

Soit "$\forall a \exists b G(a,b)$ avec $G(a,b)$ sous-forme Cnf", une instance de co-2-Qbf-Dnf, posons : - $H = s$, - $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, G(a,b) \vee s\}$ avec s une nouvelle variable propositionnelle.

  Les thèses préférées pour la cardinalité de E sont alors de la forme $Y = \{l_1, \ldots, l_n, G(a,b) \vee s\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i$. Toutes les remarques faites pour Exi-T-Cnf restent valables ici dans le cadre des thèses préférées pour la cardinalité.
  On a alors :

  $$\forall a \exists b G(a,b) \text{ satisfiable}$$
  $$\Rightarrow$$
  $$\forall \{l_1, \ldots, l_n\}, \{l_1, \ldots, l_n\} \text{ consistant avec } G(a,b)$$
  $$\Rightarrow$$
  $$\forall Y \text{ thèse préférée pour la cardinalité de } E, Y \text{ n'infère pas } s,$$

  et réciproquement :

$$\forall Y \text{ thèse préférée pour la cardinalité de } E,\ Y \text{ n'infère pas } s$$
$$\Rightarrow$$
$$\forall\{l_1,\ldots,l_n\},\ \{l_1,\ldots,l_n\} \text{ n'infère pas } \neg G(a,b)$$
$$\Rightarrow$$
$$\forall\{l_1,\ldots,l_n\},\ \{l_1,\ldots,l_n\} \text{ consistant avec } G(a,b)$$
$$\Rightarrow$$
$$\forall a \exists b G(a,b) \text{ satisfiable}$$

On en conclut donc que co-2QBF-DNF $\propto$ co-EXI-CAR-CNF et donc que EXI-CAR-CNF est Σ_2^p-complet.

- pour ARG-CAR-CNF, comme pour ARG-T-CNF (voir théorème 4.1.27 page 114), on garde la même transformation en utilisant le problème EXI-CAR-MIXTE qui est Σ_2^p-complet pour les mêmes raisons que celles évoquées lors de la démonstration de complétude de EXI-T-MIXTE. On prouve ainsi que ARG-CAR-CNF $\in \Delta_3^p$ et est Σ_2^p-difficile. Tout comme pour ARG-CAR, je ne suis pas parvenue à prouver que ARG-CAR-CNF est Π_2^p -difficile.

$\square$

Théorème 4.1.32 (Complexité de UNI (EXI, ARG)-LEX-CNF)

- UNI-LEX-CNF *est* Δ_2^p*-complet*
- EXI-LEX-CNF *est* Σ_2^p*-complet*
- ARG-LEX-CNF $\in \Delta_3^p$ *et est* Σ_2^p *-difficile*

Preuve : Étude d'appartenance.

Comme pour UNI (EXI, ARG)-CAR, tous les algorithmes sont conservés, et on obtient les mêmes résultats d'appartenance que dans le cas général :

- UNI-LEX-CNF appartient au plus à la classe de complexité Δ_2^p,
- EXI-LEX-CNF appartient au plus à la classe de complexité Σ_2^p,
- ARG-LEX-CNF appartient au plus à la classe de complexité Δ_3^p.

Étude de complétude.

- pour UNI-LEX-CNF, on peut garder la même transformation que celle du cas général (voir théorème 4.1.16 page 97) qui permet de passer de UNI-CAR-CNF à UNI-LEX-CNF, puis en utilisant le fait que UNI-CAR-CNF est Δ_2^p-complet, on obtient que UNI-LEX-CNF est Δ_2^p-complet.
- pour EXI-LEX-CNF, on peut garder la même transformation que celle du cas général (voir théorème 4.1.17 page 103) qui permet de passer de EXI-CAR-CNF à EXI-LEX-CNF, puis en utilisant les résultats sur EXI-CAR-CNF, on obtient que EXI-LEX-CNF est Σ_2^p-complet.
- pour ARG-LEX-CNF, on peut garder la même transformation que celle du cas général (voir théorème 4.1.18 page 105) qui permet de passer de ARG-CAR-CNF à ARG-LEX-CNF, puis en utilisant les résultats sur ARG-CAR-CNF, on prouve que ARG-LEX-CNF $\in \Delta_3^p$ et est Σ_2^p-difficile. Par contre, comme pour ARG-LEX, je ne suis pas parvenue à prouver que ARG-LEX-CNF est Π_2^p -difficile.

$\square$

Complexité de UNI (EXI, ARG)-E-CNF Il n'existe pas à ma connaissance de travaux sur ces relations d'inférence dans le cas de formules CNF. Mon étude ne portant pas particulièrement sur la logique des défauts, je ne m'étendrais pas sur ce sujet.

4.1.3.3 Cas d'une base de Horn

Ici, on part de l'hypothèse que la base étudiée $(E, <)$ est un ensemble de *conjonctions de clauses de Horn* et que la formule traitée H est une clause de Horn. Rappelons qu'une clause de Horn est une formule de la forme :

- soit $A_1 \ldots A_n \rightarrow B$
- soit $A_1 \ldots A_n \rightarrow$ (avec A_i et B des littéraux positifs).

Donc, dans toute la suite, une *base de Horn* sera définie comme un ensemble de conjonctions de clauses de Horn et pas comme un ensemble de clauses de Horn (appelé par opposition *base de Horn stricte*). Ainsi, on pourra garder un aspect syntaxique conforme à nos choix initiaux : la formule $a \wedge b$ n'étant pas traitée comme l'ensemble $\{a, b\}$, ce qui n'aurait pas été possible avec une base de Horn stricte.

Toutefois, notons que le problème de l'existence ou non d'un modèle dans une base de Horn telle que nous l'avons définie est exactement le même que dans une base de Horn stricte. Ce qui va nous permettre de profiter du fait que, dans le cas de Horn strict, les problèmes GSAT et UNGSAT (notés alors SAT-HORN et UNSAT-HORN) deviennent polynomiaux. Ceci implique que les problèmes d'inférence classique deviennent polynomiaux eux aussi, que ce soit dans le cas d'une base de Horn stricte ou pas. On a en effet :

- $B \vdash H \Leftrightarrow B \cup \{\neg H\}$ inconsistant $\Leftrightarrow B \cup \{\neg H\}$ non satisfiable,
- $B \not\vdash H \Leftrightarrow B \cup \{\neg H\}$ consistant $\Leftrightarrow B \cup \{\neg H\}$ satisfiable.

Ainsi, tant que la formule H est une clause de Horn, $\neg H$ peut être vue soit comme un ensemble de clauses de Horn, soit comme une conjonction de clauses de Horn, et dans les deux cas, les problèmes d'inférence classique se ramènent alors aux problèmes SAT-HORN et UNSAT-HORN.

On constate rapidement que tous les problèmes étudiés dans la section 4.1.2 page 75 passent dans la classe de complexité inférieure.

Remarquons aussi que les résultats donnés dans cette section permettent de fixer une borne supérieure pour la complexité des problèmes étudiés dans le cas où la base E est une base de Horn *stricte* et plus seulement une base de Horn. Par contre, tous les résultats de complétude seront à revoir si on veut affiner ce résultat.

Notation : dans le cas d'une base de Horn, les problèmes p-m seront notés p-m-HORN.

Théorème 4.1.33 (Complexité de UNI (EXI, ARG)-T-HORN)

- UNI-T-HORN *est co-*NP*-complet*
- EXI-T-HORN *est* NP*-complet*
- ARG-T-HORN $\in \Delta_2^p$ - (NP $\cup$ *co-*NP) *si* NP $\neq$ *co-*NP

Preuve : Étude d'appartenance.

Tous les algorithmes sont conservés et comme la complexité des problèmes d'inférence a diminué, on a alors :

- UNI-T-HORN appartient au plus à la classe de complexité co-NP,
- EXI-T-HORN appartient au plus à la classe de complexité NP,
- ARG-T-HORN appartient au plus à la classe de complexité Δ_2^p.

Étude de complétude.

- pour Uni-T-Horn, on va utiliser la même méthode que pour Sbr-Horn (voir annexe A page 235), c'est-à-dire prouver que Sat $\propto$ co-Uni-T-Horn[23].

  Soit "C" une instance de Sat, avec $C = \{C_j\}$ pour $j = 1 \dots m$ ensemble de clauses quelconques à satisfaire, et $V(C) = \{x_1, \dots, x_n\}$ l'ensemble des variables propositionnelles de C, posons :

  - $H = \neg s$.
  - $E = \{p, x_1, \dots, x_n, y_1, \dots, y_n, \neg z_1, \dots, \neg z_n, \neg s\}$ avec $y_1, \dots, y_n$, $z_1, \dots, z_n$, s nouveaux symboles de variables propositionnelles, et avec la formule $p = (z_1 \dots z_n \rightarrow s) \wedge \bigwedge_{i=1\dots n}((\neg x_i \vee \neg y_i) \wedge (y_i \rightarrow z_i) \wedge (x_i \rightarrow z_i)) \wedge \bigwedge_{j=1\dots m} C_j[y]$,

  avec la notation suivante : $C_j[y]$ = la clause C_j dans laquelle on remplace tous les x_i positifs par des $\neg y_i$.

  Remarques préliminaires :

  - Il est inutile de définir un ordre entre les formules de la base.
  - L'idée de cette démonstration est issue d'un article de Eiter et Gottlob (voir [EG92]).
  - Le but est de transformer un problème portant sur des clauses quelconques en un problème portant sur des conjonctions de clauses de Horn. Il faut donc transformer les clauses initiales en clauses de Horn ou en conjonctions de clauses de Horn. Pour cela, on introduit une nouvelle variable propositionnelle y_i pour chaque x_i de C, y_i jouera le rôle de $\neg x_i$. Cela pose alors un problème de cohérence : il ne faut pas que x_i et y_i puissent avoir la même valeur de vérité. J'introduis alors une nouvelle variable propositionnelle z_i assurant le fait que x_i et y_i ne soient pas vraies en même temps ; la maximalité pour l'inclusion assure pour sa part qu'au moins une des deux variables, x_i ou y_i, est vraie, quel que soit i. La dernière variable rajoutée s sera vraie quand tous les z_i seront vrais, donc quand pour tous les x_i et y_i, on aura valeur-vérité $(x_i) \neq$ valeur-vérité (y_i).
  - En transformant C_j en $C_j[y]$, on obtient une clause de Horn, quel que soit j.
  - On a alors la propriété suivante :

    Propriété 4.1.24 *Si la valeur de vérité de x_i est différente de la valeur de vérité de y_i $\forall i$, alors C sera satisfaite ssi $C[y]$ est satisfaite.*
  - De plus, si on calcule les sous-bases maximales consistantes de E, on trouve que :

    Propriété 4.1.25 *Si C est satisfiable alors il existe Y sous-base maximale consistante de E telle que $Y \vdash z_i$ ($\forall i$) et $Y \vdash s$. Y n'inférera pas $\neg s$.*
  - La seule sous-base maximale consistante inférant s est une sous-base contenant la formule p, puisque la variable s n'apparaît que dans p.

  Prouvons maintenant que Sat$(C) \Leftrightarrow$ co-Uni-T-Horn (E, H).

  - Sat implique co-Uni-T-Horn :

    C satisfiable

    $\Leftrightarrow$

    il existe une interprétation Φ de $\{x_1, \dots, x_n\}$ dans laquelle C est vraie

    $\Rightarrow$

    Posons $W = \{x_i$ tels que $\Phi(x_i)$ soit *vrai*$\} \cup \{yj$ tels que $\Phi(xj)$ soit *faux*$\}$.

    $\forall i\ \Phi(x_i) \neq \Phi(y_i)$ puisque $y_i \equiv \neg x_i$.

    On a donc : $\forall j\ C_j[y]$ satisfaite puisque C_j est satisfaite et que x_i et y_i n'ont pas la même valeur de vérité $\forall i$.

[23]Notons qu'il s'agit ici du problème Sat portant sur la satisfiabilité d'ensembles de clauses quelconques, pas de clauses de Horn !

On peut ainsi en déduire que p est satisfaite, donc que $Y = \{p\} \cup W$ est maximale consistante dans E[24].

$\Rightarrow$

il existe une sous-base maximale consistante Y de E telle que Y infère tous les z_i et infère s.

$\Rightarrow$

il existe une sous-base maximale consistante Y de E telle que Y n'infère pas $\neg s$.

$\Leftrightarrow$

co-Uni-T-Horn $(E,\ H)$

- co-Uni-T-Horn implique Sat :

co-Uni-T-Horn $(E,\ H)$

$\Leftrightarrow$

il existe Y sous-base maximale consistante de E telle que Y n'infère pas $\neg s$.

Donc Y est inconsistante avec $\neg s$, puisque $\neg s \in E$, donc Y infère s.

Y contient p, donc Y contient un certain ensemble $\{x_i\} \cup \{y_i\}$ tel que dans une interprétation satisfaisant Y donc p, on aura :

- $\forall i$ la valeur de vérité de x_i est différente de celle de y_i,

- les $C_j[y]$ seront satisfaites.

Posons comme interprétation Φ :

$\Phi(x_i) = vrai$ si x_i appartient à Y et $faux$ sinon

$\Rightarrow$

Dans l'interprétation Φ qui satisfait Y, les $C_j[y]$ sont satisfaites et comme les x_i et les y_i n'ont pas la même valeur de vérité

$\Rightarrow$

les C_j sont satisfaites elles aussi

$\Leftrightarrow$

C satisfiable

Le problème Sat se ramène donc polynomialement à co-Uni-T-Horn. On en déduit que le problème co-Uni-T-Horn est NP-complet, et donc que Uni-T-Horn est co-NP-complet.

- pour Exi-T-Horn, on va faire le même raisonnement que celui pour co-Uni-T-Horn, c'est-à-dire prouver que Sat $\propto$ Exi-T-Horn.

Soit "C" une instance de Sat, avec $C = \{C_j\}$ pour $j = 1 \ldots m$ ensemble de clauses quelconques à satisfaire, soit $V(C) = \{x_1, \ldots, x_n\}$ l'ensemble des variables propositionnelles de C, posons : - $H = s$. - $E = \{p, x_1, \ldots, x_n, y_1, \ldots, y_n, \neg z_1, \ldots, \neg z_n, \neg s\}$ avec $y_1, \ldots, y_n$, $z_1, \ldots, z_n$, s nouveaux symboles de variables propositionnelles, et avec la formule $p = (z_1 \ldots z_n \rightarrow s) \wedge \bigwedge_{i=1\ldots n}((\neg x_i \vee \neg y_i) \wedge (y_i \rightarrow z_i) \wedge (x_i \rightarrow z_i)) \wedge \bigwedge_{j=1\ldots m} C_j[y]$, avec la notation suivante : $C_j[y]$ = la clause C_j dans laquelle on remplace tous les x_i positifs par des $\neg y_i$.

Les remarques préliminaires fournies pour la démonstration de Uni-T-Horn restent valables.

Prouvons maintenant que Sat$(C) \Leftrightarrow$ Exi-T-Horn (E, H).

- Sat implique Exi-T-Horn :

[24] Cette étape du raisonnement reste toujours la même pour les démonstrations sur les problèmes à clauses de Horn montrant l'existence d'une transformation polynomiale entre Sat et le problème traité. Elle ne sera donc pas répétée.

C satisfiable
$\Rightarrow$
de la même façon que pour co-UNI-T-HORN, il existe une sous-base maximale consistante Y de E définie par
$Y = \{p\} \cup \{x_i \text{ tels que } \Phi(x_i) \text{ soit } vrai\} \cup \{yj \text{ tels que } \Phi(xj) \text{ soit } faux\}$
et telle que Y infère tous les z_i et infère s.
$\Leftrightarrow$
EXI-T-HORN (E, H)

- EXI-T-HORN implique SAT :

EXI-T-HORN (E, H)
$\Leftrightarrow$
il existe Y sous-base maximale consistante de E telle que Y infère s.
Y contient p, donc Y contient un certain ensemble $\{x_i\} \cup \{y_i\}$ tel que dans une interprétation satisfaisant Y donc p, on aura :
-$\forall i$ la valeur de vérité de x_i est différente de celle de y_i,
- les $C_j[y]$ seront satisfaites.
Posons comme interprétation Φ :
$\Phi(x_i) = vrai$ si x_i appartient à Y et $faux$ sinon
$\Rightarrow$
Dans l'interprétation Φ qui satisfait Y, les $C_j[y]$ sont satisfaites et comme les x_i et les y_i n'ont pas la même valeur de vérité
$\Rightarrow$
les C_j sont satisfaites elles aussi
$\Leftrightarrow$
C satisfiable

Le problème SAT se ramène donc polynomialement à EXI-T-HORN. On en déduit que le problème EXI-T-HORN est NP-complet.

- pour ARG-T-HORN, on ne peut pas garder la même démonstration que pour ARG-T (voir théorème 4.1.3 page 78). En effet les transformations polynomiales utilisées pour passer de EXI-T à ARG-T et de co-EXI-T à ARG-T ne conservent pas la propriété Horn des formules utilisées :
  - $H \rightarrow G$ ne sera une clause de Horn que si H est un littéral positif (rappelons que G est une nouvelle variable propositionnelle),
  - $\neg H$ ne sera une clause de Horn que si H est un littéral positif ou négatif.

Il va donc falloir étudier un nouveau problème appelé EXI-T-HORN-POSITIF :

instance : E une base de Horn, H un littéral positif,
question : H est-elle conséquence faible de E en utilisant les thèses définies à partir de E ?

L'algorithme 4.33 à appliquer à ce problème est identique à celui de EXI-T et EXI-T-HORN.

Algorithme 4.33 : EXI-T-HORN-POSITIF (E, H)
début
  deviner un sous-ensemble Y de E
  vérifier que Y est une thèse de E
  vérifier que Y infère H
fin

Cet algorithme est non déterministe (à cause du "deviner") polynomial, puisque E est une base de Horn et que H est un littéral positif (donc une clause de Horn). Le problème EXI-T-HORN-POSITIF appartient à la classe de complexité NP.

EXI-T-HORN-POSITIF est-il NP-complet ? Il suffit de se ramener à la démonstration de complétude faite pour EXI-T-HORN et de constater qu'elle reste valable en intégralité pour EXI-T-HORN-POSITIF puisqu'on posait : $H = s$ (s étant un littéral positif). J'en conclus que EXI-T-HORN-POSITIF est NP-complet.
À partir de là, je peux reprendre les transformations polynomiales définies pour ARG-T et les appliquer respectivement à EXI-T-HORN-POSITIF et co-EXI-T-HORN-POSITIF. Ce qui donne :

| Soit "E, H" une instance de EXI-T-HORN-POSITIF, posons :
$f(E) = E \cup \{H \rightarrow G\}$ et $f(H) = G$
(G nouveau symbole propositionnel) |

$f(E)$ est bien une base de Horn et $f(H)$ est une clause de Horn. Puis avec :

| Soit "E, H" une instance de co-EXI-T-HORN-POSITIF, posons :
$f(E) = E \cup \{\neg H\}$ et $f(H) = \neg H$ |

Ici aussi $f(E)$ est une base de Horn et $f(H)$ est une clause de Horn.
Ainsi, on obtient que ARG-T-HORN appartient à la classe de complexité Δ_2^p - (NP $\cup$ co-NP) si NP $\neq$ co-NP.

□

Théorème 4.1.34 (Complexité de UNI (EXI, ARG)-S-HORN)

- UNI-S-HORN *est* P
- EXI-S-HORN *est* NP-*complet*
- ARG-S-HORN $\in \Delta_2^p$ - *(*NP $\cup$ *co*-NP*) si* NP $\neq$ *co*-NP

Preuve : Étude d'appartenance et de complétude.

Toutes les démonstrations sont conservées et comme la complexité de l'inférence classique pour un ensemble de conjonctions de clauses de Horn (pour UNI-S-HORN), de EXI-T-HORN (pour EXI-S-HORN) et de ARG-T-HORN (pour ARG-S-HORN) a diminué, on a alors :

- UNI-S-HORN est désormais de complexité P,
- EXI-S-HORN est NP-complet.
- ARG-S-HORN appartient à la classe de complexité Δ_2^p - (NP $\cup$ co-NP) si NP $\neq$ co-NP.

□

Théorème 4.1.35 (Complexité de UNI (EXI, ARG)-BO-HORN)

- UNI-BO-HORN *est* P
- EXI-BO-HORN *est* NP-*complet*
- ARG-BO-HORN $\in \Delta_2^p$ - *(*NP $\cup$ *co*-NP*) si* NP $\neq$ *co*-NP

Preuve : Étude d'appartenance.

Tous les algorithmes sont conservés et comme la complexité des problèmes d'inférence a diminué, on a alors :

- UNI-BO-HORN est désormais de complexité P,
- EXI-BO-HORN appartient au plus à la classe de complexité NP,

- Arg-Bo-Horn appartient au plus à la classe de complexité Δ_2^p.

Étude de complétude.

- pour Exi-Bo-Horn, on va utiliser la même méthode que pour Exi-T-Horn (voir théorème 4.1.33 page 120), c'est-à-dire prouver que Sat $\propto$ Exi-Bo-Horn.

  Soit "C" une instance de Sat, avec $C = \{C_j\}$ pour $j = 1 \ldots m$ ensemble de clauses quelconques à satisfaire, soit $V(C) = \{x_1, \ldots, x_n\}$ l'ensemble des variables propositionnelles de C, posons :
  - $H = s$.
  - $E = \{p, x_1, \ldots, x_n, y_1, \ldots, y_n, \neg z_1, \ldots, \neg z_n, \neg s\}$ avec $y_1, \ldots, y_n$, $z_1, \ldots, z_n$, s nouveaux symboles de variables propositionnelles, et avec la formule $p = (z_1 \ldots z_n \rightarrow s) \wedge \bigwedge_{i=1 \ldots n}((\neg x_i \vee \neg y_i) \wedge (y_i \rightarrow z_i) \wedge (x_i \rightarrow z_i)) \wedge \bigwedge_{j=1 \ldots m} C_j[y]$,

  avec la notation suivante : $C_j[y]$ = la clause C_j dans laquelle on remplace tous les x_i positifs par des $\neg y_i$. Il est inutile de définir un ordre entre les formules de la base.

  Les remarques préliminaires fournies pour Uni-T-Horn restent valables ici. On constate en plus que :

  - Comme il n'y a pas d'ordre entre les formules de E, cela signifie que pour toute sous-base consistante Y de E telle que $Y \neq E$, $a(Y) = 1$.
  - E est inconsistant ; en effet pour que E soit consistant, il faut que tous les x_i, tous les y_i soient à *vrai*, tous les z_i et s soient à *faux*, et que p soit à *vrai*, or pour que p soit *vrai*, cela impose que soit x_i, soit y_i soit *faux*, $\forall i$; on a donc une incohérence.
  - La seule sous-base consistante préférée dans l'ordre "Best-Out"[25] inférant s est une sous-base contenant la formule p et un ensemble de x_i et y_i tel que les valeurs de vérité des x_i soient différentes des valeurs de vérité des y_i, puisque la variable s n'apparaît que dans p.

  Prouvons maintenant que Sat$(C) \Leftrightarrow$ Exi-Bo-Horn (E, H).

  - Sat implique Exi-Bo-Horn :

    C satisfiable

    $\Rightarrow$

    il existe une sous-base consistante Y de E telle que Y infère tous les z_i et infère s.

    $\Leftrightarrow$

    Exi-Bo-Horn (E, H)
  - Exi-Bo-Horn implique Sat :

    Exi-Bo-Horn (E, H)

    $\Leftrightarrow$

    il existe Y sous-base consistante préférée dans l'ordre "Best-Out" de E telle que Y infère s. Donc Y contient p, donc Y contient un ensemble $\{x_i\} \cup \{y_i\}$ tel que dans une interprétation satisfaisant Y donc p, on aura :

    - $\forall i$ la valeur de vérité de x_i est différente de celle de y_i,

    - les $C_j[y]$ seront satisfaites.

    Posons comme interprétation Φ : $\Phi(x_i) = vrai$ si $x_i \in Y$ et *faux* sinon

    $\Rightarrow$

    Dans l'interprétation Φ qui satisfait Y, les $C_j[y]$ sont satisfaites et comme les x_i et les y_i n'ont pas la même valeur de vérité

[25] Étant donné qu'il n'y a qu'une seule strate dans E, toute sous-base consistante est préférée pour l'ordre "Best-Out".

$$\Rightarrow$$
les C_j sont satisfaites elles aussi
$$\Leftrightarrow$$
C satisfiable

Le problème SAT se ramène donc polynomialement à EXI-BO-HORN. Et on en déduit que le problème EXI-BO-HORN est NP-complet.

- pour ARG-BO-HORN, on ne peut pas garder la même démonstration que pour ARG-BO (voir théorème 4.1.9 page 85). En effet, comme pour ARG-T, les transformations polynomiales utilisées pour passer de EXI-BO à ARG-BO et de co-EXI-BO à ARG-BO ne conservent pas la "Hornitude" des formules utilisées (voir remarques sur la démonstration du théorème 4.1.33 page 120 pour ARG-T-HORN). Il va donc falloir étudier un nouveau problème appelé EXI-BO-HORN-POSITIF :

  instance : E une base de Horn, H un littéral positif,
  question : H est-elle conséquence faible de $(E, <)$ en utilisant l'ordre "Best Out" sur les sous-bases consistantes ?

  Je démontre comme pour EXI-T-HORN-POSITIF (voir théorème 4.1.33 page 120) que EXI-BO-HORN-POSITIF est NP-complet. À partir de là, je peux reprendre les transformations polynomiales définies pour ARG-BO et les appliquer respectivement à EXI-BO-HORN-POSITIF et co-EXI-BO-HORN-POSITIF :

Soit "$(E,<)$, H" une instance de EXI-BO-HORN-POSITIF, posons : - $f(H) = G$ (G nouveau symbole propositionnel), - $f((E,<)) = (E \cup \{H \rightarrow G\}, <)$ avec $H \rightarrow G$ constituant seule la première strate.

  On a bien $f((E,<))$ qui est une base de Horn et $f(H)$ qui est une clause de Horn. Puis, avec :

Soit "$(E,<), H$" une instance de co-EXI-BO-HORN-POSITIF, posons : - $f(H) = \neg H$, - $f((E,<)) = (E \cup \{\neg H\}, <)$ avec $\neg H$ constituant seule la dernière strate.

  Ici aussi $f((E,<))$ est une base de Horn et $f(H)$ est une clause de Horn.
  Ainsi, on obtient que ARG-BO-HORN appartient à la classe de complexité Δ_2^p - (NP $\cup$ co-NP) si NP $\neq$ co-NP.

□

Théorème 4.1.36 (Complexité de UNI (EXI, ARG)-INCL-HORN)

- UNI-INCL-HORN *est co*-NP-*complet*
- EXI-INCL-HORN *est* NP-*complet*
- ARG-INCL-HORN $\in \Delta_2^p$ - *(*NP $\cup$ *co*-NP*) si* NP $\neq$ *co*-NP

Preuve : Étude d'appartenance.

Tous les algorithmes sont conservés et comme la complexité des problèmes d'inférence a diminué, on a alors :

- UNI-INCL-HORN appartient au plus à la classe de complexité co-NP,
- EXI-INCL-HORN appartient au plus à la classe de complexité NP,
- ARG-INCL-HORN appartient au plus à la classe de complexité Δ_2^p.

Étude de complétude.

- pour Uni-Incl-Horn, on garde exactement la même démonstration que pour Uni-Incl (voir théorème 4.1.10 page 87), mais comme maintenant le problème Uni-T-Horn est co-NP-complet, on obtient que Uni-Incl-Horn est co-NP-complet.
- pour Exi-Incl-Horn, on garde exactement la même démonstration que pour Exi-Incl (voir théorème 4.1.11 page 88), mais comme maintenant le problème Exi-T-Horn est NP-complet, on obtient que Exi-Incl-Horn est NP-complet.
- pour Arg-Incl-Horn, on garde exactement la même démonstration que pour Arg-Incl (voir théorème 4.1.12 page 89), mais comme maintenant le problème Arg-T-Horn utilise les problèmes co-Exi-T-Horn-Positif et Exi-T-Horn-Positif qui sont respectivement co-NP-complet et NP-complet, on obtient donc que Arg-Incl-Horn appartient à la classe de complexité Δ_2^p - (NP $\cup$ co-NP) si NP $\neq$ co-NP.

□

Les multi-ensembles Jusqu'à présent, tous les problèmes utilisés dans les démonstrations de complexité étaient définis pour des ensembles de croyances. Or, pour étudier le cas particulier des clauses de Horn pour les problèmes utilisant la cardinalité, il est indispensable d'introduire une nouvelle notion : *les multi-ensembles*. En effet, dans les démonstrations proposées pour les théorèmes 4.1.37 page 130 et 4.1.38 page 137, je vais être amenée à répéter des formules de la base initiale ce qui n'est pas compatible avec la notion d'ensemble. Notons toutefois que tous les résultats obtenus jusqu'ici sur des ensembles sont conservés dans leur intégralité quand on travaille sur des multi-ensembles. En effet, le cas des ensembles est une restriction du cas des multi-ensembles donc toutes les preuves de complétude restent valables, quant aux preuves d'appartenance, on constate que les seuls algorithmes dépendants de cet aspect sont les algorithmes utilisant la cardinalité et ils peuvent être modifiés pour intégrer cet aspect.

Définition 4.1.10 *E est un multi-ensemble de formules propositionnelles ssi :*

- *E est un ensemble de couples (Φ_i, p_i) avec :*
  - *Φ_i une formule propositionnelle,*
  - *p_i est le poids de Φ_i, c'est-à-dire le nombre d'occurrences de Φ_i ;*
- *il ne peut pas y avoir dans E deux couples (Φ_i, p_i) et (Φ_j, p_j) tels que $\Phi_i = \Phi_j$ et $p_i \neq p_j$.*

Définition 4.1.11 *$(E, <) = E_1 \cup \ldots \cup E_n$ est un multi-ensemble stratifié de formules propositionnelles ssi :*

- *$(E, <)$ est un ensemble stratifié de couples (Φ_i, p_i) avec :*
  - *Φ_i une formule propositionnelle,*
  - *p_i est le poids de Φ_i, c'est-à-dire le nombre d'occurrences de Φ_i ;*
- *il ne peut pas y avoir dans $(E, <)$ deux couples (Φ_i, p_i) et (Φ_j, p_j) tels que $\Phi_i = \Phi_j$ et $p_i \neq p_j$.*

Définition 4.1.12 *Les multi-ensembles étant définis comme des ensembles de couples, on conserve la notion d'inclusion classique : soit E un multi-ensemble, S est un "sous-ensemble" de E ssi $S \subseteq E$. Idem pour le cas stratifié.*

Exemples : Soit $E = \{(A,3),(B,1),(C,2)\}$, E est un multi-ensemble correspondant à la collection de formules propositionnelles suivante : $\{A, A, A, B, C, C\}$. Soit $E = \{(A,3),(B,1),(A,2)\}$, E n'est pas un multi-ensemble. Soit $E = \{(A,3),(B,1),(A,3)\}$, E est le multi-ensemble : { (A,3), (B,1)}.

Cette notion de multi-ensemble est essentielle en cardinalité car dire qu'une formule Φ peut être répétée p fois change totalement le calcul des sous-bases préférées pour la cardinalité.

Définition 4.1.13 *Soit S un sous-ensemble du multi-ensemble E, on note $|S|^*$ et on appelle cardinal de S*

$|S|^* = \sum_i p_i$ *tel que* $(\Phi_i, p_i) \in S$.

Définition 4.1.14 *On note E^* l'ensemble de formules propositionnelles issu du multi-ensemble E :*

$E^* = \{\Phi_i$ *telle que* $(\Phi_i, p_i) \in E\}$

Définition 4.1.15 *Soit S un sous-ensemble du multi-ensemble E, S est une sous-base consistante de E ssi S^* est une sous-base consistante de E^*.*

Soit S un sous-ensemble du multi-ensemble E, S est une thèse de E ssi S^* est une thèse de E^*.

Définition 4.1.16 *Soit S un sous-ensemble du multi-ensemble E, S est une sous-base préférée pour la cardinalité de E ssi :*

- *S est une sous-base consistante de E,*
- *il n'existe pas S' sous-base consistante de E telle que $|S|^* < |S'|^*$.*

Définition 4.1.17 *Soit S un sous-ensemble du multi-ensemble $(E,<) = E_1 \cup \ldots \cup E_n$, S est une sous-base préférée pour l'ordre lexicographique de $(E,<)$ ssi $\forall k = 1 \ldots n$, $S_1 \cup \ldots \cup S_k$ est une sous-base préférée pour la cardinalité du multi-ensemble $E_1 \cup \ldots \cup E_k$.*

Je peux alors redéfinir les problèmes d'inférence non-monotone liés à la cardinalité :

Définition 4.1.18 *Les problèmes d'inférence non-monotone* Uni *(*Exi, Arg*)*-Car *sur des multi-ensembles sont :*

> **instance :** *E un multi-ensemble de formules propositionnelles, H une formule propositionnelle,*
>
> **question :** *H est-elle une conséquence forte (faible, argumentative) de E en utilisant l'ordre basé sur la cardinalité sur les thèses de E ?*

Ces problèmes sont notés Uni *(*Exi, Arg*)*-Car-Me.

Définition 4.1.19 *Les problèmes d'inférence non-monotone* Uni *(*Exi, Arg*)*-Lex *sur des multi-ensembles sont :*

> **instance :** *E un multi-ensemble de formules propositionnelles, H une formule propositionnelle,*
>
> **question :** *H est-elle une conséquence forte (faible, argumentative) avec préférences de E en utilisant l'ordre lexicographique sur les thèses de E ?*

Ces problèmes sont notés Uni *(*Exi, Arg*)*-Lex-Me.

Algorithme 4.34 : UNI-CAR-ME (E, H)
début

> $k \leftarrow 0$
> $nfmax \leftarrow (\sum_i p_i$ tel que $(\Phi_i, p_i) \in E)$
> **si** MAX-GSAT-ME (E, $nfmax$) **alors**
> > $k \leftarrow nfmax$
>
> **sinon**
> > $nfmin \leftarrow 0$
> > **tant que** $((nfmax$ - $nfmin) > 1)$ **faire**
> > > $k \leftarrow (nfmax + nfmin)$ div 2 *(* div = division entière *)*
> > > **si** MAX-GSAT-ME (E, k) **alors**
> > > > $nfmin \leftarrow k$
> > >
> > > **sinon**
> > > > $nfmax \leftarrow k$
>
> **si** NGAT-CAR-ME (E,H,k) **alors**
> > $échec \leftarrow vrai$
>
> **sinon**
> > $échec \leftarrow faux$
>
> vérifier que l'on n'a pas échec ($échec = faux$)

fin

Algorithme 4.35 : MAX-GSAT-ME (E, k)
début

> deviner une interprétation M
> $p \leftarrow 0$
> **pour** chaque formule G de E **faire**
> > **si** M satisfait G **alors**
> > > $p \leftarrow p + p_G$ *(* p_G = poids de G *)*
>
> vérifier que $p \geq k$

fin

Algorithme 4.36 : NGSAT-CAR-ME (E, H, k)
début

> deviner une interprétation M
> *(* vérifier qu'elle satisfait k formules de E *)*
> $k' \leftarrow 0$
> **pour** chaque formule G de E **faire**
> > **si** M satisfait G **alors**
> > > $k' \leftarrow k' + p_G$ *(* p_G = poids de G *)*
>
> vérifier que $k' = k$
>
> vérifier que M ne satisfait pas H

fin

Algorithme 4.37 : EXI-CAR-ME (E, H)
début
deviner un sous-ensemble Y de E
(vérifier que Y est une thèse préférée pour la cardinalité de E *)*
$échec \leftarrow faux$
$k \leftarrow |Y|^*$
si il existe une sous-base consistante Z de E telle que $|Z|^* > k$ **alors**
 $échec \leftarrow vrai$

si non *échec* **alors**
 vérifier que Y infère H
fin

Notons que les algorithmes utilisant la cardinalité et apparaissant dans les preuves d'appartenance peuvent être modifiés comme suit :

- algorithme 4.34 page précédente pour UNI-CAR-ME (recherche dichotomique pour conserver un algorithme polynomial) ;
- algorithme 4.35 page précédente pour MAX-GSAT-ME ;
- algorithme 4.36 page précédente pour NGSAT-CAR-ME ;
- algorithme 4.37 pour EXI-CAR-ME ;
- algorithme 4.38 pour MAX-GSAT-STRICT-ME ;

Algorithme 4.38 : MAX-GSAT-STRICT-ME (E, k)
début
deviner une interprétation M
$p \leftarrow 0$
pour chaque formule G de E **faire**
 si M satisfait G **alors**
  $p \leftarrow p + p_G$ *(* p_G = poids de G *)*
vérifier que $p > k$
fin

- algorithmes 4.39 page ci-contre et 4.40 page suivante pour UNI-LEX-ME (recherche dichotomique pour conserver un algorithme polynomial) ;
- algorithme 4.41 page 132 pour MAX-GSAT-ARRAY-ME ;
- algorithme 4.42 page 132 pour NGSAT-LEX-ME ;
- algorithme 4.43 page 132 pour EXI-LEX-ME ;

Tous ces algorithmes donnent la même borne maximum pour la complexité que celle obtenue par les algorithmes correspondant au cas des ensembles.

Théorème 4.1.37 (Complexité de UNI (EXI, ARG)-CAR-HORN)

- UNI-CAR-HORN $\in \Delta_2^p$ *mais* UNI-CAR-ME-HORN *est* Δ_2^p*-complet*
- EXI-CAR-HORN $\in \Sigma_2^p$
- ARG-CAR-HORN $\in \Delta_3^p$

Preuve : Étude d'appartenance.

Dans tous les algorithmes proposés, que ce soit pour UNI-CAR ou pour EXI-CAR, j'utilise non seulement un oracle traitant l'inférence classique, mais aussi des oracles

Algorithme 4.39 : UNI-LEX-ME $((E,<),\ H)$
début

$E' \leftarrow \{(H \rightarrow G,1)\} \cup E \cup \{(\neg G,1)\}$
$k \leftarrow <0,0,\ldots,0>$ *(* vecteur de dimension n' = nombre de strates dans E' *)*
pour $n_s \leftarrow 1$ à n' **faire**
  $nfmax \leftarrow (\sum_i p_i$ tel que $(\Phi_i, p_i) \in E'_{n_s})$
  $k[n_s] \leftarrow nfmax$
  si (non MAX-GSAT-ARRAY-ME $(E',\ k)$) **alors**
    $nfmin \leftarrow 0$
    $k[n_s] \leftarrow 0$
    tant que $((nfmax$ - $nfmin) > 1)$ **faire**
      $k[n_s] \leftarrow (nfmax + nfmin)$ div 2 *(* div = division entière *)*
      si MAX-GSAT-ARRAY-ME $(E',\ k)$ **alors**
        $nfmin \leftarrow k[n_s]$
      sinon
        $nfmax \leftarrow k[n_s]$

vérifier que $k[n']$ est différent de 1
fin

Algorithme 4.40 : UNI-LEX-ME $((E,<),\ H)$ (autre version)
début

(calcul du vecteur k *)*
$k \leftarrow <0,0,\ldots,0>$ *(* vecteur de dimension n (nombre de strates dans E) *)*
pour $n_s \leftarrow 1$ à n **faire**
  $nfmax \leftarrow (\sum_i p_i$ tel que $(\Phi_i, p_i) \in E_{n_s})$
  $k[n_s] \leftarrow nfmax$
  si (non MAX-GSAT-ARRAY-ME $(E,\ k)$) **alors**
    $nfmin \leftarrow 0$
    $k[n_s] \leftarrow 0$
    tant que $((nfmax$ - $nfmin) > 1)$ **faire**
      $k[n_s] \leftarrow (nfmax + nfmin)$ div 2 *(* div = division entière *)*
      si MAX-GSAT-ARRAY-ME $(E,\ k)$ **alors**
        $nfmin \leftarrow k[n_s]$
      sinon
        $nfmax \leftarrow k[n_s]$

si NGAT-LEX-ME (E,H,k) **alors**
  $échec \leftarrow vrai$
sinon
  $échec \leftarrow faux$

vérifier que l'on n'a pas échec ($échec = faux$)
fin

Algorithme 4.41 : MAX-GSAT-ARRAY-ME $((E,<),\ k)$
début

deviner une interprétation M
$i \leftarrow 1$ *(* strate la plus prioritaire *)*
$\acute{e}chec \leftarrow faux$
tant que $i \leq n$ (nombre de strates de E) et (non $\acute{e}chec$) **faire**
  $p \leftarrow 0$
  pour chaque formule G de $(E,<)$ **faire**
    si M satisfait G **alors**
      $p \leftarrow p + p_G$ *(* p_G = poids de G *)*
  si $p \geq k[i]$ **alors**
    $i \leftarrow i + 1$
  sinon
    $\acute{e}chec \leftarrow vrai$
vérifier que l'on n'a pas échec ($\acute{e}chec = faux$)

fin

Algorithme 4.42 : NGSAT-LEX-ME $((E,<),\ H,\ k)$
début

deviner une interprétation M
(vérifier qu'elle satisfait $k[i]$ formules de chaque strate i de E *)*
$k' \leftarrow <0, 0, \ldots, 0>$ *(* k' de même dimension n que k *)*
pour chaque strate i de 1 à n **faire**
  pour chaque formule G de la strate i **faire**
    si M satisfait G **alors**
      $k'[i] \leftarrow k'[i] + p_G$ *(* p_G = poids de G *)*
vérifier que $k' = k$

vérifier que M ne satisfait pas H

fin

Algorithme 4.43 : EXI-LEX-ME $((E,<),\ H)$
début

deviner un sous-ensemble Y de $(E,<)$
(vérifier que Y est une thèse préférée lexicographiquement de $(E,<)$ *)*
$\acute{e}chec \leftarrow faux$
$i \leftarrow 1$ *(* strate la plus prioritaire *)*
tant que non $\acute{e}chec$ et $i < n + 1$ **faire**
  $k \leftarrow |Y_1|^* + |Y_2|^* + \ldots + |Y_i|^*$
  si il existe une sous-base consistante Z de $E_1 \cup E_2 \cup \ldots \cup E_i$ telle que $|Z|^* > k$ **alors**
    $\acute{e}chec \leftarrow vrai$
  sinon
    $i \leftarrow i + 1$
si non $\acute{e}chec$ **alors**
  vérifier que Y infère H

fin

appelés MAX-GSAT, NGSAT-CAR (pour UNI-CAR) ou MAX-GSAT-STRICT (pour EXI-CAR). Le problème d'inférence classique dans le cas de conjonctions de clauses de Horn devient polynomial. Par contre, on doit étudier les problèmes MAX-GSAT, NGSAT-CAR et MAX-GSAT-STRICT dans le cas de conjonctions de clauses de Horn (problèmes notés alors MAX-SAT-HORN, NSAT-CAR-HORN et MAX-SAT-STRICT-HORN)[26].

- Les algorithmes de MAX-SAT-HORN, NSAT-CAR-HORN et MAX-SAT-STRICT-HORN sont les mêmes que ceux de MAX-GSAT, NGSAT-CAR et MAX-GSAT-STRICT (voir les démonstrations des théorèmes 4.1.13 page 90 et 4.1.14 page 94). Les problèmes MAX-SAT-HORN, NSAT-CAR-HORN et MAX-SAT-STRICT-HORN appartiennent donc à la classe de complexité NP. Le résultat est le même pour le cas des multi-ensembles : MAX-SAT-ME-HORN, NSAT-CAR-ME-HORN et MAX-SAT-STRICT-ME-HORN appartiennent donc à la classe de complexité NP.
- Complétude de MAX-SAT-HORN : on ne peut plus utiliser la même transformation que pour MAX-GSAT puisque SAT devient polynomial quand il s'agit de clauses de Horn. J'utilise donc un autre problème NP-complet MAX-2-SAT : "Soit C collection de n clauses à 2 littéraux maximum, soit un entier $k \leq n$, existe-t-il une assignation M satisfaisant au moins k clauses prises dans C ?" (voir [GJ79]). Remarque : Si $k = n$ ou k fixe ($k = n - k_0$, avec k_0 une constante) alors MAX-2-SAT devient polynomial.
  Mais comme ce problème est défini sur une collection de clauses et non sur un ensemble de clauses, je vais donc démontrer la complétude du problème MAX-SAT-ME-HORN au lieu de démontrer celle de MAX-SAT-HORN.

Soit une collection C de n clauses à 2 littéraux maximum, décomposons C en : $CH = \{$les clauses de C qui sont déjà sous forme de clauses de Horn$\}$ et $C \setminus CH$ et posons : $E = CH \cup CNH$ avec : $CNH = \{$les clauses définies de la manière suivante : $\forall c$ clause de C de la forme $\rightarrow a_1a_2$ ($c \in C \setminus CH$), les clauses de Horn $\rightarrow a_1$, $\rightarrow a_2$, $a_1a_2 \rightarrow\ \in CNH\}$.

  Remarquons que les clauses $\rightarrow a_1$, $\rightarrow a_2$, $a_1a_2 \rightarrow$ ainsi générées peuvent ne pas être uniques. C'est donc ici que l'utilisation d'un multi-ensemble prend tout son sens. Bien que E ne soit pas un multi-ensemble tel que je l'ai défini, il est bien évidemment très simple de compter le nombre d'occurrences de chaque formule de E (ce qui se fait en un temps polynomial) et de créer le multi-ensemble correspondant à E.
  Soit p le nombre de clauses de C qui ne sont pas des clauses de Horn ($p = |C \setminus CH|$), soit M une assignation des variables apparaissant dans C, on a les propriétés suivantes :
  - Soit c clause de C sous la forme $\rightarrow a_1a_2$, M satisfait $c \Leftrightarrow M$ satisfait exactement 2 clauses de l'ensemble $\{\rightarrow a_1, \rightarrow a_2, a_1a_2 \rightarrow\}$.
  - Soit c clause de C sous la forme $\rightarrow a_1a_2$, M ne satisfait pas $c \Leftrightarrow M$ satisfait exactement 1 clause de l'ensemble $\{\rightarrow a_1, \rightarrow a_2, a_1a_2 \rightarrow\}$ (c'est la clause $a_1a_2 \rightarrow$).

  Soit M une assignation des variables de C, définissons k_M = nombre de clauses de C satisfaites par M. On a alors : $k_M = k_{0_M} + k_{1_M}$ avec k_{0_M} = nombre de clauses de Horn de C qui sont satisfaites par M et k_{1_M} = nombre de clauses de C sous forme de $\rightarrow a_1a_2$ qui sont satisfaites par M. Calculons k''_M = nombre de clauses de E satisfaites par M :

  $$k''_M = k_{0_M} + 2k_{1_M} + (p - k_{1_M}) = k_{0_M} + k_{1_M} + p = k_M + p$$

[26] Les démonstrations données s'appliquent aussi au cas d'une base de Horn stricte.

Soit k entier $\leq n$ (nombre de clauses de C), posons :
$f(k) = k' = k + p$

Montrons alors que MAX-2-SAT $\propto$ MAX-SAT-ME-HORN :

$$\text{MAX-2-SAT } (C,k)$$
$$\Leftrightarrow$$
$$\exists M \text{ assignation satisfaisant au moins } k \text{ clauses de } C\text{, donc } k_M \geq k$$
$$\Leftrightarrow$$
$$\exists M \text{ assignation satisfaisant au moins } k+p \text{ clauses de } E\text{,}$$
$$\text{puisque } k''_M = k_M + p \geq k + p$$
$$\Leftrightarrow$$
$$\exists Y \text{ sous-base consistante de } E \text{ telle que } |Y| \geq k + p$$
$$\Leftrightarrow$$
$$\exists Y \text{ sous-base consistante de } E \text{ telle que } |Y| \geq k' = f(k)$$
$$\Leftrightarrow$$
$$\text{MAX-SAT-ME-HORN } (E,f(k))$$

Le problème MAX-SAT-ME-HORN est donc NP-complet.

- Complétude du problème NSAT-CAR-HORN : il est inutile de prouver la complétude de ce problème puisqu'il ne sert d'oracle que pour un algorithme dans lequel on utilise aussi l'oracle MAX-SAT-HORN dont on vient de montrer que la version multi-ensemble est NP-complète. Et comme on sait que NSAT-CAR-HORN appartient à la classe de complexité NP, il se transforme donc polynomialement en MAX-SAT-ME-HORN. L'algorithme utilisant ces oracles est donc un algorithme faisant appel à un oracle non déterministe polynomial.
- Complétude du problème MAX-SAT-STRICT-HORN : on peut prouver que sa version multi-ensemble est NP-complète en utilisant la transformation suivante.

Soit "E, k" une instance de MAX-SAT-ME-HORN, posons :
- $f(E) = E$ (donc une seule strate),
- $f(k) = k - 1$.

On a alors :

il existe une interprétation satisfaisant au moins k formules de E
$$\Leftrightarrow$$
il existe une interprétation satisfaisant strictement plus de $k-1$ formules de E.

On a donc MAX-SAT-ME-HORN $\propto$ MAX-SAT-STRICT-ME-HORN, et MAX-SAT-STRICT-ME-HORN est NP-complet.

J'en conclus que les problèmes d'inférence utilisant l'ordre basé sur la cardinalité n'ont donc pas changé de complexité dans le cas de multi-ensembles :

- UNI-CAR-ME-HORN appartient au plus à la classe de complexité Δ_2^p,
- EXI-CAR-ME-HORN appartient au plus à la classe de complexité Σ_2^p,
- ARG-CAR-ME-HORN appartient au plus à la classe de complexité Δ_3^p.

Or le cas des ensembles étant une restriction du cas des multi-ensembles, je peux aussi conclure que :

- UNI-CAR-HORN appartient au plus à la classe de complexité Δ_2^p,
- EXI-CAR-HORN appartient au plus à la classe de complexité Σ_2^p,
- ARG-CAR-HORN appartient au plus à la classe de complexité Δ_3^p.

Étude de complétude.

- pour UNI-CAR-HORN, on ne peut pas garder la démonstration du cas général (voir théorème 4.1.13 page 90), puisque la transformation utilisée ne génère pas des clauses de Horn. Je ne conserve alors qu'un résultat d'appartenance.

Par contre, il existe une transformation polynomiale permettant de passer du problème UNI-LEX-HORN au problème UNI-CAR-ME-HORN.

> Soit "$(E,<), H$" une instance de UNI-LEX-HORN, posons :
> - $f(H) = H$,
> - $f(E)$ = le multi-ensemble produit par l'algorithme 4.44[a].
>
> [a] Cet algorithme est issu d'un algorithme présenté dans [DLS94].

Algorithme 4.44 : Construction de $f(E)$

début

$f(E) \leftarrow \{(\Phi_i, 1), \Phi_i \in E_n\}$ *(* E_n = strate la moins prioritaire *)*

$p \leftarrow 1$

pour chaque strate E_i de $(E,<)$, $i = n-1, \ldots, 1$ **faire**

$p \leftarrow p \times (1 + |E_{i+1}|)$

$f(E) \leftarrow f(E) \cup \{(\Phi_i, p)$ pour toutes les formules $\Phi_i \in E_i\}$

fin

Remarques :

- L'algorithme permettant le calcul de $f(E)$ est un algorithme polynomial.
- $f(E)$ est un multi-ensemble non stratifié.
- À chaque strate i de $(E,<)$ correspond un poids unique p_i. En conséquence, toutes les formules de cette strate ont le même poids dans $f(E)$.
- Quand on applique cet algorithme à un sous-ensemble de E, les poids de chaque strate sont toujours calculés d'après $(E,<)$.

