



THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Évaluation de logiques non-monotones pour la formalisation du droit

Moreira Machado, Carlos

Award date:
1995

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix - Namur

Institut d'informatique

Evaluation de logiques
non-monotones pour la
formalisation du droit

Mémoire de fin d'études présenté par :

Moreira Machado Carlos

promoteur : **Pierre-Yves Schobbens**

En vue de l'obtention du diplôme de
licencié et maître en informatique

Anée académique 1994-1995.

Remerciements

Je tiens tout d'abord à adresser mes remerciements à Monsieur P-Y. Schobben, promoteur de ce mémoire, pour l'aide précieuse qu'il m'a apportée notamment en ce qui concerne les questions techniques. .

Je suis également très reconnaissant à Monsieur D. Poulin, Professeur à l'Université de Montréal en Faculté de droit, pour la culture générale en intelligence artificielle appliquée au droit qu'il m'a permis d'acquérir durant mon stage de fin d'études au centre de recherche en droit public de l'Université de Montréal.

Je remercie également toutes les personnes qui d'une manière ou d'une autre, m'ont permis de produire ce mémoire.

Ma gratitude va aussi à tout le corps professoral et à la direction de l'Institut d'informatique de Facultés Universitaires Notre-Dame de la Paix pour la formation qu'ils m'ont permis d'acquérir.

Carlos Moreira Machado

Table des matières

<i>Introduction</i>	11
---------------------------	----

<i>Chapitre premier : Problématique</i>	13
---	----

Le droit, l'interprétation et l'IA	13
---	----

1. Le caractère inéluctable d'indétermination de la règle de droit.....	13
2. Polémique sur l'interprétation des lois.....	14
2.1 Le courant positiviste.....	14
2.2 Le courant non positiviste.....	17
3. Les obstacles à la formalisation du droit.....	19
3.1 La texture ouverte.....	19
3.1.1 La sous-détermination.....	20
3.1.2 La surdétermination.....	21
3.1.3 L'invalidation.....	21
3.1.4 L'imprécision (le flou).....	22
3.2 Les exceptions aux règles.....	22

Aperçu de quelques projets en IA et droit	25
--	----

4. Le droit modélisé dans un programme logique.....	25
4.1 The British Nationality Act.....	25
4.2 The Pension Rules of the Indian Central Civil Service.....	28
5. Différentes techniques pour traiter la texture ouverte.....	28
5.1 Pondérations.....	28
5.2 Résoudre uniquement les cas simples.....	30
6. Modélisation de l'argumentation juridique.....	31
6.1 Modélisation à base de cas et de règles.....	31
6.1.1 Le système HYPO.....	32
6.2 Un modèle computationnel d'argumentation dialectique.....	36
7. Représentation des connaissances juridiques.....	42
7.1 La DataLex Legal Workstation.....	42
7.2 Prolex.....	43

<i>Chapitre 2 : Formalisations du raisonnement non-monotone</i>	49
---	----

Concepts de base	50
-------------------------------	----

1. Propriétés méta-théoriques.....	51
------------------------------------	----

Logiques à préférence de modèles	53
---	----

2. L'hypothèse de monde clos.....	54
3. La circonscription.....	55
3.1 Forme syntaxique.....	55
3.2 Forme sémantique.....	57

Logiques à base de cohérence.....	60
4. Logique de défauts.....	61
5. Logique modale non-monotone.....	66
6. Logique à cohérence maximale.....	68
6.1 Théorie de Poole.....	70
6.2 Théorie de Brewka.....	72
Problèmes des logiques non-monotones.....	73
7. Effort d'unification.....	73
8. Problèmes d'implémentations.....	74
9. Systèmes à processus non-monotone.....	75
10. Système prouveur pour la logique de défauts.....	77
11. Conclusion.....	78
Chapitre 3 : Théorie de Prakken et améliorations.....	79
Théorie de Prakken	79
1. Approche basée sur l'extension	80
1.1 Méthode générale.....	80
1.2 Extension de la logique de défauts avec priorités de Brewka.....	83
1.2.1 Théorie de défauts avec priorités	83
1.2.2 Extension d'une théorie de défauts avec priorités.....	84
1.2.3 Principes juridiques de résolution de conflits.....	86
1.2.4 Formalisation des principes de résolution.....	87
Améliorations.....	90
2. Représentation.....	90
3. Définitions.....	91
4. Circuits de défauts arcs-points.....	92
5. Circuits de défauts quelconques	97
5.1 Circuit quelconque isolé.....	97
5.1.1 Circuit sans défauts points qui contredisent.....	99
5.1.2 Les défauts points peuvent se contredire.....	101
5.2 Ensemble de circuits quelconques disjoints.....	103
5.3 Ensemble de circuits quelconques liés.....	106
5.3.1 Circuits liés par des arcs.....	107
5.3.2 Circuits liés par des points.....	113
Conclusion	119

Introduction

Dès le début des années 70, les juristes découvrent l'intelligence artificielle et l'intelligence artificielle découvre le droit. Dès lors, les deux parties en ont profité puisque le droit est un domaine permettant l'étude de beaucoup de notions examinées en intelligence artificielle (noté I.A.) comme : le raisonnement sur des concepts partiellement définis, le méta-raisonnement, le raisonnement invalidable, ...