Propriété 4.1.26 *Y est une thèse préférée lexicographiquement de $(E,<)$ ssi $f(Y)$ est une thèse préférée de $f(E)$ pour l'ordre basé sur la cardinalité.*

Preuve : La démonstration se fait en 3 étapes :

- montrer qu'*à tout sous-ensemble de $(E,<)$ correspond par la fonction f un sous-ensemble de $f(E)$, et vice-versa* :
  c'est évident, d'après l'algorithme donné, sachant que les poids de chaque strate sont fixés en fonction de la base $(E,<)$;
- montrer que *si Y est une thèse préférée lexicographiquement de $(E,<)$ alors $f(Y)$ est une thèse préférée pour l'ordre basé sur la cardinalité de $f(E)$* :
  soit $(E,<) = E_1 \cup \ldots \cup E_n$ une base stratifiée, soit A un sous-ensemble de $(E,<)$, d'après l'algorithme de construction de $f(E)$, le cardinal de $f(A)$ se calcule de la façon suivante :

$$\begin{aligned}
|f(A)|^* &= |f(A_1)|^* + |f(A_2)|^* + \ldots + |f(A_n)|^* \\
&= (p_1 \times |A_1|) + (p_2 \times |A_2|) + \ldots + (p_n \times |A_n|) \\
&= [\prod_{j=n-1}^{1} (1 + |E_{j+1}|)] \times |A_1| \\
&\quad + [\prod_{j=n-1}^{2} (1 + |E_{j+1}|)] \times |A_2| \\
&\quad \vdots \\
&\quad + [\prod_{j=n-1}^{n-1} (1 + |E_{j+1}|)] \times |A_{n-1}| \\
&\quad + |A_n|
\end{aligned}$$

soient A et B deux sous-ensembles de $(E,<)$ tels que A est préféré à B pour l'ordre lexicographique sur $(E,<)$, on a alors par définition : il existe i un numéro de strate de $(E,<)$ tel que $|A_i| > |B_i|$ et $\forall j < i$, $|A_j| = |B_j|$;
ainsi, on peut dire que, pour cette strate i, $|A_i| = |B_i| + x$ où $x \geq 1$;
d'autre part, on a toujours : $\forall k$, en particulier pour $k > i$, $|B_k| - |A_k| \leq |E_k|$;
calculons alors la différence $|f(A)|^* - |f(B)|^*$:

$$\begin{aligned}
|f(A)|^* - |f(B)|^* &= [\prod_{j=n-1}^{i}(1+|E_{j+1}|)] \times (|B_i| + x - |B_i|)\\
&+[\prod_{j=n-1}^{i+1}(1+|E_{j+1}|)] \times (|A_{i+1}| - |B_{i+1}|)\\
&\vdots\\
&+[\prod_{j=n-1}^{n-1}(1+|E_{j+1}|)] \times (|A_{n-1}| - |B_{n-1}|)\\
&+(|A_n| - |B_n|)
\end{aligned}$$

ce qui implique que :

$$\begin{aligned}
|f(A)|^* - |f(B)|^* &\geq [\prod_{j=n-1}^{i}(1+|E_{j+1}|)]\\
&-[\prod_{j=n-1}^{i+1}(1+|E_{j+1}|)] \times |E_{i+1}|\\
&\vdots\\
&-[\prod_{j=n-1}^{n-1}(1+|E_{j+1}|)] \times |E_{n-1}|\\
&-|E_n|\\
|f(A)|^* - |f(B)|^* &\geq 1 + |E_n| - |E_n|\\
|f(A)|^* - |f(B)|^* &\geq 1
\end{aligned}$$

donc $f(A)$ est préféré pour la cardinalité à $f(B)$.

- montrer que *si $f(Y)$ est une thèse préférée pour l'ordre basé sur la cardinalité de $f(E)$ alors Y est une thèse préférée lexicographiquement de* $(E,<)$:
  soit $f(A)$ une thèse préférée pour l'ordre basé sur la cardinalité de $f(E)$, raisonnons par l'absurde en supposant que A n'est pas une thèse préférée lexicographiquement de $(E,<)$;
  alors il existe B thèse préférée lexicographiquement à A; on sait alors que $f(B)$ est préférée pour la cardinalité à $f(A)$ (voir démonstration précédente) ;
  on a donc une contradiction avec l'hypothèse initiale disant que $f(A)$ est une thèse préférée pour l'ordre basé sur la cardinalité de $f(E)$;
  on en déduit ainsi que A est une thèse préférée lexicographiquement de $(E,<)$.

□

Grâce à cette propriété, on prouve que :
UNI-LEX-HORN $(E, H) \Leftrightarrow$ UNI-CAR-ME-HORN $(f(E), f(H))$,
ce qui implique que UNI-LEX-HORN $\propto$ UNI-CAR-ME-HORN.
Or on démontre (voir le théorème 4.1.38) que UNI-LEX-HORN est Δ_2^p-complet.
Ce qui me permet de conclure que UNI-CAR-ME-HORN est Δ_2^p-complet.

- pour EXI-CAR-HORN, on ne peut pas garder la même démonstration que pour EXI-CAR (voir théorème 4.1.14 page 94). Toutefois, je n'ai pas pu trouver une transformation polynomiale d'un problème Σ_2^p-complet en EXI-CAR-HORN. Je ne conserve alors qu'un résultat d'appartenance.
- pour ARG-CAR-HORN, on retrouve les mêmes difficultés que celles rencontrées pour ARG-CAR (voir théorème 4.1.15 page 96) alourdies encore par le fait que je ne sais rien sur la complétude de EXI-CAR-HORN. Donc, comme pour EXI-CAR-HORN, je ne conserve qu'un résultat d'appartenance.

□

Théorème 4.1.38 (Complexité de UNI (EXI, ARG)-LEX-HORN)

- UNI-LEX-HORN *est* Δ_2^p*-complet*
- EXI-LEX-HORN $\in \Sigma_2^p$
- ARG-LEX-HORN $\in \Delta_3^p$

Preuve : Étude d'appartenance.

L'utilisation de l'ordre lexicographique sur une base stratifiée de Horn présente les mêmes particularités que l'utilisation de l'ordre basé sur la cardinalité dans une base sans strate. J'ai déjà étudié l'oracle MAX-GSAT-STRICT dans le cadre de UNI (EXI, ARG)-CAR-HORN, il reste à étudier les oracles appelés MAX-SAT-ARRAY-HORN et NSAT-LEX-HORN[27].

- Les algorithmes de MAX-SAT-ARRAY-HORN et de NSAT-LEX-HORN sont les mêmes que ceux de MAX-GSAT-ARRAY et de NGSAT-LEX (voir la démonstration du théorème 4.1.16 page 97). Les problèmes MAX-SAT-ARRAY-HORN et NSAT-LEX-HORN appartiennent donc à la classe de complexité NP. Le résultat est le même pour le cas des multi-ensembles : MAX-SAT-ARRAY-ME-HORN et NSAT-LEX-ME-HORN appartiennent donc à la classe de complexité NP.
- Complétude du problème MAX-SAT-ARRAY-HORN : on peut utiliser la même transformation que dans le cas général mais uniquement sur les multi-ensembles[28], puisque MAX-SAT-ME-HORN est toujours une restriction de MAX-SAT-ARRAY-ME-HORN. On a donc MAX-SAT-ME-HORN $\propto$ MAX-SAT-ARRAY-ME-HORN, donc MAX-SAT-ARRAY-ME-HORN est NP-complet.
- Complétude du problème NSAT-LEX-HORN : comme pour NSAT-CAR-HORN dans UNI-CAR-HORN, il est inutile de montrer la complétude de ce problème, puisqu'il est toujours utilisé en collaboration avec l'oracle MAX-SAT-ARRAY-HORN dont la version multi-ensemble est NP-complète.

J'en conclus que les problèmes d'inférence utilisant l'ordre lexicographique n'ont donc pas changé de complexité dans le cas des multi-ensembles :

- UNI-LEX-ME-HORN appartient au plus à la classe de complexité Δ_2^p,

[27] Ici aussi, comme pour MAX-SAT-HORN, NSAT-CAR-HORN et MAX-SAT-STRICT-HORN (voir preuve du théorème 4.1.37 page 130), les démonstrations données s'appliquent au cas d'une base de Horn stricte.
[28] On ne sait rien sur la complétude de MAX-SAT-HORN.

- EXI-LEX-ME-HORN appartient au plus à la classe de complexité Σ_2^p,
- ARG-LEX-ME-HORN appartient au plus à la classe de complexité Δ_3^p.

Or le cas des ensembles étant une restriction du cas des multi-ensembles, je peux aussi conclure que :

- UNI-LEX-HORN appartient au plus à la classe de complexité Δ_2^p,
- EXI-LEX-HORN appartient au plus à la classe de complexité Σ_2^p,
- ARG-LEX-HORN appartient au plus à la classe de complexité Δ_3^p.

Étude de complétude.

- pour UNI-LEX-HORN, je ne peux pas utiliser les démonstrations du cas général (voir théorème 4.1.16 page 97), puisque je ne sais rien sur UNI-CAR-HORN et que la transformation à partir du problème ALM ne génère pas de clauses de Horn. Essayons donc avec d'autres problèmes Δ_2^p-complets. Montrons, par exemple, que ACM[29] $\propto$ UNI-LEX-HORN.

  > Soit "$C = \{C_1, \ldots, C_m\}$ un ensemble de clauses ($X = \{x_1, \ldots, x_n\}$ les variables de C), $k = \{1, \ldots, m\}$ un entier" une instance de ACM, posons :
  > - $H = C_k[y] \wedge s$.
  > - $(E, <) =$
  > $\{p_1, \ldots, p_m\} \cup \{x_1, \ldots, x_n, y_1, \ldots, y_n\} \cup \{\neg z_1, \ldots, \neg z_n, \neg s\}$ avec $y_1, \ldots, y_n$, $z_1, \ldots, z_n$, s nouveaux symboles de variables propositionnelles, et avec les formules
  > $p_j = (z_1 \ldots z_n \rightarrow s) \wedge \bigwedge_{i=1\ldots n}((\neg x_i \vee \neg y_i) \wedge (y_i \rightarrow z_i) \wedge (x_i \rightarrow z_i)) \wedge C_j[y]$,
  > avec la notation suivante : $C_j[y]$ = la clause C_j dans laquelle on remplace tous les littéraux positifs x_i par des littéraux négatifs $\neg y_i$.
  > L'ordre $<$ entre les formules de la base est le suivant : les p_j plus prioritaires que les x_i et les y_i, eux-mêmes plus prioritaires que les $\neg z_i$ et le $\neg s$.

  Notons $Y = \{y_1, \ldots, y_n\}$.
  La plupart des remarques préliminaires faites pour UNI-T-HORN restent valables ici. On a alors les propriétés suivantes :

  Propriété 4.1.27 *Si la valeur de vérité de x_i est différente de la valeur de vérité de y_i $\forall i$ alors C sera satisfaite ssi $C[y]$ est satisfaite.*

  $\forall V$ assignation de X, notons $D(V)$ l'ensemble des clauses de C satisfaites par V. De même, $\forall V'$ assignation de $X \cup Y$, notons $D(V')$ l'ensemble des clauses de $C[y]$ satisfaites par V'.

  Propriété 4.1.28 *Si, quel que soit i la valeur de vérité de x_i est différente de la valeur de vérité de y_i alors :*
  il existe une assignation V de X card-maximale pour C satisfaisant l'ensemble de clauses $D(V)$ ssi il existe une assignation V' de $X \cup Y$ card-maximale pour $C[y]$ satisfaisant $D(V)[y]$

  Preuve : Toute la preuve repose sur la propriété 4.1.27. L'hypothèse de base est que la valeur de vérité de x_i est différente de la valeur de vérité de y_i, $\forall i$.

  - Il existe une assignation V de X satisfaisant l'ensemble de clauses $D(V) \subseteq C \Leftrightarrow$ (à cause de la propriété 4.1.27 et de l'hypothèse de

[29] Ce problème est issu de [EG93] et a déjà été utilisé pour la preuve du théorème 4.1.13 page 90.

base) il existe une assignation V' satisfaisant l'ensemble $D(V)[y] \subseteq C[y]$ correspondant à $D(V)$. Remarquons au passage que $D(V') = D(V)[y]$ et que $|D(V)| = |D(V)[y]|$.

- V est une assignation de X card-maximale pour $C \Leftrightarrow$ il n'existe pas d'assignation W de X telle que $|D(V)| < |D(W)|$.
  Construisons l'assignation V' telle que $V'(x_i) = V(x_i)\ \forall x_i \in X$ et $V'(y_i) \neq V(x_i)\ \forall y_i \in Y$, on a ainsi une assignation V' de $X \cup Y$ satisfaisant $D(V)[y]$. Raisonnons par l'absurde et supposons que l'assignation V' n'est pas card-maximale. Il existe alors une autre assignation de $X \cup Y$ notée W' qui est card-maximale pour $C[y]$, on a donc $|D(V')| < |D(W')|$. Donc, l'assignation W correspondant à W' (c'est à dire la restriction de W' à X) aura la propriété suivante : $|D(V)| < |D(W)|$, ce qui est en contradiction avec l'hypothèse V card-maximale.
  De la même façon, on prouve que si V' est une assignation de $X \cup Y$ card-maximale pour $C[y]$ alors l'assignation V de X correspondante est card-maximale pour C.

□

De plus :

- Si on calcule les thèses préférées lexicographiquement T de $(E, <)$, on trouve que : s'il existe une C_j satisfiable alors il existe une thèse T telle que $T \vdash z_i$ $(\forall i)$ et $T \vdash s$.
- Les seules thèses préférées lexicographiquement inférant s sont les sous-bases contenant le plus de formules p_j, puisque la variable s n'apparaît que dans les p_j.
- D'autre part, j'ai toujours les propriétés utilisées pour la démonstration de complétude de Uni-Car.

Prouvons maintenant que Acm $\propto$ Uni-Lex-Horn.

- Acm implique Uni-Lex-Horn :

  Toute assignation V_c de X card-maximale pour C vérifie : $V_c(C_k) = vrai.$
  Étendons chaque V_c à Y de telle façon que $V_c(x_i) \neq V_c(y_i)$.
  $\Rightarrow$
  $\forall T$ thèse préférée lexicographiquement de E :

  - T est constituée de 3 strates T_1, T_2 et T_3 :
    - T_1 contient le maximum de p_j, T_1 est maximale pour la cardinalité dans E_1 (première strate de E),
    - T_2 contient des x_i et des y_i (exactement n en tout puisque à cause des formules p_j, on doit avoir la valeur de x_i différente de la valeur de y_i $\forall i$),
    - T_3 est vide (puisque à cause des formules p_j, on doit avoir tous les z_i et s) ;

    donc T infère s.
  - $\exists V'$ une assignation de $X \cup Y$ telle que l'ensemble des clauses satisfaites par V' est T_1 (cf. propriété 4.1.7 page 93). V' satisfait les formules p_j de T_1, donc les formules $C_j[y]$ correspondantes.

  $\Rightarrow$
  V' est card-maximale pour E_1 (cf. propriété 4.1.9 page 93) et pour $C[y]$
  $\Rightarrow$
  $\exists V$ assignation de X card-maximale pour C (cf. propriété 4.1.28 page ci-contre). Or toutes les assignations card-maximales pour C satisfont C_k.

$\Rightarrow$

V' satisfait $C_k[y]$ (cf. propriété 4.1.28 page 138)

$\Rightarrow$

T_1 contient la formule p_k

$\Rightarrow$

T_1 infère $C_k[y]$

$\Rightarrow$

toutes les thèses préférées lexicographiquement T de $(E, <)$ infèrent s et $C_k[y]$, donc infèrent $C_k[y] \wedge s$.

- Uni-Lex-Horn implique Acm :

  quelle que soit T thèse préférée lexicographiquement de $(E, <)$, T infère $C_k[y] \wedge s$, donc T infère $C_k[y]$ et s.

  $\Rightarrow$

  $\forall T$, T_1 (première strate de T) contient la formule p_k (puisque $T \vdash C_k[y]$) et toute assignation de $X \cup Y$ satisfaisant T vérifie le fait que la valeur de x_i est différente de la valeur de y_i $\forall i$ (puisque $T \vdash s$).

  $\Rightarrow$

  $\forall V_c$ assignation de X card-maximale pour C, il existe V' une assignation de $X \cup Y$ card-maximale pour $C[y]$ (cf. propriété 4.1.28 page 138)

  $\Rightarrow$

  $D(V')$ est un sous-ensemble maximal pour la cardinalité pour $C[y]$ (cf. propriété 4.1.9 page 93)

  $\Rightarrow$

  l'ensemble $P_{D(V')}$ des formules p_j correspondant à $D(V')$ est maximal pour la cardinalité pour E_1 (première strate de E)

  $\Rightarrow$

  $P_{D(V')}$ est un des T_1 (première strate des T préférées lexicographiquement)

  $\Rightarrow$

  $P_{D(V')}$ contient p_k

  $\Rightarrow$

  $D(V')$ contient $C_k[y]$

  $\Rightarrow$

  V' satisfait $C_k[y]$

  $\Rightarrow$

  Toute assignation V_c card-maximale pour C vérifie : $V_c(C_k) = vrai$

  Le problème Acm se ramène donc polynomialement à Uni-Lex-Horn. On en déduit que Uni-Lex-Horn est Δ_2^p-complet.

- pour Exi-Lex-Horn, je ne peux pas utiliser la démonstration du cas général (voir théorème 4.1.17 page 103), puisque je ne sais rien sur Exi-Car-Horn. Toutefois, je n'ai pas pu trouver une autre transformation polynomiale d'un problème Σ_2^p-complet en Exi-Lex-Horn. Je ne conserve alors qu'un résultat d'appartenance.
- pour Arg-Lex-Horn, je ne peux pas utiliser la démonstration du cas général (voir théorème 4.1.18 page 105), puisque je ne sais rien sur Arg-Car-Horn. Donc, comme pour Exi-Lex-Horn, je ne conserve qu'un résultat d'appartenance.

□

Définition 4.1.20 *Une théorie des défauts de Horn peut être définie de différentes façons. Voici deux définitions possibles :*

- définition de Stillman donnée dans [Sti90]
  - *l'ensemble des prémisses W est un ensemble de clauses de Horn,*

- *l'ensemble des défauts D est un ensemble de défauts de la forme $a : b/b$ avec a un unique littéral positif, et b un unique littéral (positif ou négatif),*
- *la formule à inférer est un littéral (positif ou négatif).*

- définition de Kautz et Selman donnée dans [KS91]
  - *l'ensemble des prémisses W est un ensemble de littéraux,*
  - *l'ensemble des défauts D est un ensemble de défauts de la forme $a_1, \ldots, an : b/b$ avec, $\forall i$, a_i littéral positif, et b un unique littéral (positif ou négatif)*[30]*,*
  - *la formule à inférer est un littéral (positif ou négatif).*

Théorème 4.1.39 (Complexité de UNI (EXI, ARG)-E-HORN)

Suivant la définition choisie pour une théorie des défauts de Horn, on obtient des résultats différents :

- Avec la définition de Stillman (voir [Sti90]) :
  - *pas d'étude de* UNI-E-HORN
  - EXI-E-HORN *est* NP-*complet*
  - ARG-E-HORN $\in \Delta_2^p$ *et est co*-NP-*difficile*
- Avec la définition de Kautz et Selman (voir [KS91]) :
  - UNI-E-HORN *est co*-NP-*complet*
  - EXI-E-HORN $\in$ P
  - ARG-E-HORN $\in$ P

Preuve : Étude d'appartenance et de complétude.

Parmi ces trois relations d'inférence, deux ont déjà été étudiées par Kautz et Selman dans [KS91] et par Stillman dans [Sti90]. Il s'agit de UNI-E-HORN et EXI-E-HORN. Je me contenterai donc de citer leurs résultats. Par contre, il n'y a eu à ma connaissance aucune étude sur ARG-E-HORN.

Signalons avant tout que la définition de Stillman est différente de celle de Kautz et Selman (voir définition 4.1.20 page ci-contre). Ils n'obtiennent donc pas les mêmes résultats[31].

Le résultat obtenu par Stillman est le suivant :

- EXI-E-HORN est NP-complet (preuve de Stillman dans [Sti90]).

Les résultats obtenus par Kautz et Selman sont les suivants :

- UNI-E-HORN est co-NP-complet (preuve de Kautz et Selman dans [KS91]).
- EXI-E-HORN est polynomial (preuve de Kautz et Selman dans [KS91]).

Ni Stillman, ni Kautz et Selman n'ont étudié ARG-E-HORN. Peut-on conserver la démonstration faite pour ARG-E (voir théorème 4.1.23 page 106) ?

- Oui, pour la preuve d'appartenance puisque l'algorithme utilisé reste inchangé avec un oracle (EXI-E-HORN) qui a diminué en complexité. Suivant la définition utilisée, ARG-E-HORN appartient donc :
  - soit à Δ_2^p (définition de Stillman),
  - soit à P (définition de Kautz et Selman).

[30] Il s'agit donc, dans les 2 cas, d'une sous-classe des défauts normaux.

[31] Il est possible de donner encore d'autres définitions de $E = (W, D)$ et de la formule à inférer dans le cas des clauses de Horn et on obtiendrait sûrement d'autres résultats.

- Non, pour la preuve de complétude. Du point de vue de Kautz et Selman, le problème de la complétude ne se pose plus puisque ARG-E-HORN est de complexité P. Et du point de vue de Stillman, je rencontre la difficulté suivante : une des transformations polynomiales effectuées dans le cadre de la relation ARG-E ne permet pas de respecter les contraintes imposées par Stillman (voir tableau ci-dessous).

Définition de Stillman	
Transformation polynomiale de EXI-E (E, H) à ARG-E $(f(E), f(H))$: - $f(H) = H$, - $f(E) = E$ avec ajout de $H \rightarrow G$ aux prémisses (G nouveau symbole propositionnel)	Soient H un littéral quelconque et G un littéral positif, $H \rightarrow G$ *n'est pas une conjonction de clauses de Horn.* (excepté si H est un littéral positif) Transformation non utilisable
Transformation polynomiale de co-EXI-E (E, H) à ARG-E $(f(E), f(H))$: - $f(E) = E$ avec ajout de $:\neg H/\neg H$ aux défauts - $f(H) = \neg H$	Soit H un littéral quelconque, le défaut $:\neg H/\neg H$ *est de la forme* $a : b/b$ *avec* b *un littéral unique.* (que H soit un littéral positif ou négatif) Transformation utilisable

Je peux donc toujours montrer que co-EXI-E-HORN $\propto$ ARG-E-HORN du point de vue de Stillman, donc que ARG-E-HORN est co-NP-difficile.
Par contre, je ne peux plus montrer que EXI-E-HORN $\propto$ ARG-E-HORN. La seule possibilité restante est d'imposer que H soit un littéral positif. Ainsi, je conserve la transformation polynomiale définie pour ARG-E et je montre que EXI-E-HORN-POSITIF $\propto$ ARG-E-HORN avec EXI-E-HORN-POSITIF problème correspondant à EXI-E-HORN en imposant que la formule inférée soit uniquement un littéral positif.
Or, on ne connaît pas la complexité de ce problème (on sait seulement qu'au pire EXI-E-HORN-POSITIF sera NP-complet puisque c'est une restriction de EXI-E-HORN). Il faudrait donc mener une étude semblable à celle de Stillman pour calculer la complexité de EXI-E-HORN-POSITIF et l'utiliser afin de calculer celle de ARG-E-HORN, ce que je ne ferai pas ici, puisque la logique des défauts n'est pas au centre de mon étude.

□

Remarques sur le raisonnement à partir de la logique des défauts : Il existe de nombreux autres cas particuliers concernant la logique des défauts (voir [Sti90] et [KS91]).

Dans [KS91], Kautz et Selman ont proposé, à partir d'un ensemble W de littéraux, toute une palette de possibilités pour l'ensemble D. Ainsi, nous venons de voir une définition portant sur l'utilisation des défauts de Horn qui permet d'obtenir un algorithme polynomial pour EXI-E-HORN.

Ils ont aussi proposé une théorie des défauts unaire et normale, c'est-à-dire dont les défauts sont de la forme $a : b/b$ avec a un unique littéral positif, et b un unique littéral (positif ou négatif), ce qui est un cas particulier des défauts de Horn. Dans ce cas là, ils ont fourni des algorithmes polynomiaux pour UNI-E et pour EXI-E. Les autres possibilités proposées ne permettent pas de réduire la complexité en deçà de la classe NP.

La hiérarchie de Kautz et Selman a été reprise par Stillman dans [Sti90] en étendant les choix possibles pour W et en rajoutant certains choix pour l'ensemble D. Nous venons de voir par exemple l'utilisation de clauses de Horn pour W avec des défauts unaires et normaux.

Stillman démontre aussi que le processus d'inférence Exi-E est polynomial dans le cas de défauts unaires, normaux et sans pré-requis et avec un ensemble W ne contenant que des clauses avec 2 littéraux au plus (rappelons que le problème de satisfaction pour des clauses de 2 littéraux au plus, appelé 2-Sat, est polynomial – voir [GJ79]).

Il serait intéressant de voir la correspondance de tous ces cas particuliers de théories des défauts dans le cas d'une base de croyances propositionnelle classique et de calculer la complexité des processus d'inférence associés. Il est probable que l'on trouverait aussi des algorithmes polynomiaux.

4.1.3.4 Cas d'une base de Horn stratifiée strictement

Dans ce dernier cas, on "mixe" le premier et le troisième cas particuliers étudiés ici. Ainsi, on espère cumuler les gains en complexité de ces deux méthodes.

Théorème 4.1.40 (Complexité de Uni (Exi, Arg)-Bo) *Dans le cas d'une base de Horn stratifiée strictement,* Uni-Bo *(respectivement* Exi-Bo, Arg-Bo*) est noté 1/*Strate-Uni-Bo-Horn *(respectivement 1/*Strate-Exi-Bo-Horn*, 1/*Strate-Arg-Bo-Horn*). Les complexités sont :*

- *1/*Strate-Uni-Bo-Horn $\in$ P
- *1/*Strate-Exi-Bo-Horn *est* NP*-complet*
- *si* NP $\neq$ *co*-NP*, 1/*Strate-Arg-Bo-Horn $\in \Delta_2^p$ - *(*NP $\cup$ *co*-NP*)*

Preuve : Étude d'appartenance.

Ici, comme dans le cas général portant sur une seule formule par strate, et contrairement à ce qui se passe pour les méthodes "Inclusion Based" et lexicographique (voir propriété 4.1.22 page 110), il n'y a pas de simplification des algorithmes (voir la démonstration du théorème 4.1.25 page 112). On garde donc les mêmes résultats d'appartenance que ceux donnés dans le cas de Horn :

- 1/Strate-Uni-Bo-Horn appartient au plus à la classe de complexité P,
- 1/Strate-Exi-Bo-Horn appartient au plus à la classe de complexité NP,
- 1/Strate-Arg-Bo-Horn appartient au plus à la classe de complexité Δ_2^p.

Étude de complétude.

- Pour 1/Strate-Exi-Bo-Horn, la démonstration du cas Horn (voir théorème 4.1.35 page 124) ne s'applique plus puisque la base E proposée n'est pas strictement stratifiée. Je propose alors une nouvelle transformation polynomiale :

Soit "E, H", une instance de Exi-Bo-Horn, avec $(E, <) = (\{\Phi_1^1, \ldots, \Phi_{k_1}^1\}, \ldots, \{\Phi_1^n, \ldots, \Phi_{k_n}^n\})$, n étant le nombre de strates de E, posons : - $f(H) = H$, - $(f(E), <) = (\{\Phi_1^1 \wedge \ldots \wedge \Phi_{k_1}^1\}, \ldots, \{\Phi_1^n \wedge \ldots \wedge \Phi_{k_n}^n\}, \{\Phi_1^1\}, \ldots, \{\Phi_{k_1}^1\}, \ldots, \{\Phi_1^n\}, \ldots, \{\Phi_{k_n}^n\})$

  Nous constatons alors que, d'après la définition de l'ordre "Best-Out" :

  Propriété 4.1.29 *Soit $N(E) = \cup_{i=1\ldots l} E_i$ le noyau de E, alors $\cup_{i=1\ldots l}\{\Phi_1^i \wedge \ldots \wedge \Phi_{k_i}^i\}$ est le noyau de $f(E)$*

  Propriété 4.1.30 *Soit $N(f(E)) = \cup_{i=1\ldots l}\{\Phi_1^i \wedge \ldots \wedge \Phi_{k_i}^i\}$ le noyau de $f(E)$, alors $\cup_{i=1\ldots l} E_i$ est le noyau de E*

  Propriété 4.1.31 *$N(E)$ et $N(f(E))$ sont logiquement équivalents.*

  On a donc :

- EXI-BO-HORN implique 1/STRATE-EXI-BO-HORN :

$\exists Y$ sous base consistante bo-préférée de E telle que $Y \vdash H$

$\Rightarrow$

$\exists Y$ sous base consistante de E telle que Y contient le noyau de E et $Y \vdash H$

$\Rightarrow$

$\exists Y'$ sous base consistante de $f(E)$ telle que $Y' = Y \cup N(f(E))$ et $Y' \vdash H$

$\Rightarrow$

$\exists Y'$ sous base consistante bo-préférée de $f(E)$ telle que $Y' \vdash H$

- 1/STRATE-EXI-BO-HORN implique EXI-BO-HORN :

$\exists Y'$ sous base consistante bo-préférée de $f(E)$ telle que $Y' \vdash H$

$\Rightarrow$

$\exists Y$ sous base consistante de E telle que $Y = Y'$ dans laquelle on remplace toutes les formules $\Phi_1^i \wedge \ldots \wedge \Phi_{k_i}^i$ par l'ensemble $\{\Phi_1^i, \ldots, \Phi_{k_i}^i\}$; cette sous-base Y infère H et contient $N(E)$

$\Rightarrow$

$\exists Y$ sous base consistante bo-préférée de E telle que $Y \vdash H$

On a ainsi EXI-BO-HORN $\propto$ 1/STRATE-EXI-BO-HORN. Le problème 1/STRATE-EXI-BO-HORN est donc NP-complet.

- Pour 1/STRATE-ARG-BO-HORN, je peux garder la même démonstration que celle du cas Horn (voir théorème 4.1.35 page 124) en partant des problèmes 1/STRATE-EXI-BO-HORN-POSITIF et co-1/STRATE-EXI-BO-HORN-POSITIF, les deux transformations polynomiales utilisées générant des sous-bases stratifiées strictement. Ces deux problèmes sont respectivement NP-complet et co-NP-complet. En effet, il suffit de reprendre la démonstration de 1/STRATE-EXI-BO-HORN en imposant que H soit un littéral positif, et on peut alors garder la même transformation polynomiale pour passer de EXI-BO-HORN-POSITIF (que l'on sait être NP-complet – voir théorème 4.1.35 page 124) à 1/STRATE-EXI-BO-HORN-POSITIF. J'en conclus que 1/STRATE-ARG-BO $\in \Delta_2^p$ - (NP $\cup$ co-NP) si NP $\neq$ co-NP.

□

Propriété 4.1.32 (UNI(EXI, ARG)-INCL **et** UNI(EXI, ARG)-LEX) *On sait que, dans le cas d'une base stratifiée strictement, il existe une unique thèse préférée à la fois pour l'ordre "Inclusion Based" et pour l'ordre lexicographique. Si on rajoute en plus la contrainte d'une base de Horn, on obtient alors un unique problème noté 1/*STRATE-HORN.

Théorème 4.1.41 (Complexité de UNI(EXI, ARG)-INCL **et** UNI(EXI, ARG)-LEX**)**

*1/*STRATE-HORN $\in$ P.

Preuve : Étude d'appartenance.

L'algorithme proposé pour 1/STRATE (voir théorème 4.1.26 page 113) est encore valable. Par contre, maintenant que nous avons des conjonctions de clauses de Horn, cet algorithme, qui fait appel à un oracle de complexité P, devient déterministe polynomial. Le problème 1/STRATE-HORN appartient donc à la classe de complexité P.
□

4.1.4 Synthèse des résultats et conclusion

Dans le tableau ci-après, sont répertoriés tous les résultats démontrés dans les sections précédentes. On retrouve ainsi les cinq cas étudiés : le cas général, le cas d'une stratification stricte, le cas d'une base de formules CNF, le cas d'une base de Horn et le cas d'une base de Horn strictement stratifiée.

Remarquons que les résultats présentés ici sont aussi valables dans le cas des multi-ensembles. Le seul problème pour lequel j'ai trouvé une différence entre le cas des ensembles et le cas des multi-ensembles est le problème Uni-Car-Horn (voir tableau des résultats).

Rappelons que les résultats obtenus ici sont valables à la fois pour les relations de classe 1 (les p-m) et pour celles de classe 2 (les G-p-m). En effet, les algorithmes proposés pour les relations p-m sont utilisables aussi pour les relations G-p-m, ce qui permet de conserver tous les résultats d'appartenance. Quant aux résultats de complétude, ils demeurent identiques puisque les relations p-m sont des restrictions des relations G-p-m.

	Cas propositionnel général	Cas d'une base stratifiée strictement	Cas d'une base de formules Cnf	Cas d'une base de Horn	Cas d'une base de Horn strictement ordonnée
Uni-T thèses non ordonnées + conséquence forte	Π_2^p-complet	non concerné	Uni-T-Cnf Π_2^p-complet	Uni-T-Horn co-NP-complet	non concerné
Exi-T thèses non ordonnées + conséquence faible	Σ_2^p-complet	non concerné	Exi-T-Cnf Σ_2^p-complet	Exi-T-Horn NP-complet	non concerné
Arg-T thèses non ordonnées + conséquence argumentative	Δ_3^p - $(\Sigma_2^p \cup \Pi_2^p)$ si $\Sigma_2^p \neq \Pi_2^p$	non concerné	Arg-T-Cnf Δ_3^p - $(\Sigma_2^p \cup \Pi_2^p)$ si $\Sigma_2^p \neq \Pi_2^p$	Arg-T-Horn Δ_2^p - (NP $\cup$ co-NP) si NP $\neq$ co-NP	non concerné
Uni-S sous-bases consistantes non ordonnées + conséquence forte	co-NP-complet	non concerné	Uni-S-Cnf co-NP-complet	Uni-S-Horn P	non concerné

EXI-S sous-bases consistantes non ordonnées + conséquence faible	Σ_2^p-complet	non concerné	EXI-S-CNF Σ_2^p-complet	EXI-S-HORN NP-complet	non concerné
ARG-S sous-bases consistantes non ordonnées + conséquence argumentative	Δ_3^p - ($\Sigma_2^p \cup \Pi_2^p$) si $\Sigma_2^p \neq \Pi_2^p$	non concerné	ARG-S-CNF Δ_3^p - ($\Sigma_2^p \cup \Pi_2^p$) si $\Sigma_2^p \neq \Pi_2^p$	ARG-S-HORN Δ_2^p - (NP $\cup$ co-NP) si NP $\neq$ co-NP	non concerné
UNI-BO sous-bases consistantes "Best-Out" ordonnées + conséquence forte	Δ_2^p - (NP $\cup$ co-NP) si NP$\neq$ co-NP	1/STRATE-UNI-BO Δ_2^p - (NP $\cup$ co-NP) si NP$\neq$ co-NP	UNI-BO-CNF Δ_2^p - (NP $\cup$ co-NP) si NP$\neq$ co-NP	UNI-BO-HORN P	1/STRATE-UNI-BO-HORN P
EXI-BO sous-bases consistantes "Best-Out" ordonnées + conséquence faible	Σ_2^p-complet	1/STRATE-EXI-BO Σ_2^p-complet	EXI-BO-CNF Σ_2^p-complet	EXI-BO-HORN NP-complet	1/STRATE-EXI-BO-HORN NP-complet
ARG-BO sous-bases consistantes "Best-Out" ordonnées + conséquence argumentative	Δ_3^p - ($\Sigma_2^p \cup \Pi_2^p$) si $\Sigma_2^p \neq \Pi_2^p$	1/STRATE-ARG-BO Δ_3^p - ($\Sigma_2^p \cup \Pi_2^p$) si $\Sigma_2^p \neq \Pi_2^p$	ARG-BO-CNF Δ_3^p - ($\Sigma_2^p \cup \Pi_2^p$) si $\Sigma_2^p \neq \Pi_2^p$	ARG-BO-HORN Δ_2^p - (NP $\cup$ co-NP) si NP $\neq$ co-NP	1/STRATE-ARG-BO-HORN Δ_2^p - (NP $\cup$ co-NP) si NP $\neq$ co-NP

UNI-INCL thèses ordonnées par méthode "Inclusion Based" + conséquence forte	Π_2^p-complet	une seule relation d'inférence pour UNI (EXI, ARG) -INCL (LEX) : 1/STRATE Δ_2^p-complet	UNI-INCL-CNF Π_2^p-complet	UNI-INCL-HORN co-NP-complet	1/STRATE-HORN P
EXI-INCL thèses ordonnées par méthode "Inclusion Based" + conséquence faible	Σ_2^p-complet	(voir UNI-INCL)	EXI-INCL-CNF Σ_2^p-complet	EXI-INCL-HORN NP-complet	1/STRATE-HORN P
ARG-INCL thèses ordonnées par méthode "Inclusion Based" + conséquence argumentative	Δ_3^p - ($\Sigma_2^p \cup \Pi_2^p$) si $\Sigma_2^p \neq \Pi_2^p$	(voir UNI-INCL)	ARG-INCL-CNF Δ_3^p - ($\Sigma_2^p \cup \Pi_2^p$) si $\Sigma_2^p \neq \Pi_2^p$	ARG-INCL-HORN Δ_2^p - (NP $\cup$ co-NP) si NP $\neq$ co-NP	1/STRATE-HORN P
UNI-LEX thèses ordonnées par ordre lexicographique + conséquence forte	Δ_2^p-complet	(voir UNI-INCL)	UNI-LEX-CNF Δ_2^p-complet	UNI-LEX-HORN Δ_2^p-complet	1/STRATE-HORN P
EXI-LEX thèses ordonnées par ordre lexicographique + conséquence faible	Σ_2^p-complet	(voir UNI-INCL)	EXI-LEX-CNF Σ_2^p-complet	EXI-LEX-HORN Σ_2^p	1/STRATE-HORN P

ARG-LEX thèses ordonnées par ordre lexicographique + conséquence argumentative	Δ_3^p et Σ_2^p -difficile	(voir UNI-INCL)	ARG-LEX-CNF Δ_3^p et Σ_2^p -difficile	ARG-LEX-HORN Δ_3^p	1/STRATE-HORN P
UNI-CAR thèses ordonnées par ordre basé sur la cardinalité + conséquence forte	Δ_2^p-complet	non concerné	UNI-CAR-CNF Δ_2^p-complet	UNI-CAR-HORN Δ_2^p (et UNI-CAR-ME-HORN Δ_2^p-complet	non concerné
EXI-CAR thèses ordonnées par ordre basé sur la cardinalité + conséquence faible	Σ_2^p-complet	non concerné	EXI-CAR-CNF Σ_2^p-complet	EXI-CAR-HORN Σ_2^p	non concerné
ARG-CAR thèses ordonnées par ordre basé sur la cardinalité + conséquence argumentative	Δ_3^p et Σ_2^p -difficile	non concerné	ARG-CAR-CNF Δ_3^p et Σ_2^p -difficile	ARG-CAR-HORN Δ_3^p	non concerné
UNI-E extensions non ordonnées + conséquence forte	Π_2^p-complet	non concerné	UNI-E-CNF non étudié	UNI-E-HORN *Définition Stillman :* non étudié *Définition Kautz et Selman :* co-NP-complet	non concerné

EXI-E extensions non ordonnées + conséquence faible	Σ_2^p-complet	non concerné	EXI-E-CNF non étudié	EXI-E-HORN *Définition Stillman :* NP-complet *Définition Kautz et Selman :* P	non concerné
ARG-E extensions non ordonnées + conséquence argumentative	Δ_3^p - ($\Sigma_2^p \cup \Pi_2^p$) si $\Sigma_2^p \neq \Pi_2^p$	non concerné	ARG-E-CNF non étudié	ARG-E-HORN *Définition Stillman :* Δ_2^p et co-NP-difficile *Définition Kautz et Selman :* P	non concerné

Que peut-on conclure de tels résultats ?

Remarquons avant tout que seule la complexité temporelle des problèmes d'inférence non-monotone a été étudiée ici. En effet, la théorie de la complexité nous apprend que le temps est une ressource plus critique que l'espace.

Ensuite, on constate rapidement que la majorité des problèmes d'inférence non-monotone dans le cas propositionnel général ont une complexité exponentielle par rapport à la taille du problème. Bien que ces complexités ne dépassent pas le troisième niveau de la hiérarchie polynomiale, elles restent prohibitives.

Cela signifie que l'interrogation d'une base de connaissances propositionnelle E quelconque pour déterminer si une formule donnée est déductible de E, par un raisonnement non-monotone, risque fort de prendre un temps beaucoup trop important pour un opérateur humain (plusieurs heures, voir plusieurs jours et même plusieurs années).

Il existe différents moyens de réduire ce temps qui se ramènent tous à restreindre le cadre du problème étudié. J'en ai étudié quatre :

- ne travailler que sur une base stratifiée strictement, ce qui réduit, le plus souvent, la complexité d'un demi-niveau de la hiérarchie polynomiale,
- ne travailler que sur des formules CNF, ce qui ne réduit en rien la complexité des problèmes étudiés,
- ne travailler que sur des formules et des bases constituées respectivement de clauses de Horn et de conjonctions de clauses de Horn, ce qui ramène la complexité au niveau inférieur de la hiérarchie polynomiale dans la plupart des cas[32] ;
- ne travailler que sur des formules et des bases strictement stratifiées et constituées respectivement de clauses de Horn et de conjonctions de clauses de Horn, ce qui rend la plupart des problèmes polynomiaux.

[32] Rappelons que les résultats donnés ici permettent aussi de fixer une borne supérieure pour la complexité des problèmes d'inférence non-monotone réduits au cas d'une base de Horn *stricte*, c'est-à-dire ne traitant que des formules et des bases constituées de clauses de Horn.

La dernière méthode est bien sûr la plus intéressante mais aussi la plus contraignante : toutes les formules propositionnelles ne peuvent pas se traduire sous forme de conjonctions de clauses de Horn (par exemple : $A \vee B$) et définir un ordre strict entre les formules n'est pas toujours possible. De plus certaines relations n'exploitant pas l'ordre entre les formules, ce cas particulier n'est pas toujours utilisable.

D'autre part, si on se contente du cas particulier des conjonctions de clauses de Horn, lui-aussi très restrictif, on reste malheureusement en majorité dans le cadre de problèmes à complexité exponentielle.

Une analyse plus poussée des résultats permet de distinguer les relations d'inférence non-monotone Uni-Bo et Uni-Lex qui, dans le pire des cas, ne dépassent pas le premier niveau (Δ_2^p, NP, co-NP) de la hiérarchie polynomiale. Malheureusement, même ces relations ont des inconvénients :

- Uni-Bo présente le problème de la noyade (voir [BCD+93]) ;
- et paradoxalement, Uni-Lex n'est pas affectée par la restriction à un langage pauvre (les conjonctions de clauses de Horn) alors que presque toutes les autres relations la rejoignent sur le premier niveau de la hiérarchie polynomiale.

Les problèmes d'inférence non-monotone sont-ils condamnés à ne jamais être résolus ?

En fait, je ne pense pas que ces résultats permettent d'affirmer que toute utilisation pratique de ces formalismes est impossible. En effet :

- Ce problème est aussi celui de la logique classique (l'inférence classique est de complexité co-NP-complet), et il existe désormais un certain nombre de mécanismes, d'algorithmes permettant d'obtenir quelques résultats (voir les travaux sur Gsat dans [LMS92b]).
- Le développement de machines massivement parallèles de plus en plus rapides, de plus en plus puissantes, se poursuit. Elles pourraient peut-être permettre de repousser la frontière de l'"inutilisabilité" pratique suffisamment loin pour pouvoir traiter des instances de taille suffisante.
- Et puis surtout les calculs de complexité effectués ici reflètent le pire des cas et bon nombre de problèmes pratiques à traiter ne sont pas dans cette catégorie.

Il faudrait pour affiner cette conclusion faire aussi une étude de complexité en moyenne, ce que je ne ferai pas ici par manque de compétence dans ce domaine.

En conclusion, pour traiter l'inférence non-monotone, il faut définir le problème le plus précisément possible, afin d'utiliser au mieux ses particularités lors de la définition des algorithmes de résolution du problème. Cela pose alors le problème de la réutilisabilité de tels algorithmes. Je pense qu'il faudra probablement en venir à la définition de diverses catégories de problèmes et à l'écriture d'algorithmes propres à chaque catégorie.

4.2 Prudence

Dans cette section, je présente une classification semblable à celle présentée par Pinkas et Loui dans [PL92]. En effet, j'étudie du point de vue de la prudence les principes UNI, EXI et ARG qui sont déjà traités dans [PL92]. Mais, alors que Pinkas et Loui les ont classés sans préciser les mécanismes de génération, je m'attache à raffiner une partie de leur schéma en introduisant l'impact de quelques mécanismes de génération sur la prudence des relations d'inférence non-monotone.

Les résultats présentés dans cette section ont fait l'objet de publications (voir [CLS94a, LS95c, CLS95]).

Les relations d'inférence étudiées ici sont celles de la classe 1, donc du type $(E, <) \mid\!\sim^{p,m} \Phi$. Rappelons toutefois que les relations de la classe 2 sont elles-aussi concernées par cette étude (voir le lien entre les deux classes en section 2.6.6 page 46).

Pour construire cette classification, j'utilise les propriétés suivantes :

Propriété 4.2.1 *Soit Y une sous-base incl-préférée de E alors Y est aussi une thèse de E.*

Preuve : Par définition d'une sous-base incl-préférée. □

Propriété 4.2.2 *Soit Y une thèse de E alors Y est aussi une sous-base consistante de E.*

Preuve : Par définition d'une thèse. □

Propriété 4.2.3 *Soit Y une sous-base car-préférée de E alors Y est aussi une thèse de E.*

Preuve : Par définition d'une sous-base car-préférée. □

Propriété 4.2.4 *Soit Y une sous-base lex-préférée de E alors Y est aussi une sous-base incl-préférée de E.*

Preuve : Voir dans [BCD+93]. □

Propriété 4.2.5 *Soit Y une sous-base incl-préférée de E alors Y est aussi une sous-base bo-préférée de E.*

Preuve : Voir dans [BCD+93]. □

Propriété 4.2.6 *Soit Y une sous-base bo-préférée de E alors il existe Y' sous-base maximale consistante de E telle que Y est incluse dans Y'.*

Preuve : Voir dans [BCD+93]. □

Propriété 4.2.7 *Soient la base E et la théorie super-normale correspondante (W, D) (voir section 2.2.3 page 16), soit Y une sous-base incl-préférée de E alors $C_n(Y)$ est une extension de (W, D).*

Preuve : Voir dans [Bre89b]. □

Propriété 4.2.8 *Soient la base E et la théorie super-normale correspondante (W, D), soit Y une extension de (W, D) alors il existe Y' sous-base incl-préférée de E telle que $Y = C_n(Y')$.*

Preuve : Voir dans [Bre89b]. □

Grâce à toutes ces propriétés, on prouve que :

- $E \mid\!\sim^{\forall,Bo} \Phi$
  $\Rightarrow$ (lien numéro 22 sur les figures 4.1 page 154 et 4.2 page 155)
  $E \mid\!\sim^{\forall,In} \Phi$
  $\Leftrightarrow$ (lien numéro 26 sur les figures 4.1 page 154 et 4.2 page 155)
  $E \mid\!\sim^{\forall,E^*} \Phi$
  $\Rightarrow$ (lien numéro 25 sur les figures 4.1 page 154 et 4.2 page 155)
  $E \mid\!\sim^{\forall,Lex} \Phi$
- $E \mid\!\sim^{\forall,S} \Phi$
  $\Rightarrow$ (lien numéro 20 sur les figures 4.1 page 154)
  $E \mid\!\sim^{\forall,T} \Phi$
  $\Rightarrow$ (lien numéro 23 sur les figures 4.1 page 154)
  $E \mid\!\sim^{\forall,Car} \Phi$
  $\Rightarrow$ (lien numéro 24 sur les figures 4.1 page 154)
  $E \mid\!\sim^{\forall,Lex} \Phi$
- $E \mid\!\sim^{\forall,T} \Phi$
  $\Rightarrow$ (lien numéro 21 sur les figures 4.1 page 154)
  $E \mid\!\sim^{\forall,In} \Phi$
- $E \mid\!\sim^{\exists,Lex} \Phi$
  $\Rightarrow$ (lien numéro 35 sur les figures 4.1 page 154 et 4.2 page 155)
  $E \mid\!\sim^{\exists,In} \Phi$
  $\Leftrightarrow$ (lien numéro 36 sur les figures 4.1 page 154 et 4.2 page 155)
  $E \mid\!\sim^{\exists,E^*} \Phi$
  $\Rightarrow$ (lien numéro 32 sur les figures 4.1 page 154 et 4.2 page 155)
  $E \mid\!\sim^{\exists,Bo} \Phi$
- $E \mid\!\sim^{\exists,Lex} \Phi$
  $\Rightarrow$ (lien numéro 34 sur les figures 4.1 page 154)
  $E \mid\!\sim^{\exists,Car} \Phi$
  $\Rightarrow$ (lien numéro 33 sur les figures 4.1 page 154)
  $E \mid\!\sim^{\exists,T} \Phi$
  $\Rightarrow$ (lien numéro 30 sur les figures 4.1 page 154)
  $E \mid\!\sim^{\exists,S} \Phi$
- $E \mid\!\sim^{\exists,In} \Phi$
  $\Rightarrow$ (lien numéro 31 sur les figures 4.1 page 154)
  $E \mid\!\sim^{\exists,T} \Phi$
- $E \mid\!\sim^{\exists,Bo} \Phi$
  $\Rightarrow$ (lien numéro 37 sur les figures 4.1 page 154)
  $E \mid\!\sim^{\exists,T} \Phi$

D'autre part, $\forall m \in \{$Bo, Incl, Lex, E*$\}$, on sait que :

- $E \mid\!\sim^{\forall,m} \Phi$
  $\Rightarrow$ (liens numéros 1, 3, 5, 7, 9, 11, 13 sur les figures 4.1 page 154 et 4.2 page 155)
  $E \mid\!\sim^{A,m} \Phi$
  $\Rightarrow$ (liens numéros 2, 4, 6, 8, 10, 12, 14 sur les figures 4.1 page 154 et 4.2 page 155)
  $E \mid\!\sim^{\exists,m} \Phi$.

- $E \mid\!\sim^{A,In} \Phi$
$\Leftrightarrow$ (lien numéro 16 sur les figures 4.1 page suivante et 4.2 page 155)
$E \mid\!\sim^{A,E^*} \Phi$

Remarques :

- Les liens de prudence entre les relations de classe 1 (les p-m) sont présentés par la figure 4.1 page suivante.
- Les résultats obtenus sur les relations de classe 1 étant aussi valables pour les relations de classe 2 (les G-p-m), cela donne le schéma 4.2 page 155.
- La relation "est plus prudente que" est une relation transitive[33]. Toutefois, tous les liens de prudence n'ont pas été tracés dans les figures 4.1 page suivante et 4.2 page 155 afin d'alléger le schéma.
- Les relations Uni-T, Exi-T et Arg-T ont déjà été comparées du point de vue de la prudence dans [BDP93].

4.3 Propriétés de déduction

Dans cette section, j'étudie les propriétés vérifiées par les relations d'inférence non-monotone de la classe 2, telles que je les ai définies dans la section 2.6 page 29 de ce document. Toutefois, cette étude peut aussi être étendue aux relations d'inférence de la classe 1 correspondantes (voir le lien entre les deux classes en section 2.6.6 page 46).

J'ai choisi de m'intéresser uniquement aux relations d'inférence de classe 2, donc du type $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$ de manière à respecter exactement la terminologie des propriétés telles qu'elles ont été définies par Kraus, Lehmann, Magidor, Gärdenfors et Makinson. Ce faisant, les seules relations présentant un intérêt sont celles utilisant la stratification (donc celles construites avec les mécanismes Bo, Incl, Lex, E – ou E*). Toutefois, il est possible d'utiliser une définition restreinte des propriétés dans lesquelles interviennent uniquement la relation $(E,<) \mid\!\sim^{p,m} \Phi$ et non plus la relation $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$. Dans ce cadre là, on peut alors traiter aussi les relations d'inférence de classe 1 définies avec les mécanismes T, S et Car[34].

Ainsi, je me limiterai aux relations du type $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$ avec $p \in$ {Uni, Exi, Arg} et $m \in$ {Bo, Incl, Lex, E*}. Certains de ces résultats ont été démontrés par d'autres chercheurs. Je vais les présenter dans une première section. Puis, je donnerai mes propres résultats en complément dans une seconde section.

Remarquons qu'ici aussi on est amené à réaliser une étude pouvant sembler répétitive, puisqu'il faut examiner chaque propriété pour chaque relation. Les lecteurs pressés pourront passer directement aux tableaux récapitulatifs donnés en section 4.3.3 page 175.

Les résultats présentés dans cette section ont fait l'objet de publications (voir [CLS94a, LS95c, CLS95]).

4.3.1 Les résultats déjà existants et leurs incidences

Les travaux de Kraus, Lehmann et Magidor [KLM90], ainsi que ceux de Gärdenfors et Makinson [GM94] fournissent la totalité des résultats concernant les relations d'inférence définies à partir

[33] En effet, la relation "est plus prudente que" est construite à partir de la relation d'inclusion qui est transitive.
[34] C'est l'option choisie par les auteurs de [BDP93, Ben94] ; ils démontrent ainsi que :

- la relation G-Arg-T vérifie la propriété d'affaiblissement droit définie par $\dfrac{\models \alpha \rightarrow \beta \,;\, E \mid\!\sim \alpha}{E \mid\!\sim \beta}$;
- la relation G-Arg-T ne vérifie pas la propriété du ET définie par $\dfrac{E \mid\!\sim \beta \,;\, E \mid\!\sim \gamma}{E \mid\!\sim \beta \wedge \gamma}$.

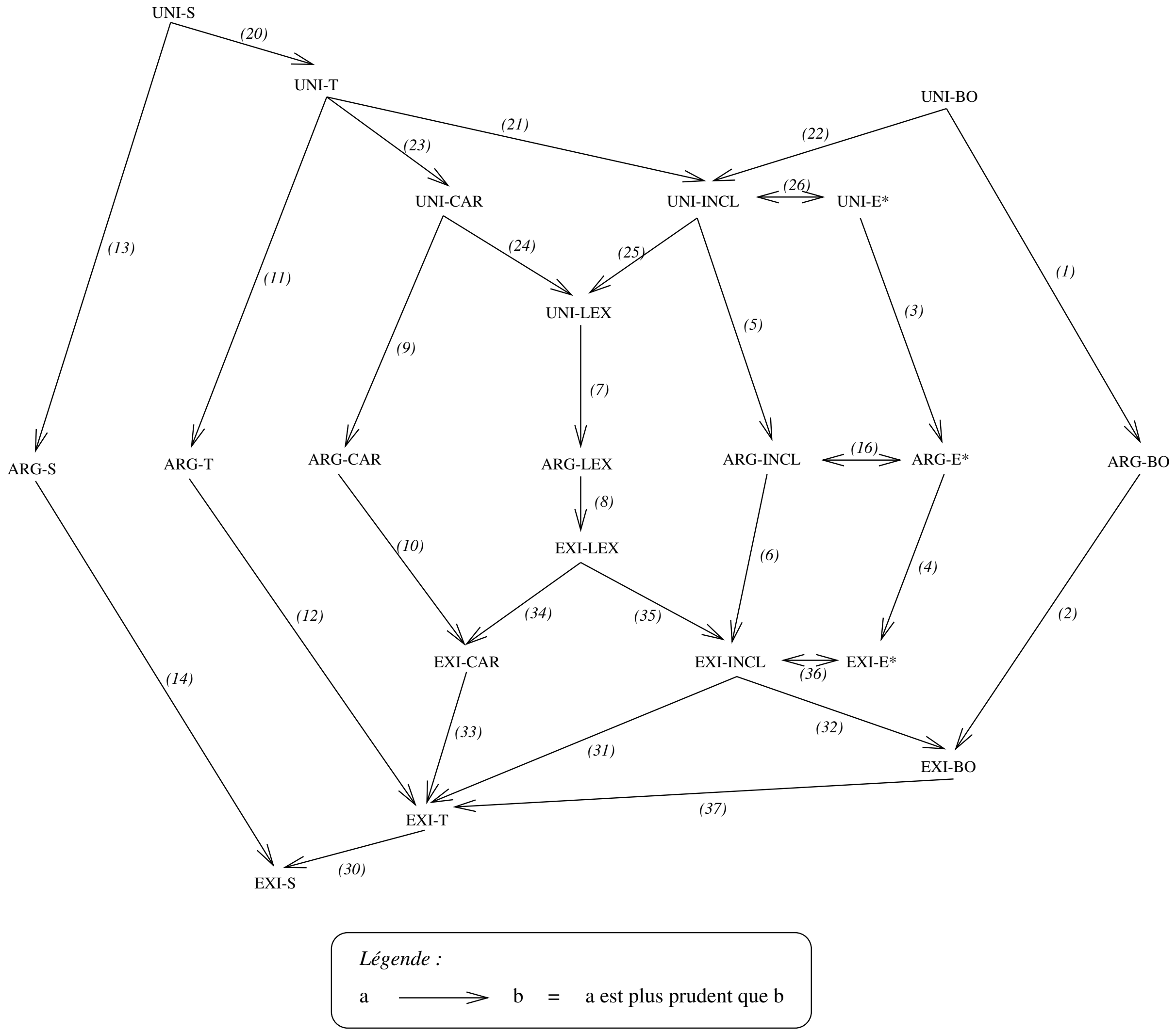


Figure 4.1 : Liens de prudence entre les relations d'inférence non-monotone de classe 1

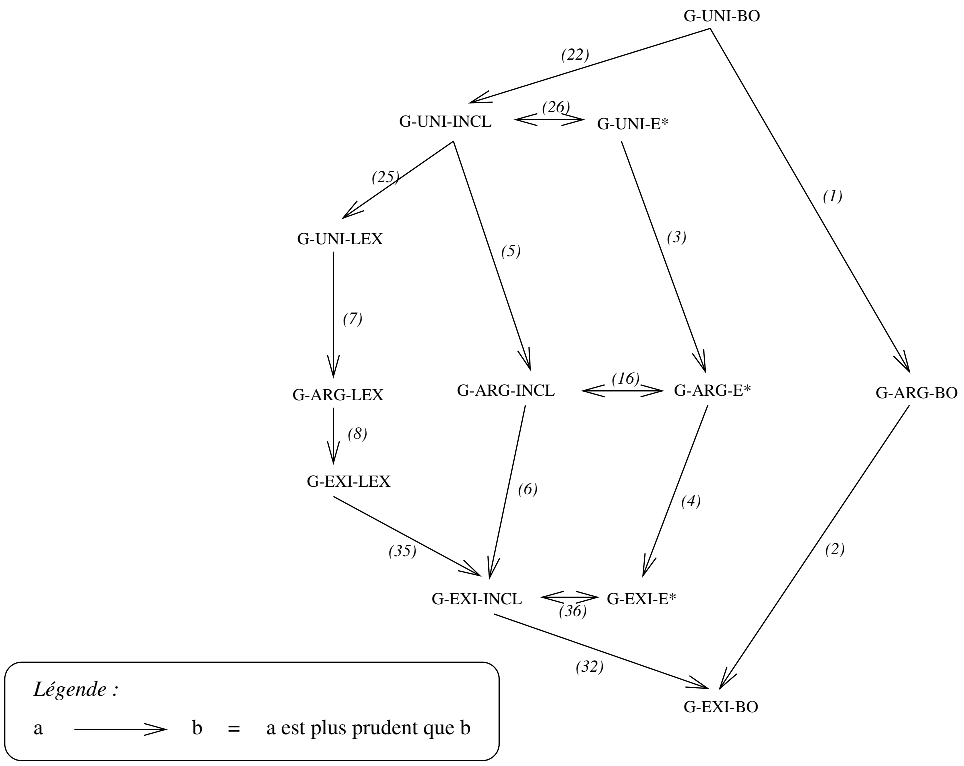


Figure 4.2 : Liens de prudence entre les relations d'inférence non-monotone de classe 2

du principe Uni. En effet, tous les théorèmes de représentation (voir section 3.4 page 55) se ramènent à l'utilisation d'une approche "sceptique" : il existe un modèle d'un type particulier tel que *toutes* les interprétations vérifient une propriété donnée.

4.3.1.1 G-Uni-Incl, G-Uni-Lex, G-Uni-Bo

En utilisant ces théorèmes de représentation, Benferhat, Cayrol, Dubois, Lang et Prade ont démontré les résultats suivants :

- G-Uni-Incl est une relation d'inférence préférentielle (voir [BCD$^+$93, Ben94]).
- G-Uni-Lex est une relation d'inférence rationnelle (voir [BCD$^+$93, Ben94]).
- G-Uni-Bo est une relation d'inférence quasiment rationnelle, puisque la relation d'inférence issue de la logique possibiliste, et notée $\vdash_\pi$, est quasiment rationnelle (voir [Ben94]) et qu'il a été démontré dans [BCD$^+$93] que $\mid\!\sim^{\forall,Bo} \Leftrightarrow \vdash_\pi$. La dénomination de "quasiment rationnelle" vient du fait que Benferhat et ses collègues n'utilisent pas la forme classique des propriétés de supra-classicité et de réflexivité mais une forme restreinte. En effet, ils ont choisi comme convention le principe appelé "Nihil ex Absurdo", c'est-à-dire qu'en présence d'une inconsistance, ils choisissent de ne rien déduire[35] ; ceci les amène donc à définir les propriétés de déduction suivantes :
  - propriété de réflexivité restreinte : si $\alpha \neq \perp$ alors $\alpha \mid\!\sim \alpha$,
  - la supra-classicité restreinte : si $\alpha \neq \perp$ alors $\dfrac{\alpha \vdash \beta}{\alpha \mid\!\sim \beta}$.

[35] L'autre principe utilisé en général consiste à dire qu'en présence d'une inconsistance, on choisit de tout déduire.

On dit alors qu'une relation d'inférence est quasiment rationnelle quand elle vérifie la monotonie rationnelle, la réflexivité restreinte et toutes les propriétés du système P excepté la réflexivité classique.

Toutes ces démonstrations s'articulent autour de la reconnaissance d'un modèle particulier permettant de définir la relation d'inférence étudiée, puis de l'application du théorème de représentation correspondant.

On a aussi les résultats suivants :

- G-Uni-Incl vérifie les propriétés de : supra-classicité, équivalence logique gauche, affaiblissement droit, ET, conditionnalisation faible, préservation de la consistance, cumulativité, OU et ne vérifie pas les propriétés de monotonie rationnelle et de monotonie rationnelle faible (voir [BCD+93]).
- G-Uni-Lex vérifie les propriétés de : supra-classicité, équivalence logique gauche, affaiblissement droit, ET, conditionnalisation faible, préservation de la consistance, cumulativité, OU, monotonie rationnelle et monotonie rationnelle faible (voir [BCD+93]) ; G-Uni-Lex vérifie donc les postulats étendus de Gärdenfors et Makinson (voir [GM94]).
- G-Uni-Bo vérifie les propriétés de : supra-classicité restreinte, équivalence logique gauche, affaiblissement droit, ET, préservation de la consistance, OU, monotonie rationnelle et monotonie rationnelle faible (voir [BCD+93, Ben94]). La conditionnalisation faible et la cumulativité n'apparaissent pas dans ces résultats. Toutefois, la conditionnalisation faible est prouvée à la fois par :
  - le théorème 2.6.1 page 43 donné dans ce document,
  - une propriété donnée dans [BDP92] qui dit que la propriété de conditionnalisation est impliquée soit par les propriétés d'équivalence logique gauche, d'affaiblissement droit, du ET, du OU, de la monotonie rationnelle, de la réflexivité restreinte et du Nihil ex Absurdo, soit par les propriétés d'équivalence logique gauche, d'affaiblissement droit, du ET, du OU, de la monotonie prudente, de la réflexivité ; or la conditionnalisation faible est un cas particulier de la conditionnalisation.

Remarquons que les relations G-Uni-Incl, G-Uni-Bo et G-Uni-Lex ne sont pas définies de manière identique dans [BCD+93] et dans ce document. Toutefois, la propriété 2.6.11 page 46 permet de se rendre compte que les deux définitions sont équivalentes et que l'on peut donc utiliser les résultats présentés dans [BCD+93] dans le cadre présent.

4.3.1.2 G-Uni-E* et G-Exi-E*

Dans [Bra93], Brass étudie un sous-ensemble de la logique des défauts (mécanisme E*) : les défauts super-normaux. Ceci correspond au cas particulier du mécanisme Incl sur une base de croyances possédant deux strates. Quatre principes sont invoqués dans ce travail dont les principes G-Uni et G-Exi.

Ainsi, Brass a démontré que :

- G-Uni-E*, dans le cas des défauts super-normaux, vérifie les propriétés suivantes :
  - la réflexivité,
  - l'affaiblissement droit,
  - l'équivalence logique gauche,
  - le ET,
  - le OU.
- G-Exi-E*, dans le cas des défauts super-normaux, vérifie les propriétés suivantes :
  - la réflexivité,

- l'affaiblissement droit,
- l'équivalence logique gauche.

Pour prouver que G-UNI-E* est préférentielle, dans le cas des défauts super-normaux, il suffit de prouver qu'elle vérifie les propriétés de coupure et de monotonie prudente.

Théorème 4.3.1 *Dans le cas des défauts super-normaux, la relation G-UNI-E* vérifie la propriété de la coupure (si $\alpha \mid\!\sim \beta$ et $\alpha \wedge \beta \mid\!\sim \gamma$ alors $\alpha \mid\!\sim \gamma$).*

> **Preuve :** On part des hypothèses $\alpha \mid\!\sim \beta$ et $\alpha \wedge \beta \mid\!\sim \gamma$.
>
> La première hypothèse implique que $\forall Y$ extension de $T = (W \cup \{\alpha\}, D)$, $Y \vdash \beta$; or les extensions sont fermées pour la conséquence logique, donc $\beta \in Y$.
>
> La seconde hypothèse implique que $\forall Y'$ extension de $T' = (W \cup \{\alpha, \beta\}, D)$, $Y' \vdash \gamma$; or les extensions Y' de T' sont construites à partir de T comme suit :
>
> - soit $Y' = C_n(Y \cup \{\beta\})$ ssi Y est β-consistante,
> - soit $Y' = C_n(S \cup \{\beta\})$ ssi S est le résultat d'une branche dont l'hypothèse manquante était "$\beta \in$ l'extension".
>
> Or, on sait que $\forall Y, \beta \in Y$, donc le premier cas revient à dire que $\forall Y$, $\exists Y'$ telle que $Y' = Y$. Et on sait que $\forall Y'$, $Y' \vdash \gamma$; donc $\forall Y$, $Y \vdash \gamma$ et on en conclut que $\alpha \mid\!\sim \gamma$. □

Théorème 4.3.2 *Dans le cas des défauts super-normaux, la relation G-UNI-E* vérifie la propriété de monotonie prudente (si $\alpha \mid\!\sim \beta$ et $\alpha \mid\!\sim \gamma$ alors $\alpha \wedge \beta \mid\!\sim \gamma$).*

> **Preuve :** On sait (voir [Bre89b]) que, dans le cas d'une théorie des défauts super-normale $T = (W, D)$, les extensions obtenues correspondent exactement aux thèses incl-préférées obtenues sur la base E composée de deux strates $E_1 = W$ et $E_2 =$ {les conséquents des défauts de D}.
>
> Or, on sait que la relation G-UNI-INCL vérifie la monotonie prudente puisqu'elle est préférentielle.
>
> On a donc : si $\forall Y$ thèse incl-préférée de $\alpha \oplus E$, $Y \vdash \beta$ et $Y \vdash \gamma$ alors $\forall Y'$ thèse incl-préférée de $(\alpha \wedge \beta) \oplus E$, $Y' \vdash \gamma$.
>
> Sachant que les Y thèses incl-préférées de $\alpha \oplus E$ sont exactement les extensions de $\alpha \oplus T = (W \cup \{\alpha\}, D)$ et que les Y' thèses incl-préférées de $(\alpha \wedge \beta) \oplus E$ sont exactement les extensions de $(\alpha \wedge \beta) \oplus T = (W \cup \{\alpha, \beta\}, D)$, on déduit que la relation G-UNI-E* vérifie la propriété de monotonie prudente[36]. □

Contre-exemple pour la monotonie rationnelle. Ici aussi, le contre-exemple donné dans [Ben94] pour G-UNI-INCL peut s'appliquer :

La théorie des défauts T est la suivante :

- $W = \{a, b \rightarrow ; a \rightarrow c\}$,
- $D = \{ : \rightarrow a / \rightarrow a \ ; \ : \rightarrow b / \rightarrow b \ ; \ : \ c \rightarrow / c \rightarrow \}$.

On obtient ainsi 2 extensions :

- $Y_1 = C_n(\{a, b \rightarrow ; a \rightarrow c ; \rightarrow a\})$;
- $Y_2 = C_n(\{a, b \rightarrow ; a \rightarrow c ; \rightarrow b ; c \rightarrow\})$;

[36] Ce type de démonstration peut aussi être utilisé pour prouver que G-UNI-E* vérifie la coupure.

Toutes ces extensions infèrent la formule $a \vee (\neg c)$ et il existe une extension qui n'infère pas $\neg c$. Posons :

- $\alpha = vrai$,
- $\beta = c$,
- $\gamma = a \vee (\neg c)$.

On a ainsi $\alpha \mid\!\sim \gamma$ et $\alpha \mid\!\not\sim (\neg\beta)$.

Si on rajoute la formule β à la base E, on obtient $(\alpha \wedge \beta) \mid\!\not\sim \gamma$, car il existe une extension de $\beta \oplus E$ qui n'infère pas γ (l'extension $Y = C_n(\{a, b \rightarrow\ ; a \rightarrow c\ ; \rightarrow b\ ; \rightarrow c\}))$.

Théorème 4.3.3 *Dans le cas des défauts super-normaux, la relation G-*UNI-E* *est préférentielle et non rationnelle.*

Contre-exemple dans le cas des défauts normaux. Prenons la théorie suivante :

- $W = \{\}$,
- $D = \{\ :\ a/a\ ; (a \vee b)\ :\ (\neg a)/(\neg a)\}$.

Il est facile de voir sur cette théorie, que la relation G-UNI-E* ne vérifie pas la monotonie prudente.

En effet, cette théorie admet une seule extension $Y = C_n\{a\}$. Cette extension permet d'inférer les formules a et $a \vee b$. Posons :

- $\alpha = vrai$,
- $\beta = a \vee b$,
- $\gamma = a$.

On a ainsi $\alpha \mid\!\sim \gamma$ et $\alpha \mid\!\sim \beta$.

Si on rajoute la formule β à la base E on obtient $(\alpha \wedge \beta) \mid\!\not\sim \gamma$, car il existe maintenant deux extensions pour $\beta \oplus E$: $Y_1 = C_n\{a\}$ et $Y_2 = C_n\{\neg a, b\})$ avec Y_2 qui n'infère pas γ.

4.3.1.3 Conclusion sur les résultats déjà existants

Dans les tableaux 4.4 page suivante et 4.5 page ci-contre, je donne tous les résultats obtenus sur les relations G-UNI-m pour $m \in \{$BO, INCL, LEX, E$^*\}$ et sur G-EXI-E* dans cette section et dans la section 2.6.4 page 43. Le tableau 4.4 page ci-contre permet de se situer par rapport aux travaux de Kraus, Lehmann et Magidor (voir [KLM90]) et le tableau 4.5 page suivante permet de se situer par rapport aux travaux de Gärdenfors et Makinson (voir [GM94]).

4.3.2 Les nouveaux résultats

Ces nouveaux résultats concernent la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 et sont présentés propriété par propriété avec les preuves correspondant aux différents cas.

4.3.2.1 La supra-classicité

Rappelons que la propriété de supra-classicité se définit par $\dfrac{\alpha \vdash \beta}{\alpha \mid\!\sim \beta}$

Relations d'inférence	Propriétés
G-Uni-Bo	système P (sauf la réflexivité) + réflexivité restreinte + monotonie rationnelle + conditionnalisation faible
G-Uni-Incl	système P + conditionnalisation faible
G-Uni-Lex	système P + monotonie rationnelle + conditionnalisation faible
G-Uni-E* (super-normaux)	système P + conditionnalisation faible
G-Exi-E* (super-normaux)	réflexivité + affaiblissement droit + équivalence logique gauche

Tableau 4.4 : Résultats existants sur les propriétés des relations G-p-m par rapport à Kraus, Lehmann et Magidor.

Relations d'inférence	Propriétés
G-Uni-Bo	postulats de base (sauf supra-classicité) + supra-classicité restreinte + monotonie rationnelle + OU
G-Uni-Incl	postulats étendus sauf monotonie rationnelle et monotonie rationnelle faible
G-Uni-Lex	postulats étendus
G-Uni-E* (super-normaux)	affaiblissement droit + équivalence logique gauche + ET + OU
G-Exi-E* (super-normaux)	affaiblissement droit + équivalence logique gauche

Tableau 4.5 : Résultats existants sur les propriétés des relations G-p-m par rapport à Gärdenfors et Makinson.

Théorème 4.3.4 $\forall p \in \{\textsc{Uni}, \textsc{Exi}, \textsc{Arg}\}$, $\forall m \in \{\textsc{Bo}, \textsc{Incl}, \textsc{Lex}, \mathrm{E}^*\}$, *la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété de supra-classicité.*

Preuve :

- Pour G-Uni-m : (pour m = Incl ou Lex, voir aussi section 4.3.1 page 153) si $\alpha \vdash \beta$ alors $\forall S$ sous-ensemble de E α-consistant, $S \cup \{\alpha\} \vdash \beta$, donc d'après les théorèmes, 2.6.4 page 37, 2.6.3 page 33, 2.6.7 page 40 et 2.6.8 page 42, on a $\alpha \mid\!\sim^{\forall,m} \beta \ \forall m$.
- Pour G-Exi-m : si $\alpha \vdash \beta$ alors, d'après le cas précédent, $\alpha \mid\!\sim^{\forall,m} \beta \ \forall m$, donc $\alpha \mid\!\sim^{\exists,m} \beta \ \forall m$.
- Pour G-Arg-m : si $\alpha \vdash \beta$ alors $\alpha \nvdash (\neg\beta)$ donc $\forall S$ sous-ensemble de E α-consistant, $S \cup \{\alpha\} \nvdash (\neg\beta)$, et avec le fait que $\alpha \mid\!\sim^{\exists,m} \beta \ \forall m$, on obtient que $\alpha \mid\!\sim^{A,m} \beta \ \forall m$.

□

4.3.2.2 La réflexivité

Rappelons que la propriété de réflexivité se définit par $\alpha \mid\!\sim \alpha$

Théorème 4.3.5 $\forall p \in \{\textsc{Uni}, \textsc{Exi}, \textsc{Arg}\}$, $\forall m \in \{\textsc{Bo}, \textsc{Incl}, \textsc{Lex}, \mathrm{E}^*\}$, *la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété de réflexivité.*

Preuve :

- Pour G-Uni-m : (pour m = Incl, Lex ou E*, voir aussi section 4.3.1 page 153) on sait que la relation vérifie la supra-classicité donc d'après [Gär91, GM94] elle vérifie aussi la réflexivité.
- Pour G-Exi-m et G-Arg-m : (pour G-Exi-E*, voir aussi section 4.3.1 page 153) idem G-Uni-m.

□

4.3.2.3 La préservation de la consistance

Rappelons que la propriété de préservation de la consistance se définit par $\dfrac{\alpha \mid\!\sim \perp}{\alpha \vdash \perp}$

(avec $\perp$ représentant la contradiction)

Théorème 4.3.6 $\forall p \in \{\textsc{Uni}, \textsc{Exi}, \textsc{Arg}\}$, $\forall m \in \{\textsc{Bo}, \textsc{Incl}, \textsc{Lex}, \mathrm{E}^*\}$, *la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété de préservation de la consistance.*

Preuve :

- Pour G-Uni-m : (pour m = Bo, Incl ou Lex, voir aussi section 4.3.1 page 153) $\forall m$, $\alpha \mid\!\sim^{\forall,m} \perp \Leftrightarrow \forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \vdash \perp \Leftrightarrow \forall Y'$ sous-base m-préférée de $\alpha \oplus E$, Y' est inconsistante ; or, vu la définition des Y' (voir section 2.2 page 7), ceci n'est possible que lorsque la formule α est inconsistante, donc $\alpha \vdash \perp$.
- Pour G-Exi-m : idem G-Uni-m.
- Pour G-Arg-m : idem G-Uni-m.

□

4.3.2.4 La conditionnalisation faible

Rappelons que la propriété de conditionnalisation faible se définit par $\dfrac{\alpha \mid\sim \beta}{\mid\sim \alpha \to \beta}$

Théorème 4.3.7 $\forall p \in$ {Uni, Exi, Arg}, $\forall m \in$ {Bo, Incl, Lex, E*}, *la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété de conditionnalisation faible.*

Preuve :

- Pour G-Uni-m : voir théorème 2.6.1 page 43 et pour $m =$ Bo, Incl ou Lex, voir aussi section 4.3.1 page 153.
- Pour G-Exi-m : $\forall m$, $\alpha \mid\sim^{\exists,m} \beta \Rightarrow \forall m$, $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta$; par définition (voir section 2.2 page 7) les Y' ne peuvent avoir que deux formes possibles :
  - soit $Y' = \{\alpha\} \cup Y$ avec Y sous-base m-préférée de E ; alors on peut appliquer le théorème de la déduction et on obtient : $\exists Y$ sous-base m-préférée de E telle que $Y \vdash \alpha \to \beta$;
  - soit $Y' = \{\alpha\} \cup S$ avec S sous-ensemble de E qui n'est pas une sous-base m-préférée de E ; S est donc issu d'une sous-base Y m-préférée de E inconsistante avec $\alpha \Rightarrow$ il existe Y sous-base m-préférée de E telle que $Y \vdash (\neg\alpha)$ $\Rightarrow$ il existe Y sous-base m-préférée de E telle que $Y \vdash (\neg\alpha) \vee \beta \Rightarrow$ il existe Y sous-base m-préférée de E telle que $Y \vdash (\alpha \to \beta)$.
- Pour G-Arg-m : $\forall m$, $\alpha \mid\sim^{A,m} \beta \Rightarrow \forall m$, $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta$ et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \not\vdash (\neg\beta)$; grâce à la démonstration de la propriété de conditionnalisation faible pour les relations d'inférence G-Exi-m, on sait déjà que "$\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta \Rightarrow \exists Y$ sous-base m-préférée de E telle que $Y \vdash (\alpha \to \beta)$" ; raisonnons par l'absurde en supposant que $\neg$ ($\forall Y$ sous-base m-préférée de E, $Y \not\vdash \neg(\alpha \to \beta)$) ; ceci implique que $\exists Y$ sous-base préférée de E telle que $Y \vdash \neg(\alpha \to \beta) \Rightarrow \exists Y$ sous-base m-préférée de E consistante avec α telle que $Y \cup \{\alpha\} \vdash \neg\beta \Rightarrow \exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \neg\beta \Rightarrow$ contradiction avec l'hypothèse "$\forall Y'$ sous-base m-préférée de $\alpha \oplus E$ $Y' \not\vdash (\neg\beta)$".

□

4.3.2.5 Le OU

Rappelons que la propriété du OU se définit par $\dfrac{\alpha \mid\sim \gamma \;;\; \beta \mid\sim \gamma}{\alpha \vee \beta \mid\sim \gamma}$

Rappel 4.3.1 *Pour* $p =$ Uni, $\forall m \in$ {Bo, Incl, Lex, E*}, *la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété du OU.*

Preuve : Voir section 4.3.1 page 153. □

Théorème 4.3.8 $\forall p \in$ {Exi, Arg}, $\forall m \in$ {Bo, Incl, Lex, E*}, *la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30,* ne vérifie pas *la propriété du OU.*

Preuve :

- Pour G-EXI-m : Il suffit de prendre un contre-exemple ; soit la base E suivante (stratifiée en deux strates) :

$$\begin{array}{|cc|} x, \alpha \rightarrow \gamma & \beta \rightarrow \gamma, x \\ \hline \rightarrow x & x \rightarrow \end{array}$$

Le calcul des sous-bases m-préférées de E fournit les résultats suivants :
  - $\forall m \in \{\text{INCL}, \text{LEX}\}$ les sous-bases m-préférées de E sont :
    - $Y_1 = \{x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\}$ et
    - $Y_2 = \{x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ x \rightarrow\}$;
  - pour $m = \text{BO}$, parmi les sous-bases m-préférées de E, on trouve[37] :
    - $Y_1 = \{x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\}$,
    - $Y_2 = \{x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ x \rightarrow\}$;
  - pour $m = \text{E}^*$, les sous-bases m-préférées de E sont :
    - $Y_1 = C_n(\{x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\})$ et
    - $Y_2 = C_n(\{x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ x \rightarrow\})$;

pour $\alpha \oplus E$, on obtient :
  - $\forall m \in \{\text{INCL}, \text{LEX}\}$ les sous-bases m-préférées de $\alpha \oplus E$ sont :
    - $Y_1' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\}$ et
    - $Y_2' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ x \rightarrow\}$;
  - pour $m = \text{BO}$, parmi les sous-bases m-préférées de $\alpha \oplus E$, on trouve :
    - $Y_1' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\}$,
    - $Y_2' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ x \rightarrow\}$;
  - pour $m = \text{E}^*$, les sous-bases m-préférées de $\alpha \oplus E$ sont :
    - $Y_1' = C_n(\{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\})$ et
    - $Y_2' = C_n(\{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ x \rightarrow\})$;

pour $\beta \oplus E$, on obtient :
  - $\forall m \in \{\text{INCL}, \text{LEX}\}$ les sous-bases m-préférées de $\beta \oplus E$ sont :
    - $Y_1'' = \{\rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\}$ et
    - $Y_2'' = \{\rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ x \rightarrow\}$;
  - pour $m = \text{BO}$, parmi les sous-bases m-préférées de $\beta \oplus E$, on trouve :
    - $Y_1'' = \{\rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\}$,
    - $Y_2'' = \{\rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ x \rightarrow\}$;
  - pour $m = \text{E}^*$, les sous-bases m-préférées de $\beta \oplus E$ sont :
    - $Y_1'' = C_n(\{\rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\})$ et
    - $Y_2'' = C_n(\{\rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ x \rightarrow\})$;

pour $(\alpha \vee \beta) \oplus E$, on obtient :
  - $\forall m \in \{\text{INCL}, \text{LEX}\}$ les sous-bases m-préférées de $(\alpha \vee \beta) \oplus E$ sont :
    - $Y_1''' = \{\rightarrow \alpha, \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\}$ et
    - $Y_2''' = \{\rightarrow \alpha, \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ x \rightarrow\}$;
  - pour $m = \text{BO}$, parmi les sous-bases m-préférées de $(\alpha \vee \beta) \oplus E$, on trouve :
    - $Y_1''' = \{\rightarrow \alpha, \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\}$,
    - $Y_2''' = \{\rightarrow \alpha, \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ x \rightarrow\}$;
  - pour $m = \text{E}^*$, les sous-bases m-préférées de $(\alpha \vee \beta) \oplus E$ sont :
    - $Y_1''' = C_n(\{\rightarrow \alpha, \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta \rightarrow \gamma, x\ ;\ \rightarrow x\})$ et

[37] Je cite ici, parmi les sous-bases BO-préférées de E, les sous-bases maximales consistantes (voir la propriété 2.6.3 page 37). Remarquons alors que si ces sous-bases n'infèrent pas une formule alors aucune autre sous-base BO-préférée n'inférera cette formule. Cette remarque s'applique très souvent dans les contre-exemples concernant le mécanisme BO, elle ne sera donc pas répétée.

- $Y_2''' = C_n(\{\rightarrow \alpha, \beta \ ; \ x, \alpha \rightarrow \gamma \ ; \ \beta \rightarrow \gamma, x \ ; \ x \rightarrow\})$;

$\forall m$, on constate que :
  - $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$ (les Y_1') ;
  - $\exists Y''$ sous-base m-préférée de $\beta \oplus E$ telle que $Y'' \vdash \gamma$ (les Y_2'') ;
  - $\nexists Y'''$ sous-base m-préférée de $(\alpha \vee \beta) \oplus E$ telle que $Y''' \vdash \gamma$.
- Pour G-ARG-m : le contre-exemple appliqué au cas G-EXI-m s'applique ici aussi et, $\forall m$, on constate que :
  - $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$ (les Y_1') et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash (\neg\gamma)$;
  - $\exists Y''$ sous-base m-préférée de $\beta \oplus E$ telle que $Y'' \vdash \gamma$ (les Y_2'') et $\forall Y''$ sous-base m-préférée de $\beta \oplus E$, $Y'' \nvdash (\neg\gamma)$;
  - $\nexists Y'''$ sous-base m-préférée de $(\alpha \vee \beta) \oplus E$ telle que $Y''' \vdash \gamma$ et $\forall Y'''$ sous-base m-préférée de $(\alpha \vee \beta) \oplus E$, $Y''' \nvdash (\neg\gamma)$.

□

4.3.2.6 La monotonie rationnelle

Rappelons que la propriété de monotonie rationnelle se définit par $\dfrac{\alpha \mid\!\sim \gamma \ ; \alpha \not\mid\!\sim \neg\beta}{\alpha \wedge \beta \mid\!\sim \gamma}$

Rappel 4.3.2 *Quel que soit $m \in \{$*INCL*, $E^*\}$, la relation d'inférence non-monotone G-*UNI*-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30,* ne vérifie pas *la propriété de monotonie rationnelle.*

Preuve : Voir section 4.3.1 page 153. □

Rappel 4.3.3 *Quel que soit $m \in \{$*BO*, *LEX*$\}$, la relation d'inférence non-monotone G-*UNI*-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété de monotonie rationnelle.*

Preuve : Voir section 4.3.1 page 153. □

Théorème 4.3.9 *Pour $p =$ *EXI*, quel que soit $m \in \{$*BO*, *INCL*, *LEX*, $E^*\}$, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété de monotonie rationnelle.*

Preuve : Elle utilise le lemme suivant :

Lemme 4.3.1 $\forall m \in \{$BO, INCL, LEX, $E^*\}$ *un mécanisme de sélection, soient α et β deux formules, soit Y une sous-base m-préférée de $\alpha \oplus E$ qui est β-consistante alors $Y \cup \{\beta\}$ est une sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$.*

Preuve :

- Pour $m =$ INCL ou LEX, d'après les théorèmes de caractérisation 2.6.3 page 33 et 2.6.7 page 40, on sait que $Y = \{\alpha\} \cup S$ avec S sous-ensemble de E, S α-consistant et S m-préféré parmi les sous-bases de E α-consistantes ; sachant que Y est β-consistante, on obtient que S est $\alpha \wedge \beta$-consistant, donc S est m-préféré parmi les sous-bases de E $\alpha \wedge \beta$-consistantes ; en effet, si ne n'était pas le cas, on aurait un sous-ensemble S' m-préféré à S qui serait $\alpha \wedge \beta$-consistant, donc α-consistant et donc S ne serait pas m-préféré parmi les sous-bases α-consistantes.

- Pour $m =$ Bo, d'après le théorème de caractérisation 2.6.4 page 37, on sait que $Y = \{\alpha\} \cup S$ avec $S \subseteq E$, S α-consistant et $N(\alpha \oplus E) \subseteq S$; sachant que Y est β-consistante, on obtient que S est $\alpha \wedge \beta$-consistant et que $N(\alpha \oplus E)$ est $\alpha \wedge \beta$-consistant ; donc $N((\alpha \wedge \beta) \oplus E) = N(\alpha \oplus E)$ et $Y \cup \{\beta\}$ est une sous-base préférée de $(\alpha \wedge \beta) \oplus E$.
- Pour $m = \mathrm{E}^*$, comme je me restreins au cas des défauts super-normaux, je peux utiliser la propriété donnée dans [Bre89b] qui prouve, dans ce cas là, qu'il est toujours possible de trouver une stratification de la base permettant d'obtenir les mêmes résultats avec le mécanisme Incl que ceux obtenus avec le mécanisme E^* ; je pose ainsi E la base stratifiée composée des deux strates E_0 et E_1 avec $E_0 = W$ consistant et $E_1 =$ {les conséquents de chaque défaut de D} ; à chaque Y sous-base incl-préférée de $\alpha \oplus E$ correspond une extension Y_E de $(W \cup \{\alpha\}, D)$ telle que $Y_E = C_n(Y)$; donc si Y_E est β-consistante alors Y est β-consistante ; sachant que la propriété est vraie pour Incl, on obtient que $Y \cup \{\beta\}$ est incl-préférée de $(\alpha \wedge \beta) \oplus E$, donc $Y_E \cup \{\beta\}$ est une extension de $(\alpha \wedge \beta) \oplus (W, D)$.

□

$\forall m \in \{$Bo, Incl, Lex, $\mathrm{E}^*\}$, $\alpha \mathrel{|\!\sim} \gamma$ et $\alpha \mathrel{|\!\not\sim} (\neg\beta) \Rightarrow \exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y \vdash \gamma$ et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash (\neg\beta)$, donc toutes les sous-bases m-préférées de $\alpha \oplus E$ sont β-consistantes $\Rightarrow \exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \cup \{\beta\} \vdash \gamma$; or par le lemme 4.3.1 page précédente, $Y' \cup \{\beta\}$ est une sous-base m-préférée de $(\alpha \wedge \beta) \oplus E \Rightarrow \exists Y''$ sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$ telle que $Y'' \vdash \gamma \Rightarrow (\alpha \wedge \beta) \mathrel{|\!\sim} \gamma$ □

Théorème 4.3.10 *Pour $p =$* Arg, *quel que soit $m \in \{$*Bo, Incl, Lex, E^**$\}$, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30,* ne vérifie pas *la propriété de monotonie rationnelle.*

Preuve : Il suffit de prendre un contre-exemple ; soit la base E stratifiée en deux strates suivante :

$$\begin{array}{|cc|} x, \alpha \rightarrow \gamma & \beta, \gamma \rightarrow x \\ \hline \rightarrow x & x \rightarrow \end{array}$$

$\forall m$, le calcul des sous-bases m-préférées de E fournit les résultats suivants :

- $\forall m \in \{$Incl, Lex$\}$ les sous-bases m-préférées de E sont :
  - $Y_1 = \{x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\}$ et
  - $Y_2 = \{x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ x \rightarrow\}$;
- pour $m =$ Bo, parmi les sous-bases m-préférées de E, on trouve :
  - $Y_1 = \{x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\}$,
  - $Y_2 = \{x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ x \rightarrow\}$;
- pour $m = \mathrm{E}^*$, les sous-bases m-préférées de E sont :
  - $Y_1 = C_n(\{x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\})$ et
  - $Y_2 = C_n(\{x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ x \rightarrow\})$;

pour $\alpha \oplus E$, on obtient :

- $\forall m \in \{$Incl, Lex$\}$ les sous-bases m-préférées de $\alpha \oplus E$ sont :
  - $Y_1' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\}$ et
  - $Y_2' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ x \rightarrow\}$;

- pour m = Bo, parmi les sous-bases m-préférées de $\alpha \oplus E$, on trouve :
  - $Y'_1 = \{\rightarrow \alpha \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; \rightarrow x\}$,
  - $Y'_2 = \{\rightarrow \alpha \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; x \rightarrow\}$;
- pour m = E*, les sous-bases m-préférées de $\alpha \oplus E$ sont :
  - $Y'_1 = C_n(\{\rightarrow \alpha \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; \rightarrow x\})$ et
  - $Y'_2 = C_n(\{\rightarrow \alpha \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; x \rightarrow\})$;

pour $(\alpha \wedge \beta) \oplus E$[38], on obtient :

- $\forall m \in$ {Incl, Lex} les sous-bases m-préférées de $(\alpha \wedge \beta) \oplus E$ sont :
  - $Y''_1 = \{\rightarrow \alpha \ ; \rightarrow \beta \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; \rightarrow x\}$ et
  - $Y''_2 = \{\rightarrow \alpha \ ; \rightarrow \beta \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; x \rightarrow\}$;
- pour m = Bo, parmi les sous-bases m-préférées de $(\alpha \wedge \beta) \oplus E$, on trouve :
  - $Y''_1 = \{\rightarrow \alpha \ ; \rightarrow \beta \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; \rightarrow x\}$,
  - $Y''_2 = \{\rightarrow \alpha \ ; \rightarrow \beta \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; x \rightarrow\}$;
- pour m = E*, les sous-bases m-préférées de $(\alpha \wedge \beta) \oplus E$ sont :
  - $Y''_1 = C_n(\{\rightarrow \alpha \ ; \rightarrow \beta \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; \rightarrow x\})$ et
  - $Y''_2 = C_n(\{\rightarrow \alpha \ ; \rightarrow \beta \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; x \rightarrow\})$;

$\forall m$, on constate que :

- $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$ (les Y'_1) et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash (\neg\gamma)$;
- $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash (\neg\beta)$;
- $\exists Y''$ sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$ telle que $Y'' \vdash \gamma$ (les Y''_1) et $\exists Y''$ sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$ telle que $Y'' \vdash (\neg\gamma)$ (les Y''_2).

□

4.3.2.7 L'équivalence logique gauche

Rappelons que la propriété d'équivalence logique gauche se définit par $\dfrac{\models \alpha \leftrightarrow \beta \ ; \alpha \mid\!\sim \gamma}{\beta \mid\!\sim \gamma}$

Théorème 4.3.11 $\forall p \in$ {Uni, Exi, Arg}, $\forall m \in$ {Bo, Incl, Lex, E*}, *la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété d'équivalence logique gauche.*

Preuve :

- Pour G-Uni-m : voir section 4.3.1 page 153.
- Pour G-Exi-m : (pour G-Exi-E*, voir aussi section 4.3.1 page 153) on sait que $\alpha \mid\!\sim \gamma \Leftrightarrow \exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$;
  or, $\forall m$ Y' est de la forme $S \cup \{\alpha\}$ avec S un sous-ensemble de E m-préféré parmi les sous-ensembles α-consistants ;
  d'autre part, on sait que α est équivalent à β, donc $\forall S$ sous-ensemble de E m-préféré parmi les sous-ensembles α-consistants, S est aussi un sous-ensemble de E m-préféré parmi les sous-ensembles β-consistants, donc $S \cup \{\beta\}$ est une sous-base m-préférée de $\beta \oplus E$;
  donc $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma \Rightarrow \exists S$ sous-ensemble de E m-préféré parmi les sous-ensembles α-consistants tel que $S \cup \{\alpha\} \vdash \gamma$; or,

[38]Rappelons que je me situe dans un cadre syntaxique. Ceci implique que la formule $\alpha \wedge \beta$ est un tout indivisible, même si elle s'exprime sous la forme des deux clauses $\rightarrow \alpha$ et $\rightarrow \beta$.

α est équivalent à β, donc $S \cup \{\alpha\}$ et $S \cup \{\beta\}$ ont exactement les mêmes conséquences logiques $\Rightarrow \exists S$ sous-ensemble de E m-préféré parmi les sous-ensembles β-consistants tel que $S \cup \{\beta\} \vdash \gamma \Rightarrow \exists Y''$ sous-base m-préférée de $\beta \oplus E$ telle que $Y'' \vdash \gamma$.

- Pour G-Arg-m : on applique le même raisonnement que pour G-Exi-m ;
on sait que, α étant équivalent à β, $\forall S$, $S \cup \{\alpha\}$ et $S \cup \{\beta\}$ ont exactement les mêmes conséquences logiques, on obtient que :
$\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$ et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash \neg\gamma \Rightarrow \exists Y''$ sous-base m-préférée de $\beta \oplus E$ telle que $Y'' \vdash \gamma$ et $\forall Y''$ sous-base m-préférée de $\beta \oplus E$, $Y'' \nvdash \neg\gamma$.

□

4.3.2.8 L'affaiblissement droit

Rappelons que la propriété d'affaiblissement droit se définit par $\dfrac{\models \alpha \rightarrow \beta\ ; \gamma \mathrel{|\!\sim} \alpha}{\gamma \mathrel{|\!\sim} \beta}$

Théorème 4.3.12 $\forall p \in \{$Uni, Exi, Arg$\}$, $\forall m \in \{$Bo, Incl, Lex, E$^*\}$, *la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété d'affaiblissement droit.*

Preuve :

- Pour G-Uni-m : voir section 4.3.1 page 153.
- Pour G-Exi-m : (pour G-Exi-E*, voir aussi section 4.3.1 page 153) on sait que $\models \alpha \rightarrow \beta \Leftrightarrow \vdash \alpha \rightarrow \beta$; on sait aussi que $\gamma \mathrel{|\!\sim} \alpha \Leftrightarrow \exists Y'$ sous-base m-préférée de $\gamma \oplus E$ telle que $Y' \vdash \alpha$; donc $\exists Y'$ sous-base m-préférée de $\gamma \oplus E$ telle que $Y' \vdash \beta$.
- Pour G-Arg-m : on sait que $\models \alpha \rightarrow \beta \Leftrightarrow \vdash \alpha \rightarrow \beta$; on sait aussi que $\gamma \mathrel{|\!\sim} \alpha \Leftrightarrow \exists Y'$ sous-base m-préférée de $\gamma \oplus E$ telle que $Y' \vdash \alpha$ et $\forall Y'$ sous-base m-préférée de $\gamma \oplus E$, $Y' \nvdash (\neg\alpha)$; donc $\exists Y'$ sous-base m-préférée de $\gamma \oplus E$ telle que $Y' \vdash \beta$ et $\forall Y'$ sous-base m-préférée de $\gamma \oplus E$, $Y' \nvdash (\neg\beta)$ (ce dernier point se prouve en raisonnant par l'absurde car s'il existe Y' sous-base m-préférée de $\gamma \oplus E$ telle que $Y' \vdash \neg\beta$ alors il existe aussi Y' sous-base m-préférée de $\gamma \oplus E$ telle que $Y' \vdash \neg\alpha$, puisque $\models \alpha \rightarrow \beta \Leftrightarrow \models \neg\beta \rightarrow \neg\alpha$).

□

4.3.2.9 La coupure

Rappelons que la propriété de la coupure se définit par $\dfrac{\alpha \mathrel{|\!\sim} \beta\ ; \alpha \wedge \beta \mathrel{|\!\sim} \gamma}{\alpha \mathrel{|\!\sim} \gamma}$

Rappel 4.3.4 *Pour* $p =$ Uni, $\forall m \in \{$Bo, Incl, Lex, E$^*\}$, *la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété de la coupure.*

Preuve : voir section 4.3.1 page 153. □

Théorème 4.3.13 $\forall p \in \{$Exi, Arg$\}$, $\forall m \in \{$Bo, Incl, Lex, E$^*\}$, *la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30,* ne vérifie pas *la propriété de la coupure.*

Preuve :

- Pour G-Exi-m : il suffit de prendre un contre-exemple ; posons la base E stratifiée en deux strates suivante :

$$\begin{array}{|cc|} x, \alpha \rightarrow \beta & \alpha, \beta \rightarrow \gamma, x \\ \hline \rightarrow x & x \rightarrow \end{array}$$

  $\forall m$, le calcul des sous-bases m-préférées de E fournit les résultats suivants :
  - $\forall m \in \{\text{Incl}, \text{Lex}\}$ les sous-bases m-préférées de E sont :
    - $Y_1 = \{x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ \rightarrow x\}$ et
    - $Y_2 = \{x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
  - pour $m =$ Bo, parmi les sous-bases m-préférées de E, on trouve :
    - $Y_1 = \{x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ \rightarrow x\}$,
    - $Y_2 = \{x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
  - pour $m = \mathrm{E}^*$, les sous-bases m-préférées de E sont :
    - $Y_1 = C_n(\{x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ \rightarrow x\})$ et
    - $Y_2 = C_n(\{x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ x \rightarrow\})$;

  pour $\alpha \oplus E$, on obtient :
  - $\forall m \in \{\text{Incl}, \text{Lex}\}$ les sous-bases m-préférées de $\alpha \oplus E$ sont :
    - $Y'_1 = \{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ \rightarrow x\}$ et
    - $Y'_2 = \{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
  - pour $m =$ Bo, parmi les sous-bases m-préférées de $\alpha \oplus E$, on trouve :
    - $Y'_1 = \{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ \rightarrow x\}$,
    - $Y'_2 = \{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
  - pour $m = \mathrm{E}^*$, les sous-bases m-préférées de $\alpha \oplus E$ sont :
    - $Y'_1 = C_n(\{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ \rightarrow x\})$ et
    - $Y'_2 = C_n(\{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ x \rightarrow\})$;

  pour $(\alpha \wedge \beta) \oplus E$, on obtient :
  - $\forall m \in \{\text{Incl}, \text{Lex}\}$ les sous-bases m-préférées de $(\alpha \wedge \beta) \oplus E$ sont :
    - $Y''_1 = \{\rightarrow \alpha \ ; \ \rightarrow \beta \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ \rightarrow x\}$ et
    - $Y''_2 = \{\rightarrow \alpha \ ; \ \rightarrow \beta \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
  - pour $m =$ Bo, parmi les sous-bases m-préférées de $(\alpha \wedge \beta) \oplus E$, on trouve :
    - $Y''_1 = \{\rightarrow \alpha \ ; \ \rightarrow \beta \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ \rightarrow x\}$,
    - $Y''_2 = \{\rightarrow \alpha \ ; \ \rightarrow \beta \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
  - pour $m = \mathrm{E}^*$, les sous-bases m-préférées de $(\alpha \wedge \beta) \oplus E$ sont :
    - $Y''_1 = C_n(\{\rightarrow \alpha \ ; \ \rightarrow \beta \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ \rightarrow x\})$ et
    - $Y''_2 = C_n(\{\rightarrow \alpha \ ; \ \rightarrow \beta \ ; \ x, \alpha \rightarrow \beta \ ; \ \alpha, \beta \rightarrow \gamma, x \ ; \ x \rightarrow\})$;

  $\forall m$, on constate que :
  - $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta$ (les Y'_1) ;
  - $\exists Y''$ sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$ telle que $Y'' \vdash \gamma$ (les Y''_2) ;
  - $\nexists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$.
- Pour G-Arg-m : le contre-exemple appliqué au cas G-Exi-m s'applique ici aussi et, $\forall m$, on constate que :
  - $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta$ (les Y'_1) et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash (\neg\beta)$;
  - $\exists Y''$ sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$ telle que $Y'' \vdash \gamma$ (les Y''_2) et $\forall Y''$ sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$, $Y'' \nvdash (\neg\gamma)$;
  - $\nexists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$ et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash (\neg\gamma)$.

$\square$

4.3.2.10 La monotonie prudente

Rappelons que la propriété de la monotonie prudente se définit par $\dfrac{\alpha \mid\!\sim \beta \; ; \alpha \mid\!\sim \gamma}{\alpha \wedge \beta \mid\!\sim \gamma}$

Rappel 4.3.5 *Pour $p =$ Uni, $\forall m \in \{$Bo, Incl, Lex, E$^*\}$, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété de la monotonie prudente.*

Preuve : voir section 4.3.1 page 153. □

Théorème 4.3.14 *Pour $p =$ Exi, $\forall m \in \{$Bo, Incl, Lex, E$^*\}$, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30,* ne vérifie pas *la propriété de la monotonie prudente.*

Preuve : Il suffit de prendre un contre-exemple ; posons la base E suivante :

$x, \alpha \rightarrow \beta$	$\alpha \rightarrow \gamma, x, \beta$
$\beta \rightarrow x$	
$\rightarrow x$	$x \rightarrow$

$\forall m$, le calcul des sous-bases m-préférées de E fournit les résultats suivants :

- $\forall m \in \{$Incl, Lex$\}$ les sous-bases m-préférées de E sont :
  - $Y_1 = \{x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; \rightarrow x\}$ et
  - $Y_2 = \{x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; x \rightarrow\}$;
- pour $m =$ Bo, parmi les sous-bases m-préférées de E, on trouve :
  - $Y_1 = \{x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; \rightarrow x\}$,
  - $Y_2 = \{x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; x \rightarrow\}$;
- pour $m =$ E*, les sous-bases m-préférées de E sont :
  - $Y_1 = C_n(\{x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; \rightarrow x\})$ et
  - $Y_2 = C_n(\{x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; x \rightarrow\})$;

pour $\alpha \oplus E$, on obtient :

- $\forall m \in \{$Incl, Lex$\}$ les sous-bases m-préférées de $\alpha \oplus E$ sont :
  - $Y_1' = \{\rightarrow \alpha \; ; \; x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; \rightarrow x\}$ et
  - $Y_2' = \{\rightarrow \alpha \; ; \; x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; x \rightarrow\}$;
- pour $m =$ Bo, parmi les sous-bases m-préférées de $\alpha \oplus E$, on trouve :
  - $Y_1' = \{\rightarrow \alpha \; ; \; x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; \rightarrow x\}$,
  - $Y_2' = \{\rightarrow \alpha \; ; \; x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; x \rightarrow\}$;
- pour $m =$ E*, les sous-bases m-préférées de $\alpha \oplus E$ sont :
  - $Y_1' = C_n(\{\rightarrow \alpha \; ; \; x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; \rightarrow x\})$ et
  - $Y_2' = C_n(\{\rightarrow \alpha \; ; \; x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; x \rightarrow\})$;

pour $(\alpha \wedge \beta) \oplus E$, on obtient :

- $\forall m \in \{$Incl, Lex$\}$ la sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$ est :
  - $Y_1'' = \{\rightarrow \alpha \; ; \; \rightarrow \beta \; ; \; x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; \rightarrow x\}$
- pour $m =$ Bo, parmi les sous-bases m-préférées de $(\alpha \wedge \beta) \oplus E$, on trouve :
  - $Y_1'' = \{\rightarrow \alpha \; ; \; \rightarrow \beta \; ; \; x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x \; ; \; \rightarrow x\}$,
  - $Y_2'' = \{\rightarrow \alpha \; ; \; \rightarrow \beta \; ; \; x, \alpha \rightarrow \beta \; ; \; \alpha \rightarrow \gamma, x, \beta \; ; \; \beta \rightarrow x\}$;

- pour m = E*, les sous-bases m-préférées de $(\alpha \wedge \beta) \oplus E$ sont :
  - $Y_1'' = C_n(\{\rightarrow \alpha\ ;\ \rightarrow \beta\ ;\ x, \alpha \rightarrow \beta\ ;\ \alpha \rightarrow \gamma, x, \beta\ ;\ \beta \rightarrow x\ ;\ \rightarrow x\})$ et
  - $Y_2'' = C_n(\{\rightarrow \alpha\ ;\ \rightarrow \beta\ ;\ x, \alpha \rightarrow \beta\ ;\ \alpha \rightarrow \gamma, x, \beta\ ;\ x \rightarrow\})$;

$\forall m$, on constate que :

- $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta$ (les Y_1') ;
- $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$ (les Y_2') ;
- $\nexists Y''$ sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$ telle que $Y'' \vdash \gamma$.