La recherche en I.A. ne consiste pas uniquement à développer des programmes informatiques; beaucoup de recherches se portent sur le développement de théories générales permettant d'exprimer, de justifier et/ou de critiquer les idées sur lesquelles des programmes peuvent être basés. Ce mémoire présente certains aspects logiques - basés en grande partie sur la thèse de Prakken¹ - de recherches menées en I.A. appliquée au raisonnement juridique.

Le but de Prakken, dans sa thèse, n'est pas d'étudier les aspects logiques de l'élaboration de juges ou avocats artificiels intelligents; mais bien d'étudier les aspects logiques de systèmes basés sur de la connaissance juridique.

La loi s'applique à un monde ouvert, changeant et imprévisible pour lequel il n'existe pas de règles définitives. C'est pourquoi les lois sont enclines à des exceptions et donc, invalidables. Dans le droit, une place importante est laissée aux conflits d'opinions; si un système à base de connaissances juridiques se veut « le plus complet possible », il devra rendre compte de ces conflits, il devra contenir des informations incohérentes.

L'étude du raisonnement invalidable et du raisonnement à partir de connaissances incohérentes a souvent été présenté (c'est peut-être moins le cas aujourd'hui) comme un défi à la pertinence de la logique pour l'I.A.. Mais, le développement récent dans l'étude logique du raisonnement de sens commun a rendu cette analyse du raisonnement invalidable et/ou sur base d'informations incohérentes - très utilisé en droit - possible.

Dans le chapitre premier de ce mémoire nous examinerons certaines structures du raisonnement juridique qui ne peuvent être analysées par des systèmes de logique classiques et qui sont donc souvent présentés comme un obstacle à des méthodes basées sur la logique. Mais nous verrons que ce n'est pas un obstacle aux méthodes basées sur la logique (non-monotone). Nous ferons ensuite, au chapitre 2, un survol des

¹ [PRA1993]

principales approches de formalisation du raisonnement de sens commun ou non-monotone, c'est à dire des logiques non-monotones. Nous verrons pour terminer au chapitre 3 comment on peut formaliser avec une logique non-monotone l'utilisation de principes et règles de résolution de conflits entre différentes règles de droits applicables. Nous tenterons dans ce chapitre de pousser plus loin la théorie de Prakken².

² [PRA1995]

Chapitre premier : Problématique

Le droit, l'interprétation et l'I.A.

Le droit, par sa tradition de débat sur le raisonnement juridique, par son mode de fonctionnement jurisprudentiel et par sa codification « structurée » des données, est un domaine offrant un grand attrait pour l'intelligence artificielle. Cependant, le droit pose encore beaucoup de problèmes aux informaticiens, quand il s'agit de représenter fidèlement des notions juridiques dans un modèle informatique. En effet, il faut pour cela pouvoir traiter l'indétermination des règles de droit, faire des analogies, manipuler des exceptions et des règles conflictuelles, ainsi que pouvoir s'adapter aux changements.

1. Le caractère inéluctable d'indétermination de la règle de droit

La règle juridique n'est pas figée comme un théorème, elle a un caractère fondamentalement dynamique. Certes, le droit est une mécanique basée sur des concepts, mais une mécanique appliquée au réel, au vivant : or ce vivant ne cesse d'évoluer, il est d'une infinie richesse et d'une variété inépuisable. La règle juridique est donc amenée à bouger, à se préciser, à se nuancer, à s'adapter, à changer pour rester attachée à la vie sociale qu'elle est censée organiser.

On en arrive donc parfois à devoir remplacer une règle, à l'abandonner au profit d'une nouvelle supposée plus adéquate et mieux adaptée. Mais dans d'autres cas, ou avant d'en arriver à ce changement, la règle « vit » par le biais de l'interprétation que lui donnent les juges et de l'application qu'ils en font aux situations concrètes qui leur sont présentées.

Le rêve d'un droit qui serait à ce point précis qu'il n'y aurait plus de place pour l'interprétation est une utopie. L'idéal d'un droit entièrement prévisible et à la portée de tous se heurte à l'objectif de souplesse de la règle de droit, ainsi qu'à l'impossibilité de tout prévoir face à laquelle se trouvent les législateurs.