$\square$

Théorème 4.3.15 *Pour p =* Arg*, $\forall m \in \{$*Bo*,* Incl*,* Lex*, E$^*\}$, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30,* ne vérifie pas *la propriété de la monotonie prudente.*

Preuve : Il suffit de prendre un contre-exemple ; posons la base E suivante :

$x, \alpha \rightarrow \beta$	$x, \alpha \rightarrow \gamma$	$\beta, \gamma \rightarrow x$
$\rightarrow x$	$x \rightarrow$	

$\forall m$, le calcul des sous-bases m-préférées de E fournit les résultats suivants :

- $\forall m \in \{$Incl, Lex$\}$ les sous-bases m-préférées de E sont :
  - $Y_1 = \{x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\}$ et
  - $Y_2 = \{x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ x \rightarrow\}$;
- pour m = Bo, parmi les sous-bases m-préférées de E, on trouve :
  - $Y_1 = \{x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\}$,
  - $Y_2 = \{x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ x \rightarrow\}$;
- pour m = E*, les sous-bases m-préférées de E sont :
  - $Y_1 = C_n(\{x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\})$ et
  - $Y_2 = C_n(\{x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ x \rightarrow\})$;

pour $\alpha \oplus E$, on obtient :

- $\forall m \in \{$Incl, Lex$\}$ les sous-bases m-préférées de $\alpha \oplus E$ sont :
  - $Y_1' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\}$ et
  - $Y_2' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ x \rightarrow\}$;
- pour m = Bo, parmi les sous-bases m-préférées de $\alpha \oplus E$, on trouve :
  - $Y_1' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\}$,
  - $Y_2' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ x \rightarrow\}$;
- pour m = E*, les sous-bases m-préférées de $\alpha \oplus E$ sont :
  - $Y_1' = C_n(\{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\})$ et
  - $Y_2' = C_n(\{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ x \rightarrow\})$;

pour $(\alpha \wedge \beta) \oplus E$, on obtient :

- $\forall m \in \{$Incl, Lex$\}$ la sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$ est :
  - $Y_1'' = \{\rightarrow \alpha\ ;\ \rightarrow \beta\ ;\ x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\}$
  - $Y_2'' = \{\rightarrow \alpha\ ;\ \rightarrow \beta\ ;\ x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ x \rightarrow\}$;
- pour m = Bo, parmi les sous-bases m-préférées de $(\alpha \wedge \beta) \oplus E$, on trouve :
  - $Y_1'' = \{\rightarrow \alpha\ ;\ \rightarrow \beta\ ;\ x, \alpha \rightarrow \beta\ ;\ x, \alpha \rightarrow \gamma\ ;\ \beta, \gamma \rightarrow x\ ;\ \rightarrow x\}$,

- $Y_2'' = \{\rightarrow \alpha \ ; \rightarrow \beta \ ; x, \alpha \rightarrow \beta \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; x \rightarrow\}$;
- pour $m = \mathrm{E}^*$, les sous-bases m-préférées de $(\alpha \wedge \beta) \oplus E$ sont :
  - $Y_1'' = C_n(\{\rightarrow \alpha \ ; \rightarrow \beta \ ; x, \alpha \rightarrow \beta \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; \rightarrow x\})$ et
  - $Y_2'' = C_n(\{\rightarrow \alpha \ ; \rightarrow \beta \ ; x, \alpha \rightarrow \beta \ ; x, \alpha \rightarrow \gamma \ ; \beta, \gamma \rightarrow x \ ; x \rightarrow\})$;

$\forall m$, on constate que :

- $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta$ (les Y_1') et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash \neg\beta$;
- $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$ (les Y_2') et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash \neg\gamma$;
- $\exists Y''$ sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$ telle que $Y'' \vdash \gamma$ (les Y_1'') et $\exists Y''$ sous-base m-préférée de $(\alpha \wedge \beta) \oplus E$ telle que $Y'' \vdash \neg\gamma$ (les Y_2'').

□

4.3.2.11 La cumulativité

Rappelons que la propriété de cumulativité se définit par $\dfrac{\alpha \mid\!\sim \beta \ ; \beta \vdash \alpha}{\alpha \mid\!\sim \gamma \Leftrightarrow \beta \mid\!\sim \gamma}$

Avant toute chose, je donne d'abord la démonstration de la propriété 4.3.1 présentée dans [GM94].

Propriété 4.3.1 *Les propriétés de coupure, monotonie prudente, équivalence logique gauche et supra-classicité impliquent la propriété de cumulativité.*

Preuve : Soient α et β, deux formules telles que $\alpha \mid\!\sim \beta$ et $\beta \vdash \alpha$.

- supposons que $\alpha \mid\!\sim \gamma$; avec $\alpha \mid\!\sim \beta$ et la monotonie prudente, on obtient que $\alpha \wedge \beta \mid\!\sim \gamma$; d'autre part, on sait que $\beta \vdash \alpha \Rightarrow \vdash \beta \rightarrow \alpha \Rightarrow \vdash (\alpha \wedge \beta) \leftrightarrow \beta$; donc en appliquant l'équivalence logique gauche, on obtient que $\beta \mid\!\sim \gamma$;
- supposons que $\beta \mid\!\sim \gamma$; avec $\beta \vdash \alpha$ et la supra-classicité, on obtient que $\beta \mid\!\sim \alpha$; donc avec $\alpha \mid\!\sim \beta$ et $\beta \mid\!\sim \gamma$, on obtient $(\alpha \wedge \beta) \mid\!\sim \gamma$; puis en appliquant la coupure, on obtient que $\alpha \mid\!\sim \gamma$.

□

Théorème 4.3.16 *Pour $p =$ Uni, quel que soit $m \in \{$Bo, Incl, Lex, $\mathrm{E}^*\}$, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété de cumulativité.*

Preuve : (pour $m =$ Incl ou Lex, voir aussi la section 4.3.1 page 153) Quel que soit $m \in \{$Bo, Incl, Lex, $\mathrm{E}^*\}$, on utilise la propriété 4.3.1. En effet, dans ce cas-là, toutes les propriétés nécessaires sont vérifiées (voir les théorèmes 4.3.2.1 page 158, 4.3.2.7 page 165, 4.3.2.9 page 166, 4.3.2.10 page 168). On peut donc en conclure que la propriété de cumulativité est vérifiée. □

Théorème 4.3.17 *Pour $p =$ Exi, quel que soit $m \in \{$Bo, Incl, Lex, $\mathrm{E}^*\}$, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30,* ne vérifie pas *la propriété de cumulativité.*

Preuve : On sait grâce à [GM94] que la propriété de cumulativité est équivalente aux propriétés de coupure et de monotonie prudente quand les postulats de base sont vérifiés.

Or, pour $p =$ EXI, $\forall m \in \{$BO, INCL, LEX, E$^*\}$, tous ces postulats de base sont vérifiés (voir les théorèmes 4.3.2.1 page 158, 4.3.2.7 page 165, 4.3.2.8 page 166, 4.3.2.12 page suivante, 4.3.2.4 page 161, 4.3.2.3 page 160, 4.3.2.13 page 173), alors que les propriétés de coupure et de monotonie prudente ne sont pas vérifiées (voir les théorèmes 4.3.2.9 page 166, 4.3.2.10 page 168). Donc la propriété de cumulativité n'est pas vérifiée. □

Théorème 4.3.18 *Pour $p =$ ARG, $\forall m \in \{$BO, INCL, LEX, E$^*\}$, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30,* ne vérifie pas *la propriété de la cumulativité.*

Preuve : Il suffit de prendre un contre-exemple ; posons la base E suivante :

$$\begin{array}{|cc|} x, \alpha \rightarrow \delta & \delta, \alpha \rightarrow \gamma, x \\ \hline \rightarrow x & x \rightarrow \end{array}$$

et posons $\beta = \delta \wedge \alpha$.

$\forall m$, le calcul des sous-bases m-préférées de E fournit les résultats suivants :

- $\forall m \in \{$INCL, LEX$\}$ les sous-bases m-préférées de E sont :
  - $Y_1 = \{x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ \rightarrow x\}$ et
  - $Y_2 = \{x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
- pour $m =$ BO, parmi les sous-bases m-préférées de E, on trouve :
  - $Y_1 = \{x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ \rightarrow x\}$,
  - $Y_2 = \{x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
- pour $m =$ E*, les sous-bases m-préférées de E sont :
  - $Y_1 = C_n(\{x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ \rightarrow x\})$ et
  - $Y_2 = C_n(\{x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ x \rightarrow\})$;

pour $\alpha \oplus E$, on obtient :

- $\forall m \in \{$INCL, LEX$\}$ les sous-bases m-préférées de $\alpha \oplus E$ sont :
  - $Y'_1 = \{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ \rightarrow x\}$ et
  - $Y'_2 = \{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
- pour $m =$ BO, parmi les sous-bases m-préférées de $\alpha \oplus E$, on trouve :
  - $Y'_1 = \{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ \rightarrow x\}$,
  - $Y'_2 = \{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
- pour $m =$ E*, les sous-bases m-préférées de $\alpha \oplus E$ sont :
  - $Y'_1 = C_n(\{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ \rightarrow x\})$ et
  - $Y'_2 = C_n(\{\rightarrow \alpha \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ x \rightarrow\})$;

pour $\beta \oplus E$, on obtient :

- $\forall m \in \{$INCL, LEX$\}$ les sous-bases m-préférées de $\beta \oplus E$ sont :
  - $Y''_1 = \{\rightarrow \beta \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ \rightarrow x\}$ et
  - $Y''_2 = \{\rightarrow \beta \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
- pour $m =$ BO, parmi les sous-bases m-préférées de $\beta \oplus E$, on trouve :
  - $Y''_1 = \{\rightarrow \beta \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ \rightarrow x\}$,
  - $Y''_2 = \{\rightarrow \beta \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ x \rightarrow\}$;
- pour $m =$ E*, les sous-bases m-préférées de $\beta \oplus E$ sont :
  - $Y''_1 = C_n(\{\rightarrow \beta \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ \rightarrow x\})$ et
  - $Y''_2 = C_n(\{\rightarrow \beta \ ; \ x, \alpha \rightarrow \delta \ ; \ \delta, \alpha \rightarrow \gamma, x \ ; \ x \rightarrow\})$;

$\forall m$, on constate que :

- $\exists Y'$ sous-base m-préferée de $\alpha \oplus E$ telle que $Y' \vdash \beta$ (les Y_1') et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash (\neg\beta)$;
- $\beta \vdash \alpha$;
- $\exists Y''$ sous-base m-préferée de $\beta \oplus E$ telle que $Y'' \vdash \gamma$ (les Y_2'') et $\forall Y''$ sous-base m-préférée de $\beta \oplus E$, $Y'' \nvdash (\neg\gamma)$;
- $\nexists Y'$ sous-base m-préferée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$ et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash (\neg\gamma)$.

$\square$

4.3.2.12 Le ET

Rappelons que la propriété du ET se définit par $\dfrac{\alpha \mid\!\sim \beta \ ; \alpha \mid\!\sim \gamma}{\alpha \mid\!\sim \beta \wedge \gamma}$

Théorème 4.3.19 $\forall p \in \{$Exi, Arg$\}$, $\forall m \in \{$Bo, Incl, Lex, E$^*\}$, *la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30,* ne vérifie pas *la propriété du ET.*

Preuve :

- Pour G-Exi-m : Il suffit de prendre un contre-exemple ; soit la base E suivante :

$x, \alpha \rightarrow \beta$	$\alpha \rightarrow \gamma, x$
$\rightarrow x$	$x \rightarrow$

  $\forall m$, le calcul des sous-bases m-préférées de E fournit les résultats suivants :
  - $\forall m \in \{$Incl, Lex$\}$ les sous-bases m-préférées de E sont :
    - $Y_1 = \{x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; \rightarrow x\}$ et
    - $Y_2 = \{x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; x \rightarrow\}$;
  - pour $m =$ Bo, parmi les sous-bases m-préférées de E, on trouve :
    - $Y_1 = \{x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; \rightarrow x\}$,
    - $Y_2 = \{x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; x \rightarrow\}$;
  - pour $m =$ E*, les sous-bases m-préférées de E sont :
    - $Y_1 = C_n(\{x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; \rightarrow x\})$ et
    - $Y_2 = C_n(\{x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; x \rightarrow\})$;

  pour $\alpha \oplus E$, on obtient :
  - $\forall m \in \{$Incl, Lex$\}$ les sous-bases m-préférées de $\alpha \oplus E$ sont :
    - $Y_1' = \{\rightarrow \alpha \ ; x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; \rightarrow x\}$ et
    - $Y_2' = \{\rightarrow \alpha \ ; x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; x \rightarrow\}$;
  - pour $m =$ Bo, parmi les sous-bases m-préférées de $\alpha \oplus E$, on trouve :
    - $Y_1' = \{\rightarrow \alpha \ ; x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; \rightarrow x\}$,
    - $Y_2' = \{\rightarrow \alpha \ ; x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; x \rightarrow\}$;
  - pour $m =$ E*, les sous-bases m-préférées de $\alpha \oplus E$ sont :
    - $Y_1' = C_n(\{\rightarrow \alpha \ ; x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; \rightarrow x\})$ et
    - $Y_2' = C_n(\{\rightarrow \alpha \ ; x, \alpha \rightarrow \beta \ ; \alpha \rightarrow \gamma, x \ ; x \rightarrow\})$;

  $\forall m$, on constate que :
  - $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta$ (les Y_1') ;
  - $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$ (les Y_2') ;
  - $\nexists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash (\beta \wedge \gamma)$.

- Pour G-Arg-m : le contre-exemple appliqué au cas G-Exi-m s'applique ici aussi et, $\forall m$, on constate que :
  - $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta$ (les Y_1') et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash (\neg\beta)$;
  - $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \gamma$ (les Y_2') et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash (\neg\gamma)$;
  - $\nexists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta \wedge \gamma$.

□

Théorème 4.3.20 *Pour p = Uni, $\forall m \in$ {Bo, Incl, Lex, E*}, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété du ET.*

Preuve : (pour m = Bo, Incl ou Lex, voir aussi section 4.3.1 page 153) La propriété du ET est dérivée des propriétés du système C (voir [KLM90]). Or, pour p = Uni, $\forall m \in$ {Bo, Incl, Lex, E*}, la relation d'inférence non-monotone G-p-m vérifie toutes ces propriétés (voir section 4.3.1 page 153). Elle vérifie donc aussi la propriété du ET.
□

4.3.2.13 La monotonie rationnelle faible

Rappelons que la propriété de monotonie rationnelle faible se définit par $\dfrac{\mid\!\sim (\alpha \to \beta)\ ;\ \mid\!\not\sim \neg\alpha}{\alpha \mid\!\sim \beta}$

Propriété 4.3.2 $\forall p \in$ {Uni, Exi, Arg}, $\forall m \in$ {Bo, Incl, Lex, E*}, *si la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 vérifie la propriété de monotonie rationnelle, alors elle vérifie aussi la propriété de monotonie rationnelle faible.*

Preuve : La monotonie rationnelle faible est un cas particulier de la monotonie rationnelle ; soit la propriété de monotonie rationnelle exprimée sous la forme suivante :

$$\frac{P \mid\!\sim R\ ;\ P \mid\!\not\sim \neg Q}{(P \wedge Q) \mid\!\sim R}$$

posons $P = vrai$, $Q = \alpha$, $R = \alpha \to \beta$, on obtient alors :

$$\frac{\mid\!\sim (\alpha \to \beta)\ ;\ \mid\!\not\sim \neg\alpha}{\alpha \mid\!\sim (\alpha \to \beta)}$$

or :

- pour p = Uni, $\alpha \mid\!\sim (\alpha \to \beta) \Rightarrow \forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \vdash (\alpha \to \beta)$ $\Rightarrow \forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \cup \{\alpha\} \vdash \beta$, or toutes les Y' sous-bases m-préférées de $\alpha \oplus E$ sont de la forme $\{\alpha\} \cup S$ avec S un sous-ensemble de E, donc $Y' = Y' \cup \{\alpha\}$; on a donc $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \vdash \beta$; donc $\alpha \mid\!\sim \beta$;
- pour p = Exi, $\alpha \mid\!\sim (\alpha \to \beta) \Rightarrow \exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash (\alpha \to \beta) \Rightarrow \exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \cup \{\alpha\} \vdash \beta$, or toutes les Y' sous-bases m-préférées de $\alpha \oplus E$ sont de la forme $\{\alpha\} \cup S$ avec S un sous-ensemble de E, donc $Y' = Y' \cup \{\alpha\}$; on a donc $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta \Rightarrow \alpha \mid\!\sim \beta$;

- pour p = Arg, $\alpha \mid\!\sim (\alpha \rightarrow \beta) \Rightarrow \exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash (\alpha \rightarrow \beta)$ et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \nvdash \neg(\alpha \rightarrow \beta) \Rightarrow \exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \cup \{\alpha\} \vdash \beta$ et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, Y' consistante avec $\neg\alpha \vee \beta$, or toutes les Y' sous-bases m-préférées de $\alpha \oplus E$ sont consistantes avec α et de la forme $\{\alpha\} \cup S$ avec S un sous-ensemble de E, donc $Y' = Y' \cup \{\alpha\}$; on a donc $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta$ et $\forall Y'$ sous-base m-préférée de $\alpha \oplus E$, Y' consistante avec $\beta \Rightarrow \alpha \mid\!\sim \beta$.

□

Théorème 4.3.21 *Pour p =* Uni*, quel que soit $m \in$ {*Bo*,* Lex*} et pour p =* Exi*, quel que soit $m \in$ {*Bo*,* Incl*,* Lex*,* E**}, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30, vérifie la propriété de monotonie rationnelle faible.*

> **Preuve :** (voir aussi section 4.3.1 page 153) C'est une simple application de la propriété 4.3.2 page précédente, puisque dans tous ces cas les relations d'inférence non-monotone vérifient la monotonie rationnelle. □

Théorème 4.3.22 *Pour p =* Uni*, $\forall m \in$ {*Incl*,* E**}, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30,* ne vérifie pas *la propriété de monotonie rationnelle faible.*

Preuve : (pour m = Incl, voir aussi section 4.3.1 page 153) Il suffit de donner un contre-exemple[39] ; soit la base E suivante :

$a, b \rightarrow$		$a \rightarrow c$
$\rightarrow a$	$\rightarrow b$	$c \rightarrow$

On obtient ainsi :

- pour m = Incl, les sous-bases m-préférées de E sont :
  - $Y_1 = \{a, b \rightarrow ; a \rightarrow c ; \rightarrow a\}$;
  - $Y_2 = \{a, b \rightarrow ; a \rightarrow c ; \rightarrow b ; c \rightarrow\}$;
- pour m = E*, les sous-bases m-préférées de E sont :
  - $Y_1 = C_n\{a, b \rightarrow ; a \rightarrow c ; \rightarrow a\}$;
  - $Y_2 = C_n\{a, b \rightarrow ; a \rightarrow c ; \rightarrow b ; c \rightarrow\}$;

$\forall m$, les sous-bases m-préférées infèrent la formule $c \rightarrow a$ et il existe une sous-base m-préférée qui n'infère pas $\neg c$. Posons :

- $\alpha = c$,
- $\beta = a$.

On a ainsi $\mid\!\sim \alpha \rightarrow \beta$ et $\not\mid\!\sim (\neg\alpha)$.

$\forall m$, si on rajoute la formule α à la base E, on obtient $\alpha \not\mid\!\sim \beta$, car il existe au moins une sous-base m-préférée de $\alpha \oplus E$ qui n'infère pas β (la sous-base m-préférée contenant $\{a, b \rightarrow ; a \rightarrow c ; \rightarrow b ; \rightarrow c\}$).

□

Théorème 4.3.23 *Pour p =* Arg*, quel que soit $m \in$ {*Bo*,* Incl*,* Lex*,* E**}, la relation d'inférence non-monotone G-p-m, définie à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30,* ne vérifie pas *la propriété de monotonie rationnelle faible.*

[39] Celui utilisé pour prouver que G-Uni-Incl ne vérifie pas la monotonie rationnelle convient parfaitement (voir dans [Ben94]).

Preuve : Il suffit de prendre un contre-exemple ; soit la base E suivante :

$$\begin{array}{|cc|} x, \alpha \rightarrow \beta & \beta \rightarrow x \\ \hline \rightarrow x & x \rightarrow \end{array}$$

$\forall m$, le calcul des sous-bases m-préférées de E fournit les résultats suivants :

- $\forall m \in \{$INCL, LEX$\}$ les sous-bases m-préférées de E sont :
  - $Y_1 = \{x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ \rightarrow x\}$ et
  - $Y_2 = \{x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ x \rightarrow\}$;
- pour $m =$ BO, parmi les sous-bases m-préférées de E, on trouve :
  - $Y_1 = \{x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ \rightarrow x\}$,
  - $Y_2 = \{x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ x \rightarrow\}$;
- pour $m = \mathrm{E}^*$, les sous-bases m-préférées de E sont :
  - $Y_1 = C_n(\{x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ \rightarrow x\})$ et
  - $Y_2 = C_n(\{x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ x \rightarrow\})$;

pour $\alpha \oplus E$, on obtient :

- $\forall m \in \{$INCL, LEX$\}$ les sous-bases m-préférées de $\alpha \oplus E$ sont :
  - $Y_1' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ \rightarrow x\}$ et
  - $Y_2' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ x \rightarrow\}$;
- pour $m =$ BO, parmi les sous-bases m-préférées de $\alpha \oplus E$, on trouve :
  - $Y_1' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ \rightarrow x\}$,
  - $Y_2' = \{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ x \rightarrow\}$;
- pour $m = \mathrm{E}^*$, les sous-bases m-préférées de $\alpha \oplus E$ sont :
  - $Y_1' = C_n(\{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ \rightarrow x\})$ et
  - $Y_2' = C_n(\{\rightarrow \alpha\ ;\ x, \alpha \rightarrow \beta\ ;\ \beta \rightarrow x\ ;\ x \rightarrow\})$;

$\forall m$, on constate que :

- $\nexists Y$ sous-base m-préférée de E telle que $Y \vdash (\neg\alpha)$, donc $\not\mid\!\sim \neg\alpha$;
- $\exists Y$ sous-base m-préférée de E telle que $Y \vdash (\alpha \rightarrow \beta)$ (les Y_1) et $\forall Y$ sous-base m-préférée de E, $Y \nvdash \neg(\alpha \rightarrow \beta)$, donc $\mid\!\sim (\alpha \rightarrow \beta)$;
- $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$ telle que $Y' \vdash \beta$ (les Y_1') et $\exists Y'$ sous-base m-préférée de $\alpha \oplus E$, $Y' \vdash (\neg\beta)$ (les Y_2''), donc $\alpha \not\mid\!\sim \beta$.

□

4.3.3 Conclusion sur les résultats

Tous les résultats rappelés et démontrés dans cette section sont récapitulés dans les tableaux 4.6 page suivante et 4.7 page 177.

On peut s'attacher à identifier des liens entre les divers résultats présentés.

Ces liens sont de différents types :

- les similarités ;
- les différences ;
- comment se situer par rapport aux classifications existantes (celles proposées dans [KLM90] ou dans [GM94]) ?
- quelle relation d'inférence non-monotone choisir quand on souhaite que le raisonnement modélisé vérifie un ensemble de propriétés donné ?

Rappelons que tous les résultats de cette section sont donnés par rapport à un ensemble de propriétés que je me suis fixée (voir la liste de ces propriétés dans la section 3.4 page 55).

PROPRIÉTÉS	Système P: réflexivité	Système P: équivalence logique gauche	Système P: affaiblissement droit	Système P: coupure	Système P: monotonie prudente	Système P: OU	monotonie rationnelle
G-UNI-BO	♣	♣	♣	♣	♣	♣	♣
G-UNI-INCL	♣	♣	♣	♣	♣	♣	♢
G-UNI-LEX	♣	♣	♣	♣	♣	♣	♣
G-UNI-E*	♣	♣	♣	♣	♣	♣	♢
G-EXI-BO	♣	♣	♣	♢	♢	♢	♣
G-EXI-INCL	♣	♣	♣	♢	♢	♢	♣
G-EXI-LEX	♣	♣	♣	♢	♢	♢	♣
G-EXI-E*	♣	♣	♣	♢	♢	♢	♣
G-ARG-BO	♣	♣	♣	♢	♢	♢	♢
G-ARG-INCL	♣	♣	♣	♢	♢	♢	♢
G-ARG-LEX	♣	♣	♣	♢	♢	♢	♢
G-ARG-E*	♣	♣	♣	♢	♢	♢	♢

Tableau 4.6 : Récapitulatif des résultats du point de vue de Kraus, Lehmann et Magidor [(♣ = propriété vérifiée, ♢ = propriété non vérifiée)]

PROPRIÉTÉS	Postulats étendus de AGM									
	Postulats de base									
	supra-classicité	équivalence logique gauche	affaiblissement droit	ET	monotonie rationnelle faible	conditionnalisation faible	préservation de la consistance	cumulativité	monotonie rationnelle	OU
G-Uni-Bo	♣	♣	♣	♣	♣	♣	♣	♣	♣	♣
G-Uni-Incl	♣	♣	♣	♣	♢	♣	♣	♣	♢	♣
G-Uni-Lex	♣	♣	♣	♣	♣	♣	♣	♣	♣	♣
G-Uni-E*	♣	♣	♣	♣	♢	♣	♣	♣	♢	♣
G-Exi-Bo	♣	♣	♣	♢	♣	♣	♣	♢	♣	♢
G-Exi-Incl	♣	♣	♣	♢	♣	♣	♣	♢	♣	♢
G-Exi-Lex	♣	♣	♣	♢	♣	♣	♣	♢	♣	♢
G-Exi-E*	♣	♣	♣	♢	♣	♣	♣	♢	♣	♢
G-Arg-Bo	♣	♣	♣	♢	♢	♣	♣	♢	♢	♢
G-Arg-Incl	♣	♣	♣	♢	♢	♣	♣	♢	♢	♢
G-Arg-Lex	♣	♣	♣	♢	♢	♣	♣	♢	♢	♢
G-Arg-E*	♣	♣	♣	♢	♢	♣	♣	♢	♢	♢

Tableau 4.7 : Récapitulatif des résultats du point de vue de Gärdenfors et Makinson
[(♣ = propriété vérifiée, ♢ = propriété non vérifiée)]

4.3.3.1 Les similarités

Propriété 4.3.3 $\forall m \in \{\text{Bo}, \text{Incl}, \text{Lex}, \text{E}^*\}$, *les relations d'inférence* G-Exi-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 vérifient toutes le même ensemble de propriétés.*

Propriété 4.3.4 $\forall m \in \{\text{Bo}, \text{Incl}, \text{Lex}, \text{E}^*\}$, *les relations d'inférence* G-Arg-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 vérifient toutes le même ensemble de propriétés.*

Propriété 4.3.5 $\forall m \in \{\text{Incl}, \text{E}^*\}$, *les relations d'inférence* G-Uni-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 vérifient toutes le même ensemble de propriétés.*

Propriété 4.3.6 $\forall m \in \{\text{Bo}, \text{Lex}\}$, *les relations d'inférence* G-Uni-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 vérifient toutes le même ensemble de propriétés.*

4.3.3.2 Les différences

Propriété 4.3.7 $\forall m \in \{\text{Bo}, \text{Lex}\}$, *les relations d'inférence* G-Exi-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 ne vérifient aucune propriété de plus*[40] *que les relations d'inférence* G-Uni-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30.*

Propriété 4.3.8 $\forall m \in \{\text{Incl}, \text{E}^*\}$, *les relations d'inférence* G-Exi-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 vérifient deux propriétés que ne vérifient pas les relations d'inférence* G-Uni-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 : la monotonie rationnelle et la monotonie rationnelle faible.*

Propriété 4.3.9 $\forall m \in \{\text{Bo}, \text{Incl}, \text{Lex}, \text{E}^*\}$, *les relations d'inférence* G-Arg-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 ne vérifient aucune propriété de plus que les relations d'inférence* G-Uni-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30.*

Propriété 4.3.10 $\forall m \in \{\text{Bo}, \text{Incl}, \text{Lex}, \text{E}^*\}$, *les relations d'inférence* G-Arg-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 ne vérifient aucune propriété de plus que les relations d'inférence* G-Exi-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30.*

Propriété 4.3.11 $\forall m \in \{\text{Bo}, \text{Incl}, \text{Lex}, \text{E}^*\}$, *les relations d'inférence* G-Exi-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 vérifient deux propriétés que ne vérifient pas les relations d'inférence* G-Arg-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 : la monotonie rationnelle et la monotonie rationnelle faible (les* G-Exi-m *sont donc plus près des postulats de base de [GM94] que les* G-Arg-m*, il ne leur manque que le ET !).*

4.3.3.3 Lien avec la classification de Kraus, Lehmann et Magidor

Propriété 4.3.12 $\forall m \in \{\text{Bo}, \text{Incl}, \text{Lex}, \text{E}^*\}$, *les relations d'inférence* G-Exi-m *et* G-Arg-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 induisent des systèmes encore plus faibles que le système C (système le plus faible présenté dans [KLM90]).*

Propriété 4.3.13 $\forall m \in \{\text{Bo}, \text{Incl}, \text{Lex}, \text{E}^*\}$, *les relations d'inférence* G-Uni-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 vérifient les propriétés du système P de [KLM90]. Ce sont des relations d'inférence préférentielles.*

Propriété 4.3.14 $\forall m \in \{\text{Bo}, \text{Lex}\}$, *les relations d'inférence* G-Uni-m *définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 vérifient les propriétés du système P de [KLM90] plus la monotonie rationnelle. Ce sont des relations d'inférence rationnelles.*

[40] Ce qui ne veut pas dire que les relations G-Exi-m vérifient toutes les propriétés des G-Uni-m.

4.3.3.4 Lien avec la classification de Gärdenfors et Makinson

Propriété 4.3.15 $\forall m \in \{$Bo, Incl, Lex, E$^*\}$, *les relations d'inférence G-Uni-m définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 sont des relations d'inférence non-monotone au sens de [GM94].*

Propriété 4.3.16 $\forall m \in \{$Bo, Incl, Lex, E$^*\}$, *les relations d'inférence G-Exi-m et G-Arg-m définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 ne sont pas des relations d'inférence non-monotone au sens de [GM94]* (il manque le ET).

Propriété 4.3.17 $\forall m \in \{$Bo, Lex$\}$, *les relations d'inférence G-Uni-m définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 vérifient l'ensemble des postulats de base de [GM94].*

Propriété 4.3.18 $\forall m \in \{$Incl, E$^*\}$, *les relations d'inférence G-Uni-m définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 ne vérifient pas l'ensemble des postulats de base de [GM94].*

Propriété 4.3.19 $\forall m \in \{$Bo, Lex$\}$, *les relations d'inférence G-Uni-m définies à l'aide des définitions 2.6.1 page 30 et 2.6.2 page 30 vérifient l'ensemble étendu des postulats de [GM94]. Ce sont des relations d'inférence comparatives.*

4.3.3.5 Choix en fonction d'un ensemble de propriétés

Propriété 4.3.20 *Pour respecter toutes les propriétés du système C de [KLM90], il faut au moins une relation du type G-Uni-m, $\forall m \in \{$Bo, Incl, Lex, E$^*\}$.*

Propriété 4.3.21 *Pour respecter toutes les propriétés du système P de [KLM90], il faut au moins une relation du type G-Uni-m, $\forall m \in \{$Bo, Incl, Lex, E$^*\}$.*

Propriété 4.3.22 *Pour respecter toutes les propriétés du système P de [KLM90] et la monotonie rationnelle, il faut au moins une relation du type G-Uni-m, avec $m \in \{$Bo, Lex$\}$.*

Propriété 4.3.23 *Pour respecter tous les postulats de base de [GM94], il faut au moins une relation du type G-Uni-m, $\forall m \in \{$Bo, Lex$\}$.*

Propriété 4.3.24 *Pour respecter tous les postulats étendus de [GM94], il faut au moins une relation du type G-Uni-m, avec $m \in \{$Bo, Lex$\}$.*

Rappelons aussi les résultats déjà existants.

Propriété 4.3.25 *(voir [BCD$^+$93]) L'ensemble des relations G-Uni-Lex est exactement l'ensemble des relations comparatives*[41].

Propriété 4.3.26 *(voir [BCD$^+$93]) L'ensemble des relations G-Uni-Incl est strictement inclus dans l'ensemble des relations préférentielles et contient strictement l'ensemble des relations comparatives.*

Les résultats de [BCD$^+$93] me permettent de représenter les relations G-Uni-Lex et G-Uni-Incl dans le schéma 4.3 page suivante.

[41] C'est-à-dire les relations définies par l'ensemble étendu des postulats de Gärdenfors et Makinson.

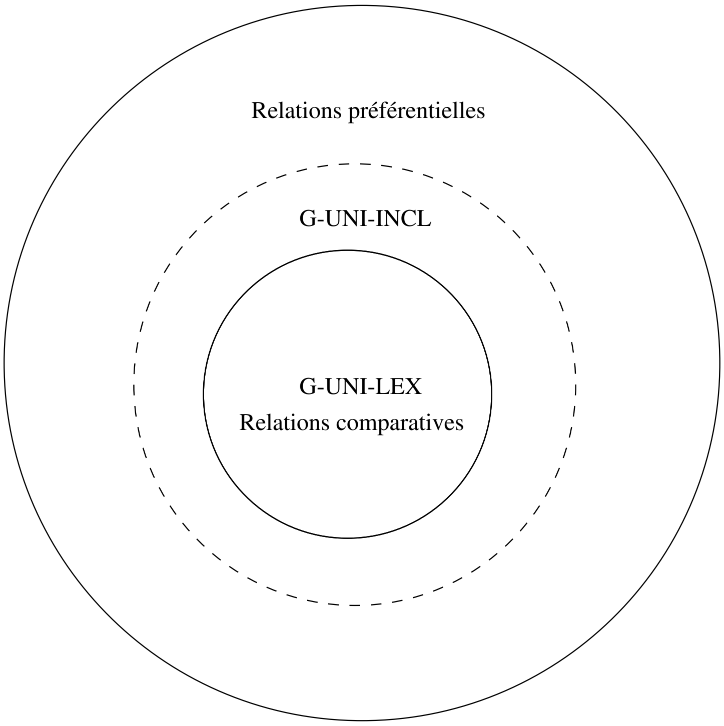


Figure 4.3 : Lien entre quelques relations G-UNI-m (point de vue des propriétés)

4.4 Réunion des trois critères de comparaison théorique

Remarquons tout d'abord que le lien entre la prudence et les propriétés de déduction a déjà été exploré par Brass en ce qui concerne les relations G-UNI-E* et G-EXI-E* (voir [Bra93]).

Ainsi, il donne les théorèmes 4.4.1 et 4.4.2.

Théorème 4.4.1 *La relation G-UNI-E* est la plus prudente, parmi les relations d'inférence définies avec le mécanisme de génération E*, qui vérifie les propriétés de réflexivité, affaiblissement droit, équivalence logique gauche, ET, OU et CWA.*

Théorème 4.4.2 *La relation G-EXI-E* est la plus prudente, parmi les relations d'inférence définies avec le mécanisme de génération E*, qui vérifie les propriétés de réflexivité, affaiblissement droit, CLC, DA, CWA et IMD.*

Les propriétés CWA, CLC, DA et IMD sont définies dans [Bra93] et sont liées pour la plupart au formalisme des défauts. Elles ne seront donc pas exposées ici.

Ces théorèmes prouvent que la relation d'inférence étudiée R_P est à la fois saine et complète pour l'ensemble de propriétés de déduction donné P : R_P vérifie P et $\forall R$, une relation d'inférence définie sur le même mécanisme de génération d'extensions que R, si R vérifie P alors R_P est plus prudente que R.

Il m'est difficile de fournir le même type de résultats puisque je manipule plusieurs sortes de mécanismes de génération et que les sous-bases ainsi générées ne sont pas toujours comparables.

Toutefois, le théorème 4.4.3, qui est une version affaiblie de ce type de résultat, est directement issu des résultats des sections 4.3 page 153 et 4.2 page 151.

Théorème 4.4.3 *La relation G-UNI-BO est la plus prudente, parmi les relations d'inférence G-p-m avec $p \in \{$UNI, EXI, ARG$\}$ et $m \in \{$BO, INCL, LEX, E*$\}$, qui vérifie l'ensemble des propriétés de déduction du système P et l'ensemble des postulats étendus.*

Dans cette section je vais confronter les différents points de vue sur les relations d'inférence non-monotone. Ce faisant, je fais la synthèse des résultats obtenus sur les relations de classe 1 (les *p-m*) et sur celles de classe 2 (les *G-p-m*). Je limite donc mon analyse aux mécanismes $m \in \{$Bo, Incl, Lex, E*$\}$ et je réduis mon domaine aux seules relations *G-p-m* qui sont les plus générales pour les mécanismes choisis.

J'exprimerai chaque point de vue sous la forme d'un pré-ordre partiel (voir les tableaux 4.8 pour la complexité, 4.9 page suivante et 4.10 page 185 pour la prudence – deux conventions différentes, 4.11 page 186 pour les propriétés) puis je synthétiserai ces tableaux en un tableau unique 4.12 page 187 ou 4.13 page 188 (suivant la convention choisie pour la prudence) qui peut se représenter sous forme d'un graphe (voir les figures 4.4 page 183 et 4.5 page 184).

La technique utilisée est celle de Pareto :

Définition 4.4.1 *Soit* $(R_1, \ldots, R_n)$ *un ensemble de n relations d'ordre ou de pré-ordre, soient* X *et* Y *deux objets à comparer dans le cadre des n relations, on dira que* X *est Pareto-supérieur à* Y *(noté* $X > Y$*) ssi* $(\forall i,\ XR_iY)$ *et* $(\exists j$ *tel qu'on n'a pas* $YR_jX)$.

4.4.1 La complexité

Je présente dans le tableau 4.8 la relation de pré-ordre entre les relations d'inférence induite par les résultats de complexité donnés dans la section 4.1 page 73.

$\ddots$ Y $\ddots$ X $\ddots$	*G*-Uni-Bo	*G*-Exi-Bo	*G*-Arg-Bo	*G*-Uni-Incl	*G*-Exi-Incl	*G*-Arg-Incl	*G*-Uni-Lex	*G*-Exi-Lex	*G*-Arg-Lex	*G*-Uni-E*	*G*-Exi-E*
G-Exi-Bo	$<$										
G-Arg-Bo	$<$	$<$									
G-Uni-Incl	$<$	$<$	$>$								
G-Exi-Incl	$<$	$=$	$>$	NC							
G-Arg-Incl	$<$	$<$	$=$	$<$	$<$						
G-Uni-Lex	$=$	$>$	$>$	$>$	$>$	$>$					
G-Exi-Lex	$<$	$=$	$>$	$>$	$=$	$>$	$<$				
G-Arg-Lex	$<$	$<$	$=$	$<$	$<$	$=$	$<$	$<$			
G-Uni-E*	$<$	$<$	$>$	$=$	$<$	$>$	$<$	$<$	$>$		
G-Exi-E*	$<$	$=$	$>$	$>$	$=$	$>$	$<$	$=$	$>$	NC	
G-Arg-E*	$<$	$<$	$=$	$<$	$<$	$=$	$<$	$<$	$=$	$<$	$<$

Légende : soient X et Y des relations d'inférence,

- "$X = Y$" $\mapsto$ X et Y ont la même complexité,
- "$X > Y$" $\mapsto$ X a une meilleure complexité que Y (donc $Y < X$),
- "X NC Y" $\mapsto$ X et Y ne sont pas comparables.

Tableau 4.8 : Complexité des relations d'inférence non-monotone

4.4.2 La prudence

Pour le point de vue de la prudence, on utilise le graphe 4.1 page 154 afin de définir la relation de pré-ordre présentée dans les tableaux 4.9 page suivante et 4.10 page 185.

J'ai choisi de tester deux conventions différentes :

- l'élément le meilleur du point de vue de la prudence est celui qui est le plus prudent (voir tableau 4.9) ;
- l'élément le meilleur du point de vue de la prudence est celui qui est le moins prudent (voir tableau 4.10 page 185).

X \ Y	G-Uni-Bo	G-Exi-Bo	G-Arg-Bo	G-Uni-Incl	G-Exi-Incl	G-Arg-Incl	G-Uni-Lex	G-Exi-Lex	G-Arg-Lex	G-Uni-E*	G-Exi-E*
G-Exi-Bo	$<$										
G-Arg-Bo	$<$	$>$									
G-Uni-Incl	$<$	$>$	NC								
G-Exi-Incl	$<$	$>$	NC	$<$							
G-Arg-Incl	$<$	$>$	NC	$<$	$>$						
G-Uni-Lex	$<$	$>$	NC	$<$	$>$	NC					
G-Exi-Lex	$<$	$>$	NC	$<$	$>$	NC	$<$				
G-Arg-Lex	$<$	$>$	NC	$<$	$>$	NC	$<$	$>$			
G-Uni-E*	$<$	$>$	NC	$=$	$>$	$>$	$>$	$>$	$>$		
G-Exi-E*	$<$	$>$	NC	$<$	$=$	$<$	$<$	$<$	$<$	$<$	
G-Arg-E*	$<$	$>$	NC	$<$	$>$	$=$	NC	NC	NC	$<$	$>$

Légende : soient X et Y des relations d'inférence,

- "$X = Y$" $\mapsto$ X et Y sont de même prudence,
- "$X > Y$" $\mapsto$ X est plus prudente que Y (donc $Y < X$),
- "X NC Y" $\mapsto$ X et Y ne sont pas comparables.

Tableau 4.9 : Prudence des relations d'inférence non-monotone – convention 1

4.4.3 Les propriétés de déduction

Pour le point de vue des propriétés de déduction, on utilise les tableaux 4.6 page 176 et 4.7 page 177 (voir section 4.3 page 153) afin de définir la relation de pré-ordre présentée dans le tableau 4.11 page 186.

4.4.4 Les trois points de vue à la fois

Les trois points de vue réunis peuvent être représentés :

- soit par le tableau 4.12 page 187 et le graphe 4.4 page ci-contre qui sont construits à partir des tableaux 4.8 page précédente, 4.9 et 4.11 page 186 ;
- soit par le tableau 4.13 page 188 et le graphe 4.5 page 184 qui sont construits à partir des tableaux 4.8 page précédente, 4.10 page 185 et 4.11 page 186.

Remarquons que le choix d'une convention pour l'échelle de la prudence a un impact non négligeable sur le résultat final.

L'analyse des résultats obtenus met en évidence trois points particuliers :

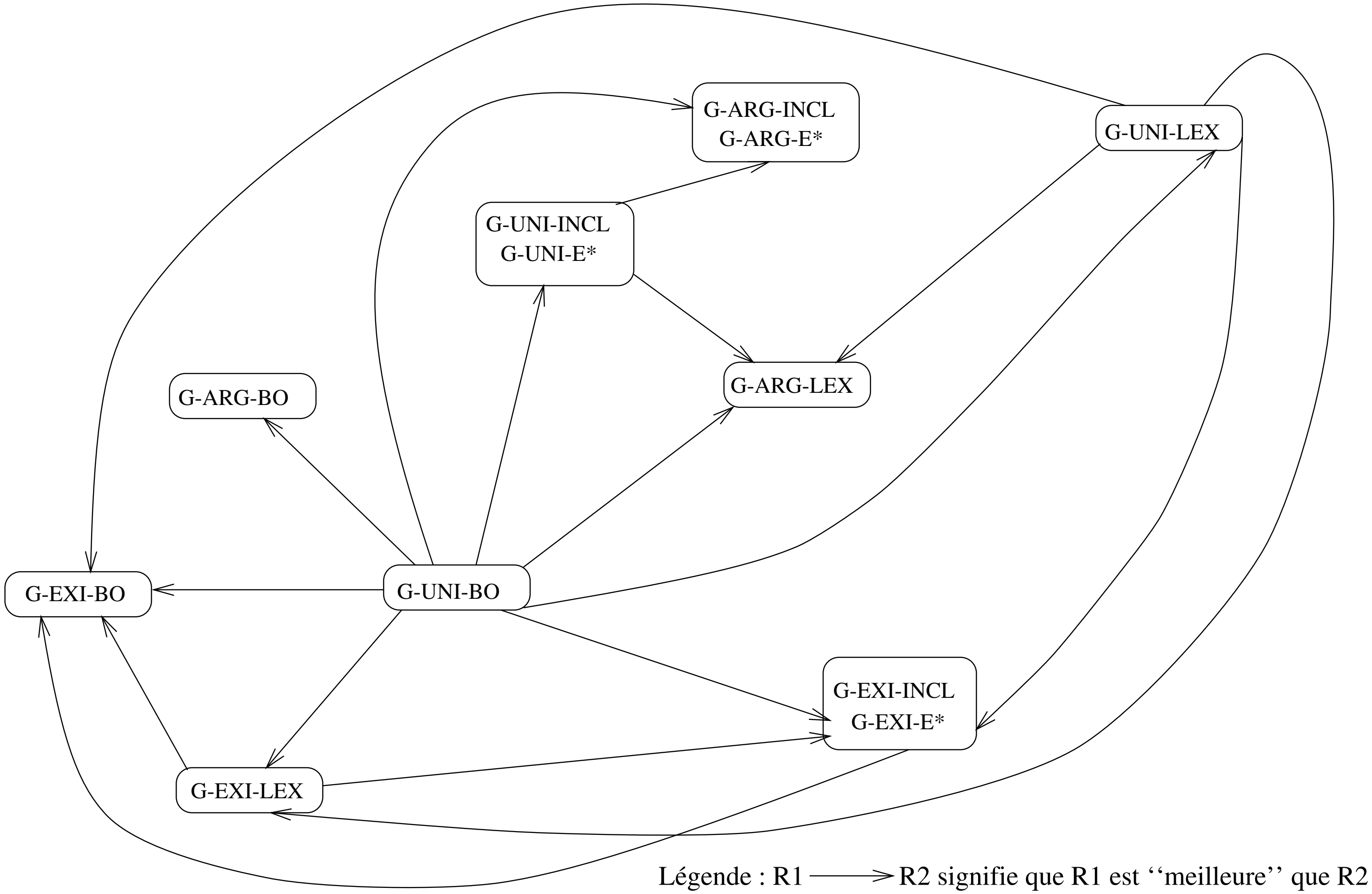


Figure 4.4 : Pré-ordre entre les relations d'inférence non-monotone (prudence – convention 1)

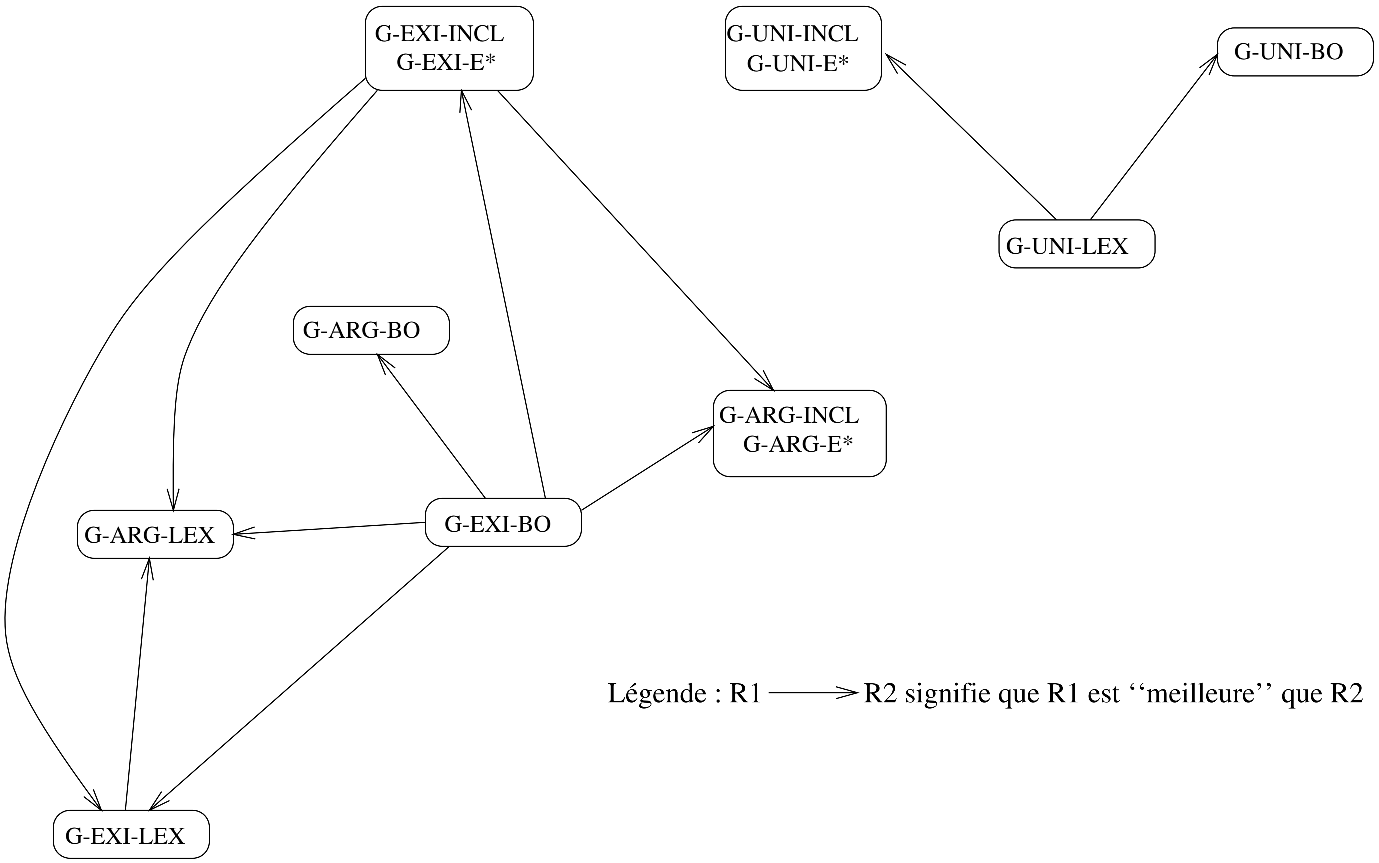


Figure 4.5 : Pré-ordre entre les relations d'inférence non-monotone (prudence – convention 2)

X \ Y	G-Uni-Bo	G-Exi-Bo	G-Arg-Bo	G-Uni-Incl	G-Exi-Incl	G-Arg-Incl	G-Uni-Lex	G-Exi-Lex	G-Arg-Lex	G-Uni-E*	G-Exi-E*
G-Exi-Bo	$>$										
G-Arg-Bo	$>$	$<$									
G-Uni-Incl	$>$	$<$	NC								
G-Exi-Incl	$>$	$<$	NC	$>$							
G-Arg-Incl	$>$	$<$	NC	$>$	$<$						
G-Uni-Lex	$>$	$<$	NC	$>$	$<$	NC					
G-Exi-Lex	$>$	$<$	NC	$>$	$<$	NC	$>$				
G-Arg-Lex	$>$	$<$	NC	$>$	$<$	NC	$>$	$<$			
G-Uni-E*	$>$	$<$	NC	$=$	$<$	$<$	$<$	$<$	$<$		
G-Exi-E*	$>$	$<$	NC	$>$	$=$	$>$	$>$	$>$	$>$	$>$	
G-Arg-E*	$>$	$<$	NC	$>$	$<$	$=$	NC	NC	NC	$>$	$<$

Légende : soient X et Y des relations d'inférence,

- "$X = Y$" $\mapsto$ X et Y sont de même prudence,
- "$X > Y$" $\mapsto$ X est moins prudente que Y (donc $Y < X$),
- "X NC Y" $\mapsto$ X et Y ne sont pas comparables.

Tableau 4.10 : Prudence des relations d'inférence non-monotone – convention 2

- l'excellence de la relation G-Uni-Bo pour chaque critère étudié dans le cas de la convention numéro 1 pour la prudence (le plus prudent est le meilleur),
- l'absence de relation globale, prenant en compte les trois critères, entre les relations G-Uni-m et les autres relations dans le cas de la convention numéro 2 pour la prudence (le moins prudent est le meilleur),
- la médiocrité des relations G-Arg-m, $\forall m \in \{$Bo, Incl, Lex, E*$\}$, dans les deux conventions choisies.

La relation G-Uni-Bo (convention 1) L'ensemble des résultats obtenus, avec la convention numéro 1 pour la prudence, semble indiquer que la relation G-Uni-Bo est la plus satisfaisante quel que soit le point de vue pris en compte :

- sa complexité est la meilleure que ce soit dans le cas général ou dans les cas particuliers étudiés (voir [CLS93]) ;
- elle vérifie toutes les propriétés étudiées ;
- elle est la plus prudente des relations utilisant la stratification de la base de croyances, donc la meilleure de ce point de vue quand on choisit la convention 1.

Malheureusement, on sait aussi que cette relation présente un inconvénient majeur appelé l'effet de noyade (voir [BCD+93]) : on ne peut pas G-Uni-Bo-inférer de conclusions à l'aide de formules situées dans une strate inférieure à celle où se produit l'inconsistance.

L'exemple classique du pingouin permet d'illustrer ce problème :

Si on a la base E suivante :

X \ Y	G-Uni-Bo	G-Exi-Bo	G-Arg-Bo	G-Uni-Incl	G-Exi-Incl	G-Arg-Incl	G-Uni-Lex	G-Exi-Lex	G-Arg-Lex	G-Uni-E*	G-Exi-E*
G-Exi-Bo	$<$										
G-Arg-Bo	$<$	$<$									
G-Uni-Incl	$<$	NC	$>$								
G-Exi-Incl	$<$	$=$	$>$	NC							
G-Arg-Incl	$<$	$<$	$=$	$<$	$<$						
G-Uni-Lex	$=$	$>$	$>$	$>$	$>$	$>$					
G-Exi-Lex	$<$	$=$	$>$	NC	$=$	$>$	$<$				
G-Arg-Lex	$<$	$<$	$=$	$<$	$<$	$=$	$<$	$<$			
G-Uni-E*	$<$	NC	$>$	$=$	NC	$>$	$<$	NC	$>$		
G-Exi-E*	$<$	$=$	$>$	NC	$=$	$>$	$<$	$=$	$>$	NC	
G-Arg-E*	$<$	$<$	$=$	$<$	$<$	$=$	$<$	$<$	$=$	$<$	$<$

Légende : soient X et Y des relations d'inférence,

- "$X = Y$" $\mapsto$ X et Y ont les mêmes propriétés (par rapport à l'ensemble de propriétés que je me suis fixée),
- "$X > Y$" $\mapsto$ X a les mêmes propriétés que Y plus quelques autres (donc $Y < X$),
- "X NC Y" $\mapsto$ X et Y ne sont pas comparables.

Tableau 4.11 : Propriétés de déduction des relations d'inférence non-monotone

X \ Y	*G*-UNI-BO	*G*-EXI-BO	*G*-ARG-BO	*G*-UNI-INCL	*G*-EXI-INCL	*G*-ARG-INCL	*G*-UNI-LEX	*G*-EXI-LEX	*G*-ARG-LEX	*G*-UNI-E*	*G*-EXI-E*
G-EXI-BO	$<$										
G-ARG-BO	$<$	NC									
G-UNI-INCL	$<$	NC	NC								
G-EXI-INCL	$<$	$>$	NC	NC							
G-ARG-INCL	$<$	NC	NC	$<$	NC						
G-UNI-LEX	$<$	$>$	NC	NC	$>$	NC					
G-EXI-LEX	$<$	$>$	NC	NC	$>$	NC	$<$				
G-ARG-LEX	$<$	NC	NC	$<$	NC	NC	$<$	NC			
G-UNI-E*	$<$	NC	NC	$=$	NC	$>$	NC	NC	$>$		
G-EXI-E*	$<$	$>$	NC	NC	$=$	NC	$<$	$<$	NC	NC	
G-ARG-E*	$<$	NC	NC	$<$	NC	$=$	NC	NC	NC	$<$	NC

Légende : soient X et Y des relations d'inférence,

- "$X = Y$" $\mapsto$ X et Y sont équivalentes,
- "$X > Y$" $\mapsto$ X est "meilleure que" Y sur les 3 critères (donc $Y < X$),
- "X NC Y" $\mapsto$ X et Y ne sont pas comparables.

Tableau 4.12 : Pré-ordre entre les relations d'inférence non-monotone (prudence – convention 1)

X \ Y	G-Uni-Bo	G-Exi-Bo	G-Arg-Bo	G-Uni-Incl	G-Exi-Incl	G-Arg-Incl	G-Uni-Lex	G-Exi-Lex	G-Arg-Lex	G-Uni-E*	G-Exi-E*
G-Exi-Bo	NC										
G-Arg-Bo	NC	$<$									
G-Uni-Incl	NC	NC	NC								
G-Exi-Incl	NC	$<$	NC	NC							
G-Arg-Incl	NC	$<$	NC	NC	$<$						
G-Uni-Lex	$>$	NC	NC	$>$	NC	NC					
G-Exi-Lex	NC	$<$	NC	NC	$<$	NC	NC				
G-Arg-Lex	NC	$<$	NC	NC	$<$	NC	NC	$<$			
G-Uni-E*	NC	NC	NC	$=$	NC	NC	$<$	NC	NC		
G-Exi-E*	NC	$<$	NC	NC	$=$	$>$	NC	$>$	$>$	NC	
G-Arg-E*	NC	$<$	NC	NC	$<$	$=$	NC	NC	NC	NC	$<$

Légende : soient X et Y des relations d'inférence,

- "$X = Y$" $\mapsto$ X et Y sont équivalentes,
- "$X > Y$" $\mapsto$ X est "meilleure que" Y sur les 3 critères (donc $Y < X$),
- "X NC Y" $\mapsto$ X et Y ne sont pas comparables.

Tableau 4.13 : Pré-ordre entre les relations d'inférence non-monotone (prudence – convention 2)

E_1	$\rightarrow P$	titi est un pingouin
	$P \rightarrow O$	un pingouin est un oiseau
E_2	$P \rightarrow \neg V$	un pingouin ne sait pas voler
	$O \rightarrow V$	un oiseau sait voler
E_3	$O \rightarrow A$	un oiseau a des ailes

Dans ce cas là, les sous-bases bo-préférées sont :

$\rightarrow P$	$\rightarrow P$	$\rightarrow P$	$\rightarrow P$	$\rightarrow P$
$P \rightarrow O$	$P \rightarrow O$	$P \rightarrow O$	$P \rightarrow O$	$P \rightarrow O$
$P \rightarrow \neg V$	$P \rightarrow \neg V$		$O \rightarrow V$	$O \rightarrow V$
$O \rightarrow A$			$O \rightarrow A$	

On constate que $E \not\mid\!\sim^{\forall,Bo} A$ alors que la formule $O \rightarrow A$ n'est pour rien dans l'inconsistance de la base E.

Notons alors que la propriété de monotonie rationnelle, qui avait été définie afin de résoudre le problème de la non-pertinence (voir [LM92]), n'y réussit donc pas complètement ; en effet, l'effet de noyade qui est un cas particulier de la non-pertinence est toujours présent alors que G-Uni-Bo vérifie la monotonie rationnelle.

Les bons résultats obtenus pour G-Uni-Bo signifient donc qu'aucun des critères étudiés ne prend en compte cet aspect de la relation. J'en conclus que l'ensemble des propriétés définissant une relation rationnelle et l'ensemble des postulats étendus ne suffisent pas à juger de l'intérêt d'une relation d'inférence non-monotone. Il faudrait trouver une nouvelle propriété qui garantirait la résolution du problème de non-pertinence, donc éviterait le problème de la noyade.

L'absence de relation globale entre les G-Uni-m et les autres (convention 2) Cette absence provient du fait que la relation de pré-ordre issue de la convention numéro 2 pour la prudence va à l'encontre de la relation de pré-ordre issue des propriétés. Nous avons donc deux critères contradictoires qui définissent deux ensembles de relations d'inférence incomparables.

Cela prouve bien l'importance du choix des relations de pré-ordre correspondant à chaque critère et composant la relation de pré-ordre globale.

Les relations G-Arg-m Commençons par remarquer que les propriétés de Gärdenfors et Makinson, tout comme les propriétés de Kraus, Lehmann et Magidor ont été écrites au départ dans l'optique d'une sémantique liée aux modèles, donc pour des relations du type G-Uni-m.

Il n'est donc pas étonnant que ce soient les relations G-Uni-m ($\forall m \in$ {Bo, Incl, Lex, E}) qui obtiennent les meilleurs résultats de ce point de vue.

D'un autre côté, il me semble que, sur l'échelle de la prudence, les extrêmes sont irréalistes :

- le principe Uni est trop exigeant, donc trop pessimiste ;
- le principe Exi n'est pas suffisamment exigeant, donc trop optimiste.

Ainsi, on se trouve devant le dilemme suivant :

- les relations G-Uni-m vérifient tout un ensemble de propriétés intéressantes mais sont trop prudentes ;
- les relations G-Arg-m ne vérifient pas suffisamment de propriétés intéressantes (propriétés définies au départ pour les G-Uni-m) mais ce sont des moyens termes intéressants du point de vue de la prudence (voir papiers sur l'argumentation [EGFK93, SL92, Cay95a]) ;
- les relations G-Exi-m vérifient à peine plus de propriétés que les G-Arg-m mais ne sont pas assez prudentes.

Ainsi, les résultats négatifs obtenus sur les relations du type G-ARG-m semblent devoir être nuancés par les faits suivants :

- les G-ARG-m sont étudiées à l'aide d'un ensemble de propriétés définies au départ pour d'autres types de relations ; une des perspectives de travail serait donc de définir un nouveau jeu de propriétés non lié à la sémantique des modèles et qui, par là-même serait utilisable pour différents principes p (par exemple, $p =$ UNI, EXI ou ARG) ; je pourrais ainsi comparer les relations G-UNI-m, G-EXI-m, G-ARG-m d'une manière un peu plus objective ;
- les G-ARG-m, qui se situent au milieu de l'échelle de la prudence, apportent un plus très net du point de vue du réalisme de la relation d'inférence non-monotone ; malheureusement cet aspect positif est aussi un inconvénient du point de vue de la complexité de calcul.

Pour illustrer l'intérêt des relations G-ARG-m, reprenons l'exemple utilisé pour présenter le problème de la noyade dans le cas de G-UNI-BO. On constate que $E \mid\!\sim^{A,Bo} A$ alors que l'on a $E \mid\!\not\sim^{\forall,Bo} A$. La simple utilisation d'une conséquence argumentative, sans changer le mécanisme de génération, permet de résoudre le problème de la noyade.

D'autre part, notons qu'il existe de plus en plus d'approches du raisonnement non-monotone utilisant la notion d'argumentation (voir par exemple [EGFK93, SL92] pour des travaux récents sur ce sujet). Dans ces travaux, on considère en général qu'une conclusion est inférée si les arguments qui la supportent "ont plus de poids" que les arguments qui la contrent. Il existe un lien entre ces approches utilisant l'argumentation et les relations d'inférence non-monotone dans le cadre d'une base de croyances non stratifiée (voir [Cay95a, Cay95b])[42]. Cela me semble être une raison supplémentaire pour continuer à étudier et à classifier les relations d'inférence non-monotone se rapprochant de tels principes.

4.4.5 Conclusion générale

Cette comparaison théorique a permis de présenter un cadre général pour les relations d'inférence non-monotone, dans lequel il est possible de :

- comparer ces relations à travers différents points de vue, et ce faisant,
- mieux comprendre les différents formalismes présentés.

Désormais, on est plus apte à préciser les intérêts de la relation d'inférence choisie par rapport à un contexte d'utilisation donné. Par exemple, dans un contexte de diagnostic reposant sur l'utilisation de modèles et basé sur la consistance[43], on choisira en priorité une relation du type G-UNI-LEX, qui vérifie un ensemble de propriétés intéressant et dont la complexité est la plus faible possible ; dans un contexte de connaissances hiérarchiques, on évitera la relation G-UNI-BO à cause du problème de la noyade ; dans un contexte de fusion de bases de connaissances, on préférera les relations G-ARG-m qui savent prendre en compte différents points de vue dans l'élaboration d'un raisonnement.

Cette étude doit être prolongée dans différentes directions :

- découverte d'une nouvelle propriété caractérisant l'absence de l'effet de noyade,
- définitions de propriétés caractérisant les relations de type G-EXI-m ou G-ARG-m,
- définition d'une échelle de "réalisme" dans laquelle la relation d'inférence la meilleure serait celle dont le réalisme serait le plus significatif (la relation ne serait ni trop optimiste en évitant la déduction de trop de formules, ni trop pessimiste en inférant suffisamment de formules) ; ainsi, je pourrais comparer les principes UNI, EXI et ARG de manière différente et, à mon avis, meilleure qu'en utilisant l'échelle de la prudence.

[42] Par contre, à ma connaissance, il n'existe pas de travaux sur ce sujet dans le cadre d'une base de croyances stratifiée.

[43] Un diagnostic basé sur la consistance est une sous-base maximale consistante ; les formules absentes de cette sous-base correspondent aux composants en panne. Dans cette approche, si on utilise le maximum de cardinalité pour sélectionner une sous-base, on minimalise le nombre de composants en panne.

Toutes ces directions de recherche visent à définir de nouveaux outils de mesure permettant de comparer de plus en plus finement les relations d'inférence.

Une autre perspective serait de synthétiser davantage les résultats présentés ici en utilisant la suggestion proposée par François Lévy dans une communication personnelle (voir section 2.2.2 page 14) : considérer les formalismes utilisés ici comme des formalismes reposant tous sur un ordre de type “lexicographique” (au sens de la définition mathématique classique 2.2.12 page 15).

Chapitre 5

Application à des problèmes de révision

En explicitant les liens existant entre divers processus de raisonnement, on constate alors que la classification présentée dans la section 4.4 page 180, pour des problèmes d'inférence non-monotone, est applicable en grande partie à d'autres types de problèmes, par exemple la révision.

5.1 Deux problématiques liées : inférence plausible, révision de bases de croyances

De nombreux travaux sont dédiés à l'étude des liens entre les processus de révision et ceux d'inférence plausible (voir par exemple [GM89, Gär90, FL94]).

Dans le contexte d'un problème de révision, on considère toujours que c'est la connaissance de l'agent qui change et pas le monde. On note $K * \Phi$ le résultat de la révision de K par la formule Φ. Généralement, dans les travaux sur la révision, la base K est une théorie (c'est-à-dire un ensemble de formules clos pour la déduction logique classique) consistante.

Les premiers travaux sur la révision ont été réalisés par Gärdenfors et Makinson (voir [AGM85, GM89]).

Ainsi, ont été définis 8 postulats appelés postulats pour la révision ou postulats AGM et notés K^{*1} à K^{*8} :

- K^{*1} : $K * \Phi$ est une théorie ;
- K^{*2} : $\Phi \in K * \Phi$;
- K^{*3} : $K * \Phi \subseteq C_n(K, \Phi)$;
- K^{*4} : si $\neg\Phi \notin K$ alors $C_n(K, \Phi) \subseteq (K * \Phi)$;
- K^{*5} : si $K * \Phi$ est inconsistant alors Φ est la contradiction ;
- K^{*6} : si $\models (\Phi \leftrightarrow \Psi)$ alors $(K * \Phi) = (K * \Psi)$;
- K^{*7} : $(K * (\Phi \wedge \Psi)) \subseteq C_n(K * \Phi, \Psi)$;
- K^{*8} : si $\neg\Psi \notin K * \Phi$ alors $C_n(K * \Phi, \Psi) \subseteq (K * (\Phi \wedge \Psi))$;

Les premiers liens entre révision et inférence plausible sont exposés dans [GM89, FL94] au travers des théorèmes de représentation suivants :

Théorème 5.1.1 *(voir démonstration dans [GM89, FL94]) Soit K une théorie, et $*$ une opération de révision. On définit une inférence $\mid\!\sim^{K,*}$ par $\alpha \mid\!\sim^{K,*} \beta$ ssi $\beta \in K * \alpha$.*

Si $*$ vérifie les postulats AGM, alors $\mid\!\sim^{K,*}$ est une inférence rationnelle qui préserve la consistance.

Théorème 5.1.2 *(voir démonstration dans [FL94]) Soit* $\mid\!\sim$ *une relation d'inférence rationnelle qui préserve la consistance, définissons :*

$$K *_{\mid\sim} \Phi = \begin{cases} \{\Psi | \Phi \mid\!\sim \Psi\} & si\ \neg\Phi \in C_n(K) \\ C_n(K \cup \{\Phi\}) & sinon \end{cases}$$

alors $*_{\mid\sim}$ *est une opération de révision qui vérifie les postulats AGM plus un postulat appelé* K^{*9} *proposé dans [FL94] qui est le suivant :*

$$\forall K \ telle\ que\ \neg\Phi \in K, K * \Phi = K_\perp * \Phi$$

(avec $K_\perp$ *= le langage complet).*

Remarque : dans [FL94], la théorie K peut être inconsistante.

Théorème 5.1.3 *(voir démonstration dans [FL94]) Soit* $\mid\!\sim$ *une relation d'inférence rationnelle qui préserve la consistance, il existe une théorie* K *et une opération de révision* $*$ *satisfaisant les postulats d'AGM plus le postulat* K^{*9} *telle que* $\Phi \mid\!\sim \Psi$ *ssi* $\Psi \in K * \Phi$.

D'autre part, il est évident que, à cause des définitions données respectivement dans les théorèmes 5.1.1 page précédente et 5.1.2 pour l'inférence non-monotone et pour l'opération de révision, on a la propriété suivante :

Propriété 5.1.1 *Les postulats* K^{*3}, K^{*4} *et* K^{*9} *sont vrais* par définition *pour l'opération de révision issue du théorème 5.1.2. De plus, le postulat* K^{*3} *implique* par définition *la propriété de conditionnalisation faible.*

Preuve : Voyons chaque postulat l'un après l'autre :

- K^{*3} est vérifié par l'opération de révision $*_{\mid\sim}$ issue du théorème 5.1.2 car : si $\neg\Phi \in K$ alors $C_n(K, \Phi)$ = langage complet, sinon $(K *_{\mid\sim} \Phi) = C_n(K, \Phi)$; donc dans les deux cas, on a $(K *_{\mid\sim} \Phi) \subseteq C_n(K, \Phi)$;
- K^{*4} est vérifié par l'opération de révision $*_{\mid\sim}$ issue du théorème 5.1.2 car : si $\neg\Phi \notin K$ alors $(K *_{\mid\sim} \Phi) = C_n(K, \Phi)$; donc, on a $C_n(K, \Phi) \subseteq (K *_{\mid\sim} \Phi)$;
- K^{*9} est vérifié par l'opération de révision $*_{\mid\sim}$ issue du théorème 5.1.2 car : $\forall K$, si $\neg\Phi \in K$ alors $(K *_{\mid\sim} \Phi) = \{\Psi$ telle que $\Phi \mid\!\sim \Psi\}$; d'autre part, $(K_\perp *_{\mid\sim} \Phi) = \{\Psi$ telle que $\Phi \mid\!\sim \Psi\}$, puisque $K_\perp$ est le langage complet, et donc que $\neg\Phi \in K_\perp$; donc, on a $(K *_{\mid\sim} \Phi) = (K_\perp *_{\mid\sim} \Phi)$;
- quand l'opération de révision $*$ vérifie le postulat K^{*3}, alors l'inférence non-monotone $\mid\!\sim^{K,*}$ issue du théorème 5.1.1 page précédente vérifie la propriété de conditionnalisation faible car : l'hypothèse $p \mid\!\sim^{K,*} q$ est équivalente à $q \in K * p$, donc, en appliquant le postulat K^{*3}, $q \in C_n(K, \Phi)$, donc $(K \cup \{p\}) \vdash q$, donc $K \vdash (p \rightarrow q)$, donc $(p \rightarrow q) \in C_n(K)$; or $C_n(K) = K * T$ (T = tautologie) ; donc $(p \rightarrow q) \in K * T$; donc $\mid\!\sim^{K,*} (p \rightarrow q)$.

□

5.2 Quelques résultats

Ainsi, on peut extraire de la littérature sur ce sujet (essentiellement [GM89, FL94]) les tableaux 5.1 page suivante et 5.2 page ci-contre qui récapitulent les liens entre les postulats et les propriétés.

K^{*1}	$\rightarrow$	Affaiblissement droit + ET
K^{*2}	$\rightarrow$	Réflexivité
K^{*3}	$\rightarrow$	Conditionnalisation faible
K^{*4}		
K^{*5}	$\rightarrow$	Préservation de la consistance
K^{*6}	$\rightarrow$	Équivalence logique gauche
K^{*7}	$\rightarrow$	Conditionnalisation
K^{*8}	$\rightarrow$	Monotonie rationnelle

Tableau 5.1 : Liens postulats de la révision – propriétés de déduction
[$X \rightarrow Y$ signifie que le postulat X de l'opération de révision $*$ sur K implique la propriété de déduction Y pour la relation d'inférence $\mid\!\sim^{K,*}$]

Affaiblissement droit + ET	$\rightarrow$	K^{*1}
Réflexivité	$\rightarrow$	K^{*2}
	(par définition)	K^{*3}
	(par définition)	K^{*4}
Préservation de la consistance	$\rightarrow$	K^{*5}
Équivalence logique gauche	$\rightarrow$	K^{*6}
Conditionnalisation	$\rightarrow$ (avec K^{*3})	K^{*7}
Monotonie rationnelle	$\rightarrow$ (avec K^{*1})	K^{*8}
	(par définition)	K^{*9}

Tableau 5.2 : Liens propriétés de déduction – postulats de la révision
[$Y \rightarrow X$ signifie que la propriété de déduction Y pour la relation d'inférence $\mid\!\sim^{K}$ implique le postulat X pour l'opération de révision $*_{\mid\sim}$ sur K]

Les choses sont donc relativement claires en ce qui concernent les relations d'inférence rationnelle. Qu'en-t-il des relations qui ne sont pas rationnelles ? Et en particulier, quels liens peut-on établir entre relations préférentielles et révision ?

On sait déjà grâce aux théorèmes 5.1.1 page 193, 5.1.2 page 194, 5.1.3 page 194, que l'on peut associer à une relation d'inférence préférentielle qui préserve la consistance, une opération de révision qui satisfait les postulats K^{*1} à K^{*7} et K^{*9}. Ce qui nous donne le théorème suivant :

Théorème 5.2.1 *Soit $\mid\!\sim$ une relation d'inférence préférentielle qui préserve la consistance, définissons :*

$$K *_{\mid\sim} \Phi = \begin{cases} \{\Psi \mid \Phi \mid\!\sim \Psi\} & si\ \neg\Phi \in C_n(K) \\ C_n(K \cup \{\Phi\}) & sinon \end{cases}$$

alors $*_{\mid\sim}$ vérifie les postulats K^{*1} à K^{*7} et K^{*9}.

> **Preuve :** Une relation d'inférence préférentielle vérifie par définition les propriétés suivantes :
>
> - la réflexivité,
> - l'équivalence logique gauche,
> - l'affaiblissement droit,
> - la coupure,
> - la monotonie prudente,
> - le OU.
>
> Donc elle vérifie le ET et la conditionnalisation (voir propriétés 3.4.3 page 65 et 3.4.4 page 65). De plus, par hypothèse, elle vérifie la préservation de la consistance. Ainsi, en utilisant le tableau 5.1 page précédente, on montre que l'opération de révision $*_{\mid\sim}$ vérifie les sept premiers postulats AGM et K^{*9}. □

Le théorème réciproque est-il vrai ?

Soit $*$ une opération de révision qui satisfait les sept premiers postulats AGM, posons $\Phi \mid\!\sim \Psi$ ssi $\Psi \in K * \Phi$. Quelles sont les propriétés de $\mid\!\sim$?

Grâce au tableau 5.1 page précédente, on sait déjà que $\mid\!\sim$ vérifie :

- la réflexivité,
- l'équivalence logique gauche,
- l'affaiblissement droit,
- le ET,
- la préservation de la consistance,
- la conditionnalisation.

En appliquant la propriété 3.4.5 page 65, on sait que $\mid\!\sim$ vérifie aussi la propriété du OU. Il lui manque donc, pour qu'elle soit préférentielle, la monotonie prudente. Or, la propriété 3.4.7 page 66 montre que la monotonie prudente provient à la fois du ET, de la préservation de la consistance, mais aussi de la monotonie rationnelle, qui n'est pas vérifiée ici.

Il n'est donc pas possible de donner la réciproque du théorème 5.2.1 à partir des éléments exposés ci-dessus. Par contre, les travaux de Rott (voir [Rot92]) mettent en évidence une caractérisation des relations d'inférence préférentielle en terme de révision. Pour cela, Rott introduit un nouveau postulat appelé K^{*8-c} :

$$\text{Si } \Psi \in K * \Phi \text{ alors } K * \Phi \subseteq K * (\Phi \wedge \Psi)$$

Ce postulat est exactement la traduction de la monotonie prudente.

Dans [Rot92], Rott démontre alors que K^{*8} implique K^{*8-c} (étant donnés les postulats K^{*1} à K^{*6}), ce qui est le pendant de la propriété 3.4.7 page 66.

On obtient ainsi le théorème 5.2.2 qui caractérise les relations d'inférence préférentielle.

Théorème 5.2.2 *Soit $\mid\!\sim$ une relation d'inférence préférentielle qui préserve la consistance, il existe une théorie K et une opération de révision $*$ satisfaisant les sept premiers postulats d'AGM plus les postulats K^{*8-c} donné dans[Rot92] et K^{*9} donné dans [FL94] telle que $\Phi \mid\!\sim \Psi$ ssi $\Psi \in K * \Phi$.*

Il reste un dernier point à explorer : comment se situe la relation Uni-Incl à rapport aux postulats de la révision ?

Ici, les principaux résultats viennent de travaux menés par Rott dans [Rot93], dans un cadre totalement différent, puisqu'il s'agissait de travaux portant sur la contraction. Dans ce contexte particulier, il propose un nouveau postulat appelé P_8, dont l'adaptation en terme de déduction donne la chose suivante :

$$\text{Si } (A \vee B) \mid\!\sim G \text{ alors } (\{\Phi | A \mid\!\sim \Phi\} \cup \{\Phi | B \mid\!\sim \Phi\} \vdash G$$

Et, en posant $\alpha \mid\!\sim^{K,*} \beta$ ssi $\beta \in K * \alpha$, on peut aussi exprimer ce postulat en terme de postulat pour la révision (noté alors K^{*8-r}) :

$$K * (A \vee B) \subseteq C_n((K * A) \cup (K * B))$$

Lors d'une communication personnelle, Rott a suggéré que ce postulat permettrait la caractérisation de la relation Uni-Incl. Ce résultat reste à démontrer.

Pour récapituler, on obtient le tableau 5.3, dans lequel PC signifie "préservation de la consistance" et MR signifie "monotonie rationnelle".

Relations d'inférence préférentielle vérifiant PC	$\supseteq$	Relations de type Uni-Incl	$\supseteq$	Relations d'inférence rationnelle vérifiant PC : Uni-Lex
$\updownarrow$		$\updownarrow$ (?)		$\updownarrow$
Système P + PC		Système P + PC + P8		Système P + PC + MR
$\updownarrow$		$\updownarrow$		$\updownarrow$
K^{*1} à K^{*7} + K^{*8-c} + K^{*9}		K^{*1} à K^{*7} + K^{*8-c} + K^{*9} + K^{*8-r}		K^{*1} à K^{*8} + K^{*9}

Tableau 5.3 : Liens révision – inférence non-monotone

Chapitre 6

Mise en œuvre de relations d'inférence non-monotone

6.1 Liste de quelques travaux sur l'algorithmique

On distingue essentiellement quatre types de travaux sur l'algorithmique du raisonnement non-monotone qui pourraient éventuellement nous être utiles, au vu des relations étudiées dans ce document :

- tout d'abord, les travaux portant sur la logique des défauts (voir [Lév91, SR91, Sat94, Men94]) qui concernent en particulier le calcul des extensions et qui permettent donc par là-même de traiter les relations Uni-E, Exi-E et Arg-E ;
- il existe aussi un prouveur et un ATMS pour la logique possibiliste (voir [Lan91]) ; ces deux outils sont intéressants pour les relations Uni-Bo, Exi-Bo et Arg-Bo ;
- dans [BCD+93], un algorithme est proposé pour les relations de type Incl et Lex ; il présuppose la connaissance des ensembles de conflits minimaux générés par un ATMS classique ;
- enfin, dans [CRS92] est proposé un algorithme de résolution de conflits utilisable pour les relations de type T et Incl.

La logique des défauts n'étant pas au centre de cette étude, les résultats algorithmiques la concernant ne seront donc pas étudiés plus avant pour l'instant. Quant aux autres résultats, ils présupposent tous l'utilisation d'un ATMS, ce qui est coûteux du point de vue temporel (voir [Pro90] pour une étude de complexité des ATMS) et du point de vue mémoire puisque, dans le pire des cas, on peut avoir un nombre de "nogoods"[1] exponentiel, ce qui sous-entend que même un traitement linéaire sur ces ensembles prendra un temps important. Je vais donc chercher à développer des algorithmes ne nécessitant pas l'usage préalable d'un tel outil.

6.2 Les spécificités algorithmiques des relations étudiées

Afin d'être en mesure d'effectuer les meilleurs choix algorithmiques possibles, étudions les spécificités des relations d'inférence non-monotone à implémenter.

Ces relations sont définies en deux étapes :

- un mécanisme de sélection des sous-bases préférées,

[1] Un "nogood" est un ensemble de formules représentant un conflit de la base initiale qui est minimal au sens de l'inclusion. Ce mot est issu de la terminologie des ATMS.

- un principe d'inférence.

Plusieurs options sont envisageables selon que l'algorithme couple ou pas ces deux étapes. L'utilisation et les résultats des différentes méthodes de couplage dépendent à la fois de la relation traitée et de la base E étudiée. En effet, dans le cas où $E \mid\!\sim^{\exists,In} \Phi$, un couplage fort (calcul des sous-bases préférées *en même temps* que la vérification de l'inférence) permettra d'obtenir une réponse plus rapidement pour la relation Exi-Incl[2] qu'un couplage faible (calcul des sous-bases *puis* test de l'inférence). De même, on voit que, dans le cas où $E \mid\!\not\sim^{\exists,In} \Phi$, ces deux méthodes sont identiques (du point de vue temps de réponse) pour implémenter la relation Exi-Incl[3].

Une autre spécificité des relations étudiées ici est l'utilisation des sous-bases consistantes maximales pour l'inclusion. Or, ces sous-bases peuvent être calculées de deux façons possibles, soit directement, soit en passant par les ensembles minimaux de conflits (voir par exemple les algorithmes proposés dans [BCD+93, Men94]) . On distingue ainsi deux approches :

Définition 6.2.1 *(Définition informelle) Il existe au moins deux approches algorithmiques possibles pour calculer les sous-bases préférées d'une base de croyances :*

- *l'approche descendante : on part des ensembles de conflits minimaux pour calculer les sous-bases préférées ; cette méthode est intéressante à condition de disposer de ces "nogoods" ;*
- *l'approche ascendante, par opposition, consiste à calculer directement les sous-bases préférées sans passer par le calcul des "nogoods".*

Il est clair, comme je l'ai déjà signalé, que l'utilisation préalable d'un ATMS est préjudiciable du point de vue de la complexité temporelle, puisque l'algorithme de recherche des "nogoods" par un ATMS est exponentiel, même dans le cas où la base ne contient que des clauses de Horn (voir [Pro90]).

6.3 Les choix

L'étude algorithmique est structurée en fonction des points suivants :

- Quelles relations vont être implémentées ?
- L'algorithme doit-il être dédié ou pas à la relation implémentée ?
- Cet algorithme utilisera-t-il un ATMS dans une première phase ?
- À quel type de couplage correspondra-t-il ?

Le premier choix est dicté par les résultats de la comparaison théorique qui permet d'élire Uni-Bo et Uni-Lex comme étant les relations les plus "intéressantes". Connaissant d'autre part l'inconvénient majeur dont souffre Uni-Bo (le problème de non-pertinence), on a choisi comme relation à implémenter Uni-Lex.

Le second choix est lié aux contraintes d'efficacité. En effet, l'avantage d'un algorithme dédié est son efficacité pour la relation choisie. Par exemple, dans le cas de la relation Uni-Lex, un algorithme dédié calculera directement les sous-bases préférées lexicographiquement au lieu de calculer d'abord les sous-bases maximales consistantes puis d'utiliser les préférences entre les formules pour déterminer les sous-bases lex-préférées. L'algorithme dédié est certes un peu plus complexe à écrire mais il paraît aussi plus efficace puisqu'il y a en général moins de sous-bases lex-préférées que de sous-bases maximales consistantes à calculer. L'inconvénient reste bien sûr le manque de réutilisabilité d'un tel algorithme. L'essentiel des algorithmes présentés ici seront dédiés à la relation Uni-Lex. Toutefois, il semble relativement aisé d'en modifier certains afin de

[2] En effet, on arrête la recherche dès qu'on a trouvé une sous-base qui infère la formule.
[3] Il faut connaître toutes les sous-bases préférées avant de pouvoir dire qu'aucune n'infère la formule.

les adapter à d'autres relations, par exemple à EXI-LEX, ARG-LEX et aux relations utilisant les mécanismes de génération CAR et BO.

Le troisième choix, comme le second, est lié à l'efficacité, qui est défavorablement affectée par l'utilisation d'un ATMS. En conséquence, aucun des algorithmes donnés dans la suite ne travaillera à partir des ensembles minimaux de conflits.

Et pour finir, j'ai testé différents couplages : tout d'abord, le couplage le plus faible (calcul des sous-bases préférées puis test de l'inférence), puis une méthode reposant sur un couplage beaucoup plus fort, et enfin, à nouveau une technique utilisant un couplage faible.

6.4 Les algorithmes

Trois algorithmes sont proposés ici. Le premier est extrêmement naïf et repose sur l'utilisation d'un prouveur correspondant au problème SAT. Le second correspond exactement à l'algorithme donné pour la preuve d'appartenance du théorème 4.1.16 page 97, donc utilise le problème MAX-SAT-ARRAY. Le troisième s'inspire de techniques utilisées dans la conception de systèmes digitaux reposant sur la notion de diagrammes de décision binaires et a donné lieu à une publication (voir [CLSS96]).

6.4.1 L'algorithme naïf

Ce premier algorithme qui utilise le problème SAT est en fait doublement exponentiel, dans le pire des cas. Je le présente quand même pour bien voir l'impact d'un couplage faible sur l'implémentation de la relation UNI-LEX. Il permet aussi d'apprécier les retombées pratiques de l'étude théorique de complexité, en comparant cet algorithme avec l'algorithme donné dans la section 6.4.2 page suivante.

On suppose que l'on dispose d'un prouveur complet pour SAT. Remarquons que cet algorithme aurait aussi pu être testé avec un prouveur incomplet de type GSAT (voir un exemple de prouveur incomplet dans [LMS92b]). Par contre, dans ce cas, on ne peut pas être certain de résoudre le problème correctement. En effet, l'impact d'une procédure aléatoire peut être négligeable sur SAT et catastrophique pour la relation UNI-LEX qui nécessite un grand nombre de résolutions du problème SAT, qui doivent toutes être correctes. Ce sujet mérite à lui seul toute une étude qui sort du cadre de cette thèse.

Cet algorithme repose sur le parcours de toute la base E, de la strate la plus prioritaire à la strate la moins prioritaire, en retenant à chaque étape les sous-bases lex-préférées par rapport aux strates déjà explorées ; ensuite, on conclut par la vérification de l'inférence de la formule Φ par chacune des sous-bases lex-préférées.

Posons les notations suivantes :

- E : la base stratifiée initiale ;
- Φ : la formule à inférer ;
- E_i : la strate courante ;
- *Vliste* : la liste des sous-bases lex-préférées de $E_1 \cup \ldots \cup E_{i-1}$ (E_i étant la strate courante) ; on dit que *Vliste* est la "vieille" liste ;
- *Nliste* : la liste des sous-bases lex-préférées $E_1 \cup \ldots \cup E_i$ (E_i étant la strate courante) ; on dit que *Nliste* est la "nouvelle" liste ;
- *Max* : maximum des cardinaux de la strate courante des sous-bases lex-préférées.

Avec ces notations, la relation UNI-LEX peut être résolue avec l'algorithme 6.1 page suivante qui prend en entrée, la base E et la formule Φ, et qui renvoie en sortie une valeur booléenne : *vrai* si la base E infère la formule Φ par la relation UNI-LEX, *faux* sinon.

Algorithme 6.1 : UNI-LEX NAÏF $((E,<), \Phi)$
début
  Vliste ← ∅
  pour chaque strate E_i de la base E **faire**
    Nliste ← ∅
    Max ← 0
    RECHERCHER-LEX-PRÉFÉRÉES-STRATE-COURANTE (E_i, *Vliste*, *Nliste*, *Max*)
    Vliste ← *Nliste*
  pour chaque sous-base de *Vliste* **faire**
    si la sous-base n'infère pas classiquement la formule Φ **alors**
      retourner *faux*
  retourner *vrai*
fin

Dans l'algorithme 6.1, la partie RECHERCHER-LEX-PRÉFÉRÉES-STRATE-COURANTE prend en entrée la strate courante E_i, la "vieille" liste de sous-bases lex-préférées et renvoie la "nouvelle" liste et la cardinalité maximum pour cette strate. Cette procédure peut être raffinée en utilisant l'algorithme 6.2.

Algorithme 6.2 : RECHERCHER-LEX-PRÉFÉRÉES-STRATE-COURANTE (E_i, *Vliste*, *Nliste*, *Max*)
début
  si strate $E_i \neq \emptyset$ **alors**
    rajouter E_i en tant que dernière strate de chaque sous-base B de *Vliste*
    pour chaque sous-base B de *Vliste* **faire**
      supprimer B de *Vliste*
      si le cardinal de la dernière strate de B est $\geq$ à *Max* **alors**
        si B est consistante **alors**
          si $B \notin$ *Nliste* **alors**
            Max ← cardinal de la dernière strate de B
            rajouter B à *Nliste*
        sinon
          pour chaque formule ϕ de la dernière strate de B **faire**
            rajouter B privée de ϕ à *Vliste*
fin

Ces algorithmes ont été implémentés en langage C (voir [LS95b]).

6.4.2 L'algorithme utilisant MAX-SAT

Ce second algorithme qui repose sur l'utilisation du problème MAX-SAT-ARRAY n'effectue qu'un nombre polynomial d'appels à une procédure de résolution d'un problème NP-complet. Il s'agit de l'algorithme donné en preuve d'appartenance de UNI-LEX à Δ_2^p (voir théorème 4.1.16 page 97).

On suppose que l'on dispose d'un prouveur complet capable de traiter le problème MAX-SAT-ARRAY qui prend en entrée une base E et un vecteur de cardinalités maximales k pour renvoyer *vrai* quand il existe une sous-base de E dont la cardinalité (strate par strate) est égale ou supérieure que celle donnée par le vecteur k et *faux* sinon.

On utilise ici une particularité de l'ordre lexicographique : toutes les sous-bases lex-préférées ont la même cardinalité strate par strate. Rajoutons la négation de la formule en tant que dernière strate. Il suffit alors qu'il existe une sous-base lex-préférée dont la cardinalité de la dernière strate est égale à 1 pour pouvoir dire qu'il existe une sous-base lex-préférée qui n'infère pas la formule Φ, et ainsi pouvoir conclure que E n'infère pas Φ par la relation UNI-LEX.

Posons les notations suivantes :

- E : la base initiale ;
- Φ : la formule à inférer ;
- E' : base résultant du rajout de la négation de la formule en tant que dernière strate ; au niveau de l'implémentation, et contrairement à la démonstration théorique du théorème 4.1.16 page 97, il est inutile de tenir compte du fait que $\neg\Phi$ pourrait déjà être dans la base ;
- n' : nombre de strates de E' ;
- k : vecteur de nombres entiers ; à chaque composante du vecteur correspond la cardinalité de la strate correspondante dans une sous-base lex-préférée ;
- n_s : numéro de la strate courante ;
- n_f : nombre courant de formules dans la strate courante ;
- $fini$: booléen (à *vrai* quand on trouve une sous-base dont la cardinalité de chaque strate correspond au vecteur k, *faux* sinon).

Cela nous donne l'algorithme 6.3 qui peut être amélioré de deux façons :

- soit en effectuant l'étape 2 par dichotomie,
- soit en remplaçant l'étape 1 par l'utilisation d'une fonction résolvant le problème MAX-GSAT-ARRAY-OPT qui n'est plus un problème de décision mais un problème d'optimisation : il prend en entrée une base de croyances et renvoie un vecteur k correspondant au vecteur de cardinalité des sous-bases lex-préférées ; on obtient alors l'algorithme 6.4.

Algorithme 6.3 : UNI-LEX AVEC MAX-SAT $((E, <), \Phi)$
début
    $E' \leftarrow E \cup \{\neg\Phi\}$ *(* Φ placée en dernière strate *)*
    $k \leftarrow <0, 0, \ldots, 0>$ *(*vecteur de dimension n' = nombre de strates dans E' *)*
1   **pour** $n_s \leftarrow 1$ à n' **faire**
        $n_f \leftarrow$ nombre de formules dans la strate E'_{n_s}
        $fini \leftarrow faux$
2   **tant que** $(n_f \geq 0)$ et (non $fini$) **faire**
            $k[n_s] \leftarrow n_f$
            si MAX-GSAT-ARRAY (E', k) **alors**
                $fini \leftarrow vrai$
            sinon
                $n_f \leftarrow n_f - 1$
    vérifier que $k[n']$ est différent de 1
fin

Algorithme 6.4 : UNI-LEX AVEC MAX-SAT AMÉLIORÉ $((E, <), \Phi)$
début
    $E' \leftarrow E \cup \{\neg\Phi\}$
    MAX-GSAT-ARRAY-OPT (E', k)
    vérifier que $k[n']$ est différent de 1   *(* n' = nombre de strates dans E' *)*
fin

6.4.3 L'approche utilisant les BDD

L'idée d'utiliser les diagrammes de décision binaires (BDD : Binary Decision Diagram) découle des mauvais résultats obtenus en complexité. En effet, les techniques BDD sont beaucoup utilisées et particulièrement efficaces dans le domaine de la conception et du test de systèmes digitaux pour résoudre des problèmes dont la complexité est au delà de NP. Certes, l'utilisation de ces BDD implique une forte consommation en mémoire puisque les graphes correspondants sont en général de grande taille (de taille exponentielle dans le nombre de variables dans le pire des cas). Mais ainsi, en sacrifiant la place mémoire, on peut espérer améliorer le temps de réponse, la construction du graphe constituant une phase de "précompilation" de la base.

6.4.3.1 Présentation des BDD

Un BDD (voir [Lee59, Bry92]) représente une fonction booléenne ou une formule Φ sous la forme d'un graphe acyclique dirigé (DAG : directed acyclic graph) utilisant des étiquettes et ne possédant qu'une racine unique. Ce graphe peut être obtenu à partir de l'arbre de décision associé à la fonction booléenne. Il possède deux sommets puits (les "feuilles" du graphe) étiquetés 0 et 1 représentant respectivement les fonctions booléennes constantes *faux* et *vrai*. Chaque sommet non terminal est étiqueté avec une variable booléenne v apparaissant dans Φ et il en part deux arcs étiquetés *then* et *else*. Le fils *then* correspond au cas où la variable v est supposée égale à *vrai* et le fils *else* correspond au cas où la variable v est supposée égale à *faux*. Ainsi les puits 1 et 0 indiquent la valeur finale de la fonction booléenne Φ comme dans les arbres de décision.

Soit $<$ un ordre total donné sur les variables, un BDD ordonné (noté OBDD) est un BDD dans lequel tous les chemins de la source vers un puits visitent les variables dans l'ordre donné.

Le OBDD réduit (appelé ROBDD) d'une formule Φ est généralement beaucoup plus petit que l'arbre de décision correspondant, puisqu'il résulte de l'application sur celui-ci des deux règles de réduction suivantes, et ce jusqu'à stabilisation (voir l'exemple tiré de [Bry92] sur figure 6.1 page suivante) :

- élimination des sommets redondants, c'est-à-dire tels que les deux fils pointent vers le même sommet ;
- et fusion de tous les sous-graphes représentant les mêmes fonctions booléennes, c'est-à-dire fusion des sommets ayant même étiquette et mêmes fils.

Ce ROBDD est unique pour un ordre donné des variables et ceci indépendamment de l'ordre d'application des règles de réduction. Avec ce processus de réduction, on diminue la taille du BDD de manière considérable, allant même parfois jusqu'à obtenir des gains exponentiels (une tautologie sur n variables est réduite au sommet 1).

Après réduction et pour un ordre des variables donné, le DAG donne une représentation canonique de la formule Φ. Ainsi, deux formules logiquement équivalentes auront le même ROBDD.

En pratique, le ROBDD n'est pas construit en réduisant un grand arbre de décision, mais en utilisant une technique incrémentale. En effet, à chaque opérateur logique correspond un opérateur ROBDD : soient Φ et Ψ deux formules logiques, la représentation de $\Phi\langle op\rangle\Psi$ ($\langle op\rangle$ étant un opérateur logique quelconque) sous la forme d'un ROBDD sera $B_\Phi\langle op_B\rangle B_\Psi$ avec $\langle op_B\rangle$ l'opérateur ROBDD correspondant à $\langle op\rangle$, et B_f le ROBDD de la formule f. La taille du ROBDD de la formule $\Phi\langle op\rangle\Psi$ est $O(S_\Phi \times S_\Psi)$ avec S_f correspondant au nombre de sommets dans le ROBDD représentant la formule f. On peut ainsi vérifier qu'une formule en implique une autre (le ROBDD de $\Phi \rightarrow \Psi$ se réduira au puits 1 si l'implication est vraie). On peut aussi traiter la quantification universelle ou existentielle sur une variable ou sur un ensemble de variables : étant donné le BDD correspondant à la formule booléenne $f(a,b)$ (a, b étant deux ensembles de variables), on peut calculer les BDD correspondant à $\exists a f(a,b)$ et $\forall a f(a,b)$ qui sont des formules booléennes sur b. La quantification existentielle (resp. universelle) sur une variable est obtenue en remplaçant chacun des sommets

x1	x2	x3	ϕ
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

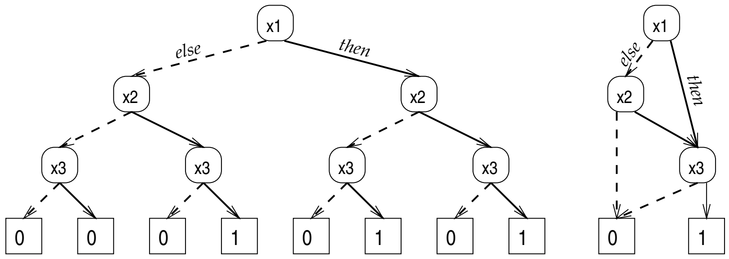


Figure 6.1 : Fonction booléenne – Arbre de décision – BDD
[de gauche à droite : la table de vérité de la formule Φ, l'arbre de décision associé et le BDD réduit ; un arc en pointillé correspond à un fils *else*, un arc en trait plein à un fils *then*]

correspondant à cette variable par le BDD représentant la disjonction (resp. conjonction) des BDD de ses fils.

Pour l'essentiel, les opérateurs sur les BDD sont de complexité temporelle linéaire dans la taille des BDD manipulés. Il est donc important de limiter cette taille autant que possible (la taille du ROBDD n'est pas nécessairement exponentielle dans le nombre de variables, bien que, dans le pire des cas, la complexité spatiale reste de l'ordre de $O(2^{|V|})$, V étant l'ensemble des variables de Φ). Des travaux sur ce sujet ont mis en évidence l'influence primordiale de l'ordre sur la taille du ROBDD. Il existe ainsi de très nombreux exemples dans des applications réelles qui peuvent être représentés de manière efficace à condition de trouver un "bon" ordre (voir [Bry92]).

6.4.3.2 Application des BDD aux relations d'inférence non-monotone

Dans un ROBDD, tout chemin de la racine à un puits représente plusieurs interprétations des variables de Φ. De même, toute interprétation correspond à un chemin unique dans le graphe. Ainsi, une interprétation satisfait la formule Φ si et seulement si le chemin correspondant dans le graphe atteint le puits 1.

Dans le cadre du raisonnement non-monotone, nous cherchons à représenter avec un ROBDD une base de formules généralement inconsistante $E = \{\phi_i\}$. Dans ce cas, le ROBDD représentant E sera réduit au puits 0. L'idée, présentée pour les CSP dynamiques dans [Jég94, JB95], est d'introduire une nouvelle variable propositionnelle h_{ϕ_i} (appelée hypothèse) pour chaque formule ϕ_i de la base E, et de modifier alors E en remplaçant chaque formule ϕ_i par la formule $h_{\phi_i} \rightarrow \phi_i$ (cette approche est classique dans le cadre des ATMS) ; la base résultante sera notée E_H, avec $H = \{h_{\phi_i}, \phi_i \in E\}$ l'ensemble des hypothèses. L'ensemble de formules ainsi obtenu est consistant et chaque interprétation le satisfaisant définit à la fois une sous-base consistante de la base initiale, puisque l'assignation à *vrai* d'une variable h_{ϕ_i} signifie que la formule ϕ_i appartient à la sous-base, et un modèle de cette sous-base. Ainsi, les sous-bases préférées correspondent à des chemins préférés dans le graphe. Cette propriété permet de ramener le problème de recherche des sous-bases préférées et de leurs modèles, au problème de recherche de "plus courts" chemins conduisant au puits 1 dans le DAG du ROBDD (cette notion de "plus court" chemin sera explicitée un peu plus loin dans le texte).

Définition d'un nouvel ordre Dans ce contexte, et quelque soit l'ordre donné $<$ sur l'ensemble V des variables apparaissant dans la base E, [JB95] montre qu'en appliquant l'ordre $<'$ obtenu en étendant $<$ par insertion des variables hypothèses h_{ϕ_i} après toutes les variables de V, la complexité spatiale du ROBDD de E_H est de $O(2^{|V|} * |H|)$, ce qui est mieux que la borne $O(2^{|V|+|H|})$ déjà connue.

Or, nous avons trouvé un autre type d'ordre encore plus efficace du point de vue occupation mémoire (voir [CLSS96]). Soit un ordre $<$ sur V, on définit l'ordre $<'$ en étendant l'ordre $<$

par insertion de chaque hypothèse h_{ϕ_i} juste après la dernière variable de ϕ_i dans l'ordre $<$. Sur l'exemple du pingouin défini avec la base stratifiée $E = E_1 \cup E_2$ et 4 formules : $\phi_1 = p$, $\phi_2 = p \rightarrow o$, $\phi_3 = p \rightarrow \neg v$ et $\phi_4 = o \rightarrow v$, avec ϕ_1 et ϕ_2 constituant la strate la plus prioritaire, et, en supposant que l'ordre initial entre les variables de E est : $p < o < v$, on rajoute les 4 variables h_1 à h_4 comme variables hypothèses pour chaque formule ϕ_i, et on obtient l'ordre : $p < h_1 < o < h_2 < v < h_3 < h_4$.

Théorème 6.4.1 *Soit une base $E = \{\phi_i\}$, soit V l'ensemble des variables utilisées dans E, $|V| = n$. On associe à chaque formule ϕ_i de E une nouvelle variable h_{ϕ_i}. Soit H l'ensemble de ces variables, appelées hypothèses, $|H| = |E| = m$. Soit $E' = \{h_{\phi_i} \rightarrow \phi_i, i = 1 \ldots m\}$.*

Étant donné un ordre sur V, on considère un ordre sur $V \cup H$ tel que toute hypothèse h_{ϕ_i} est située immédiatement après les variables apparaissant dans ϕ_i. Pour toute hypothèse h, on note n_h le nombre de variables de V qui précèdent h dans l'ordre.