La loi procède donc par notions, par concepts, généraux et abstraits afin de pouvoir être appliquée à plusieurs situations dans le temps et dans l'espace, lesquelles sont concrètes et donc particulières. Quand la loi parle d'un arbre, elle ne parle pas de tel arbre situé à tel endroit, mais de la notion d'arbre¹. Cette généralité et cette abstraction de la règle de droit ont pour corollaire le caractère inéluctable d'indétermination juridique. Face à une situation concrète, on se demande si elle est bien visée par telles ou telles lois qui sont générales et ne mentionnent pas la situation concrète en question. Il faut se livrer à un travail d'interprétation.

Mais l'indétermination à laquelle il faut faire face peut être plus ou moins grande et de ce fait, le rôle de l'interprète peut lui aussi être plus ou moins grand. En fait, le degré d'indétermination varie selon les domaines du droit et selon que la règle fait appel à des concepts plus ou moins indéterminés, plus ou moins flous.

A propos des ces concepts non définis du droit, le courant positiviste et le courant non-positiviste se sont opposés quant à la façon de les interpréter².

2. Polémique sur l'interprétation des lois

2.1 *Le courant positiviste*

L'école utilitarienne, appartenant au courant positiviste, prônait la distinction entre **ce qu'est la loi**, c'est-à-dire, ce qui est écrit dans les textes de loi, et **ce qu'elle devrait être**, c'est-à-dire, ce qu'on voudrait qu'elle soit par référence à certains standards moraux.

Pour Austin³, qui appartenait à cette école de pensée, l'existence de la loi est une chose et son mérite ou démérite en est une autre. Selon lui, toutes les lois humaines doivent respecter la loi divine, mais si ce n'est pas le cas, les lois restent néanmoins des lois, elles n'en deviennent pas pour autant invalides. D'autres auteurs utilitariens, comme Bentham, tenaient les mêmes propos mais sans faire référence à Dieu, parlant plutôt de principes d'utilité.

¹ Notons que lorsque la loi utilise des concepts ayant un sens particulier (exemples: vente, ASBL, usufruit, ...), elle doit, normalement, les définir.

² L'opposition qui est relatée dans le titre qui suit est tirée de [HAR1958] et [FUL1958].

³ Cité dans [HAR1958]

Pour les utilitariens, l'insistance sur la distinction entre ce qu'est la loi et ce qu'elle devrait être, permet d'éviter deux dangers, l'anarchisme et le réactionnisme. En effet, si on ne fait pas cette distinction, et que l'on considère que ce qui prime c'est ce qu'on pense que la loi devrait être, alors on risque de voir la loi et son autorité dissoutes (anarchisme). Et si on pense plutôt que la loi est la loi, et que l'on met de côté toute considération morale, alors on risque de voir disparaître la « morale » au profit de la loi, puisqu'elle est ce qu'elle doit être (réactionnaire). C'est l'insistance sur cette distinction entre la loi et la morale qui a permis d'atténuer, dans la jurisprudence, la croyance selon laquelle la loi était un idéal, était ce qu'elle devait être, et qu'elle devait être appliquée à la lettre.

Mais la théorie utilitarienne comprend également :

- ☞ une doctrine selon laquelle pour comprendre la vraie nature de la loi, il faut en faire une étude socio-historique, ainsi qu'en analyser le vocabulaire spécifique, et même entreprendre des études purement analytiques des concepts de droit;
- ☞ et une doctrine ou théorie impérativiste selon laquelle la loi est un ensemble d'ordres.

Cette dernière doctrine fut fortement critiquée et réfutée par de nombreux auteurs, notamment par Hart. Qu'est-ce qu'un ordre? C'est, dit-il, l'expression par une personne du désir qu'une autre personne fasse ou ne fasse pas une certaine chose, accompagnée par une menace de punition en cas de désobéissance. Selon lui, pour qu'un ordre soit une loi, il faut qu'il soit général et donné par un « souverain ». Un « souverain » est une personne ou un groupe qui à l'habitude d'être obéi par la société, mais qui lui ne doit obéir à aucune autre personne. Présenté comme ça, la loi est un ordre donné par un commandant, non commandé, de la société. Mais dans cet aspect des choses, on oublie d'établir une certaine connexion avec la morale. En effet, la situation décrite ci-dessus est équivalente à celle d'un homme armé qui dirait à sa victime: « la bourse ou la vie? », à la différence près, qu'ici, « l'homme armé » menace un grand nombre de personnes.