Alors, le BDD ordonné réduit de la base E' utilisant un tel ordre possède un nombre de sommets non terminaux inférieur à :

$$(2^n - 1) + \sum_{h \in H} (2^{n_h})$$

Preuve : Elle s'appuie sur le fait qu'un BDD ordonné réduit est *défini* comme la fermeture par deux règles de réduction, de l'arbre de décision binaire ordonné (cf. figure 6.1 page précédente). Chacune de ces règles diminue le nombre de sommets dans le BDD d'une unité, le BDD réduit étant obtenu par application répétée de ces deux règles, dans un ordre quelconque, et jusqu'à stabilisation.

Nous allons montrer qu'une application limitée de ces règles à l'arbre de décision binaire ordonné de E' permet d'atteindre un BDD ordonné et partiellement réduit dont la taille est majorée par $(2^n - 1) + \sum_{h \in H}(2^{n_h})$. Le BDD totalement réduit étant nécessairement plus petit, cela conclura la preuve.

Considérons l'arbre de décision binaire ordonné de E'. Soit $h_{\phi_i} \in H$, l'hypothèse associée à la formule ϕ_i. Soit un des sommets $s_{h_{\phi_i}}$ correspondant à cette hypothèse dans l'arbre de décision. Étant donné que h_{ϕ_i} est située après les variables de ϕ_i, deux situations sont possibles :

1. l'interprétation des variables de ϕ_i satisfait ϕ_i : dans ce cas, la formule $h_{\phi_i} \rightarrow \phi_i$ est satisfaite indépendamment de la valeur de h_{ϕ_i}. Comme h_{ϕ_i} n'apparaît dans aucune autre formule de E', les deux sous-arbres gauche et droit de $s_{h_{\phi_i}}$ sont identiques et l'on peut supprimer le sommet $s_{h_{\phi_i}}$ (ainsi qu'une copie du sous-arbre) par application des règles de réduction ;
2. l'interprétation des variables de ϕ_i viole ϕ_i : dans ce cas, si h_{ϕ_i} est interprétée à vrai, la formule $h_{\phi_i} \rightarrow \phi_i$ est violée, indépendamment des valeurs que peuvent prendre les variables qui suivent h_{ϕ_i} dans l'ordre. Toutes les feuilles du sous-arbre de $s_{h_{\phi_i}}$ correspondant à l'interprétation à vrai de h_{ϕ_i} sont donc des sommets 0. On peut alors, par application répétée des deux règles de réduction, supprimer ce sous-arbre pour le remplacer directement par le sommet 0.

Nous possédons maintenant tous les éléments pour finir notre preuve, par récurrence : dans le cas où $m = 0$, il est clair que le nombre de sommets non terminaux est majoré par $(2^n - 1)$.

Supposons donc que notre théorème s'applique sur tout BDD représentant un ensemble de formules de la forme $\{h_\phi \rightarrow \phi\}$, convenablement ordonné, et comportant strictement moins de m hypothèses.

Considérons l'arbre de décision binaire associé à E', comportant $2^{n+m} - 1$ sommets non terminaux. Soit α la première hypothèse dans l'ordre sur $V \cup H$. D'après les remarques précédentes, il est possible d'appliquer les deux règles de réduction de façon à ce que chaque sommet s_α correspondant à α disparaisse ou ait un fils égal à 0 (sommet terminal). Le nombre de sommets non terminaux dans le BDD partiellement réduit ainsi obtenu se décompose alors en :

1. $(2^{n_\alpha} - 1)$ sommets pour les n_α variables non hypothèses qui précèdent α dans l'ordre ;
2. un maximum de 2^{n_α} sommets correspondant à l'hypothèse α ;
3. 2^{n_α} sous-arbres dont les racines correspondent à la variable qui suit α dans l'ordre.

Ces 2^{n_α} sous-arbres sont des arbres de décision binaire ordonnés. Soit I l'interprétation partielle donnée par le chemin menant de la racine de l'arbre à la racine d'un de ces sous-arbres. Quel que soit un sous-arbre parmi les 2^{n_α} situés sous le niveau de α, ce sous-arbre représente l'arbre de décision binaire de la formule E'_I obtenue à partir de E' par substitution aux variables de l'interprétation I de leur valeur de vérité dans I. Du fait de l'ordre sur $V \cup H$, E'_I contient des formules constantes 0 ou 1 et des formules de la forme $h_\phi \rightarrow \phi$. Les formules 1, toujours satisfaites, n'influent pas sur la valeur de vérité de E'_I. Donc :

1. soit E'_I contient une formule 0 et, dans ce cas, sa valeur de vérité est toujours égale à 0, le sous-arbre de décision binaire de E'_I pouvant être réduit au sommet 0 par application des règles de réduction ;
2. soit E'_I ne contient pas de formule 0 et satisfait alors toutes les conditions d'application de l'hypothèse de récurrence ; il comporte alors $m-1$ hypothèses et respecte la condition sur l'ordre (une hypothèse est située après les variables de la formule associée) ; on sait donc que l'on peut réduire le sous-arbre en un BDD ordonné et partiellement réduit dont le nombre de sommets non terminaux est majoré par $(2^{n-n_\alpha} - 1) + \sum_{h \in H - \{\alpha\}} (2^{n_h - n_\alpha})$ (c'est l'hypothèse de récurrence).

Donc, dans tous les cas, on peut donc réduire chaque sous-arbre en un BDD ordonné partiellement réduit dont le nombre de sommets non terminaux est majoré par

$$(2^{n-n_\alpha} - 1) + \sum_{h \in H - \{\alpha\}} (2^{n_h - n_\alpha})$$

L'ensemble des réductions permet donc d'aboutir à un BDD ordonné et partiellement réduit comportant un nombre de sommets non terminaux majoré par :

$$\begin{aligned}
S &= \overbrace{[2^{n_\alpha} - 1]} + \overbrace{[2^{n_\alpha}]} + \overbrace{[2^{n_\alpha}.((2^{n-n_\alpha} - 1) + \sum_{h \in H - \{\alpha\}} (2^{n_h - n_\alpha}))]} \\
&= [2^{n_\alpha} - 1] + [2^{n_\alpha}] + [2^n - 2^{n_\alpha} + \sum_{h \in H - \{\alpha\}} (2^{n_h})] \\
&= 2^{n_\alpha} - 1 + 2^n + \sum_{h \in H - \{\alpha\}} (2^{n_h}) \\
&= (2^n - 1) + \sum_{h \in H} (2^{n_h})
\end{aligned}$$

□

On notera que dans le cas où toutes les variables hypothèses sont insérées après toutes les variables de V comme le proposait [JB95], on trouve un majorant de sommets non terminaux égal à $(2^n - 1) + \sum_{h \in H} 2^n = 2^n(m+1) - 1$ et qui correspond donc au majorant indiqué dans [JB95]. Dans tous les autres cas, le majorant que nous obtenons est meilleur.

Les algorithmes Maintenant qu'on dispose d'un ROBDD de taille "optimisée", il reste à expliciter la notion de "plus court" chemin dans ce ROBDD. L'idée proposée dans [JB95] pour le traitement des CSP incohérents consiste à pondérer les arcs *else* de chaque sommet correspondant à une hypothèse. Ce poids représente le coût du rejet de cette hypothèse, donc le coût du rejet de la clause correspondante. Ainsi, de façon habituelle, la longueur d'un chemin sera définie comme étant la somme des poids des arcs de ce chemin. La relation à implémenter étant UNI-LEX, les sous-bases lex-préférées de la base E correspondent à des chemins préférés dans le ROBDD de E et un chemin préféré est un "plus court" chemin correspondant à un choix judicieux de poids permettant de prendre en compte la notion de cardinalité par strate qui est à la base de la définition de la relation UNI-LEX. Ainsi, on utilise l'algorithme 6.5 issu d'un algorithme présenté dans [DLS94] (déjà utilisé en partie au travers de l'algorithme 4.44 page 135) que nous avons repris et amélioré, afin de calculer les poids associés à chaque formule ϕ_i de la base stratifiée $E = E_1 \cup \ldots \cup E_n$ (E_1 étant la strate la plus prioritaire, E_n la strate la moins prioritaire) :

Algorithme 6.5 : CALCULPOIDS($E, <$)
début
  poids des formules de $E_n \leftarrow 1$
  pour chaque strate E_i de E_{n-1} à E_1 **faire**
    poids des formules de $E_i \leftarrow$ poids des formules de $E_{i+1} \times (|E_{i+1}| + 1)$
fin

Ainsi, avec cet algorithme, *le poids de toute formule ϕ_i de la strate E_k est strictement supérieur à la somme des poids de toutes les formules des strates moins prioritaires que E_k. Donc, les sous-bases de coût minimal sont aussi les sous-bases lex-préférées.*

Comme dans [JB95], on peut alors utiliser un algorithme linéaire en temps dérivé de l'algorithme de Bellman (voir [CLR90]) pour calculer le plus court chemin menant au puits 1 dans le DAG. Il s'agit d'une procédure récursive parcourant le BDD en profondeur d'abord qui calcule, mémorise et réutilise le coût des fils pour calculer celui du père[4] (voir l'algorithme 6.6).

Algorithme 6.6 : COÛT(F)
début
  *(***** On initialise* COÛT*(1) à 0 et* COÛT*(0) au maximum possible *****)*
  $t \leftarrow$ COÛT(*then*) *(* appel récursif si* COÛT*(then) non connu utilisation immédiate sinon *)*
  $e \leftarrow$ COÛT(*else*) *(* appel récursif si* COÛT*(else) non connu utilisation immédiate sinon *)*
  si F est une hypothèse **alors** $r \leftarrow$ coût de rejet de F
  sinon $r \leftarrow 0$
  $e \leftarrow e + r$
  COÛT(F) $\leftarrow \min(t,e)$ *(* mémorisation et retour de valeur *)*
fin

Nous avons modifié cet algorithme afin qu'il construise en même temps un nouveau ROBDD R'_E qui ne contiendra plus que les chemins les plus courts menant au puits 1 (les autres chemins sont simplement dirigés vers le puits 0). Ce nouveau ROBDD est dit "dégraissé". Cette procédure de "dégraissage" récursive utilise les versions dégraissées des fils pour calculer celle du père (voir l'algorithme 6.7 page ci-contre). On utilise pour cela une fonction essentielle de manipulation des BDD notée ITE (If Then Else) qui prend en paramètre une variable x et deux ROBDD B_f et B_g et qui retourne le ROBDD représentant la formule $(\neg x \vee f) \wedge (x \vee g)$ (si x alors f sinon g).

Le nouveau ROBDD R'_E créé avec cet algorithme peut alors être utilisé pour résoudre les problèmes utilisant le mécanisme de génération LEX et ceci à moindre coût. Ainsi, on peut dire que R'_E

[4] Le coût d'un sommet n'est calculé qu'une fois, puis on le mémorise de façon à le réutiliser quand on retombe sur ce sommet.

Algorithme 6.7 : DEG(F)
début

*(***** On initialise* COÛT*(1) à 0 et* COÛT*(0) au maximum ****)*
*(***** On initialise* DEG*(1) (version dégraissée du puits 1) *****)*
*(***** avec un pointeur vers le puits 1, et* DEG*(0) *****)*
*(***** (version dégraissée du puits 0) avec un pointeur vers le puits 0 *****)*
$t \leftarrow$ COÛT($then$)
$e \leftarrow$ COÛT($else$)
si F est une hypothèse **alors** $r \leftarrow$ coût de rejet de F
sinon $r \leftarrow 0$
$e \leftarrow e + r$
COÛT(F) $\leftarrow \min(t,e)$
si $t >$ COÛT(F) **alors** $n_t \leftarrow$ puits 0
sinon $n_t \leftarrow$ DEG($then$) *(* DEG(then) : version dégraissée du fils then *)*
si $e >$ COÛT(F) **alors** $n_e \leftarrow$ puits 0
sinon $n_e \leftarrow$ DEG($else$) *(* DEG(else) : version dégraissée du fils else *)*
DEG(F) $\leftarrow$ ITE (la variable de F, n_t, n_e)

fin

représente une version “compilée” de la base (chaque chemin de la racine au puits 1 représentant une sous-base préférée avec un modèle associé). Dans le cas de UNI-LEX, il suffit de quantifier existentiellement R'_E sur toutes les variables hypothèses (ce qui peut se faire lors du dégraissage) et l'on obtient alors un nouveau ROBDD R''_E qui représente l'ensemble de tous les modèles de toutes les sous-bases préférées de E[5]. Ainsi, la base E infère la formule Φ par la relation UNI-LEX si et seulement si cette formule est *classiquement* inférée par R''_E.

Ce processus est illustré avec l'exemple du pingouin défini avec deux strates ($E = E_1 \cup E_2$) et 4 formules : $\phi_1 = p$, $\phi_2 = p \rightarrow o$, $\phi_3 = p \rightarrow \neg v$ et $\phi_4 = o \rightarrow v$, avec ϕ_1 et ϕ_2 constituant la strate la plus prioritaire. L'ordre initial entre les variables de E est : $p < o < v$. On rajoute les 4 variables h_1 à h_4 comme variables hypothèses pour chaque formule ϕ_i. En utilisant notre technique, on obtient un nouvel ordre : $p < h_1 < o < h_2 < v < h_3 < h_4$. Puis on calcule les poids et on obtient 1 comme poids pour les formules de la strate E_2, donc pour les variables h_3 et h_4, et 3 comme poids pour les formules de la strate E_1, donc pour les variables h_1 et h_2. La figure 6.2 page suivante présente successivement :

- le ROBDD R_E, avec en gras le poids des arcs *else* (qui sont en pointillé) et en italique la longueur du plus court chemin de chaque sommet vers le puits 1 ;
- le ROBDD R'_E issu de R_E en ne sélectionnant que les chemins les plus courts, donc en redirigeant les autres vers le puits 0 ;
- le ROBDD R''_E obtenu après la quantification existentielle des variables h_1, h_2, h_3 et h_4.

On voit avec R'_E que la base E possède deux sous-bases lex-préférées qui rejettent respectivement ϕ_3 et ϕ_4. De plus, R''_E représente implicitement les deux modèles de ces deux sous-bases. Ainsi, on peut prouver o et p mais pas v.

En fait, l'implémentation BDD utilisée pour les tests est un peu particulière. Il s'agit d'un “package” réalisé par Bryant et Brace (voir [BRB90]) dans lequel la représentation du BDD s'appuie sur la notion d'*arcs inversés* : une fonction booléenne référencée via un arc inversé représente la fonction opposée (voir sur la figure 6.3 page 211 l'exemple des pingouins exprimé sous la forme d'un BDD avec arcs inversés). Ainsi, on n'a plus qu'un seul puits (le puits 1) et tous les chemins mènent à ce puits. Dans ce cas, seuls nous intéressent les plus courts chemins qui, depuis la racine, parcourent un nombre pair d'arcs inversés.

[5]En fait, il s'agit d'un cas simple de quantification puisque les variables hypothèses ont toujours pour fils *then* un sommet terminal (voir la preuve du théorème 6.4.1 page 206 concernant le nouvel ordre).

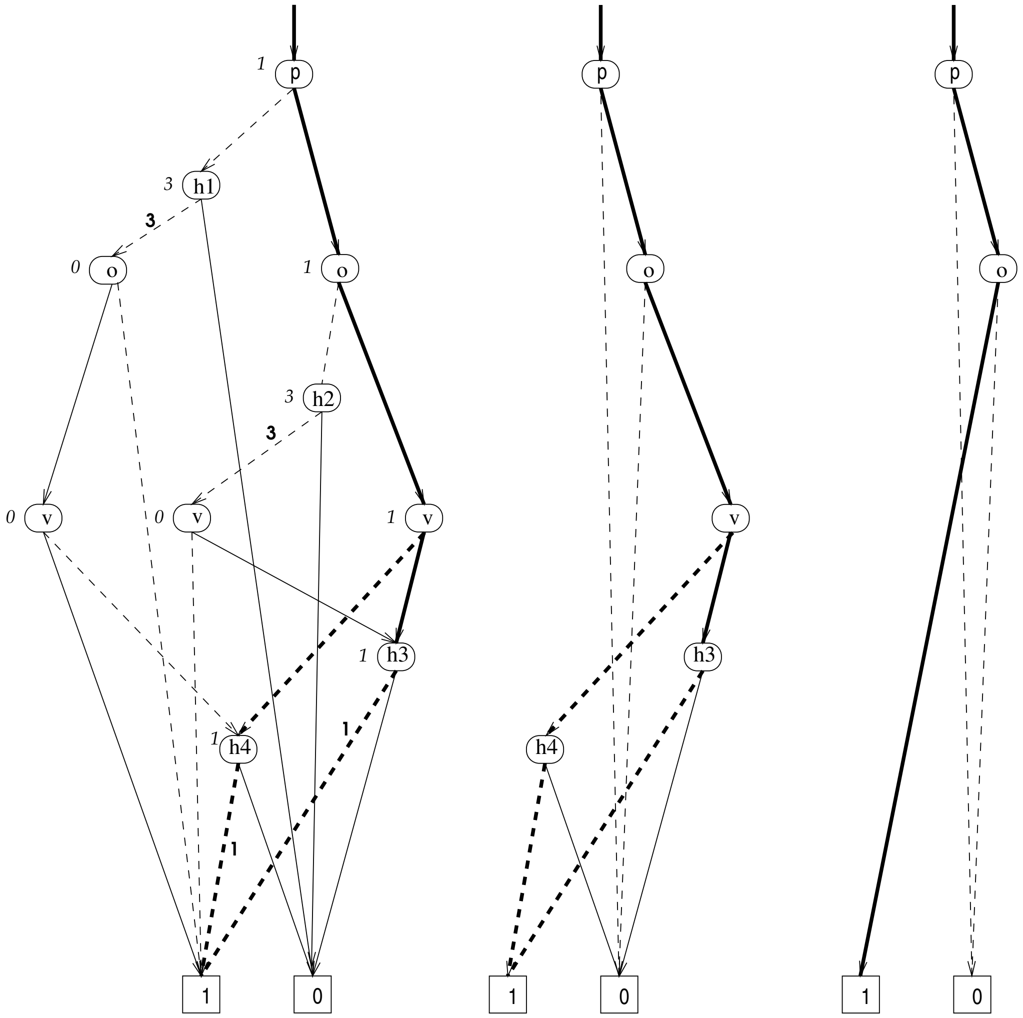
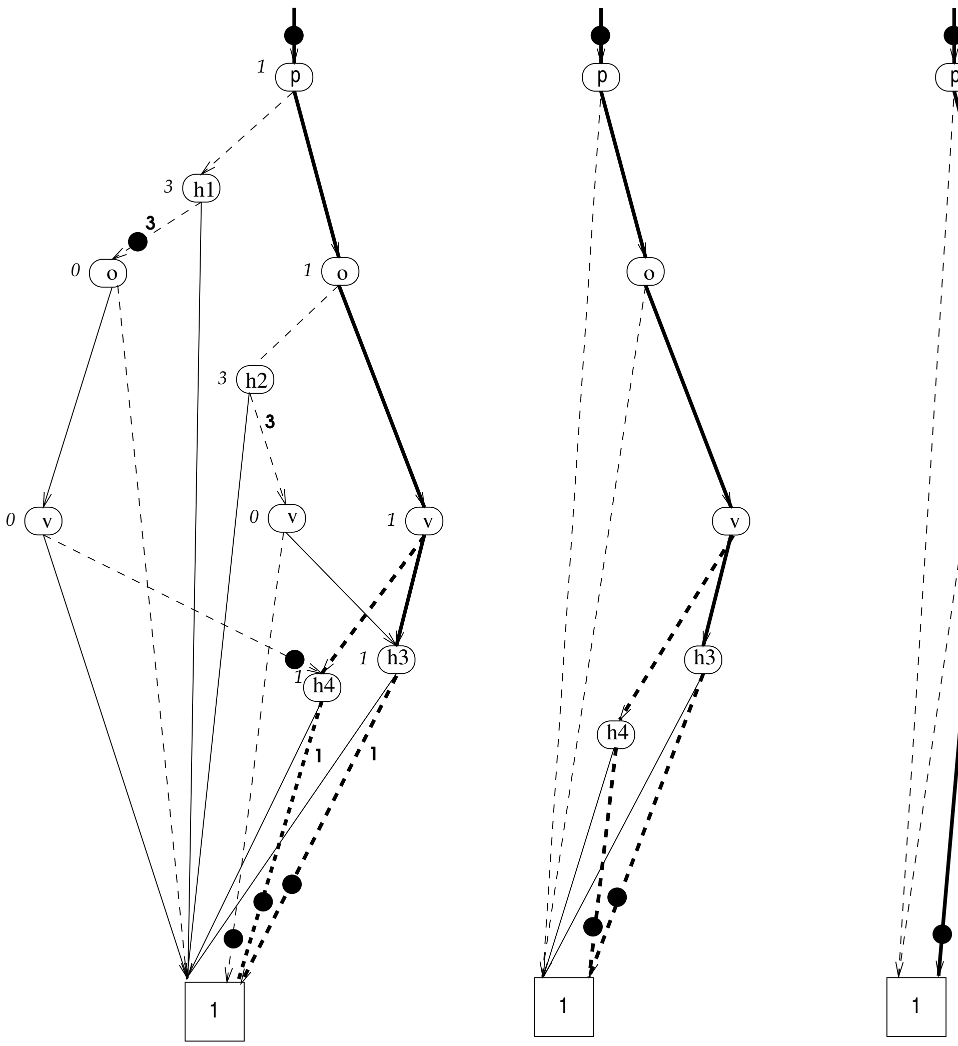


Figure 6.2 : Les ROBDD R_E, R'_E et R''_E de l'exemple sur les pingouins



Un rond noir sur un arc signifie que l'arc est inversé

Figure 6.3 : Les ROBDD avec arcs inversés R_E, R'_E et R''_E de l'exemple sur les pingouins

Dans le cas de BDD avec arcs inversés, les algorithmes donnés précédemment se compliquent, puisqu'il faut maintenir en permanence à la fois le plus court chemin vers "1" (le puits 1 atteint par un nombre *pair* d'arcs inversés) et celui vers "0" (le puits 1 atteint par un nombre *impair* d'arcs inversés). L'algorithme de calcul du plus court chemin 6.6 page 208 devient alors l'algorithme 6.8.

Algorithme 6.8 : Coût0 (F) et Coût1 (F)
début
*(***** Coût0(sommet) = chemin le plus court de ce sommet au puits 1 *****)*
*(***** passant par un nombre* impair *d'arcs inversés *****)*
*(***** Coût1(sommet) = chemin le plus court de ce sommet au puits 1 *****)*
*(***** passant par un nombre* pair *d'arcs inversés *****)*
*(***** On initialise Coût1(1) à 0 et Coût0(1) au maximum *****)*
si le fils *then* n'est pas inversé **alors** $t_0 \leftarrow$ Coût0(*then*)
sinon $t_0 \leftarrow$ Coût1(*then*)
si le fils *then* n'est pas inversé **alors** $t_1 \leftarrow$ Coût1(*then*)
sinon $t_1 \leftarrow$ Coût0(*then*)
si le fils *else* n'est pas inversé **alors** $e_0 \leftarrow$ Coût0(*else*)
sinon $e_0 \leftarrow$ Coût1(*else*)
si le fils *else* n'est pas inversé **alors** $e_1 \leftarrow$ Coût1(*else*)
sinon $e_1 \leftarrow$ Coût0(*else*)
si F est une hypothèse **alors** $r \leftarrow$ coût de rejet de F
sinon $r \leftarrow 0$
$e_0 \leftarrow e_0 + r$
$e_1 \leftarrow e_1 + r$
Coût0(F) $\leftarrow \min(t_0, e_0)$
Coût1(F) $\leftarrow \min(t_1, e_1)$
fin

Il est possible d'améliorer cet algorithme. En effet, on peut montrer que pour tout sommet soit Coût0, soit Coût1 est nul (la preuve se fait par induction : c'est vrai pour les deux puits 0 et 1, si c'est vrai pour les deux fils ce sera vrai pour le père s'il s'agit d'un nœud de type variable (à cause du *min*), et, s'il s'agit d'un nœud de type hypothèse, étant donné que l'on ne rajoute aucun coût sur le fils *then*, ce sera vrai aussi).

Quant à l'algorithme de dégraissage 6.7 page 209, dans le cas d'arcs inversés, il se transforme en l'algorithme 6.9 page suivante.

6.5 Les résultats expérimentaux

6.5.1 L'implémentation des algorithmes

Les algorithmes décrits précédemment ont été codés en langage C (voir l'implémentation des différentes méthodes dans [LS95b]).

Le prouveur utilisé pour Sat ainsi que la procédure correspondant à Max-Sat-Array ont été implémentés en langage C par Thierry Castell, étudiant en $3^{\text{ème}}$ cycle dans la même équipe que moi.

L'algorithme utilisé pour Sat est un algorithme basé sur la procédure de Davis et Putnam (voir [DP60]) utilisant l'heuristique présentée dans [DABC93]. Il s'agit de l'une des méthodes complètes les plus performantes à ce jour.

Le problème Max-Sat-Array est, quant à lui, codé par un algorithme dérivé de l'algorithme de Davis et Putnam et utilisant l'heuristique de [DABC93] avec un principe de séparation–évaluation[6]

[6]En anglais : "Branch and Bound"

Algorithme 6.9 : DEG0 (F) et DEG1 (F)
début

*(***** COÛT0(sommet) = chemin le plus court de ce sommet au puits 1 *****)*
*(***** passant par un nombre* impair *d'arcs inversés *****)*
*(***** COÛT1(sommet) = chemin le plus court de ce sommet au puits 1 *****)*
*(***** passant par un nombre* pair *d'arcs inversés *****)*
*(***** DEG0(sommet) = version dégraissée menant du sommet vers le puits 1 *****)*
*(***** passant par un nombre* impair *d'arcs inversés *****)*
*(***** DEG1(sommet) = version dégraissée menant du sommet vers le puits 1 *****)*
*(***** passant par un nombre* pair *d'arcs inversés *****)*
*(***** On initialise COÛT1(1) à 0 et COÛT0(1) au maximum *****)*
*(***** et DEG0(1) et DEG1(1) pointent vers le puits 1 *****)*
si le fils *then* n'est pas inversé **alors** $t_0 \leftarrow$ COÛT0(*then*)
sinon $t_0 \leftarrow$ COÛT1(*then*)
si le fils *then* n'est pas inversé **alors** $t_1 \leftarrow$ COÛT1(*then*)
sinon $t_1 \leftarrow$ COÛT0(*then*)
si le fils *else* n'est pas inversé **alors** $e_0 \leftarrow$ COÛT0(*else*)
sinon $e_0 \leftarrow$ COÛT1(*else*)
si le fils *else* n'est pas inversé **alors** $e_1 \leftarrow$ COÛT1(*else*)
sinon $e_1 \leftarrow$ COÛT0(*else*)
si F est une hypothèse **alors** $r \leftarrow$ coût de rejet de F
sinon $r \leftarrow 0$
$e_0 \leftarrow e_0 + r$
$e_1 \leftarrow e_1 + r$
COÛT0(F) $\leftarrow \min(t_0, e_0)$
COÛT1(F) $\leftarrow \min(t_1, e_1)$
si $t_0 >$ COÛT0(F) **alors** $n_{t_0} \leftarrow$ puits 1
sinon
  si le fils *then* n'est pas inversé **alors** $n_{t_0} \leftarrow$ DEG0(*then*)
  sinon $n_{t_0} \leftarrow$ le complément de DEG1(*then*)
si $e_0 >$ COÛT0(F) **alors** $n_{e_0} \leftarrow$ puits 1
sinon
  si le fils *else* n'est pas inversé **alors** $n_{e_0} \leftarrow$ DEG0(*else*)
  sinon $n_{e_0} \leftarrow$ le complément de DEG1(*else*)
si $t_1 >$ COÛT1(F) **alors** $n_{t_1} \leftarrow$ puits 0
sinon
  si le fils *then* n'est pas inversé **alors** $n_{t_1} \leftarrow$ DEG1(*then*)
  sinon $n_{t_1} \leftarrow$ le complément de DEG0(*then*)
si $e_1 >$ COÛT1(F) **alors** $n_{e_1} \leftarrow$ puits 0
sinon
  si le fils *else* n'est pas inversé **alors** $n_{e_1} \leftarrow$ DEG1(*else*)
  sinon $n_{e_1} \leftarrow$ le complément de DEG0(*else*)
DEG0(F) $\leftarrow$ ITE (la variable de F, n_{t_0}, n_{e_0})
DEG1(F) $\leftarrow$ ITE (la variable de F, n_{t_1}, n_{e_1})

fin

(en utilisant le nombre de clauses falsifiées après réduction de la base comme évaluation). Ici, l'idée de base est d'associer à chaque formule un poids et ainsi de maximiser non pas le nombre de formules satisfaites mais la somme des poids des formules satisfaites. Il s'agit d'une méthode complète. Cet algorithme a été développé par Thierry Castell. Notons qu'il existe dans la littérature d'autres exemples d'algorithmes pour Max-Sat, qui utilisent en particulier des méthodes incomplètes (voir par exemple [HJ90, LMS92b]).

Dans le cas des BDD, j'ai étendu le "package" BDD distribué par Bryant et Brace (voir [BRB90]) en rajoutant la prise en compte des poids pour les variables hypothèses et en implémentant les algorithmes utilisant les arcs inversés donnés dans la section 6.4.3.2 page 205.

6.5.2 Les jeux de tests

J'ai exécuté ces programmes sur 4 jeux de test :

- L'exemple classique des pingouins représenté par une base de 6 clauses, 3 strates et avec les 4 variables suivantes :
  - p = être un pingouin,
  - o = être un oiseau,
  - v = voler,
  - a = avoir des ailes.

  La base utilisée dans cet exemple est présentée dans le tableau 6.1.

E_1	$\rightarrow p$ $p \rightarrow o$
E_2	$o \rightarrow v$ $p \rightarrow \neg v$
E_3	$o \rightarrow a$ $a \rightarrow v$

Tableau 6.1 : Exemple des pingouins : la base de croyances

- L'exemple suivant porte sur l'attribution d'une augmentation de salaire, et est représenté par une base de 31 clauses, 7 strates avec les 17 variables suivantes :
  - AA = augmentation attribuée,
  - AD = augmentation disponible,
  - ADT = employé est en avance dans son travail,
  - AP = augmentation possible,
  - BN = employé est bien noté,
  - BV = employé est bien vu de son chef,
  - CMV = chef est mal vu de la hiérarchie,
  - D = employé est démotivé,
  - HFA = employé fait des horaires insuffisants,
  - HFO = employé fait des horaires importants,
  - HMO = employé fait des horaires moyens,
  - JDT = employé est à jour dans son travail,
  - RDT = employé est en retard dans son travail,
  - S = employé est en surcharge,
  - SFA = employé a un salaire faible,

- SFO = employé a un salaire fort,
- SMO = employé a un salaire moyen.

La base utilisée dans cet exemple est présentée dans le tableau 6.2.

E_1	$\rightarrow$ SFA $\rightarrow$ HFA $\rightarrow$ RDT $\rightarrow$ S $\rightarrow$ D $\rightarrow$ AD SMO $\rightarrow$ SFO $\rightarrow$ HMO $\rightarrow$ HFO $\rightarrow$ JDT $\rightarrow$ ADT $\rightarrow$ BV $\rightarrow$ CMV $\rightarrow$
E_2	AP, AD $\rightarrow$ AA HFA, BN $\rightarrow$ HMO $\rightarrow$ BN HFO $\rightarrow$ BN
E_3	SFA $\rightarrow$ AP SMO $\rightarrow$ AP SFO, AP $\rightarrow$ BN, RDT $\rightarrow$ JDT $\rightarrow$ BN ADT $\rightarrow$ BN
E_4	BN $\rightarrow$ AP AP $\rightarrow$ BN BV $\rightarrow$ BN BN $\rightarrow$ BV
E_5	S $\rightarrow$ AP
E_6	D $\rightarrow$ AP
E_7	CMV $\rightarrow$ AP

Tableau 6.2 : Exemple du salaire : la base de croyances (31 clauses)

- L'exemple sur le salaire augmenté de quelques clauses ; celui-ci sera représenté par une base de 77 clauses, 9 strates et toujours les mêmes 17 variables. La base utilisée dans cet exemple est présentée dans le tableau 6.3 page suivante.
- Des exemples de bases de 3-clauses tirées aléatoirement avec la procédure suivante (voir l'implémentation en langage C [LS95b]) : on fixe 2 paramètres, le nombre de variables n, le nombre de clauses l, et pour un n et un l donnés, on génère aléatoirement l clauses de longueur 3. Chaque clause est produite en choisissant aléatoirement un ensemble de 3 variables parmi les n disponibles et en changeant le signe de chacune d'entre elles avec une probabilité de 0.5. Ce modèle de génération est bien connu dans la communauté étudiant la résolution du problème Sat et a été introduit par [LMS92a]. Depuis, de nombreux travaux ont mis en évidence un seuil de 4.25 pour le ratio l/n autour duquel la probabilité de satisfiabilité des instances de bases 3-clauses générées aléatoirement s'inverse brusquement (avant ce seuil, les instances générées sont presque toutes consistantes et après elles sont presque toutes inconsistantes). On sait aussi qu'autour de ce seuil se situent des instances pour lesquelles

E_1	→ SFA
	→ HFA
	→ RDT
	→ S
	→ D
	→ AD
	SMO →
	SFO →
	HMO →
	HFO →
	JDT →
	ADT →
	BV →
	CMV →
E_2	HFO, HFA →
	HFO, HMO →
	HFA, HMO →
	SFA, SFO →
	SFO, SMO →
	SMO, SFA →
	JDT, ADT →
	JDT, RDT →
	RDT, ADT →

E_3	SFO → HFO, HMO
	SMO → HFO, HMO, HFA
	SFA → HFO, HMO, HFA
	SFO → JDT, RDT, ADT
	SMO → JDT, RDT, ADT
	SFA → JDT, RDT, ADT
	HFA → SFA, SMO, SFO
	HFO → SFA, SMO, SFO
	HMO → SFA, SMO, SFO
	HFA → RDT, JDT, ADT
	HFO → RDT, JDT, ADT
	HMO → RDT, JDT, ADT
	ADT → SFA, SFO, SMO
	JDT → SFA, SFO, SMO
	RDT → SFA, SFO, SMO
	ADT → HFA, HFO, HMO
	JDT → HFA, HFO, HMO
	RDT → HFA, HFO, HMO
	D, SFO →
	D, SMO →
	SFA → D
	HFO → D
	HMO, D →
	HFA → D
	RDT → D
	JDT, D →
	ADT, D →
	D → RDT
	D → HFA
	HFO → S
	S, HMO →
	S, HFA →
	RDT → S
	S, JDT →
	S, ADT →
	S → RDT
	S → HFO

E_4	AP, AD → AA
	HFA, BN →
	HMO → BN
	HFO → BN
E_5	SFA → AP
	SMO → AP
	SFO, AP →
	BN, RDT →
	JDT → BN
	ADT → BN
E_6	BN → AP
	AP → BN
	BV → BN
	BN → BV
E_7	S → AP
E_8	D → AP
E_9	CMV → AP

Tableau 6.3 : Exemple du salaire : la base de croyances (77 clauses)

la résolution du problème SAT est particulièrement difficile.

Tous les tests ont été effectués sur une Sparc Server 1000 (qui, malgré son nom, est moins rapide qu'une Sparc 10), et les résultats donnés sont de 3 types :

- Les résultats récapitulés dans le tableau 6.4 concernent uniquement l'approche BDD et portent à la fois sur le temps de réponse (en minute et seconde) et sur l'occupation mémoire (en nombre de sommets pour la taille du BDD). Ils sont issus de [CLSS96]. Ainsi, on compare l'ordre pour les variables proposé dans [JB95] avec l'ordre présenté ici en section 6.4.3.2 page 205. Les temps d'exécution mesurés correspondent à la définition du BDD, puis à son dégraissage. On constate que l'ordre proposé ici est bien meilleur que l'ordre de [JB95], puisque le gain en taille varie avec un rapport de 4 à 9 suivant les exemples. Il semble que ce rapport soit croissant avec la taille de la base initiale.

Test	Taille de R_E		Tps réponse avec R_E		Taille R''_E	Tps réponse avec R''_E	
	ordre [JB95]	notre ordre	ordre [JB95]	notre ordre		ordre [JB95]	notre ordre
Pingouins	12	10	≈ 0	≈ 0	4	≈ 0	≈ 0
Salaire 31 f.	252 932	62 028	1'07"	4.7"	17	12.8"	3.1"
Salaire 77 f.	748 461	83 040	4'25"	14.3"	17	38"	4.2"

Tableau 6.4 : Résultats sur les BDD

- Les résultats concernant les temps de réponse des 3 approches sur les trois premiers exemples, sont présentés dans le tableau 6.5. Les temps donnés ici correspondent au processus complet de la relation UNI-LEX : à partir d'une base E et d'une clause Φ, on mesure le temps nécessaire pour vérifier si $E \mid\!\sim^{\forall,lex} \Phi$. Dans le cas de l'exemple des pingouins, la clause Φ est $\rightarrow a$. Dans ce cas des exemples sur le salaire, la clause Φ est $\rightarrow$ AA[7]. Ces résultats confirment ce que la théorie nous a appris : les résultats obtenus par la méthode de résolution naïve sont très vite catastrophiques, alors que ceux obtenus en utilisant MAX-SAT sont de loin les meilleures. La méthode reposant sur les BDD est un moyen terme pouvant être intéressant dans une optique de précompilation de la base initiale.

Test	Rés. naïve	Rés. avec MAX-SAT	Résolution avec BDD	
			Tps. construction + dégraissage	Tps. vérification
Pingouins	≈ 0	≈ 0	≈ 0	≈ 0
Salaire 31 f.	0.06"	≈ 0	7.8"	≈ 0
Salaire 77 f.	arrêt après 26'40"	0.02"	18.5"	≈ 0

Tableau 6.5 : Comparaison des trois méthodes sur quelques exemples choisis

- Quant aux tests aléatoires effectués avec chacune des trois méthodes, leurs résultats sont présentés sur les figures 6.4 page 219 à 6.7 page 222 en utilisant une échelle verticale logarithmique. Ces tests ont été effectués avec trois valeurs de nombre de variables (5, 10 et 20), pour des ratios allant de 2 à 8 par pas de 0.2, et avec 50 échantillons par point. Rappelons que les bases ainsi générées sont, avec une forte probabilité, consistantes pour un ratio inférieur à 4.25 et inconsistantes pour un ratio supérieur à 4.25. D'autre part, nous avons interrompu l'exécution du test courant dès que le temps de réponse dépassait quelques centaines de secondes. Dans ce cas, aucun résultat n'est donné et la courbe est alors interrompue.
  - La figure 6.4 page 219 porte sur des temps qui correspondent au temps nécessaire pour vérifier si $E \mid\!\sim^{\forall,lex} \Phi$, à partir d'une base E et d'une clause Φ données. Ici, la clause Φ est une clause de 3 littéraux tirée aléatoirement (en utilisant la même technique et les

[7] Nous informons nos lecteurs que l'employé a finalement obtenu son augmentation de salaire !

mêmes paramètres que pour le choix de la base). Ces résultats généralisent ceux obtenus sur les trois exemples précédents, c'est-à-dire les excellents résultats de la méthode utilisant MAX-SAT, suivie de près par la méthode reposant sur les BDD (l'écart entre les deux méthodes se creusant au fur et à mesure que le nombre de variables augmente). Quant à la méthode naïve, ses performances ne lui permettent pas de traiter plus d'une dizaine de variables dès que la base devient fortement inconsistante.

- La figure 6.5 page 220 est relative aux temps de création, de dégraissage et de vérification de l'implication avec la méthode des BDD. Ces résultats confirment l'intérêt de cette technique dans une optique de précompilation de la base, puisque le temps de vérification de l'implication, une fois obtenu le BDD dégraissé, est tout à fait négligeable. Par contre, le temps de création, et dans une moindre mesure celui de dégraissage, sont importants dès que l'on manipule des bases de croyances de grande taille.
- La figure 6.6 page 221 concerne la taille des BDD : celle après la création (donc, avant dégraissage) et celle après dégraissage. On y voit l'impact impressionnant du dégraissage sur le BDD initial. On constate aussi que, dès que la base devient fortement inconsistante, le BDD dégraissé correspond, dans la majorité des cas, à une seule sous-base lex-préférée ayant un seul modèle (le nombre de sommets tend vers le nombre de variables). Cela prouve à quel point le mécanisme de génération utilisant l'ordre lexicographique est sélectif.
- La figure 6.7 page 222 reprend les résultats de la figure 6.4 page suivante en présentant un autre point de vue : l'impact du nombre de variables sur chaque méthode. Il s'avère que la technique la plus robuste est encore celle utilisant MAX-SAT.

6.5.3 Conclusion sur l'étude algorithmique

Les résultats obtenus mettent en évidence quelques points importants :

- ils identifient clairement la méthode utilisant MAX-SAT comme étant la meilleure, celle reposant sur les BDD venant ensuite, et en dernière position la méthode naïve qui est catastrophique ; ce résultat montre bien l'intérêt pratique d'une étude en complexité, l'algorithme s'appuyant sur MAX-SAT découlant de cette étude ;
- ils mettent en évidence le fait que, dans la méthode utilisant les BDD, c'est le temps de création de la base qui est le plus coûteux ; le temps de dégraissage est beaucoup plus faible et celui de vérification est proche de 0 ; l'utilisation des BDD comme moyen de précompilation de la base est donc justifiée dans la pratique ;
- ils montrent aussi que, contrairement au problème SAT, aucune des méthodes utilisées ne fait apparaître le moindre seuil.

Les perspectives sont alors multiples. On peut envisager la transformation des algorithmes présentés ici afin de pouvoir représenter d'autres relations d'inférence non-monotone. Cela devrait être facile à faire pour les méthodes naïve et BDD, puisqu'elles résultent d'un couplage faible. Par contre, l'algorithme utilisant MAX-SAT va être beaucoup plus difficile à transformer.

Bien que les résultats sur les BDD ne soient pas exceptionnels, cette méthode mérite encore quelques études. En effet, elle présente un avantage certain par rapport aux autres méthodes : elle permet de traiter une base de formules propositionnelles quelconques, alors que les autres techniques se limitent à des bases de clauses. Les améliorations possibles de cette méthode peuvent porter sur les performances en temps de construction (par exemple, en utilisant MAX-SAT pour construire le BDD dégraissé directement), ou bien en taille du BDD produit (par exemple, en améliorant l'ordre utilisé sur les variables pour, entre autres, minimiser l'impact de l'insertion des variables hypothèses, ou en étudiant un ordre d'introduction des formules durant la construction du BDD).

D'autre part, l'utilisation des BDD montre par expérimentation que, dans le cas d'une base de croyances fortement inconsistante, le mécanisme de génération LEX ne sélectionne en général

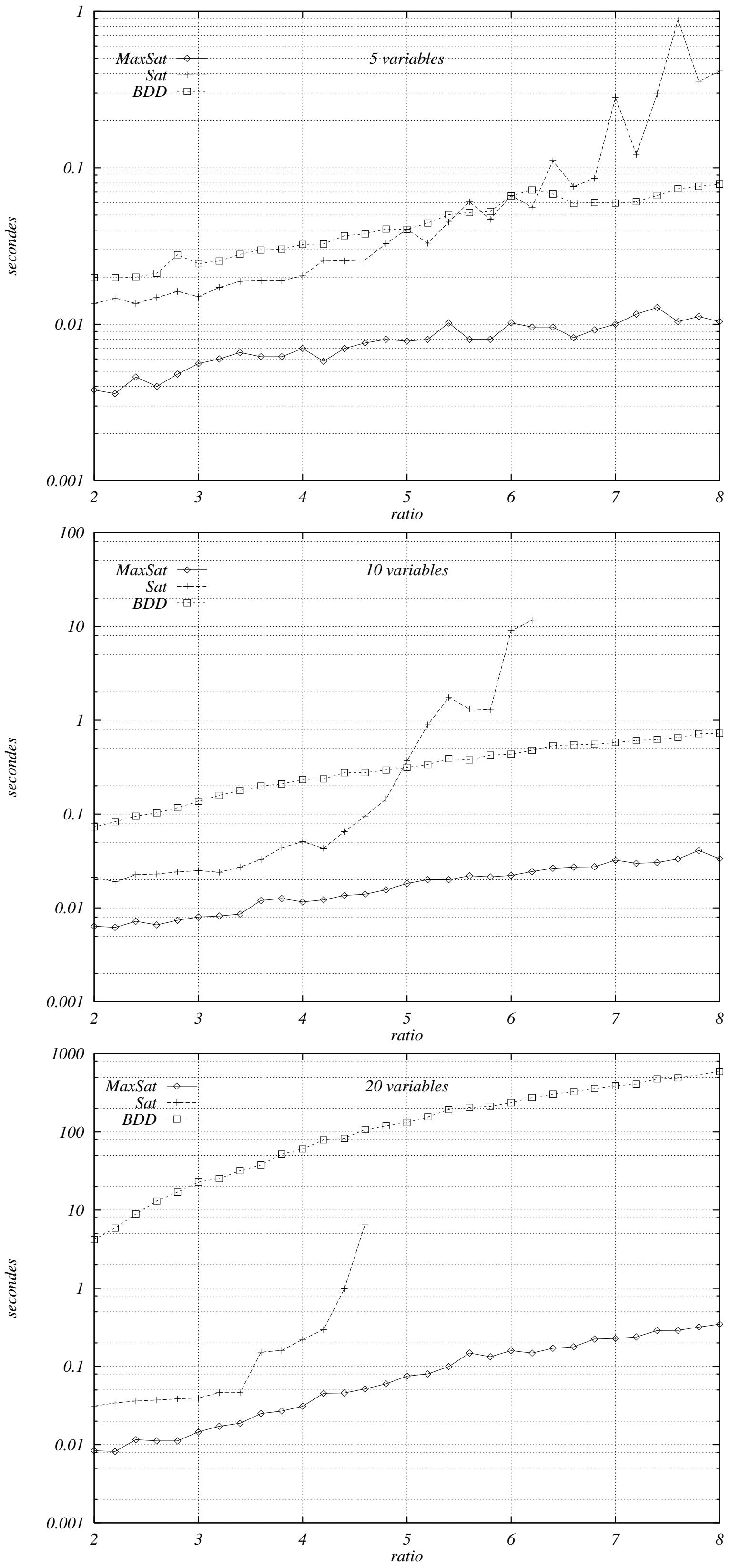


Figure 6.4 : Résultats des trois approches sur tests aléatoires – 5, 10 et 20 variables

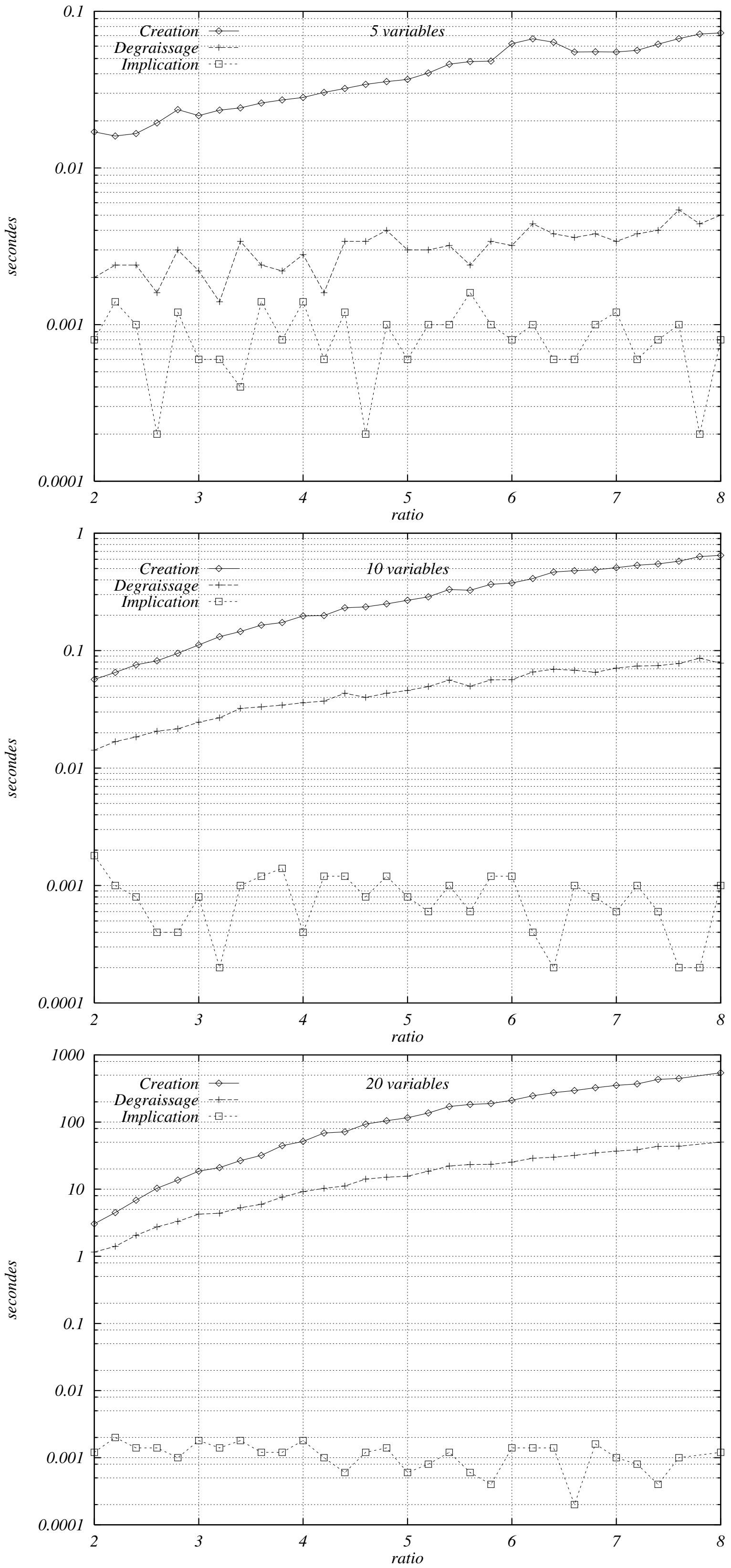


Figure 6.5 : Les différents temps obtenus avec les BDD sur tests aléatoires – 5, 10 et 20 variables

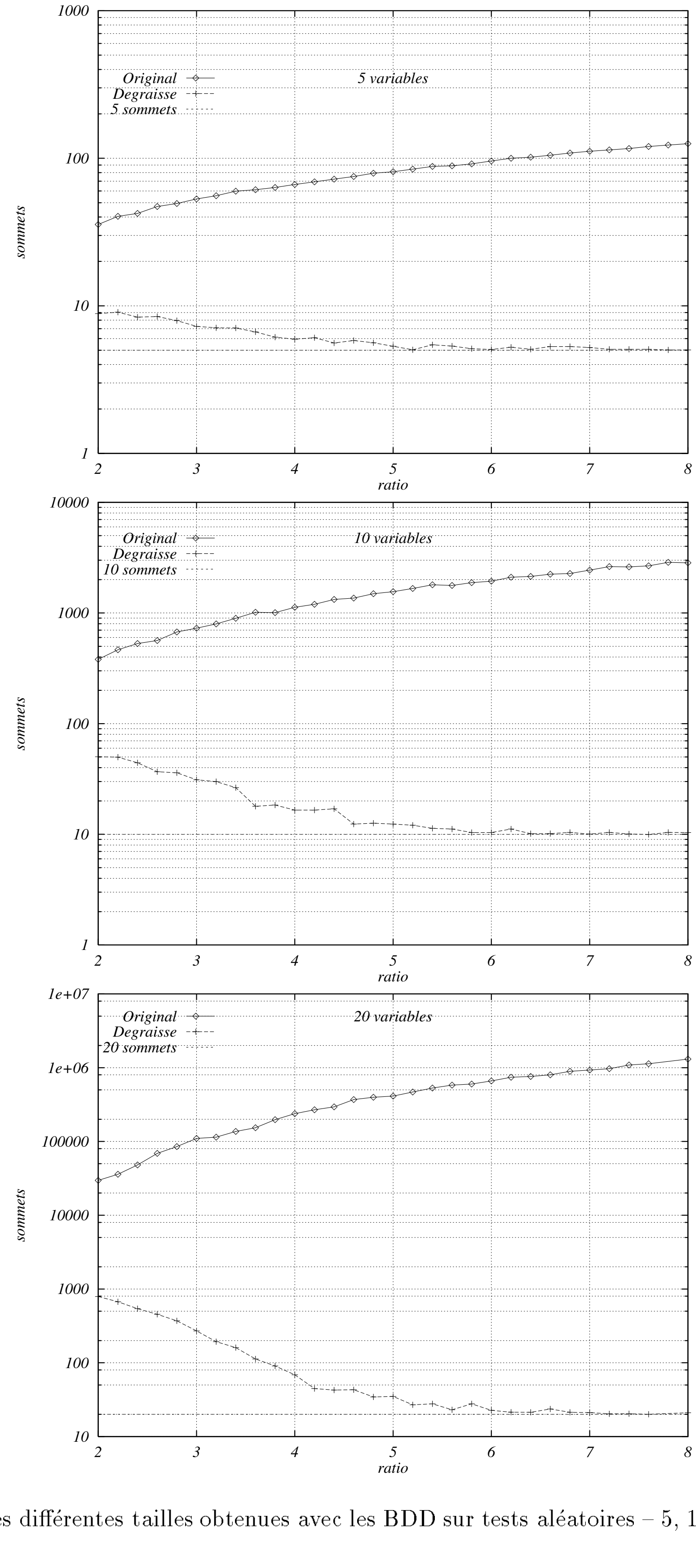


Figure 6.6 : Les différentes tailles obtenues avec les BDD sur tests aléatoires – 5, 10 et 20 variables

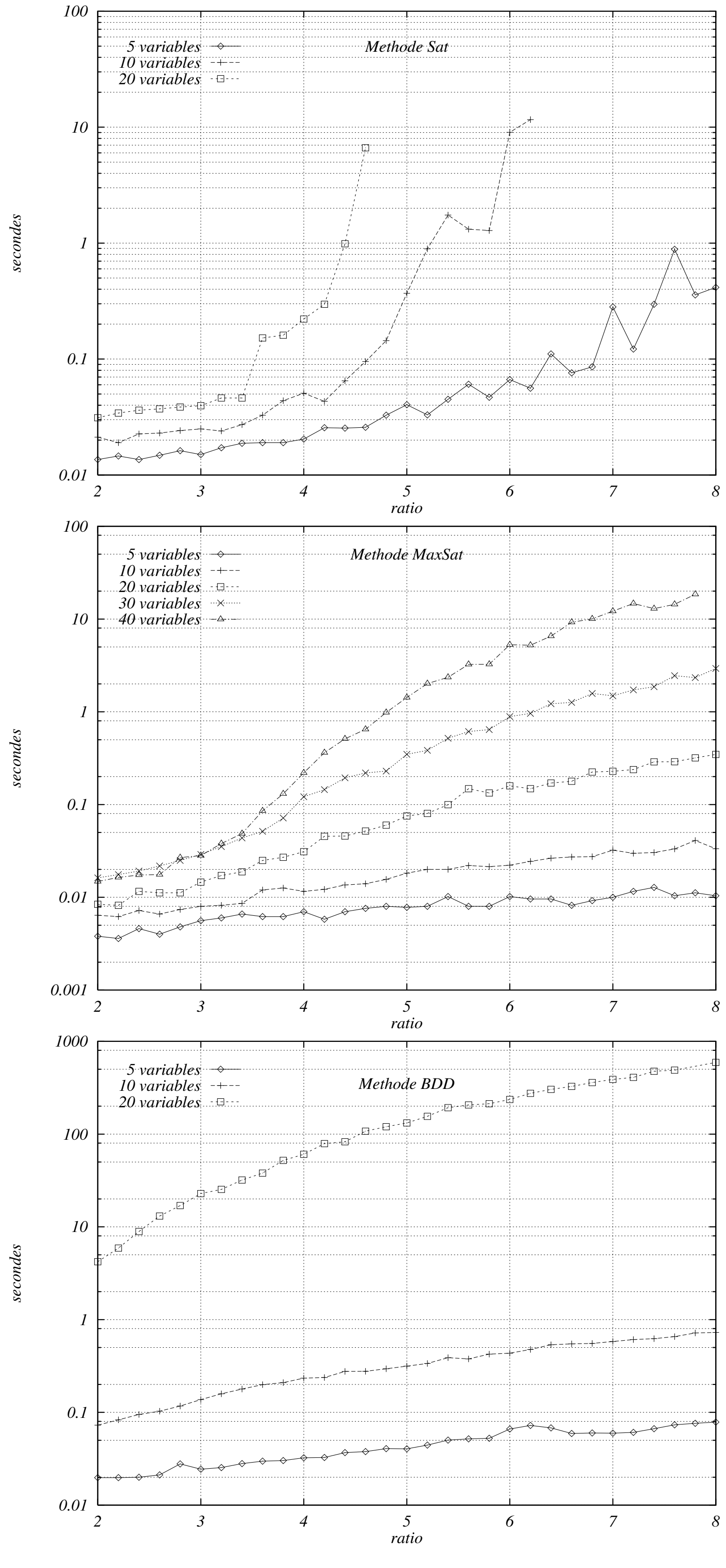


Figure 6.7 : Impact du nombre de variables sur le temps de réponse des trois méthodes

qu'une seule sous-base ayant un seul modèle. Cela paraît excessif, et incite à l'étude d'autres relations plus graduelles (par exemple, ne pas prendre en compte que les sous-bases lex-préférées, mais aussi celles qui leur sont le plus proches dans l'ordre lexicographique).