On ne peut pas dire que la loi est un ensemble d'ordres donnés par un homme armé. Cette théorie impérativiste oublie la plupart des éléments caractéristiques du droit, en voici quelques-uns. Dans nos sociétés, la seule « personne » qui pourrait tenir le rôle de « souverain » est le législateur. Or celui-ci doit obéir à des règles fondamentales acceptées (de tous) comme les modalités à suivre pour pouvoir modifier la constitution. Un autre argument contre la doctrine impérativiste, c'est que le droit est composé de différents types de lois. Certaines d'entre elles demandent à l'homme d'agir, ou de

s'abstenir d'agir, d'une certaine façon⁴ et d'autres permettent la création des structures de droits et de devoirs⁵. Ces dernières sont mises à la disposition des gens, ils en bénéficient ou pas selon leur bon vouloir, ce qui n'est pas le cas des ordres. Qui plus est, nous vivons dans une société de droits et de devoirs qui ne peuvent provenir d'ordres, puisqu'un ordre, qu'on lui obéisse ou non, ne confère aucun droit.

Mais l'invalidité de cette doctrine ne permet pas de déduire, comme certains⁶ l'ont fait, que le reste de la théorie utilitarienne s'en trouve invalidée.

Revenons-en maintenant à l'interprétation. Pour Hart, si on veut pouvoir communiquer les uns avec les autres, si on veut pouvoir exprimer qu'un certain comportement est régi par une règle de droit, il faut que les termes (concepts) généraux aient des instances courantes pour lesquelles il ne fait aucun doute que c'est d'elles qu'il s'agit. Il faut par conséquent que les termes généraux aient un « **noyau** » fixe de signification qui est généralement accompagné d'une zone de « **pénombre** » comprenant des cas discutables pour lesquels on ne sait pas s'ils sont, ou pas, visés par le terme.

Illustrons ces concepts par un exemple. Supposons une règle interdisant de pénétrer avec un véhicule dans un parc. On voit bien que cette règle vise les automobiles, les cyclomoteurs, ... (ces instances représentent le « **noyau** » du concept « **véhicule** »); mais qu'en est-il des vélos, des patins à roulettes, des voitures d'enfants, des avions, ... (ces instances-ci représentent la « **pénombre** » du concept « **véhicule** »).

Donc, selon Hart, la rationalité à laquelle on fait appel pour régler des cas qui appartiennent à la « **pénombre** » d'un concept, et plus généralement, d'une règle de droit, ne peut être une fonction logique des prémisses du cas. C'est plutôt l'affaire d'un jugement moral sur "ce qui devrait être". Et c'est là, selon lui, que les utilitariens ont commis une erreur. Pour interpréter les termes généraux d'une loi, il faut tenir compte des valeurs morales et des conséquences sociales de cette interprétation.

Ainsi, pour Hart, le lien entre la loi, la morale et les principes de justice est aussi nécessaire que la connexion entre la loi et ses sanctions. Dans la loi, il y a certaines règles chevauchant sur des principes moraux de base, comme celles interdisant le meurtre, la violence, le vol ..., qui, si elles n'existaient pas, rendraient inutile l'existence de toute autre règle.

⁴ Exemple : l'interdiction de stationner du le code de la route, l'obligation de porter secours aux personnes en danger du code pénal.

⁵ Exemple : la loi des contrats, la loi sur les A.S.B.L.

⁶ Comme Salmond dans [SAL1893].

2.2 Le courant non positiviste

Fuller⁷, disciple de l'école non positiviste, s'oppose à Hart en plusieurs points. Mais il considère que Hart a néanmoins permis de faire comprendre à tout le monde, que ce qui était important, c'était de savoir comment on pourrait, au mieux, définir et servir l'idéal de la fidélité à la loi. Par ailleurs, il est d'accord avec Hart, ce n'est pas en définissant la loi comme étant un ensemble d'ordres qu'on y parviendra. Si on veut définir cette fidélité, il faut forcément faire référence à la morale.

Fuller distingue deux relations différentes, mais interagissantes, entre la loi et la morale. Il y a la **moralité externe** à la loi, c'est l'attitude morale qui permet, donne la compétence, de faire des lois. Et puis, il y a celle qui a été négligée par Hart, la **moralité interne** à la loi. S'il y a un danger que l'affaiblissement de la partition « loi/morale » permette l'entrée de « morale immorale » dans la loi, la solution au problème ne se trouve pas dans le positivisme. Un juge poursuivant des buts jugés mauvais ne se justifierait jamais en faisant référence à des principes supérieurs (eux aussi mauvais), mais sans doute, en se réfugiant derrière une maxime comme « la loi est la loi » et en expliquant sa décision de telle façon qu'elle apparaisse comme exigée par la loi.