Dans un cadre plus général, tous les tests effectués ici seraient à reprendre pour analyser l'impact de nouveaux paramètres, par exemple le nombre de strates, le nombre de sous-bases préférées, le nombre de modèles préférés de ces sous-bases. Il faudrait aussi étudier l'évolution de l'écart entre la borne théorique et la taille réelle suivant l'évolution des paramètres (nombre de variables, ratios...). Et pour finir (momentanément, j'espère), il serait aussi intéressant d'approfondir l'étude des méthodes développées dans d'autres domaines du raisonnement non-monotone (par exemple, celles utilisées en logique des défauts). Peut-être, découvrirait-on des aspects et techniques utilisables pour coder les relations qui nous intéressent.

Chapitre 7

Conclusions et perspectives

Le cadre du raisonnement non-monotone en intelligence artificielle a vu se développer depuis quelques années de très nombreux formalismes ayant chacun leurs spécificités, tant théoriques, qu'applicatives.

Il m'a semblé alors, qu'à cette étape de la recherche sur un domaine aussi prolifique, il était important de "faire le point". Ce travail est donc une étude de synthèse de quelques formalismes déjà existants. Il ne s'agit pas de créer de nouveaux formalismes pour faire du raisonnement non-monotone, mais plutôt d'en identifier un ou plusieurs particulièrement performants (du point de vue pratique comme du point de vue théorique).

J'ai choisi de restreindre mon centre d'intérêt aux formalismes traitant de l'aspect déductif du raisonnement non-monotone et en particulier à ceux résultant de la combinaison de l'inférence classique et de préférences (voir [DLP91, CRS92, Leh92, BCD$^+$93, BDP93]). Ainsi, mon travail s'applique au traitement d'une base inconsistante, puisque les préférences peuvent servir entre autres à choisir en cas de conflits, et il concerne les méthodes dites de " restauration de la cohérence" puisque, face à une inconsistance, elles cherchent à restaurer la cohérence avec éventuellement le risque de rendre inaccessible certaines informations[1].

Dans ce contexte, on peut identifier différentes relations d'inférence pouvant toutes être vues à travers le même cadre unificateur : pour chacune d'entre elles, on identifie un ensemble de sous-bases consistantes préférées et on applique un principe d'inférence permettant de définir l'inférence non-monotone à partir de l'inférence classique sur les sous-bases préférées. Ainsi, je me suis trouvée face à une vingtaine de relations d'inférence non-monotone que j'ai étudiées suivant différents points de vue : la complexité du temps de calcul, la prudence et la validité des propriétés de déduction (voir [KLM90, GM94]). Ces critères ont été choisis en fonction de leur importance :

- La complexité en temps de calcul est essentielle dès qu'on aborde l'aspect d'une mise en pratique de la théorie (voir [Neb95]) ;
- la prudence, comme la validité des propriétés de déduction sont relatives à la capacité de la relation d'inférence à déduire le maximum de conclusions significatives suivant les données dont on dispose.

[1] De manière générale, on constate qu'il existe diverses attitudes possibles face à l'inconsistance (voir [BDP95] pour une présentation de ce sujet) :

- soit chercher à restaurer la consistance, avec éventuellement le risque de rendre inaccessible certaines informations ; dans [BDP95], ces techniques sont dites "basées sur la cohérence" ;
- soit chercher à gérer l'inconsistance sans rien jeter, c'est-à-dire inférer de nouvelles conclusions à partir de la base inconsistante en les justifiant par de "bonnes raisons", de "bons arguments" ; dans [BDP95], ces techniques sont dites "fondatrices".

.

Quelques travaux existaient déjà sur ces sujets (voir [Neb91, Got92, EG92, EG93] sur le chapitre de la complexité, voir [PL92, Bra93] pour la prudence, voir [KLM90, Leh92, BCD+93, GM94] pour les propriétés de déduction) mais le plus souvent ces études ne concernaient que quelques relations et rarement sur tous ces critères à la fois. Mon travail a donc consisté à compléter le plus possible ces différents travaux, en particulier sur les relations d'inférence non-monotone issues de la cardinalité et sur celles issues de la conséquence argumentative moins étudiées du fait de leur apparition récente. Dans le même ordre d'idée, j'ai cherché à approfondir l'aspect algorithmique de ces relations en réalisant quelques implémentations pratiques.

La première étape de l'étude est purement théorique et a permis d'obtenir de nombreux résultats :

- Premièrement, dans le cadre de la complexité, j'apporte de nombreux résultats jusqu'alors inconnus puisque sur les 81 résultats présentés en section 4.1.4 page 144, 74 sont nouveaux. J'ai aussi montré que certaines restrictions pouvaient améliorer quelque peu cet aspect mais ces gains restent limités puisqu'ils réduisent d'autant l'expressivité de notre langage (seule l'utilisation de conjonctions de clauses de Horn et d'un ordre strict et total comme relation de préférence permet de se ramener à des problèmes de complexité P).
- Deuxièmement, je présente un ensemble complet de résultats concernant la prudence et les propriétés de déduction des relations d'inférence non-monotone permettant ainsi de situer les différentes relations étudiées par rapport aux types de relations considérées comme intéressantes dans ce domaine (relations préférentielles ou relations rationnelles).

Cette étude a permis de distinguer deux relations (Uni-Bo et Uni-Lex) particulièrement performantes pour tous les critères étudiés, mais elle est surtout intéressante par le fait qu'elle m'amène à remettre en question certains des critères de comparaison utilisés. Par exemple, la prédominance de l'ensemble de propriétés de déduction, considéré à ce jour comme primordial pour avoir une "bonne" relation, me paraît excessive et ceci pour deux raisons :

- cet ensemble de propriétés est insuffisant pour détecter certains problèmes importants (voir par exemple le problème de non-pertinence pour Uni-Bo) ;
- les propriétés de cet ensemble peuvent ne pas être vérifiées par une relation sans que pour autant cette relation soit sans intérêt (voir le cas des relations issues de la conséquence argumentative).

Mis à part le fait que cet ensemble devrait être complété par de nouvelles propriétés (certaines existent déjà et d'autres sont à découvrir), je pense, et ce sera là une des perspectives de ces travaux, qu'il serait souhaitable de trouver des critères de comparaison beaucoup plus complets et objectifs. Ce n'est pas le cas actuellement avec le critère de prudence extrêmement difficile à exploiter et avec les propriétés de déduction qui favorisent certains types de relations d'inférence, puisqu'ayant été définies à l'origine pour ces relations là.

Toutefois, je ne suis pas certaine que cela soit la seule solution. En effet, une autre perspective serait probablement de "prendre le problème dans l'autre sens", c'est-à-dire de chercher par rapport à une application donnée quelles sont les propriétés souhaitables pour une relation traitant cette application. J'ai choisi ici la première méthode car mon étude n'est pas centrée sur un problème donné mais s'inscrit plutôt dans un cadre général.

Parmi les prolongements possibles, je vois aussi le même type d'étude dans le cas de relations d'inférence définies à l'aide d'un ordre partiel (ici, je me suis contentée d'explorer des relations d'inférence définies à l'aide d'un ordre total). Sur cet aspect, quelques travaux existent déjà (voir [Rya92, CRS92, Men94]), et il serait bon de voir si les résultats théoriques obtenus dans le cas d'un ordre total s'étendent à un ordre partiel.

Une autre possibilité à explorer serait d'étudier l'impact sur les résultats théoriques de la définition d'un autre opérateur $\oplus$ (l'opérateur qui permet de définir la relation $\Psi \mid\!\sim^{p,m}_{E,<} \Phi$ à partir de la relation $(E,<) \mid\!\sim^{p,m} \Phi$). En effet, la définition utilisée ici est celle se rapprochant le plus de l'aspect actuellement envisagé dans les problèmes de révision.

La seconde étape de cette étude concerne l'algorithmique, et les résultats obtenus portent sur les performances temporelles et spatiales des 3 algorithmes implémentés. Cette dernière étape rejoint l'étape de synthèse théorique en mettant en valeur, du point de vue pratique les différences de complexité théorique. On "sent" bien alors tout l'intérêt d'une étude de complexité théorique, préalable indispensable à une quelconque réalisation pratique. Ici, les perspectives sont évidemment de développer des algorithmes pour d'autres relations, puisque je me suis limitée à étudier ceux pour Uni-Lex. De plus, l'utilisation de ces différents algorithmes sur des exemples réels me paraît essentielle pour bien expliciter les avantages et les inconvénients de chaque relation.

Enfin, je pense qu'une telle étude, à la fois théorique et pratique, serait à mener sur d'autres types de relations, en particulier sur des relations "fondatrices" au sens de [BDP95], puisqu'on sait déjà qu'il existe quelques liens avec l'approche "basée sur la cohérence" qui a été au centre de mon travail (voir [Cay95a]).

Bibliographie

[AGM85] Alchourrón (C. E.), Gärdenfors (P.) et Makinson (D.). – On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic*, vol. 50, 1985, pp. 510–530.

[BCD+93] Benferhat (Salem), Cayrol (Claudette), Dubois (Didier), Lang (Jérôme) et Prade (Henri). – Inconsistency management and prioritized syntax-based entailment. *In : Proc. of the 13^{th} IJCAI*, éd. par Bajcsy (Ruzena). pp. 640–645. – Chambéry, France, 1993.

[BDLP92] Benferhat (Salem), Dubois (Didier), Lang (Jérôme) et Prade (Henri). – Hypothetical reasoning in possibilistic logic : basic notions and implementations issues. *In : Advances in Fuzzy Systems : Applications and Theory - vol 1*, éd. par Wang (P.Z.) et K.F.Loe. – World Scientific Publishing Comp., 1992.

[BDP92] Benferhat (Salem), Dubois (Didier) et Prade (Henri). – Representing default rules in possibilistic logic. *In : Proc. of the 3^{rd} KR*, éd. par Allen (J.), Fikes (R.) et Sandewall (E.). pp. 671–684. – Cambridge, MA, 1992.

[BDP93] Benferhat (Salem), Dubois (Didier) et Prade (Henri). – Argumentative inference in uncertain and inconsistent knowledge bases. *In : Proc. of the 9^{th} UAI*, éd. par Heckerman (David) et Mamdani (Abe). pp. 411–419. – Washington, DC, 1993.

[BDP94a] Benferhat (Salem), Dubois (Didier) et Prade (Henri). – Représentation des connaissances conditionnelles. *In : Proc. of the 3^{rd} Maghrebian Conference on Software Engineering and Artificial Intelligence MCSEAI*, éd. par IFIP et AFCET, pp. 79–88. – Rabat, Marocco, 1994.

[BDP94b] Benferhat (Salem), Dubois (Didier) et Prade (Henri). – *Some syntactic approaches to the handling of inconsistent knowledge bases: a comparative study.* – Rapport de recherche n° 94-55R, Institut de Recherche en Informatique de Toulouse (I.R.I.T.), décembre 1994.

[BDP95] Benferhat (Salem), Dubois (Didier) et Prade (Henri). – How to infer from inconsistent beliefs without revising? *In : Proc. of the 14^{th} IJCAI*, éd. par Mellish (Chris S.). pp. 1449–1455. – Montréal, Canada, 1995.

[Ben94] Benferhat (Salem). – *Raisonnement non-monotone et traitement de l'inconsistance en logique possibiliste.* – Institut de Recherche en Informatique de Toulouse (I.R.I.T.), 118, route de Narbonne 31062 Toulouse cedex. France, Thèse de PhD, Université Paul Sabatier, février 1994.

[Bra91] Brass (Stefan). – Deduction with supernormal defaults. *In : Proc. of the 2^{nd} Int. Workshop on Nonmonotonic and Inductive Logics (NIL'91), LNAI 659*, éd. par Brewka (G.), Jantke (K. P.) et Schmitt (P. H.). pp. 153–174. – Springer-Verlag.

[Bra93] Brass (Stefan). – On the semantics of supernormal defaults. *In : Proc. of the 13^{th} IJCAI*, éd. par Bajcsy (Ruzena). pp. 578–583. – Chambéry, France, 1993.

[BRB90] Brace (Karl S.), Rudell (Richard R.) et Bryant (Randal E.). – Efficient implementation of a BDD package. *In : 27^{th} ACM/IEEE Design Automation Conference*, pp. 40–45.

[Bre89a] Brewka (Gerhard). – Nonmonotonic logics – a brief overview. *In : AICOM, 2 (vol. 2)*, pp. 88–97.

[Bre89b] Brewka (Gerhard). – Preferred subtheories : An extended logical framework for default reasoning. *In : Proc. of the 11^{th} IJCAI*, éd. par Sridharan (N.S.). pp. 1043–1048. – Detroit, MI, 1989.

[Bry92] Bryant (R. E.). – Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, vol. 24, n° 3, septembre 1992, pp. 293–318.

[Cay90] Cayrol (Claudette). – *Assumption-Based Reasoning : A Uniform Framework for Non-Monotonic Reasoning.* – Rapport de recherche n° 90-29R, Institut de Recherche en Informatique de Toulouse (I.R.I.T.), novembre 1990.

[Cay92] Cayrol (Claudette). – Un modèle logique général pour le raisonnement révisable. *Revue d'Intelligence Artificielle*, vol. 6, n° 3, 1992, pp. 255–284.

[Cay95a] Cayrol (Claudette). – On the relation between argumentation and non-monotonic coherence-based entailment. *In : Proc. of the 14^{th} IJCAI*, éd. par Mellish (Chris S.). pp. 1443–1448. – Montréal, Canada, 1995.

[Cay95b] Cayrol (Claudette). – *Reasoning from inconsistent belief bases: On the relation between argumentation and coherence-based non-monotonic entailment.* – Rapport de recherche n° 95-09R, France, Institut de Recherche en Informatique de Toulouse (I.R.I.T.), avril 1995.

[CLR90] Cormen (Thomas H.), Leiserson (Charles E.) et Rivest (Ronald L.). – *Introduction to algorithms.* – MIT Press, 1990. ISBN : 0-262-03141-8.

[CLS93] Cayrol (Claudette) et Lagasquie-Schiex (Marie-Christine). – *Comparaison de relations d'inférence non-monotone : étude de complexité.* – Rapport de recherche n° 93-23R, France, Institut de Recherche en Informatique de Toulouse (I.R.I.T.), septembre 1993.

[CLS94a] Cayrol (Claudette) et Lagasquie-Schiex (Marie-Christine). – *Classification de relations d'inférence non-monotone : la prudence et les propriétés de déduction.* – Rapport de recherche n° 94-49R, France, Institut de Recherche en Informatique de Toulouse (I.R.I.T.), novembre 1994.

[CLS94b] Cayrol (Claudette) et Lagasquie-Schiex (Marie-Christine). – On the complexity of non-monotonic entailment in syntax-based approaches. *In : Proc. of the 11^{th} ECAI workshop on Algorithms, Complexity and Commonsense Reasoning.* – Amsterdam, Nederland, 1994.

[CLS95] Cayrol (Claudette) et Lagasquie-Schiex (Marie-Christine). – Non-monotonic syntax-based entailment: A classification of consequence relations. *In : Lecture Notes in Artificial Intelligence 946 (Proc. of ECSQARU-95)*, éd. par Froidevaux (C.) et Kohlas (J.). pp. 107–114. – Fribourg, Switzerland, 1995.

[CLSS96] Cayrol (Claudette), Lagasquie-Schiex (Marie-Christine) et Schiex (Thomas). – Non-monotonic reasoning: from complexity to algorithms. *In : Proc. of the 4^{th} International Symposium on Artificial Intelligence and Mathematics.* – Fort Lauderdale, Florida, 1996.

[CRS92] Cayrol (Claudette), Royer (Véronique) et Saurel (Claire). – Management of preferences in assumption-based reasoning. *In : Advanced methods in AI. Lecture notes in computer science 682*, éd. par Yager (R.) et Bouchon (B.), pp. 13–22. – Springer Verlag, 1992. Extended version in Technical Report IRIT-CERT, 92-13R (University Paul Sabatier Toulouse).

[DABC93] Dubois (O.), André (P.), Boufkhad (Y.) et Carlier (J.). – Sat versus unsat. *In : Proc. of the 2nd Challenge Workshop.* – position paper, 1993.

[DLP91] Dubois (Didier), Lang (Jérôme) et Prade (Henri). – Inconsistency in possibilistic knowledge bases - to live or not to live with it. *In : Fuzzy logic for the Management of Uncertainty*, éd. par Zadeh (L.A.) et Kacprzyk (J.), pp. 335–351. – Wiley and sons, 1991.

[DLS94] Dupin de Saint Cyr (Florence), Lang (Jérôme) et Schiex (Thomas). – Gestion de l'inconsistance dans les bases de connaissances : une approche syntaxique basée sur la logique des pénalités. *In : Actes de RFIA'94*, pp. 507–518. – France, 1994.

[DP60] Davis et Putnam. – A computing procedure for quantification theory. *JACM*, vol. 7, 1960, pp. 201–215.

[EG92] Eiter (Thomas) et Gottlob (Georg). – On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artificial Intelligence*, vol. 57, 1992, pp. 227–270.

[EG93] Eiter (Thomas) et Gottlob (Georg). – The complexity of logic-based abduction. *In : Proc. of the 10th Symposium on Theoretical Aspects of Computing STACS*, éd. par Enjalbert (P.), Finkel (A.) et Wagner (K. W.). pp. 70–79. – Würzburg, Germany, 1993. Long version to appear in *Journal of the ACM.*

[EGFK93] Elvang-Goransson (M.), Fox (J.) et Krause (P.). – Dialectic reasoning with inconsistent information. *In : Proc. of the 9th UAI*, éd. par Heckerman (David) et Mamdani (Abe). pp. 114–121. – Washington, DC, 1993.

[FL94] Freund (Michael) et Lehmann (Daniel). – *Belief revision and rational inference.* – Rapport de recherche n° 94-16, Israel, Leibniz Center for Research in Computer Science. Hebrew University of Jerusalem, juillet 1994.

[Gär90] Gärdenfors (Peter). – Belief revision and nonmonotonic logic : Two sides of the same coin. *In : Proc. of the 9th ECAI*, éd. par Aiello (Luigia Carlucci). pp. 768–773. – Stockhlom, Sweden, 1990.

[Gär91] Gärdenfors (Peter). – Nonmonotonic inferences based on expectations: A preliminary report. *In : Proc. of the 2nd KR*, éd. par Allen (J.), Fikes (R.) et Sandewall (E.). pp. 585–590. – Cambridge, MA, 1991.

[Gef92] Geffner (H.). – *Default reasoning: Causal and Conditional Theories.* – MIT Press, 1992.

[GJ79] Garey (Michael R.) et Johnson (David S.). – *Computers and Intractability : A Guide to the Theory of NP-completeness.* – New York, W.H. Freeman and Company, 1979.

[GM89] Gärdenfors (Peter) et Makinson (David). – Relations between the logic of theory change and nonmonotonic logic. *Lecture Notes in Compuer Science*, vol. 465, 1989, pp. 185–205.

[GM94] Gärdenfors (Peter) et Makinson (David). – Nonmonotonic inference based on expectations. *Artificial Intelligence*, vol. 65, 1994, pp. 197–245.

[Got92] Gottlob (Georg). – Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, vol. 2, n° 3, 1992, pp. 397–425.

[HC68] Hughes (G. E.) et Cresswell (M. J.). – *An introduction to modal logic.* – London and New-York, Methuen, 1968.

[HJ90] Hansen (Pierre) et Jaumard (Brigitte). – Algorithms for the maximum satisfiability problem. *Computing*, vol. 44, 1990, pp. 279–303.

[JB95] Jégou (Philippe) et Bouquet (Fabrice). – Solving over-constrained CSP using weighted OBDD. *In : Proceedings of the CP'95 Workshop on Over-Constrained Systems.* – Cassis, France, septembre 1995.

[Jég94] Jégou (Philippe). – A logical approach to solve dynamic CSPs: Preliminary report. *In : Proceedings ECAI'94 Workshop on Constraint Satisfaction Issues raised by Practical Applications*, pp. 87–94. – Amsterdam, The Netherlands, août 1994.

[Joh90] Johnson (David S.). – A catalog of complexity classes. *In : Handbook of Theoretical Computer Science*, éd. par van Leeuwen (Jan), chap. 2, pp. 67–161. – Elsevier, 1990.

[KLM90] Kraus (Sarit), Lehmann (Daniel) et Magidor (Menachem). – Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, vol. 44, 1990, pp. 167–207.

[KS91] Kautz (Henry A.) et Selman (Bart). – Hard problems for simple default logics. *Artificial Intelligence*, vol. 49, 1991, pp. 243–279.

[Lan91] Lang (Jérôme). – *Logique possibiliste : aspects formels, déduction automatique et applications.* – Thèse de PhD, Université Paul Sabatier Toulouse, 1991.

[Lee59] Lee (C. Y.). – Representation of switching circuits by binary-decision programs. *Bell System Tech. J.*, vol. 38, n° 4, juillet 1959, pp. 985–999.

[Leh92] Lehmann (Daniel). – *Another Perspective on Default Reasoning.* – Rapport de recherche n° 92-12, Israel, Leibniz Center for Research in Computer Science. Hebrew University of Jerusalem, juillet 1992.

[Lév91] Lévy (François). – Computing extensions of default theories. *In : Lecture Notes in Artificial Intelligence 548 (Proc. of ECSQAU-91)*, éd. par Kruse (R.) et Siegel (P.). pp. 219–226. – Marseille France, 1991.

[LM92] Lehmann (Daniel) et Magidor (Menachem). – What does a conditional knowledge base entail ? *Artificial Intelligence*, vol. 55, 1992, pp. 1–60.

[LMS92a] Levesque (Hector), Mitchell (David) et Selman (Bart). – Hard and easy distributions of SAT problems. *In : Proc. of AAAI-92*, pp. 459–465. – San Jose, CA, 1992.

[LMS92b] Levesque (Hector), Mitchell (David) et Selman (Bart). – A new method for solving hard satisfiability problems. *In : Proc. of AAAI-92*, pp. 440–446. – San Jose, CA, 1992.

[LS94] Lagasquie-Schiex (Marie-Christine). – La complexité des relations d'inférence non-monotone dans les approches syntaxiques. *In : Proc. of the 2^{nd} RJCIA*, pp. 65–72. – Marseille, France, 1994.

[LS95a] Lagasquie-Schiex (Marie-Christine). – *Contribution à l'étude des relations d'inférence non-monotone combinant inférence classique et préférences.* – Thèse, Université Paul Sabatier, IRIT, 1995.

[LS95b] Lagasquie-Schiex (Marie-Christine). – Implémentation de quelques algorithmes en raisonnement non-monotone. – Document annexe à [LS95a], 1995.

[LS95c] Lagasquie-Schiex (Marie-Christine). – Une classification des relations d'inférence non-monotone dans les approches syntaxiques. *In : Proc. des 5*$^{\text{ème}}$ *journées nationales du PRC-GDR Intelligence Artificielle*, p. 373. – Nancy, France, 1995.

[Luk90] Lukaszewicz (W.). – *Non-Monotonic Reasoning: Formalization of Commonsense Reasoning.* – Ellis Horwood series in Artificial Intelligence, 1990.

[Men94] Mengin (Jérôme). – *Raisonnement par défaut : résolution de conflits et priorités.* – Thèse de PhD, Université Paris Sud, 1994.

[Neb91] Nebel (Bernhard). – Belief revision and default reasoning: Syntax-based approaches. *In : Proc. of the 2*nd *KR*, éd. par Allen (J.A.), Fikes (R.) et Sandewall (E.). pp. 417–428. – Cambridge, MA, 1991.

[Neb95] Nebel (Bernhard). – Artificial intelligence: A computational perspective. – 1995. Version préliminaire, à paraître dans “Essentials in Knowledge Representation”, édité par G. Brewka.

[Pea90] Pearl (Judea). – System Z : A natural ordering of defaults with tractable applications to default reasoning. *In : Proc. of the 3*rd *Conference of Theoretical Aspects of Reasoning about Knowledge*, éd. par Vardi (M.). pp. 121–135. – Morgan-Kaufmann.

[PL92] Pinkas (Gadi) et Loui (Ronald P.). – Reasoning from inconsistency : A taxonomy of principles for resolving conflict. *In : Proc. of the 3*rd *KR*, éd. par Allen (J.A.), Fikes (R.) et Sandewall (E.). pp. 709–719. – Cambridge, MA, 1992.

[Poo88] Poole (David). – A logical framework for default reasoning. *Artificial Intelligence*, vol. 36, 1988, pp. 27–47.

[Pro90] Provan (Gregory M.). – The computational complexity of multiple-context truth maintenance systems. *In : Proc. of the 8*th *ECAI*, pp. 522–527. – Stockholm, 1990.

[Rei80] Reiter (R.). – A logic for default reasoning. *Artificial Intelligence*, vol. 13, n° 1-2, 1980, pp. 81–132.

[RM70] Rescher (N.) et Manor (R.). – On inference from inconsistent premises. *Theory and Decision*, vol. 1, 1970, pp. 179–219.

[Rot92] Rott (Hans). – Preferential belief change using generalized epistemic entrenchment. *Journal of logic, Language and Information*, vol. 1, 1992, pp. 45–78.

[Rot93] Rott (Hans). – Belief contraction in the context of the general theory of rational choice. *Journal of Symbolic Logic*, vol. 58, 1993, pp. 1426–1450.

[Rya92] Ryan (Mark). – Representing defaults as sentences with reduced priority. *In : Proc. of the 3*rd *KR*, éd. par Allen (J.A.), Fikes (R.) et Sandewall (E.). pp. 649–660. – Cambridge, MA, 1992.

[Sat94] Satoh (Ken). – A top down proof procedure for default logic by using abduction. *In : Proc. of the 11*th *ECAI*, éd. par Cohn (A. G.). pp. 65–69. – Amsterdam, The Nederlands, 1994.

[Sho87] Shoham (Yoav). – Nonmonotonic logics: Meaning and utility. *In : Proc. of the 10*th *IJCAI*, éd. par Dermott (John Mac). pp. 388–393. – Milan, Italy, 1987.

[SL92] Simari (G.R.) et Loui (R.P.). – A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, vol. 53, 1992, pp. 125–157.

[SR91] Schwind (Camilla B.) et Risch (Vincent). – A tableau-based characterisation for default logic. *In : Lecture Notes in Artificial Intelligence 548 (Proc. of ECSQAU-91)*, éd. par Kruse (R.) et Siegel (P.). pp. 310–317. – Marseille France, 1991.

[Sti90] Stillman (Jonathan). – It's not my default : The complexity of membership problems in restricted propositional default logics. *In : Proc. of the 8^{th} AAAI*, éd. par Press (MIT). pp. 571–578. – Boston, MA, 1990.

[Sto77] Stockmeyer (Larry J.). – The polynomial-time hierarchy. *Theoretical Computer Science*, vol. 3, 1977, pp. 1–22.

Annexe A

Théorèmes et définitions utilisés pour les démonstrations de complexité

Dans cette annexe, je répertorie les principaux théorèmes et définitions utilisés lors des démonstrations de complexité. Pour chacun des théorèmes, je donne soit la référence où trouver la preuve, soit la preuve elle-même.

Remarque : dans le cas des théorèmes donnant la complexité des processus de révision FMR, SBR, PBR, UBR auxquels je me suis référée lors de mes propres démonstrations de complexité, j'explicite dans une première partie les démonstrations présentées initialement par Nebel dans [Neb91], car les mécanismes mis en œuvre m'ont beaucoup inspirée pour mes propres démonstrations. Puis je calcule la complexité de ces mêmes processus dans un cas particulier (le cas de Horn) en utilisant la technique donnée par Eiter et Gottlob dans [EG92].

A.1 Les théorèmes et définitions généraux

Définition A.1.1 *Une transformation polynomiale f est une fonction calculable en un temps polynomial et telle que $\forall x,\ x \in L_1 \Leftrightarrow f(x) \in L_2$ (ceci est noté $L_1 \propto L_2$).*

Corollaire A.1.1 *Dans le cadre de problèmes de décision, $L_1 \propto L_2 \Leftrightarrow$ co-$L_1 \propto$ co-L_2.*

Preuve :

$L_1 \propto L_2$

$\Leftrightarrow$

il existe une transformation polynomiale f telle que ($x \in L_1 \Leftrightarrow f(x) \in L_2$) pour tout x

$\Leftrightarrow$

il existe une transformation polynomiale f telle que ($x \notin L_1 \Leftrightarrow f(x) \notin L_2$) pour tout x

Or nous sommes dans le cadre de problèmes de décision. Ce qui signifie que les langages L_1 et L_2 représentent des instances positives de problèmes, c'est-à-dire des instances dont la solution est oui. Donc dire que $x \notin L_1$ est équivalent à dire que $x \in$ co-L_1. On a donc :

$\Leftrightarrow$

il existe une transformation polynomiale f telle que ($x \in$ co-$L_1 \Leftrightarrow f(x) \in$ co-L_2) pour tout x

$\Leftrightarrow$

co-$L_1 \propto$ co-L_2 □

Corollaire A.1.2 *Dans le cadre de problèmes de décision, pour toute classe de complexité X de la hiérarchie polynomiale, on a : L est X-complet $\Leftrightarrow$ co-L est co-X-complet.*

Preuve :

L est X-complet

$\Leftrightarrow$

$L \in$ X et $\forall L' \in$ X, $L' \propto L$. Or $L' \propto L \Leftrightarrow$ co-$L' \propto$ co-L. On a donc :

$L \in$ X et $\forall L' \in$ X, co-$L' \propto$ co-L

$\Leftrightarrow$

co-$L \in$ co-X et $\forall$ co-$L' \in$ co-X, co-$L' \propto$ co-L

$\Leftrightarrow$

co-L est co-X-complet. □

Théorème A.1.1 *Si L est* NP*-complet et co-L* $\in \mathsf{NP}$ *alors* $\mathsf{NP} =$ *co-*NP

Preuve : (voir [GJ79])

Théorème A.1.2 *Soit X une classe de complexité de la hiérarchie polynomiale,* $\forall L_1, L_2 \in X$*,* L_1 *étant X-complet, si* $L_1 \propto L_2$*, alors* L_2 *est X-complet.*

Preuve :

Pour toute classe X de la hiérarchie polynomiale, L_2 sera X-complet ssi $L_2 \in$ X et pour tout $L \in$ X on a $L \propto L_2$.

Or on sait déjà que $L_2 \in$ X puisque cela fait partie des hypothèses. Il suffit de montrer que pour tout $L \in$ X on a $L \propto L_2$.

On sait que L_1 est X-complet, donc que pour tout $L \in$ X on a $L \propto L_1$. Et on sait que $L_1 \propto L_2$.

D'autre part, on sait que la relation $\propto$ est transitive (voir [GJ79]).

On a donc pour tout $L \in$ X, $L \propto L_2$.

L_2 est donc X-complet. □

Théorème A.1.3 *Soit X une classe de complexité de la hiérarchie polynomiale, si L est X-complet et co-L* $\in X$ *alors* $X =$ *co-*X[1]*.*

Preuve :

1. $L \in$ X-complet $\Leftrightarrow$ co-$L \in$ co-X-complet $\Leftrightarrow$ pour tout $L' \in$ co-X on a $L' \propto$ co-L
   Or on sait par hypothèse que co-$L \in$ X donc comme $L \in$ X-complet, on a co-$L \propto L$.
   La transformation polynomiale étant transitive, on a donc pour tout $L' \in$ co-X, $L' \propto L \Rightarrow$
   pour tout $L' \in$ co-X, $L' \in$ X $\Rightarrow$ co-X inclus dans X

[1] Ce théorème est une généralisation du théorème A.1.1 cité précédemment.

2. $L \in$ X-complet $\Leftrightarrow$ co-$L \in$ co-X-complet.
   D'autre part co-$L \in$ X donc $L \in$ co-X et co-L étant co-X-complet, alors $L \propto$ co-L.
   Avec de plus le fait que $L \in$ X-complet $\Leftrightarrow$ pour tout $L' \in$ X on a $L' \propto L$.
   La transformation polynomiale étant transitive, on a donc pour tout $L' \in$ X, L' $\propto$ co-L $\Rightarrow$ pour tout $L' \in$ X, $L' \in$ co-X $\Rightarrow$ X inclus dans co-X
3. En conclusion : X = co-X.

□

Théorème A.1.4 *Soit k un entier représentant un niveau de la hiérarchie polynomiale ($k \geq 1$), les deux propriétés suivantes sont équivalentes :*
(1) si $\Sigma^p_{k-1} \neq \Pi^p_{k-1}$, $L \in \Delta^p_k$ - $(\Sigma^p_{k-1} \cup \Pi^p_{k-1})$,
(2) $L \in \Delta^p_k$ et est Σ^p_{k-1}-difficile et Π^p_{k-1}-difficile.

Preuve :

- $(1) \Rightarrow (2)$:
  $(1) \Leftrightarrow L \in \Delta^p_k$ - $(\Sigma^p_{k-1} \cup \Pi^p_{k-1})$, si $\Sigma^p_{k-1} \neq \Pi^p_{k-1}$. Rappelons que : si $\Sigma^p_{k-1} = \Pi^p_{k-1}$ alors la hiérarchie polynomiale s'effondre sur Σ^p_{k-1}. Tant que l'on suppose vraie la conjecture $\Sigma^p_{k-1} \neq \Pi^p_{k-1}$, on suppose par là-même que les classes Δ^p_k et $(\Sigma^p_{k-1} \cup \Pi^p_{k-1})$ sont distinctes. Ainsi, si L appartient à la différence Δ^p_k - $(\Sigma^p_{k-1} \cup \Pi^p_{k-1})$, alors, par définition de la hiérarchie polynomiale, L est $(\Sigma^p_{k-1} \cup \Pi^p_{k-1})$-difficile. On a ainsi : $L \in \Delta^p_k$ et est Σ^p_{k-1}-difficile et Π^p_{k-1}-difficile, ce qui est exactement la propriété (2).
- $(2) \Rightarrow (1)$:
  $(2) \Leftrightarrow L \in \Delta^p_k$ et est Σ^p_{k-1}-difficile et Π^p_{k-1}-difficile. Donc, soit L' problème Σ^p_{k-1}-complet, on a : $L' \propto L$ et co-$L' \propto L$. Prenons maintenant comme hypothèse que :
  - $L \in \Sigma^p_{k-1}$:
    Utilisons le théorème A.1.3 page ci-contre donné ci-dessus. On sait que co-L' $\propto L$, donc avec l'hypothèse $L \in \Sigma^p_{k-1}$, il est possible de résoudre co-L' avec un algorithme Σ^p_{k-1}, ce qui revient à dire que co-$L' \in \Sigma^p_{k-1}$.
    D'autre part, on sait que L' est Σ^p_{k-1}-complet. Nous obtenons alors : L' est Σ^p_{k-1}-complet, co-$L' \in \Sigma^p_{k-1}$, on a donc $\Sigma^p_{k-1} = \Pi^p_{k-1}$.
  - $L \in \Pi^p_{k-1}$:
    On sait que $L_1 \propto L_2 \Leftrightarrow$ co-$L_1 \propto$ co-L_2, $\forall$ L_1 et L_2. De plus on sait que $L' \propto L$ donc co-$L' \propto$ co-L.
    Ainsi, avec l'hypothèse $L \in \Pi^p_{k-1}$, on a co-$L \in \Sigma^p_{k-1}$, et il est donc possible de résoudre co-L' avec un algorithme Σ^p_{k-1}, ce qui revient à dire que co-$L' \in \Sigma^p_{k-1}$.
    Nous pouvons donc à nouveau appliquer le même théorème : L' est Σ^p_{k-1}-complet, co-$L' \in \Sigma^p_{k-1}$, on a donc $\Sigma^p_{k-1} = \Pi^p_{k-1}$.

  Nous avons ainsi, si $\Sigma^p_{k-1} \neq \Pi^p_{k-1}$, $L \in \Delta^p_k$ - $(\Sigma^p_{k-1} \cup \Pi^p_{k-1})$, ce qui est exactement la propriété (1).

□

A.2 Les théorèmes et définitions pour la révision

A.2.1 Complexité de FMR

Définition A.2.1 *Le problème* FMR *se définit de la manière suivante : soit Z un ensemble de formules, soient y et x deux formules, Z révisé par y infère-t-il x ? L'opérateur de révision est* *défini comme suit : l'ensemble "Z révisé par y" = $Z \cup \{y\}$ si Z est consistant avec y, sinon y. Voir [Neb91] pour plus de détail.*

Théorème A.2.1 *Dans le cas propositionnel général, le processus de révision appelé* FMR *appartient à la classe de complexité Δ_2^p - (NP $\cup$ co-NP), si NP $\neq$ co-NP.*

Preuve : (voir [Neb91])

Il faut calculer la complexité du problème $P : x \in C_n(Z)$ révisé par y en utilisant la méthode FMR ?

Étude 1ère partie : l'appartenance à une classe.

L'algorithme A.1 est un algorithme possible pour FMR.

Algorithme A.1 : FMR (Z, x, y)
début
  si Z n'infère pas $\neg y$ **alors**
    $Z \cup y$ infère-t-il x ?
  else
    y infère-t-il x ?
fin

On sait d'autre part que le problème $BC \vdash H$ est de complexité co-NP et donc que le co-problème associé $BC \nvdash H$ est de complexité NP. Cela nous donne donc pour FMR un algorithme polynomial faisant appel à un oracle NP et à un oracle co-NP. Ce qui ne correspond à rien de connu dans la hiérarchie polynomiale. Par contre, on peut sans problème modifier cet algorithme de manière à n'utiliser plus qu'un oracle NP[2] (voir algorithme A.2).

Algorithme A.2 : FMR (Z, x, y) seconde possibilité
début
  si Z n'infère pas $\neg y$ **alors**
    si $Z \cup y$ n'infère pas x **alors**
      $\textit{échec} \leftarrow \textit{vrai}$
    else
      $\textit{échec} \leftarrow \textit{faux}$
  else
    si y n'infère pas x **alors**
      $\textit{échec} \leftarrow \textit{vrai}$
    else
      $\textit{échec} \leftarrow \textit{faux}$
  vérifier que l'on n'a pas échec ($\textit{échec} = \textit{faux}$)
fin

Dans cet algorithme qui est polynomial, on utilise un oracle de complexité NP.

Le problème FMR est donc dans la classe de complexité $\mathsf{P}^{\mathrm{NP}} = \Delta_2^p$.

Étude 2ème partie : la complétude ?

On raffine l'appartenance à la classe Δ_2^p en vérifiant si FMR $\in$ NP ou co-NP.

[2] Cette technique est systématiquement utilisée y compris pour des oracles de complexité X quelconque et elle justifie le fait que la hiérarchie polynomiale ne soit définie que pour des oracles de classe NP, Σ_2^p, Σ_3^p, ..., Σ_k^p.

Pour cela, on va essayer de comparer FMR à des problèmes connus de NP et co-NP. Les problèmes les plus connus sont GSAT et UNGSAT (satisfaisabilité et insatisfaisabilité d'une formule en logique classique propositionnelle). On constate alors que :

- Le problème GSAT se ramène au problème FMR par une transformation polynomiale :
  Soit H une formule propositionnelle, posons : $Z = \{H\}$, $y = vrai$, et $x = H$[3].

  $$H \text{ est satisfiable} \Leftrightarrow H \in C_n(\{H\}) \text{ révisé par } vrai.$$

  En conclusion, GSAT se transforme polynomialement en FMR (noté GSAT $\propto$ FMR).
- Le problème UNGSAT (complémentaire de GSAT) se ramène au problème FMR de manière polynomiale (on applique le même raisonnement que précédemment). Soit H une formule propositionnelle, posons : $Z = \varnothing$, $y = H$ et $x = \bot$.

  $$H \text{ est insatisfiable}$$
  $$\Leftrightarrow$$
  $$\text{il n'existe pas d'ensemble de formules consistant } M \text{ tel que } M \vdash H$$
  $$\Leftrightarrow$$
  $$H \vdash \bot.$$
  $$\Leftrightarrow$$
  $$\bot \in C_n(\varnothing) \text{ révisé par } H.$$

  En conclusion, on a UNGSAT $\propto$ FMR.

En utilisant le théorème A.1.4 page 237 donné ci-dessus, on prouve que, si NP $\neq$ co-NP, FMR $\in \Delta_2^p$ - (NP $\cup$ co-NP). □

Définition A.2.2 *On dira qu'un processus de révision* REV(Z, y, x) *est appliqué au cas Horn ssi Z est un ensemble de conjonctions de clauses de Horn, y est une conjonction de clauses de Horn et x est une clause de Horn.*

Définition A.2.3 *On dira qu'un processus de révision* REV(Z, y, x) *est appliqué au cas Horn strict ssi Z est un ensemble de clauses de Horn, y et x sont des clauses de Horn.*

Théorème A.2.2 *Dans le cas Horn (voir définition A.2.2), le processus de révision appelé* FMR *appartient à la classe de complexité* P *et sera noté* FMR-HORN.

Preuve : L'algorithme pour FMR-HORN est le même que celui de FMR. C'est un algorithme polynomial, dans lequel on utilise un oracle qui est désormais de complexité P puisque le problème de satisfaction d'un ensemble de conjonctions de clauses de Horn est polynomial.

Le problème FMR-HORN est donc dans la classe de complexité P. □

---

[3] Cette construction est la conséquence du raisonnement suivant. Tout d'abord, on remarque que $BC \vdash H$ est de complexité co-NP (donc se ramène à un problème UNGSAT), tandis que $BC \nvdash H$ est de complexité NP (donc se ramène à un problème GSAT) ; ainsi l'algorithme de FMR donné ci-dessus peut s'exprimer de la façon suivante :

1. *Si* GSAT *alors*
2. UNGSAT *numéro 1*
3. *sinon* UNGSAT *numéro 2*

Pour que GSAT se ramène à cet algorithme, il suffit que :

- ce problème GSAT corresponde au problème GSAT du "Si",
- le problème UNGSAT numéro 1 corresponde à un problème donc la réponse est oui quand la réponse au problème GSAT est oui,
- le problème UNGSAT numéro 2 corresponde à un problème donc la réponse est non quand la réponse au problème GSAT est non.

Théorème A.2.3 *Dans le cas Horn strict (voir définition A.2.3 page précédente), le processus de révision appelé* FMR *appartient à la classe de complexité* P *et sera noté* FMR-HORNS.

Preuve : La preuve est identique à celle du cas Horn puisque les problèmes de satisfaction d'un ensemble de clauses de Horn et d'un ensemble de conjonctions de clauses de Horn sont identiques.

Le problème FMR-HORNS est donc dans la classe de complexité P. □

A.2.2 Complexité de SBR

Définition A.2.4 *Le problème* SBR *se définit de la manière suivante : soit Z un ensemble de formules, soient y et x deux formules, Z révisé par y infère-t-il x ? L'opérateur de révision est défini comme suit : l'ensemble "Z révisé par y" $= (\cap_{Y \in (Z \downarrow \neg y)} C_n(Y)) + y$, avec l'opérateur $\downarrow$ défini informellement par $Z \downarrow y = \{$ensembles de formules n'impliquant pas y et maximaux pour l'inclusion$\}$, la notation $C_n(A)$ signifiant l'ensemble des conséquences logiques de A.*

Voir [Neb91] pour plus de détail, en particulier pour la définition formelle de $Z \downarrow y$. Remarquons que les Y sont les thèses de Z consistantes avec y.

Théorème A.2.4 *Dans le cas propositionnel général, le processus de révision appelé* SBR *est Π_2^p-complet.*

Preuve : (voir [Neb91])

Il faut calculer la complexité du problème P : $x \in C_n(Z)$ révisé par y en utilisant la méthode SBR ?

Dans la méthode SBR, le problème "$x \in C_n(Z)$ révisé par y" consiste à prouver que : "quel que soit Y un sous-ensemble de Z tel que Y n'infère pas $\neg y$ et est maximal au sens de l'inclusion (Y est dit maximal-y-consistant), alors $Y \cup \{y\}$ infère x".

Il va être délicat de calculer la complexité de ce problème sachant qu'il faut passer en revue tous les sous-ensembles Y de Z vérifiant les propriétés énoncées ci-dessus. On va donc plutôt calculer la complexité du co-problème dans lequel il suffit de trouver un sous-ensemble Y vérifiant les propriétés mais n'inférant pas x.

Étude 1ère partie : l'appartenance à une classe.

L'algorithme non déterministe A.3 est une des solutions possibles pour implémenter co-SBR.

Algorithme A.3 : SBR (Z, x, y)
début
  deviner un sous-ensemble Y de Z
  vérifier qu'il est maximal pour l'inclusion et n'infère pas $\neg y$
    (* c'est-à-dire que quel que soit z appartenant à $Z - Y$,
    on a $Y \cup \{z\}$ qui infère $\neg y$ *)
  vérifier que $Y \cup \{y\}$ n'infère pas x
fin

Dans cet algorithme qui est non déterministe (à cause du "deviner") polynomial, on utilise un oracle de complexité NP (même raisonnement que pour FMR).

Le problème co-SBR appartient donc à la classe de complexité $\mathsf{NP}^{\mathrm{NP}} = \Sigma_2^p$, donc le problème SBR appartient à co-$\Sigma_2^p = \Pi_2^p$.

Étude 2ème partie : la complétude ?

On connaît des problèmes Σ_2^p-complets. L'un des plus connus est le problème 2-QBF : "vérifier la satisfaisabilité de la formule $\exists a \forall b H(a,b)$ avec $a = (a_1, \ldots, a_n)$ et $b = (b_1, \ldots, b_m)$".

Pour montrer que co-SBR est Σ_2^p-complet, il suffit de montrer que le problème 2-QBF se ramène polynomialement à co-SBR. Et on pourra alors déduire que co-SBR est Σ_2^p-complet en utilisant le théorème A.1.2 page 236.

Comment faire le lien entre co-SBR et 2-QBF ? On cherche à démontrer que le problème 2-QBF se ramène polynomialement à co-SBR. Le principe est le même que celui utilisé pour FMR. Il faut montrer qu'il existe une fonction f telle que l'image par f d'une instance positive du problème numéro 1 "$\exists a \forall b H(a,b)$ est satisfiable" est une instance positive du problème numéro 2 "Z révisé par y avec la méthode SBR n'infère pas x". Posons donc :

$$x = \neg H(a,b),\ y = vrai,\ Z = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, \neg H(a,b)\}.$$

- on définit ainsi une fonction f calculable en un temps polynomial,
- on a dans Z toutes les "valeurs" possibles pour les variables propositionnelles $a_1, \ldots, a_n$ dont on cherche s'il existe une assignation,
- on n'impose aucune contrainte sur les variables propositionnelles $b_1, \ldots, b_m$, dont toutes les assignations doivent être prises en compte,
- on a dans Z la négation de la formule dont on cherche à prouver la satisfaisabilité.

Montrons alors que : "$\{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, \neg H(a,b)\}$ révisé par *vrai* avec la méthode SBR n'infère pas $\neg H(a,b)$" équivaut à "$\exists a \forall b H(a,b)$ est satisfiable" :

$\{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, \neg H(a,b)\}$ révisé par *vrai* avec la méthode SBR
n'infère pas $\neg H(a,b)$

$\Leftrightarrow$

il existe un sous-ensemble Y de Z tel que Y soit maximal pour l'inclusion et n'infère pas $\neg\ (vrai)$ = contradiction = $\perp$ (donc maximal consistant) et
tel que Y n'infère pas $\neg H(a,b)$

$\Leftrightarrow$

vu la forme de Z choisie et pour que Y soit maximal consistant et n'infère pas $\neg H(a,b)$, alors $Y = \{l_1, \ldots, l_n\}$ avec $l_i = a_i$ ou $\neg a_i$, $\forall i = 1 \ldots n$.