Fuller s'oppose fermement à Hart quand celui-ci dit que la seule différence entre les lois anglaises et les lois nazies est que ces dernières poursuivaient des buts considérés comme odieux par les Anglais (et bien d'autres). Selon Fuller, il y a bien plus. On sait que les Nazis n'hésitaient pas à faire usage de lois rétroactives pour remédier à leurs irrégularités juridiques. Ils utilisaient également des « lois secrètes », notamment pour légaliser les massacres dans les camps de concentration. De plus, ils étaient aussi toujours prêts à mettre de côté leurs « lois » quand elles les dérangent. Pour Fuller, une dictature qui a une forme si peu juridique, lui permettant de s'écarter de la moralité de l'ordre et de la moralité de la loi elle-même, cesse d'être un système juridique. La loi n'est pas loi simplement parce qu'elle est appelée loi !

Sur le problème de l'interprétation des concepts généraux, Fuller s'oppose aussi à Hart. Il doute que l'on puisse déterminer un agrégat de choses vers lequel des termes généraux, comme « véhicule », pointent, afin de leur assigner un « noyau » relativement constant quel que soit le contexte dans lequel ils apparaissent. Pour Fuller, interpréter ce n'est pas donner un sens à un mot isolé, mais c'est donner un sens à une phrase, un paragraphe, une page et parfois à tout un texte.

⁷ [FUL1958]

Si la règle qui exclut les véhicules du parc semble facile à appliquer dans certains cas, c'est parce qu'on voit clairement le but général de cette règle. Dans le cas d'une banale automobile, on est capable d'appliquer la règle, sans se poser de questions, parce qu'on voit clairement que le but de cette règle est, par exemple, de préserver la tranquillité du parc, ou d'éviter que les flâneurs ne courent un risque.

Fuller propose un exemple très clair pour appuyer sa critique. On se trouve face à un règlement interdisant de dormir dans une gare. Supposons ensuite que deux hommes soient arrêtés et amenés devant un juge. L'un d'eux parce qu'il s'était endormi en attendant son train, qui devait arriver à 1h du matin, et qui avait 2h de retard. L'autre parce qu'il se baladait dans la gare, avec une couverture et un oreiller, en y cherchant un endroit pour passer la nuit. Lequel de ces cas représente l'instance « standard » du mot dormir ? Et si on laissait de côté cette question, et que l'on relâcherait le premier et punirait le deuxième, violerait-on le devoir de fidélité à la loi ? Fuller dit que non !

Pour Fuller il est impossible d'interpréter un mot isolé dans une loi sans en connaître le but. Pour bien interpréter et comprendre une loi, il faut se mettre à la place du législateur et se demander quels étaient ses intentions, ses buts. C'est à la lumière du « devrait être » que l'on doit décider de ce qui « est ».

Fuller conclut que malgré le fait que le positivisme ait pris un tournant morbide au travers du nazisme, il ne faut pas croire que la peur d'une interprétation intentionnelle de la loi et de ses institutions juridiques est injustifiée. Selon lui, la fidélité à la loi exige que le juge ait un rôle créatif, limité par le respect des « intentions » de la loi (qui peuvent être explicites ou fonction de ses relations avec d'autres lois), limite qu'il ne peut franchir. Le problème est que cette limite possède aussi sa zone de pénombre.

On semble admettre aujourd'hui⁸, que lorsqu'un juge se trouve face à de l'indétermination, il lui appartient d'identifier, dans la réalité sociale, les comportements de référence, les standards de bonne conduite. Mais cette identification n'est pas nécessairement chose facile, le juge est parfois amené, dans une certaine mesure, à déterminer lui même ce qu'est le bon comportement. Parfois, lorsque les concepts utilisés sont vraiment flous, le juge doit alors se référer - explicitement ou implicitement - à des principes « moraux » au sens large; pour cela, il doit sinon forger lui-même des maximes, du moins souvent choisir parmi divers principes concurrents.

⁸ C'est un des points mis en avant dans l'introduction au droit, de Michel COIPEL (Professeur de droit à la faculté des sciences économiques des FUNDP), donnée en première candidature en économie aux FUNDP.

3. Les obstacles à la formalisation du droit

La caractéristique d'ouverture des concepts juridiques dont nous venons de parler pose de sérieux problèmes aux ordinateurs, bien plus qu'aux hommes. Par son jugement, l'homme est capable de rattacher des cas pratiques à des concepts du droit. Mais l'ordinateur, pour en faire « autant », devra disposer d'informations supplémentaires, comme des décisions judiciaires sur des cas antérieurs et/ou d'avis d'experts, ... Mais cette information est souvent incomplète, vague ou d'une telle incohérence que la façon dont les juristes l'utilisent dans leur raisonnement ne peut être modélisée par des méthodes standards. Les raisons de ce problème vont être examinées au travers d'une distinction entre quatre types logiques⁹ d'ouverture des concepts juridiques.

3.1 La texture ouverte

Afin de faire face à la nature imprévisible du monde que le droit régit et au fait que le législateur ne peut tout prévoir, les règles de droit laissent souvent, délibérément ouverte la question de savoir quelles circonstances factuelles classer comme instances d'un concept particulier. Cette caractéristique des concepts juridiques est appelée dans la littérature : **texture ouverte** et elle est vue comme un problème à l'application du raisonnement déductif, comme le fait remarquer Hart¹⁰ :

If a penumbra of uncertainty must surround all legal rules, then their application to specific cases in the penumbral area cannot be a matter of logical deduction, and so deductive reasoning, which for generations has been cherished as the very perfection of human reasoning, cannot serve as a model for what judges, or indeed anyone, should do in bringing particular cases under general rules. In this area men cannot live by deduction alone.

Ce qui veut dire que le problème posé par la texture ouverte n'est pas une question de logique, mais une question de contenu. C'est savoir si une situation particulière peut être classée comme une instance d'un concept juridique - par définition, général - sur base d'une connaissance juridique reconnue.

⁹ Ces quatre types logiques de texture ouverte sont tirés de [PRA1993]

¹⁰ [HAR1958]

3.1.1 La sous-détermination

Nous en avons déjà parlé, mais il convient de rementionner ici ce qui fut appelé par Hart la « pénombre » et le « noyau » de la texture ouverte¹¹ :

...if, as in the most elementary form of law, we are to express our intentions that a certain type of behaviour be regulated by rules, the general words we use (...) must have some standard instance in which no doubt are felt about its application. There must be a core of settled meaning, but there will be, as well, a penumbra of debate cases in which words are neither obviously applicable nor obviously ruled out.

Le fait que dans les cas qui tombent dans la « pénombre » d'un concept, il y ait une incertitude par rapport à sa classification comme instance ou non du concept, représente un type de texture ouverte appelée **sous-détermination**.

L'exemple des véhicules que Hart¹² donne, est un exemple d'indétermination. Comme cela a déjà été dit, il s'agit d'un règlement qui interdit l'utilisation de véhicules dans un parc. Supposons maintenant que Pierre circule dans le parc avec une automobile et Jean avec une planche à roulettes. Pour savoir à qui s'applique le règlement, il faut déterminer si une automobile et une planche à roulettes sont des véhicules. Le bon sens dit que l'automobile est, sans aucun doute possible, un véhicule, mais ce n'est pas aussi évident pour la planche à roulettes. La première fois qu'une planche à roulettes est apparue dans le parc, il n'y avait aucune connaissance juridique disponible à propos de ce problème, la planche à roulettes était un nouveau phénomène que le législateur n'avait pas prévu.

Le problème est donc de savoir si une planche à roulettes doit être classée comme une instance du concept « véhicule » ou non. Comme nous l'avons déjà dit, c'est une question de contenu, cela ne pose aucune difficulté de nature logique. Il reste néanmoins que le concept de véhicule n'a pas une définition complète et que donc, en logique standard, on ne pourra donner de réponse à la question : une planche à roulettes peut-elle rentrer dans un parc?

¹¹ [HAR1958]

¹² [HAR1958]

3.1.2 La surdétermination

L'ouverture délibérée de la règle de droit laisse la porte ouverte à des conflits d'opinions sur des questions de classification de cas concrets dans des concepts. En effet, les règles de droit ne doivent pas seulement être appliquées, elles sont également des objets du discours qui doivent aussi être mis en question. Dès lors, après un certain temps de conflit d'opinion sur un résultat apporté à un problème de texture ouverte, la doctrine se développe dans différentes directions, créant ainsi des sources d'informations contradictoires. On est alors face à ce qu'on appelle : un problème de **surdétermination**.

Dans de tels cas, tant que l'on n'est pas arrivé à un accord, les systèmes informatiques basés sur les connaissances juridiques devront contenir les différentes alternatives. La base de donnée sera alors incohérente, et le système devra pouvoir gérer cette incohérence en constituant différents sous-ensembles cohérents de prémisses qui, permettront la déduction d'une ou plusieurs conclusions.

3.1.3 L'invalidation

On trouve dans le droit des règles communément acceptées des juristes à propos des classes d'instances d'un concept juridique. Mais cette classification est parfois invalidée par l'apparition de nouvelles circonstances. On se trouve face à une situation où l'antécédent d'une condition est vrai, mais où il existe des raisons de faire une exception à la règle.

Reprenons le concept de véhicule. En Allemagne, le législateur a tenté de définir ce concept en introduisant un nouveau concept, celui d'« un objet ordinairement destiné au transport ». Sur cette base, ils ont pu conclure qu'une paire de patins à roulettes n'était pas un véhicule, et que donc, on pouvait circuler en patins dans les parcs¹³. Mais, depuis quelques années, les patins à roulettes sont de plus en plus utilisés comme moyen de transport pouvant atteindre des vitesses supérieures à celles des vélos. On voit donc, qu'il est fort possible que la règle : « les patins à roulettes ne sont pas un moyen de transport normal » ou celle qui dit que « les objets qui ne sont pas normalement destinés au transport ne sont pas des véhicules » peut être sujette à des exceptions.