$\Leftrightarrow$

$\exists \{l_1, \ldots, l_n\}$ qui est inconsistant avec $\neg H(a,b)$

$\Leftrightarrow$

$\exists \{l_1, \ldots, l_n\}$ qui infère $H(a,b)$

$\Leftrightarrow$

il existe un modèle des variables propositionnelles $a_1, \ldots, a_n$ tel que sans la moindre contrainte sur les variables propositionnelles $b_1, \ldots, b_m$ (donc pour tout modèle de $b_1, \ldots, b_m$) on a $H(a,b)$ vraie

$\Leftrightarrow$

$\exists a \forall b H(a,b)$ est satisfiable

Le problème 2-QBF se ramène donc polynomialement à co-SBR. Et on en déduit en utilisant le théorème A.1.2 page 236 que le problème co-SBR est Σ_2^p-complet et SBR est Π_2^p-complet. □

Théorème A.2.5 *Dans le cas Horn (voir définition A.2.2 page 239), le processus de révision appelé* SBR *est co-*NP*-complet (il sera noté* SBR-HORN*).*

Preuve :

Étude 1ère partie : l'appartenance à une classe.

L'algorithme pour co-SBR-HORN est le même que celui de co-SBR. Il s'agit d'un algorithme non déterministe polynomial (à cause du "deviner"), dans lequel on utilise un oracle de complexité P.

Le problème co-SBR-HORN appartient donc à la classe de complexité NP, donc le problème SBR-HORN appartient à co-NP.

Étude 2ème partie : la complétude ?

On connaît des problèmes NP-complets. L'un des plus connus est le problème SAT.

Pour montrer que co-SBR-HORN est NP-complet, il suffit de montrer que le problème SAT se ramène polynomialement à co-SBR-HORN.

Comment faire le lien entre co-SBR-HORN et SAT ? Nous allons nous inspirer de la démonstration utilisée par Eiter et Gottlob dans [EG92]. Soit $C = \{C_j\}$ pour $j = 1 \dots m$ ensemble de clauses quelconques à satisfaire, soit $V(C) = \{x_1, \dots, x_n\}$ l'ensemble des variables propositionnelles apparaissant dans C, posons :

> $Z = \{x_1, \dots, x_n, y_1, \dots, y_n, \neg z_1, \dots, \neg z_n, \neg s\}$ avec $y_1, \dots, y_n$, $z_1, \dots, z_n$, s nouveaux symboles de variables propositionnelles,
>
> $x = \neg s$,
>
> $y = (z_1 \dots z_n \rightarrow s) \wedge \bigwedge_{i=1 \dots n}((\neg x_i \vee \neg y_i) \wedge (y_i \rightarrow z_i) \wedge (x_i \rightarrow z_i)) \wedge \bigwedge_{j=1 \dots m} C_j[y]$, avec la notation suivante : $C_j[y]$ = la clause C_j dans laquelle on remplace tous les x_i positifs par des $\neg y_i$.

Quelques explications :

- Le but est de transformer un problème portant sur des clauses quelconques en un problème portant sur des conjonctions de clauses de Horn. Il faut donc transformer les clauses initiales en conjonctions de clauses de Horn. Pour cela, on introduit une nouvelle variable propositionnelle y_i pour chaque x_i de C, cette y_i jouera le rôle de $\neg x_i$. Cela pose alors un problème de cohérence : x_i et y_i ne doivent pas avoir la même valeur de vérité. Nous introduisons alors une nouvelle variable propositionnelle z_i assurant le fait que x_i et y_i ne soient pas vraies en même temps ; la maximalité pour l'inclusion assure pour sa part qu'au moins une des deux variables, x_i ou y_i, est vraie, quel que soit i. La dernière variable rajoutée s sera vraie quand tous les z_i seront vraies, donc quand pour tous les x_i et y_i, on aura valeur-vérité $(x_i) \neq$ valeur-vérité (y_i).
- En transformant C_j en $C_j[y]$, on obtient une clause de Horn, quel que soit j.
- On a alors la propriété suivante :

  Propriété A.2.1 *Si la valeur de vérité de x_i est différente de la valeur de vérité de y_i $\forall i$ alors C sera satisfaite ssi $C[y]$ est satisfaite.*

- De plus, si on calcule les sous-bases maximales consistantes Y de Z révisé avec y, on trouve que :

  Propriété A.2.2 *Si C est satisfiable alors il existe Y tel que $Y \vdash z_i$ $(\forall i)$ et $Y \vdash s$. Un tel Y n'inférera pas $\neg s$.*

Preuve :
C satisfiable $\Leftrightarrow$ il existe une interprétation Φ dans laquelle C est vraie.
Posons :

$$W = \{x_i \text{ tels que } \Phi(x_i) \text{ soit } vrai\} \cup \{yj \text{ tels que } \Phi(xj) \text{ soit } faux\}.$$

$\forall i$, $\Phi(x_i) \neq \Phi(y_i)$ puisque $y_i = \neg x_i$.
On a donc : $\forall j$ $C_j[y]$ satisfaite puisque C_j est satisfaite et que x_i et y_i n'ont pas la même valeur de vérité $\forall i$.
On peut ainsi en déduire que y est satisfaite, donc que $Y = W \cup \{y\}$ est maximale consistante pour Z révisé par y.
Donc, il existe une sous-base maximale consistante Y de Z révisé par y de la forme :
$Y = \{x_i$ tels que $\Phi(x_i)$ soit *vrai*$\} \cup \{y_i$ tels que $\Phi(x_i)$ soit *faux*$\} \cup \{y\}$.
Or une telle sous-base infère tous les z_i et infère s.
Donc Y n'infère pas $\neg s$

- Z est un ensemble de conjonctions de clauses de Horn.
- y est un ensemble de conjonctions de clauses de Horn puisque les $C_j[y]$ sont des clauses de Horn.
- x est une clause de Horn.

Montrons maintenant que SAT implique co-SBR-HORN et que co-SBR-HORN implique SAT.

- SAT implique co-SBR-HORN :

C satisfiable
$\Rightarrow$
il existe une sous-base maximale consistante Y ($Y = W \cup \{y\}$) de Z révisé par y telle que Y infère tous les z_i et infère s et Y n'infère pas $\neg s$
$\Leftrightarrow$
co-SBR-HORN

- co-SBR-HORN implique SAT :

co-SBR-HORN
$\Leftrightarrow$
il existe Y une sous-base maximale consistante de Z révisé par y telle que Y n'infère pas $\neg s$.
Or $\neg s$ appartient à Z, donc si Y n'infère pas $\neg s$, cela signifie que Y est inconsistante avec $\neg s$, donc que Y infère s, donc que Y contient y, que $\forall i$ la valeur de vérité de x_i est différente de celle de y_i et que les $C_j[y]$ sont satisfaites par la même interprétation que celle satisfaisant Y.
Posons comme interprétation Φ qui satisfait Y :
$\Phi(x_i) = vrai$ si x_i appartient à $Y - \{y\}$ et *faux* sinon
$\Rightarrow$
Dans l'interprétation Φ, les $C_j[y]$ sont satisfaites et comme les x_i et les y_i n'ont pas la même valeur de vérité
$\Rightarrow$
les C_j sont satisfaites elles aussi
$\Leftrightarrow$
C satisfiable

Le problème SAT se ramène donc polynomialement à co-SBR-HORN. Et on en déduit que le problème co-SBR-HORN est NP-complet. Donc le problème SBR-HORN est co-NP-complet. □

A.2.3 Complexité de PBR

Définition A.2.5 *Le problème* PBR *se définit de la manière suivante : soit Z un ensemble stratifié de formules, soient y et x deux formules, Z révisé par y infère-t-il x ? L'opérateur de révision est défini comme suit : l'ensemble "Z révisé par y" $= C_n((\vee(Z \Downarrow \neg y)) \wedge y)$, avec l'opérateur $\Downarrow$ défini informellement par $Z \Downarrow y = \{$ensembles de formules n'impliquant pas y et préférés par rapport à l'ordre induit par la stratification de $Z\}$, la notation $C_n(A)$ signifiant l'ensemble des conséquences logiques de A.*

Voir [Neb91] pour plus de détail, en particulier pour la définition formelle de $Z \Downarrow y$. Remarquons que l'ordre utilisé dans le calcul de $Z \Downarrow \neg y$ correspond exactement à l'ordre "Inclusion Based" sur les sous-bases consistantes avec y.

Théorème A.2.6 *Dans le cas propositionnel général, le processus de révision appelé* PBR *est Π_2^p-complet.*

Preuve :(voir [Neb91])

Il faut calculer la complexité du problème P : $x \in C_n(Z)$ révisé par y en utilisant la méthode PBR ?

Dans la méthode PBR, le problème "$x \in C_n(Z)$ révisé par y" consiste à prouver que : "soit Z une base stratifiée, quel que soit Y sous-ensemble de Z qui n'infère pas $\neg y$ et qui est préféré pour l'ordre "Inclusion Based" (voir définition de PBR), alors $Y \cup \{y\}$ infère x".

Étude 1$^{\text{ère}}$ partie : l'appartenance à une classe.

Posons $Z = Z_1 \cup \ldots \cup Z_n$ avec Z_i = classe d'équivalence (ensemble des formules ayant la même priorité).

L'algorithme A.4 est un des algorithmes possibles pour co-PBR.

Algorithme A.4 : PBR (Z, x, y)
début
- deviner un sous-ensemble Y de Z
- vérifier qu'il est préféré pour l'ordre "Inclusion Based" et n'infère pas $\neg y$
  (* c'est-à-dire que quel que soit Z_i,
  on a $Y_i = Y \cap Z_i$ qui est maximal pour l'inclusion parmi les
  sous-ensembles de Z_i tels que $(\cup_{j \geq i} Y_j)$ n'infère pas $\neg y$ *).
- vérifier que $Y \cup \{y\}$ n'infère pas x.

fin

Dans cet algorithme qui est non déterministe (à cause du "deviner") polynomial, on utilise un oracle de complexité NP (même raisonnement que pour FMR).

La complexité de l'algorithme de co-PBR est donc au plus $\mathsf{NP}^{\mathrm{NP}} = \Sigma_2^p$, donc la complexité de PBR est au plus co-$\Sigma_2^p = \Pi_2^p$.

Étude 2$^{\text{ème}}$ partie : la complétude ?

On sait que le problème SBR est Π_2^p-complet.

On vient de montrer que le problème PBR est Π_2^p.

D'autre part, il existe une transformation polynomiale évidente du problème SBR dans le problème PBR : on considère la base de connaissances Z du problème SBR comme étant une base avec priorité dans laquelle toutes les formules ont la même priorité ; Z est donc aussi une base pour PBR. On a ainsi SBR $\propto$ PBR.

On en déduit que le problème PBR est Π_2^p-complet. □

Théorème A.2.7 *Dans le cas Horn (voir définition A.2.2 page 239), le processus de révision appelé* PBR *est co-*NP*-complet (il sera noté* PBR-HORN*).*

Preuve :

Étude 1ère partie : l'appartenance à une classe.

L'algorithme pour co-PBR-HORN est le même que celui de co-PBR. Il s'agit d'un algorithme non déterministe (à cause du "deviner") polynomial, dans lequel on utilise un oracle de complexité P.

Le problème co-PBR-HORN appartient donc à la classe de complexité NP, et le problème PBR-HORN appartient à co-NP.

Étude 2ème partie : la complétude ?

On sait que le problème SBR-HORN est co-NP-complet.

On vient de montrer que le problème PBR-HORN est co-NP.

D'autre part, il existe une transformation polynomiale évidente du problème SBR-HORN dans le problème PBR-HORN : la même que celle qui permet de passer de SBR à PBR. On a ainsi SBR-HORN $\propto$ PBR-HORN.

On en déduit que le problème PBR-HORN est NP-complet. □

A.2.4 Complexité de UBR

Définition A.2.6 *Le problème* UBR *est défini de manière identique au problème* PBR *(voir définition A.2.5 page ci-contre) mais en imposant que la base Z soit stratifiée strictement (une seule formule par strate).*

Théorème A.2.8 *Dans le cas propositionnel général, le processus de révision appelé* UBR *appartient à la classe de complexité Δ_2^p - (NP $\cup$ co-NP), si NP $\neq$ co-NP.*

Preuve : (voir [Neb91])

Il faut calculer la complexité du problème $P : x \in C_n(Z)$ révisé par y en utilisant la méthode UBR ?

Dans la méthode UBR, le problème "$x \in C_n(Z)$ révisé par y" consiste à prouver que : "soit Z une base stratifiée telle que chaque strate ne contienne qu'une seule formule, quel que soit Y sous-ensemble de Z qui n'infère pas $\neg y$ et qui est préféré pour l'ordre "Inclusion Based", alors $Y \cup \{y\}$ infère x".

Étude 1ère partie : l'appartenance à une classe.

$Z = Z_1 \cup \ldots \cup Z_n$ avec $Z_i = \{z_i\}$ = singleton (on a une formule par niveau de priorité et pas plus).

Au pire, UBR aura la même complexité que PBR. On peut toutefois espérer que le fait que chaque classe d'équivalence Z_i soit un singleton simplifie la résolution du problème.

L'algorithme A.5 page suivante est une des solutions possibles pour implémenter UBR.

Remarque : cet algorithme peut paraître plus compliqué que ceux de SBR ou PBR. Cela vient du fait que le calcul des thèses préférées est intégré directement à l'algorithme, ce qui n'était pas le cas pour les algorithmes précédents. En fait, nous allons voir que sa complexité est moindre.

Ici, on a un algorithme polynomial dans lequel on fait appel à un oracle de complexité NP. On peut donc en déduire que le problème UBR appartient à la classe des problèmes de complexité $\mathsf{P}^{\mathrm{NP}} = \Delta_2^p$.

Algorithme A.5 : UBR (Z, x, y)
début
  initialiser X à $\varnothing$
  initialiser i à 1
1 **si** $X \cup Z_i$ n'infère pas $\neg y$ **alors**
    $X \leftarrow X \cup Z_i$
  incrémenter i
  si on a traité tous les niveaux de priorité **alors**
    vérifier que $X \cup \{y\}$ infère x
  else
    recommencer à partir de l'étape 1 de l'algorithme
fin

Étude $2^{\text{ème}}$ partie : la complétude ?

L'étude de la complétude est plus difficile. Essayons de raffiner l'appartenance à la classe Δ_2^p en vérifiant si UBR $\in$ NP ou co-NP.

Pour cela, on va essayer de comparer UBR à des problèmes connus de NP et co-NP. Les problèmes les plus classiques sont GSAT et UNGSAT (satisfaisabilité et insatisfaisabilité d'une formule en logique classique propositionnelle). On constate alors que :

- le problème GSAT se ramène au problème UBR de manière polynomiale (même raisonnement que pour la démonstration de FMR) :
  Soit H formule propositionnelle, posons : $Z = \{H\}$, $y = vrai$, $x = H$. On a alors :
  $$H \text{ est satisfiable} \Leftrightarrow H \in C_n(H) \text{ révisé par } vrai.$$
  Remarque : La base $Z = \{H\}$ est bien une base avec priorité non ambiguë car elle ne contient qu'une seule classe d'équivalence représentant la seule formule de la base qui peut être vue comme la formule à la fois la plus prioritaire et la moins prioritaire.
  En conclusion, on a GSAT qui se transforme polynomialement en UBR (noté GSAT $\propto$ UBR).
- le problème UNGSAT (complémentaire de GSAT) se ramène au problème UBR de manière polynomiale (même raisonnement que pour la démonstration de FMR) :
  Soit H formule propositionnelle, posons : $Z = \varnothing$, $y = H$, $x = \perp$. On a alors :
  $$H \text{ est insatisfiable} \Leftrightarrow \perp \in C_n(\varnothing) \text{ révisé par } H.$$
  Remarque : La base $Z = \varnothing$ est bien une base avec priorité non ambiguë car elle ne contient aucune formule.
  En conclusion, on a UNGSAT $\propto$ UBR.

En reprenant le même raisonnement que pour FMR, on prouve que, si NP $\neq$ co-NP, UBR $\in \Delta_2^p$ - (NP $\cup$ co-NP).

□

Théorème A.2.9 *Dans le cas Horn (voir définition A.2.2 page 239), le processus de révision appelé* UBR *appartient à la classe de complexité* P *(il sera noté* UBR-HORN*).*

Preuve :

L'algorithme pour UBR-HORN est le même que celui de UBR. C'est un algorithme polynomial, dans lequel on utilise un oracle qui est désormais de complexité P puisque le problème de satisfaction d'un ensemble de conjonctions de clauses de Horn est polynomial.

Le problème UBR-HORN est donc dans la classe de complexité P. □

Théorème A.2.10 *Dans le cas Horn strict (voir définition A.2.3 page 239), le processus de révision appelé* Ubr *appartient à la classe de complexité* P *(il sera noté* Ubr-HornS*).*

Preuve : La preuve est la même que pour Ubr-Horn. □

A.2.4.1 Remarques d'ordre général sur les processus de révision et d'inférence

Il existe entre les processus de révision et les processus d'inférence non-monotone des liens étroits (voir le chapitre 5 page 193). Typiquement, il y a une correspondance entre le problème "Z révisé par y infère-t-il x ?" et le problème "$(E, <)$ infère-t-il x ? avec $(E, <)$ = résultat de l'opération de révision de Z par y".

Par exemple, on voit que le processus Sbr correspond exactement au problème Uni-T, que même que le processus Pbr correspond au problème Uni-Incl et que Ubr correspond au problème 1/Strate.

Ce sont toutes ces correspondances qui expliquent la présence de processus de révision dans ce document dédié aux processus d'inférence. Elles justifient les similarités d'algorithmes et de démonstrations.

Annexe B

Tableau récapitulatif des transformations polynomiales utilisées

Dans ce chapitre, je récapitule toutes les transformations polynomiales utilisées dans les sections 4.1.2 page 75 et 4.1.3 page 110 lors des démonstrations de complexité.

Ce récapitulatif va se faire sous la forme d'un tableau défini comme suit :

- les lignes correspondent aux problèmes Q étudiés dans ce document qui sont :
  - soit les problèmes correspondant aux relations d'inférence non-monotone dont j'ai étudié la complexité,
  - soit les problèmes rencontrés lors des démonstrations d'appartenance de ces relations d'inférence non-monotone ;
- les colonnes correspondent aux différents cas traités :
  - le cas général,
  - le cas d'une base stratifiée strictement,
  - le cas des formules CNF,
  - le cas des conjonctions de clauses de Horn (appelé cas de Horn) ;
  - le cas d'une base de conjonctions de clauses de Horn stratifiée strictement ;
- à l'intersection d'une ligne et d'une colonne, on trouve alors le ou les problèmes Q' ainsi que la transformation utilisée pour démontrer que $Q' \propto Q$ dans le but de prouver la complétude de Q ou de raffiner l'appartenance de Q à une classe de complexité.

	Cas propositionnel général	Cas d'une base stratifiée strictement	Cas des formules CNF	Cas de Horn	Cas d'une base de Horn stratifiée strictement
UNI-T	*Pb :* 2-QBF *Ins :* "$\exists a \forall b G(a,b)$" *Transformation :* - $H = \neg G(a,b)$, - $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, \neg G(a,b)\}$.	Non concerné par ce cas.	*Pb :* 2QBF-DNF *Ins :* "$\exists a \forall b G(a,b)$" *Transformation :* - $H = \neg G(a,b)$ - $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, \neg G(a,b)\}$.	*Pb :* SAT *Ins :* "C" avec $C = \{C_j\}$ pour $j = 1 \ldots m$ ensemble de clauses quelconques à satisfaire, et $V(C) = \{x_1, \ldots, x_n\}$ l'ensemble des variables propositionnelles de C *Transformation :* - $H = \neg s$ - $E = \{p, x_1, \ldots, x_n, y_1, \ldots, y_n, \neg z_1, \ldots, \neg z_n, \neg s\}$ avec $y_1, \ldots, y_n, z_1, \ldots, z_n$, s nouveaux symboles de variables propositionnelles, et avec la formule $p = (z_1 \ldots z_n \rightarrow s) \wedge \bigwedge_{i=1 \ldots n}((\neg x_i \vee \neg y_i) \wedge (y_i \rightarrow z_i) \wedge (x_i \rightarrow z_i)) \wedge \bigwedge_{j=1 \ldots m} C_j[y]\}$, avec la notation suivante : $C_j[y]$ = la clause C_j dans laquelle on remplace tous les x_i positifs par des $\neg y_i$	Non concerné par ce cas.

EXI-T	*Pb :* 2-QBF *Ins :* "$\exists a \forall b G(a,b)$" *Transformation :* - $H = G(a,b)$, - $E = \{a_1, \dots, a_n, \neg a_1, \dots, \neg a_n\}$	Non concerné par ce cas.	*Pb :* co-2QBF-DNF *Ins :* "$\forall a \exists b G(a,b)$ avec $G(a,b)$ sous-forme CNF" *Transformation :* - $H = s$, - $E = \{a_1, \dots, a_n, \neg a_1, \dots, \neg a_n, G(a,b) \vee s\}$ avec s une nouvelle variable propositionnelle	*Pb :* SAT *Ins :* "C" avec $C = \{C_j\}$ pour $j = 1 \dots m$ ensemble de clauses quelconques à satisfaire, et $V(C) = \{x_1, \dots, x_n\}$ l'ensemble des variables propositionnelles de C *Transformation :* - $H = s$ - $E = \{p, x_1, \dots, x_n, y_1, \dots, y_n, \neg z_1, \dots, \neg z_n, \neg s\}$ avec $y_1, \dots, y_n, z_1, \dots, z_n$, s nouveaux symboles de variables propositionnelles, et avec la formule $p = (z_1 \dots z_n \rightarrow s) \wedge \bigwedge_{i=1\dots n}((\neg x_i \vee \neg y_i) \wedge (y_i \rightarrow z_i) \wedge (x_i \rightarrow z_i)) \wedge \bigwedge_{j=1\dots m} C_j[y]\}$, avec la notation suivante : $C_j[y]$ = la clause C_j dans laquelle on remplace tous les x_i positifs par des $\neg y_i$	Non concerné par ce cas.
EXI-T-MIXTE	(relation spéciale étudiée pour la démonstration de complétude de ARG-T-CNF)		Attention !! Il s'agit du cas où E est une base de formules CNF et où H est une formule DNF. *Pb :* co-2QBF-DNF *Ins :* "$\forall a \exists b G(a,b)$ avec $G(a,b)$ sous-forme CNF" *Transformation :* - $H = s$, - $E = \{a_1, \dots, a_n, \neg a_1, \dots, \neg a_n, G(a,b) \vee s\}$ avec s une nouvelle variable propositionnelle		

ARG-T	*Pb :* EXI-T *Ins :* "E, H" *Transformation :* - $f(H) = G$ - $f(E) = E \cup \{H \rightarrow G\}$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans E) *Pb :* co-EXI-T *Ins :* "E, H" *Transformation :* - $f(H) = \neg H$ - $f(E) = E \cup \{\neg H\}$	Non concerné par ce cas.	*Pb :* EXI-T-MIXTE *Ins :* "E, H" *Transformation :* - $f(H) = G$ - $f(E) = E \cup \{H \rightarrow G\}$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans E) *Pb :* co-EXI-T-MIXTE *Ins :* "E, H" *Transformation :* - $f(H) = \neg H$ - $f(E) = E \cup \{\neg H\}$	*Pb :* EXI-T-HORN-POSITIF (cas particulier de EXI-T-HORN dans lequel H est un littéral positif) *Ins :* "E, H" *Transformation :* - $f(H) = G$ - $f(E) = E \cup \{H \rightarrow G\}$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans E) *Pb :* co-EXI-T-HORN-POSITIF *Ins :* "E, H" *Transformation :* - $f(H) = \neg H$ - $f(E) = E \cup \{\neg H\}$	Non concerné par ce cas.
UNI-S	Pas de transformation.	Non concerné par ce cas.	Pas de transformation.	Pas de transformation.	Non concerné par ce cas.
EXI-S	*Pb :* EXI-T *Ins :* "E, H" *Transformation :* identité	Non concerné par ce cas.	*Pb :* EXI-T-CNF *Ins :* "E, H" *Transformation :* identité	*Pb :* EXI-T-HORN *Ins :* "E, H" *Transformation :* identité	Non concerné par ce cas.
ARG-S	*Pb :* ARG-T *Ins :* "E, H" *Transformation :* identité	Non concerné par ce cas.	*Pb :* ARG-T-CNF *Ins :* "E, H" *Transformation :* identité	*Pb :* ARG-T-HORN *Ins :* "E, H" *Transformation :* identité	Non concerné par ce cas.
UNI-BO	*Pb :* GSAT *Ins :* "G" *Transformation :* - $H = G$ - $E = \{G\}$ *Pb :* UNGSAT *Ins :* "G" *Transformation :* - $H = \neg G$ - $E = \varnothing$	idem cas général	*Pb :* SAT *Ins :* "G" *Transformation :* - $H = G$ - $E = \{G\}$ *Pb :* UNSAT *Ins :* "G" *Transformation :* - $H = s$ - $E = \{\neg G \rightarrow s\}$ avec s un nouveau symbole propositionnel	Pas de transformation.	Pas de transformation.

Exi-Bo	*Pb :* 2-Qbf, *Ins :* "$\exists a \forall b G(a,b)$" *Transformation :* - $H = G(a,b)$, - $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n\}$	*Pb :* Exi-S *Ins :* "E, H" *Transformation :* - $f(H) = H$ - $(f(E),<) = \{\perp\} \cup E_{strat}$ avec $\perp$ symbolisant la contradiction et constituant seul la première strate et E_{strat} l'ensemble constitué des formules de E ordonnées strictement dans n'importe quel ordre	*Pb :* co-2Qbf-Dnf *Ins :* "$\forall a \exists b G(a,b)$ avec $G(a,b)$ sous-forme Cnf" *Transformation :* - $H = s$, - $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, G(a,b) \vee s\}$ avec s une nouvelle variable propositionnelle	*Pb :* Sat *Ins :* "C" avec $C = \{C_j\}$ pour $j = 1 \ldots m$ ensemble de clauses quelconques à satisfaire, et $V(C) = \{x_1, \ldots, x_n\}$ l'ensemble des variables propositionnelles de C *Transformation :* - $H = s$ - $E = \{p, x_1, \ldots, x_n, y_1, \ldots, y_n, \neg z_1, \ldots, \neg z_n, \neg s\}$ avec $y_1, \ldots, y_n$, $z_1, \ldots, z_n$, s nouveaux symboles de variables propositionnelles, et avec la formule $p = (z_1 \ldots z_n \rightarrow s) \wedge \bigwedge_{i=1\ldots n}((\neg x_i \vee \neg y_i) \wedge (y_i \rightarrow z_i) \wedge (x_i \rightarrow z_i)) \wedge \bigwedge_{j=1\ldots m} C_j[y]\}$, avec la notation suivante : $C_j[y]$ = la clause C_j dans laquelle on remplace tous les x_i positifs par des $\neg y_i$	*Pb :* Exi-Bo-Horn *Ins :* "$(E,<), H$" avec $(E,<) = (\{\Phi^1_1, \ldots, \Phi^1_{k_1}\}, \ldots, \{\Phi^n_1, \ldots, \Phi^n_{k_1}\})$ *Transformation :* - $f(H) = H$ - $(f(E),<) = (\{\Phi^1_1 \wedge \ldots \wedge \Phi^1_{k_1}\}, \ldots, \{\Phi^n_1 \wedge \ldots \wedge \Phi^n_{k_1}\}, \{\Phi^1_1\}, \ldots, \{\Phi^1_{k_1}\}, \ldots, \{\Phi^n_1\}, \ldots, \{\Phi^n_{k_1}\})$
Exi-Bo-Mixte	(relation spéciale étudiée pour la démonstration de complétude de Arg-Bo-Cnf)		Attention !! Il s'agit du cas où E est une base de formules Cnf et où H est une formule Dnf. *Pb :* co-2Qbf-Dnf *Ins :* "$\forall a \exists b G(a,b)$ avec $G(a,b)$ sous-forme Cnf" *Transformation :* - $H = s$, - $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, G(a,b) \vee s\}$ avec s une nouvelle variable propositionnelle		

ARG-BO	*Pb :* EXI-BO *Ins :* "$(E,<),H$" *Transformation :* - $f(H)=G$ - $f((E,<)) = (E\cup\{H\rightarrow G\},<)$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans $(E,<)$) et la formule $H \rightarrow G$ placée seule en première strate *Pb :* co-EXI-BO *Ins :* "$(E,<),H$" *Transformation :* - $f(H)=\neg H$ - $f((E,<)) = (E\cup\{\neg H\},<)$ avec $\neg H$ constituant seule la dernière strate	*Pb :* 1/STRATE-EXI-BO *Ins :* "$(E,<),H$" *Transformation :* - $f(H)=G$ - $f((E,<)) = (E\cup\{H\rightarrow G\},<)$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans $(E,<)$) et la formule $H \rightarrow G$ placée seule en première strate *Pb :* co-1/STRATE-EXI-BO *Ins :* "$(E,<),H$" *Transformation :* - $f(H)=\neg H$ - $f((E,<)) = (E\cup\{\neg H\},<)$ avec $\neg H$ constituant seule la dernière strate	*Pb :* EXI-BO-MIXTE *Ins :* "$(E,<),H$" *Transformation :* - $f(H)=G$ - $f((E,<)) = (E\cup\{H\rightarrow G\},<)$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans $(E,<)$) et la formule $H \rightarrow G$ placée seule en première strate *Pb :* co-EXI-BO-MIXTE *Ins :* "$(E,<),H$" *Transformation :* - $f(H)=\neg H$ - $f((E,<)) = (E\cup\{\neg H\},<)$ avec $\neg H$ constituant seule la dernière strate	*Pb :* EXI-BO-HORN-POSITIF (cas particulier de EXI-BO-HORN dans lequel H est un littéral positif) *Ins :* "$(E,<),H$" *Transformation :* - $f(H)=G$ - $f((E,<)) = (E\cup\{H\rightarrow G\},<)$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans $(E,<)$) et la formule $H \rightarrow G$ placée seule en première strate *Pb :* co-EXI-BO-HORN-POSITIF *Ins :* "$(E,<),H$" *Transformation :* - $f(H)=\neg H$ - $f((E,<)) = (E\cup\{\neg H\},<)$ avec $\neg H$ constituant seule la dernière strate	*Pb :* 1/STRATE-EXI-BO-HORN-POSITIF (cas particulier de 1/STRATE-EXI-BO-HORN dans lequel H est un littéral positif) *Ins :* "$(E,<),H$" *Transformation :* - $f(H)=G$ - $f((E,<)) = (E\cup\{H\rightarrow G\},<)$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans $(E,<)$) et la formule $H \rightarrow G$ placée seule en première strate *Pb :* co-1/STRATE-EXI-BO-HORN-POSITIF *Ins :* "$(E,<),H$" *Transformation :* - $f(H)=\neg H$ - $f((E,<)) = (E\cup\{\neg H\},<)$ avec $\neg H$ constituant seule la dernière strate

UNI-INCL	*Pb :* UNI-T *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	Une seule relation d'inférence pour UNI (EXI, ARG) - INCL (LEX) notée 1/STRATE : *Pb :* ALM (cf. [EG93]) *Ins :* "$C = \{C_1, \ldots, C_m\}$ satisfiable, $X = \{x_1, \ldots, x_n\}$, $O(X) =< x_1, \ldots, x_n >$" *Transformation :* - $H = x_n$ - $(E, <) = \{C_1 \wedge \ldots \wedge C_m, x_1, \ldots, x_n\}$ l'ordre $<$ étant le suivant : la formule $CC = C_1 \wedge \ldots \wedge C_m$ est plus prioritaire que la formule x_1 elle-même plus prioritaire que la formule x_2 elle-même plus prioritaire que … plus prioritaire que x_n.	*Pb :* UNI-T-CNF *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	*Pb :* UNI-T-HORN *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	Une seule relation d'inférence pour UNI (EXI, ARG) - INCL (LEX) notée 1/STRATE-HORN. Pas de transformation.
EXI-INCL	*Pb :* EXI-T *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	(voir dans UNI-INCL)	*Pb :* EXI-T-CNF *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	*Pb :* EXI-T-HORN *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	(voir dans UNI-INCL)
ARG-INCL	*Pb :* ARG-T *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	(voir dans UNI-INCL)	*Pb :* ARG-T-CNF *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	*Pb :* ARG-T-HORN *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	(voir dans UNI-INCL)

Uni-Lex	*Pb :* Uni-Car *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	(voir dans Uni-Incl)	*Pb :* Uni-Car-Cnf *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	*Pb :* Acm *Ins :* "$C = \{C_1, \ldots, C_m\}$ un ensemble de clauses ($X = \{x_1, \ldots, x_n\}$ les variables de C), $k = \{1, \ldots, m\}$ un entier" *Transformation :* - $H = C_k[y] \wedge s$ - $(E, <) = \{p_1, \ldots, p_m\} \cup \{x_1, \ldots, x_n, y_1, \ldots, y_n\} \cup \{\neg z_1, \ldots, \neg z_n, \neg s\}$ avec $y_1, \ldots, y_n, z_1, \ldots, z_n, s$ nouveaux symboles de variables propositionnelles, et avec les formules $p_j = (z_1 \ldots z_n \rightarrow s) \wedge \bigwedge_{i=1 \ldots n}((\neg x_i \vee \neg y_i) \wedge (y_i \rightarrow z_i) \wedge (x_i \rightarrow z_i)) \wedge C_j[y]$, avec la notation suivante : $C_j[y]$ = la clause C_j dans laquelle on remplace tous les littéraux positifs x_i par des littéraux négatifs $\neg y_i$. L'ordre $<$ entre les formules de la base est le suivant : les p_j plus prioritaires que les x_i et les y_i, eux-mêmes plus prioritaires que les $\neg z_i$ et le $\neg s$	(voir dans Uni-Incl)
Exi-Lex	*Pb :* Exi-Car *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	(voir dans Uni-Incl)	*Pb :* Exi-Car-Cnf *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	non étudié	(voir dans Uni-Incl)

ARG-LEX	*Pb :* ARG-CAR *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	(voir dans UNI-INCL)	*Pb :* ARG-CAR-CNF *Ins :* "E, H" *Transformation :* identité (considérons toutes les formules de E en une seule strate)	non étudié	(voir dans UNI-INCL)
MAX-GSAT	*Pb :* GSAT *Ins :* "G" *Transformation :* - $k = 1$ - $E = \{G\}$	Non concerné par ce cas.	*Pb :* SAT *Ins :* "G" *Transformation :* - $k = 1$ - $E = \{G\}$	Pour la version multi-ensemble : *Pb :* MAX-2-SAT *Ins :* "C, k" avec C collection de n clauses à 2 littéraux maximum et k entier $\leq n$ (nombre de clauses de C) *Transformation :* - $f(k) = k + p$ avec $p = \|C \setminus CH\|$ - $E = CH \cup CNH$ avec $CH =$ {les clauses de C qui sont déjà sous forme de clauses de Horn} et CNH = {les clauses définies de la manière suivante : $\forall c$ clause de C de la forme $\rightarrow a_1 a_2$ ($c \in C \setminus CH$), les clauses de HORN $\rightarrow a_1$, $\rightarrow a_2$, $a_1 a_2 \rightarrow \in CNH$}	Non concerné par ce cas.
NGSAT-CAR	*Pb :* GSAT *Ins :* "G" *Transformation :* - $E = \{G\}$ - $H = \neg G$ - $k = 1$	Non concerné par ce cas.	*Pb :* SAT *Ins :* "G" *Transformation :* - $E = \{G\}$ - $H = \neg G$ - $k = 1$	non étudié	Non concerné par ce cas.
MAX-GSAT-STRICT	*Pb :* GSAT *Ins :* "G" *Transformation :* - $k = 0$ - $E = \{G\}$	Non concerné par ce cas.	*Pb :* SAT *Ins :* "G" *Transformation :* - $k = 0$ - $E = \{G\}$	Pour la version multi-ensemble : *Pb :* MAX-SAT-ME-HORN *Ins :* "E, k" *Transformation :* - $f(k) = k - 1$ - $f(E) = E$ (donc une seule strate)	Non concerné par ce cas.

MAX-GSAT-ARRAY	*Pb :* MAX-GSAT *Ins :* "E, k" *Transformation :* - $f(k) =< k >$ - $f(E) = E$ avec $f(E)$ une base stratifiée constituée d'une seule strate	Non concerné par ce cas.	idem cas général	Pour la version multi-ensemble : *Pb :* MAX-SAT-ME-HORN *Ins :* "E, k" *Transformation :* - $f(k) =< k >$ - $f(E) = E$ avec $f(E)$ une base stratifiée constituée d'une seule strate	Non concerné par ce cas.
NGSAT-LEX	*Pb :* NGSAT-CAR *Ins :* "E, H, k" *Transformation :* - $f(E) = E$ avec $f(E)$ une base stratifiée constituée d'une seule strate - $f(H) = H$ - $f(k) =< k >$	Non concerné par ce cas.	idem cas général	non étudié	Non concerné par ce cas.
UNI-CAR	*Pb :* ACM *Ins :* "$C = \{C_1, \ldots, C_m\}$, $k \in \{1, \ldots, m\}$" *Transformation :* - $H = C_k$ - $E = \{C_1, \ldots, C_m\}$	Non concerné par ce cas.	*Pb :* ACM *Ins :* "$C = \{C_1, \ldots, C_m\}$, $k \in \{1, \ldots, m\}$" *Transformation :* - $H = C_k$ - $E = \{C_1, \ldots, C_m\}$	Pour la version multi-ensemble : *Pb :* "UNI-LEX-HORN" *Transformation :* - $f(H) = H$ - $f(E)$ = le résultat de l'algorithme donné en section 4.1.37 page 130	Non concerné par ce cas.
EXI-CAR	*Pb :* 2-QBF *Ins :* "$\exists a \forall b G(a, b)$" *Transformation :* - $H = G(a, b)$ - $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n\}$	Non concerné par ce cas.	*Pb :* co-2QBF-DNF *Ins :* "$\forall a \exists b G(a, b)$ avec $G(a, b)$ sous-forme CNF" *Transformation :* - $H = s$ - $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, G(a, b) \vee s\}$ avec s une nouvelle variable propositionnelle	non étudié	Non concerné par ce cas.

EXI-CAR-MIXTE	(relation spéciale étudiée pour la démonstration de complétude de ARG-CAR-CNF)		Attention !! Il s'agit du cas où E est une base de formules CNF et où H est une formule DNF. *Pb :* co-2QBF-DNF *Ins :* "$\forall a \exists b G(a,b)$ avec $G(a,b)$ sous-forme CNF" *Transformation :* - $H = s$ - $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n, G(a,b) \vee s\}$ avec s une nouvelle variable propositionnelle		
ARG-CAR	*Pb :* EXI-CAR *Ins :* "E, H" *Transformation :* - $f(H) = G$ - $f(E) = E \cup \{H \rightarrow G\}$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans E)	Non concerné par ce cas.	*Pb :* EXI-CAR-MIXTE *Ins :* "E, H" *Transformation :* - $f(H) = G$ - $f(E) = E \cup \{H \rightarrow G\}$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans E)	non étudié	Non concerné par ce cas.
UNI-E	(voir [Got92])	Non concerné par ce cas.	non étudié	(voir [KS91])	Non concerné par ce cas.
EXI-E	(voir [Got92])	Non concerné par ce cas.	non étudié	(voir [KS91] et [Sti90])	Non concerné par ce cas.
ARG-E	*Pb :* EXI-E *Ins :* "$E = (W, D), H$" *Transformation :* - $f(H) = G$ - $f(E) = (W \cup \{H \rightarrow G\}, D)$ avec G nouvelle variable propositionnelle (G n'apparaît pas dans W, ni dans D) *Pb :* co-EXI-E *Ins :* "$E = (W, D), H$" *Transformation :* - $f(H) = \neg H$ - $f(E) = (W, D \cup \{: \neg H / \neg H\}$	Non concerné par ce cas.	non étudié	*Définition de Stillman :* (voir [Sti90]) *Pb :* co-EXI-E-HORN *Ins :* "$E = (W, D), H$" *Transformation :* - $f(H) = \neg H$ - $f(E) = (W, D \cup \{: \neg H / \neg H\}$	Non concerné par ce cas.

Index

Symboles

2-Qbf . . . *voir* problèmes de référence
2-Qbf-Dnf . . . *voir* problèmes de référence

A

Acm . . . *voir* problèmes de référence
affaiblissement droit . . . *voir* propriétés de déduction
algorithme . . . *voir* complexité
Alm . . . *voir* problèmes de référence
approche
    sémantique . . . 2
    syntaxique . . . 3
assertion conditionnelle . . . 55

B

base de Horn . . . 120
base de Horn stricte . . . 120
BDD "Binary Decision Diagram" . . . 204
    ordonné . . . 204
    réduit . . . 204
boucle . . . *voir* propriétés de déduction

C

clauses de Horn . . . 120, 124, 126, 130, 137, 140
complexité
    algorithme . . . 50
    hiérarchie polynomiale . . . 52
    instance d'un problème . . . 50
    langage formel d'un problème . . . 51
    machine
        de Turing . . . 51
    machines
        déterministes . . . 51
        déterministes à oracle . . . 52
        non déterministes . . . 52
        non déterministes à oracle . . . 52
    problème . . . 50
        complémentaire . . . 51
        décidable . . . 51
        de complexité NP^X . . . 52
        de complexité P^X . . . 52
        de décision . . . 50
        indécidable . . . 51
        non déterministe polynomial . . . 52
        polynomial . . . 52
        X-complet . . . 52
        X-difficile . . . 52
    temporelle d'un algorithme . . . 51
    transformation polynomiale . . . 52
conditionnalisation . . . *voir* propriétés de déduction
conditionnalisation faible . . . *voir* propriétés de déduction
condition de régularité . . . 57
conséquence argumentative avec préférences . . . *voir* principe d'inférence argumentatif
conséquence argumentative . . . *voir* principe d'inférence argumentatif
conséquence faible . . . *voir* principe d'inférence existentiel
conséquence faible avec préférences . . . *voir* principe d'inférence existentiel
conséquence forte *voir* principe d'inférence universel
conséquence forte avec préférences . . . *voir* principe d'inférence universel
contraposition . . . *voir* propriétés de déduction
coupure . . . *voir* propriétés de déduction
cumulativité . . . *voir* propriétés de déduction

D

DAG "directed acyclic graph" . . . 204

E

EHD ("easy half deduction") . . . *voir* propriétés de déduction
ensemble régulier . . . 57
équivalence . . . *voir* propriétés de déduction
équivalence logique gauche . . . *voir* propriétés de déduction
expectations . . . *voir* sous-ensembles de croyances
extensions . . . *voir* sous-ensembles de croyances

F

fermeture . . . *voir* propriétés de déduction
fermeture rationnelle . . . 61
formules CNF . . . 114–119

G

Gsat . . . *voir* problèmes de référence

H

hiérarchie polynomiale . . . *voir* complexité

I

instance d'un problème . . . *voir* complexité

L

langage compact 55
langage formel d'un problème ...... *voir* complexité

M

machines déterministes .... *voir* complexité
machines déterministes à oracle ..... *voir* complexité
machines non déterministes .... *voir* complexité
machines non déterministes à oracle *voir* complexité
machine de Turing .... *voir* complexité
Max-2-Sat .... *voir* problèmes de référence
Max-Gsat .... *voir* problèmes de référence
Max-Gsat-Array .... *voir* problèmes de référence
Max-Gsat-Strict .... *voir* problèmes de référence
modèle 55
- cumulatif 57
- cumulatif ordonné 58
- préférentiel 60
- préférentiel adéquat 71
- rangé 61
- simplement cumulatif 63
- simplement préférentiel 63

monotonie .... *voir* propriétés de déduction
monotonie prudente ... *voir* propriétés de déduction
monotonie rationnelle .. *voir* propriétés de déduction
monotonie rationnelle faible .... *voir* propriétés de déduction
MPC (modus ponens) . *voir* propriétés de déduction
multi-ensemble .... *voir* sous-ensembles de croyances

N

Ngsat-Car .... *voir* problèmes de référence
Ngsat-Lex .... *voir* problèmes de référence
Nihil ex Absurdo ...... *voir* propriétés de déduction

O

OBDD .... *voir* BDD ordonné
ordre élitiste .... *voir* ordre entre sous-ensembles de croyances
ordre "Best-Out" .... *voir* ordre entre sous-ensembles de croyances
ordre "Inclusion Based" .... *voir* ordre entre sous-ensembles de croyances
ordre basé sur la cardinalité .... *voir* ordre entre sous-ensembles de croyances
ordre démocratique .... *voir* ordre entre sous-ensembles de croyances
ordre entre sous-ensembles de croyances
- élitiste 11
- "Best-Out" 19, 48, 82, 83, 85, 111, 116, 124, 143
- "Inclusion Based" 14, 48, 87–89, 112, 117, 126, 144
- basé sur la cardinalité .. 14, 48, 90, 94, 96, 118, 130
- démocratique 10
- lexicographique .. 14, 48, 97, 103, 104, 119, 137

ordre lexicographique .... *voir* ordre entre sous-ensembles de croyances

P

postulats AGM .... *voir* révision
postulats de révision .... *voir* révision
postulats étendus ...... *voir* propriétés de déduction
postulats de base ...... *voir* propriétés de déduction
préservation de la consistance .... *voir* propriétés de déduction
principe d'inférence
- argumentatif 27, 78, 81, 106, 114, 115, 119, 120, 124, 140
- argumentatif avec préférences 28, 85, 89, 96, 104, 111, 112, 116–119, 124, 126, 130, 137, 143, 144
- existentiel .. 27, 77, 81, 106, 114, 115, 119, 120, 124, 140
- existentiel avec préférences 28, 83, 88, 94, 103, 111, 112, 116–119, 124, 126, 130, 137, 143, 144
- universel ... 27, 75, 81, 105, 114, 115, 119, 120, 124, 140
- universel avec préférences 28, 82, 87, 90, 97, 111, 112, 116–119, 124, 126, 130, 137, 143, 144

problème .... *voir* complexité
problème de révision .... *voir* révision
problèmes de référence
- 2-Qbf 53
- 2-Qbf-Dnf 114
- Acm 93
- Alm 102
- Gsat 53
- Max-2-Sat 133
- Max-Gsat 90
- Max-Gsat-Array 99
- Max-Gsat-Strict 95
- Ngsat-Car 92
- Ngsat-Lex 100
- Sat 53
- Ungsat 53

problème complémentaire .... *voir* complexité
problème décidable .... *voir* complexité
problème de complexité NP^X .... *voir* complexité
problème de complexité P^X .... *voir* complexité
problème de décision .... *voir* complexité
problème indécidable .... *voir* complexité
problème non déterministe polynomial .... *voir* complexité
problème polynomial .... *voir* complexité
problème X-complet .... *voir* complexité
problème X-difficile .... *voir* complexité
propriétés de déduction d'une relation d'inférence
- affaiblissement droit ("right weakening") ... 56, 67, 166
- boucle 58
- conditionnalisation 59, 68
- conditionnalisation faible 45, 68, 161
- contraposition 57
- coupure 56, 68, 166
- cumulativité 68, 170
- D 59
- EHD ("easy half deduction") 57
- équivalence 57
- équivalence logique gauche 56, 67, 165
- ET 57, 67, 172
- fermeture 67
- monotonie 57
- monotonie prudente 56, 68, 168
- monotonie rationnelle 61, 68, 163
- monotonie rationnelle faible 68, 173
- MPC (modus ponens) 57

Nihil ex Absurdo . . . 155
OU . . . 59, 68, 161
postulats de Gärdenfors et Makinson
étendus . . . 68
de base . . . 67
préservation de la consistance . . . 68, 160
propriété numéro 15 de Kraus, Lehmann, Magidor . . . 58
propriété numéro 19 de Kraus, Lehmann, Magidor . . . 59
propriété numéro 20 de Kraus, Lehmann, Magidor . . . 60
propriété numéro 21 de Kraus, Lehmann, Magidor . . . 60
propriété numéro 22 de Kraus, Lehmann, Magidor . . . 60
propriété numéro 9 de Kraus, Lehmann, Magidor . . . 57
réciprocité . . . 68
réflexivité . . . 56, 67, 160
réflexivité restreinte . . . 155
rationalité de la négation . . . 60
rationalité disjonctive . . . 60, 68
S . . . 59, 68
supra-classicité . . . 67, 158
supra-classicité restreinte . . . 155
transitivité . . . 57
propriété D . . . *voir* propriétés de déduction
propriété ET . . . *voir* propriétés de déduction
propriété numéro 15 de Kraus, Lehmann, Magidor *voir* propriétés de déduction
propriété numéro 19 de Kraus, Lehmann, Magidor *voir* propriétés de déduction
propriété numéro 20 de Kraus, Lehmann, Magidor *voir* propriétés de déduction
propriété numéro 21 de Kraus, Lehmann, Magidor *voir* propriétés de déduction
propriété numéro 22 de Kraus, Lehmann, Magidor *voir* propriétés de déduction
propriété numéro 9 de Kraus, Lehmann, Magidor *voir* propriétés de déduction
propriété OU . . . *voir* propriétés de déduction
propriété S . . . *voir* propriétés de déduction

R

réciprocité . . . *voir* propriétés de déduction
réflexivité . . . *voir* propriétés de déduction
réflexivité restreinte . . . *voir* propriétés de déduction
révision
postulats . . . 193
problème . . . 193
rationalité de la négation . . . *voir* propriétés de déduction
rationalité disjonctive . . . *voir* propriétés de déduction
RBDD . . . *voir* BDD réduit
relation d'inférence . . . 55
expectative . . . 24
comparative . . . 70
opération d'inférence expectative . . . 69
rationnelle . . . 61
relation de conséquence
cumulative . . . 56
cumulative monotone . . . 63
monotone . . . 63
préférentielle . . . 59
relation de conséquence cumulative . . . *voir* relation d'inférence
relation de conséquence cumulative monotone . . *voir* relation d'inférence
relation de conséquence monotone . . . *voir* relation d'inférence
relation de conséquence préférentielle . . . *voir* relation d'inférence
ROBDD . . . *voir* BDD ordonné réduit

S

Sat . . . *voir* problèmes de référence
sous-base consistante . . . *voir* sous-ensembles de croyances
sous-base consistante maximale pour l'inclusion *voir* sous-ensembles de croyances
sous-base consistante maximale pour la cardinalité *voir* sous-ensembles de croyances
sous-base fortement maximale consistante . . . *voir* sous-ensembles de croyances
sous-base maximale consistante . . . *voir* sous-ensembles de croyances
sous-base préférée pour l'ordre lexicographique . *voir* sous-ensembles de croyances
sous-ensembles de croyances
expectations . . . 67
ensembles d'expectations . . . 67
ordres d'expectations . . . 67
extensions . . . 17, 48, 105, 106, 119, 140
multi-ensemble . . . 127
cardinal . . . 128
sous-base consistante . . . 128
sous-base préférée lexicographiquement . 128
sous-base préférée pour la cardinalité . . . 128
sous-ensemble . . . 127
stratifié . . . 127
thèse . . . 128
sous-base
consistante . . 8, 48, 81–83, 85, 111, 115, 116, 124, 143
consistante maximale pour l'inclusion (maximale consistante) . . . 8
consistante maximale pour la cardinalité . 14
fortement maximale consistante . . . 12
préférée pour l'ordre lexicographique . . . 15
sous-théorie préférée . . . 9
thèses . 8, 48, 75, 77, 78, 87–90, 94, 96, 97, 103, 104, 112, 114, 117–120, 126, 130, 137, 144
sous-théorie préférée . . . *voir* sous-ensembles de croyances
supra-classicité . . . *voir* propriétés de déduction
supra-classicité restreinte . . . *voir* propriétés de déduction
système C . . . *voir* système d'inférence
système CL . . . *voir* système d'inférence
système CM . . . *voir* système d'inférence
système d'inférence
système C . . . 56
système CL . . . 58
système CM . . . 63
système M . . . 63
système P . . . 59
système R . . . 61
système M . . . *voir* système d'inférence
système P . . . *voir* système d'inférence
système R . . . *voir* système d'inférence

T

thèses .... *voir* sous-ensembles de croyances
transformation polynomiale .... *voir* complexité
transitivité .... *voir* propriétés de déduction

U

Ungsat .... *voir* problèmes de référence