Cette caractéristique du langage juridique a été appelée par Hart l'« **invalidation des concepts juridiques** ».

¹³ Il est vrai que le terme « ordinairement » ne fait que de déplacer la texture ouverte, le législateur a néanmoins dû penser que cette expression était plus précise.

3.1.4 L'imprécision (le flou)

En droit, il est également couramment fait usage de variables d'**imprécision**, ce sont des termes comme : raisonnable, suffisant, approprié, ... Ce genre de termes se retrouve plus particulièrement dans des textes où l'on prend en compte des facteurs qui peuvent avoir différentes valeurs comme : le revenu, l'âge, le loyer, ... Quand un juge doit se prononcer sur l'appréciation d'un de ces termes, il énonce sa décision de la manière suivante : « d'un côté on a A, B et C, mais d'un autre côté on a X, Y et Z et dans le cas qui nous concerne, ces derniers facteurs l'emportent sur les premiers ... ». Tout ce qu'on peut tirer de telles justifications, c'est que les facteurs ont un certain poids, que leur présence ou absence influence le résultat dans une certaine mesure.

Exemple 1

Un spécialiste de la loi sur les locations d'immeubles doit, par exemple, savoir que la présence d'un ascenseur dans un immeuble augmente les chances que cette immeuble convienne à un futur locataire âgé. Mais il ne peut dire : « s'il y a un ascenseur, et que le futur locataire est âgé et ..., alors l'immeuble lui convient ».

Pour modéliser ces connaissances, en tenant compte du poids des différents facteurs, on peut utiliser des modèles basés sur des techniques numériques, comme les logiques floues. On peut par de telles techniques arriver à dire, par exemple, qu'une personne de 1m60 peut être considérée comme grande avec un degré de vérité de 0.1; qu'une personne de 1m80 peut être considérée comme grande avec un degré de vérité de 0.5 et qu'une personne de 1m95 peut être considérée comme grande avec un degré de vérité de 0.9. De telles techniques représentent une façon de rompre le silence du système pour des cas où on a de nouvelles valeurs pour certains facteurs.

Tous ces problèmes liés à la texture ouverte du langage juridique ne sont pas les seules difficultés que doivent surmonter les systèmes informatiques qui veulent modéliser le raisonnement juridique. D'autres difficultés, liées à l'existence d'exceptions aux règles de droit, sont également de nature à rejeter la logique classique comme formalisme approprié au droit.

3.2 Les exceptions aux règles

Un des problèmes posés par l'existence d'exceptions aux règles de droit est que bien souvent, la formulation de la règle et de ses exceptions sont faites à des endroits différents. En plus de cette séparation, on trouve souvent des conflits entre différentes

lois (et exceptions); ces conflits doivent alors se résoudre, si possible, à l'aide de méta-principes basé sur la hiérarchie¹⁴ des normes, sur leur date de promulgation¹⁵ et/ou sur la spécificité¹⁶ des lois.

De par son caractère fondamentalement dynamique lié à son « devoir » de rester branchée sur la vie sociale, la règle juridique est appelée à se modifier, à évoluer et ce de manière imprévisible. Donc, si l'on veut favoriser la validation et la maintenance du système juridique informatique, il faut que la structure du formalisme soit aussi proche que possible de la source originale. C'est ce qu'on appelle une formalisation isomorphique. Pour cela, il faut maintenir la séparation d'origine entre la règle et ses exceptions. Ce qui revient à dire que le système ne doit pas raisonner de façon classique (logique classique), mais de façon non-monotone¹⁷.

Le fait que les règles de droit sont souvent mises en question - notamment à cause de l'interprétation de leur texture ouverte par des juristes d'intérêts contradictoires et/ou d'opinions morales différentes - fait qu'elles sont souvent sujettes à des exceptions qui ne sont pas explicitement décrites dans la législation.

Une règle peut ne pas être appliquée pour préserver sa fidélité (son esprit). C'est le cas quand on permet à des anciens combattants d'exposer une Jeep de la seconde guerre mondiale dans un parc, malgré la règle interdisant de pénétrer dans le parc avec un véhicule.

Une règle peut également être suspendue parce qu'elle irait à l'encontre d'un principe général de droit. Si on reprend la même règle sur les véhicules et le parc, on permettra à une ambulance de rentrer dans ce parc pour aller chercher un blessé ou un malade en vertu d'une loi du code pénal (belge) selon laquelle il faut porter assistance à toute personne en danger.

Cette invalidité des règles de droit cause les mêmes problèmes que la séparation structurelle des règles et de leurs exceptions dans la législation elle-même. Ici, le problème est qu'il faut raisonner avec des principes et des buts, c'est-à-dire de la méta-connaissance.

¹⁴ Lex Superior: « Lex Superio Derogat Legi Inferiori ».

¹⁵ Lex Posterior: « Lex Posterior Derogat Legi Priori »

¹⁶ Lex Specialis: « Lex Speciali Derogat Legi Generali ».

¹⁷ Sans entrer dans les détails (pour le moment) raisonner de façon non-monotone c'est permettre de changer de conclusion au vu de nouvelles informations.

On a donc ici une nouvelle caractéristique du raisonnement juridique, c'est le fait de devoir raisonner avec de la méta-connaissance, ce qui requiert, pour un système informatique juridique, la possibilité d'exprimer la connaissance à différents niveaux.

Dès lors, idéalement, le caractère invalidable de la règle de droit requiert non seulement des méthodes de raisonnement non-monotone, mais aussi des méthodes de raisonnement au niveau de méta-connaissances.

Cette complication du modèle - apportée par l'introduction de méta-connaissances - peut être « contournée », comme le propose Prakken¹⁸, en considérant que les principes et buts de la loi sont implicitement reflétés par la « jurisprudence » au sens large. Si des principes dictent une certaine interprétation d'un concept ou l'application d'une exception à une règle, alors cela se reflétera dans des décisions concrètes. Si le système peut avoir connaissance de telles décisions, il n'aura qu'à vérifier si le cas sur lequel il travaille ressemble à un des cas de la base et si c'est le cas, à donner la priorité à la décision du cas retrouvé plutôt qu'à une lecture littérale de la règle.

Dans la pratique, ce n'est pas chose facile que de représenter - utilement et efficacement - des cas de jurisprudence dans une base de connaissance. Il faut que l'information soit adéquatement généralisée pour permettre une application exacte et pertinente.

On voit de toute façon que ce n'est pas un problème formel, mais un problème de représentation des connaissances à un niveau d'abstraction approprié. Logiquement, comme les cas ne sont pas de la méta-connaissance, ils peuvent être représentés au même niveau de langage que les règles auxquelles ils font exception.

En résumé, si on veut formaliser correctement le raisonnement juridique, il faudra un formalisme qui permette de traiter les problèmes liés à la texture ouverte, c'est à dire qu'il faut pouvoir :

- ☞ donner des solutions, ou des pistes de solutions, à des problèmes pour lesquels on n'a pas assez d'informations;
- ☞ traiter une base de donnée qui contienne des informations incohérentes;
- ☞ traiter l'invalidation possible des règles de droit;
- ☞ gérer les variables d'imprécision.

¹⁸ Qui adopte la vue de Gardner dans [GAR1987]

Mais en plus de cela, il faut que ce formalisme soit isomorphe, afin de conserver la structure des sources du droit, et qu'il permette la représentation de méta-connaissances ou qu'il gère une base de cas.

Aperçu de quelques projets en IA et droit

Nous allons maintenant présenter quelques programmes existants utilisant quelques-unes des caractéristiques non-standard développées précédemment. (La plupart des systèmes décrits ici ont un comportement non-monotone dû au fait que l'on compare des arguments alternatifs générés par une base de connaissance incohérente. On peut aussi ajouter que ces systèmes manquent pour la plupart d'une base formelle.)

4. Le droit modélisé dans un programme logique

4.1 *The British Nationality Act*

Plus qu'une approche particulière, l'implémentation du *British Nationality Act*¹⁹ est un exemple d'une approche générale de la formalisation du droit en un système informatique. On y formalise la loi en un programme logique composé d'un ensemble de formules écrites dans un langage logique pour lequel il existe un prouveur de théorèmes.

La technique utilisée par M.J. Sergot et son équipe consiste à représenter d'abord la loi sous forme d'un arbre ET/OU ayant à la racine la conclusion à laquelle le système est dédié (est-ce un citoyen britannique ou non?) et comme branches les conditions de cette conclusion. Dans cet arbre, chaque niveau (sauf les deux premiers) est un raffinement en sous-conditions des conditions du niveau supérieur. Le problème que pose cette démarche est de savoir où s'arrêter dans le raffinement, de savoir où placer les entrées que l'utilisateur du système devra fournir. Une fois cet arbre construit, il est alors

¹⁹[SER1986]

aisément transformé en clauses de Horn étendues²⁰, qui sont elles-mêmes facilement traduisibles en Prolog.

Exemple 2

L'arbre représenté à la Figure 1 donne les différentes conditions (conjonctions et disjonctions) à remplir pour pouvoir obtenir la nationalité anglaise. On voit, entre autres, qu'on peut l'acquérir à la naissance à condition de remplir les conditions du sous-arbre de gauche ou l'ensemble de conditions du sous-arbre de droite.

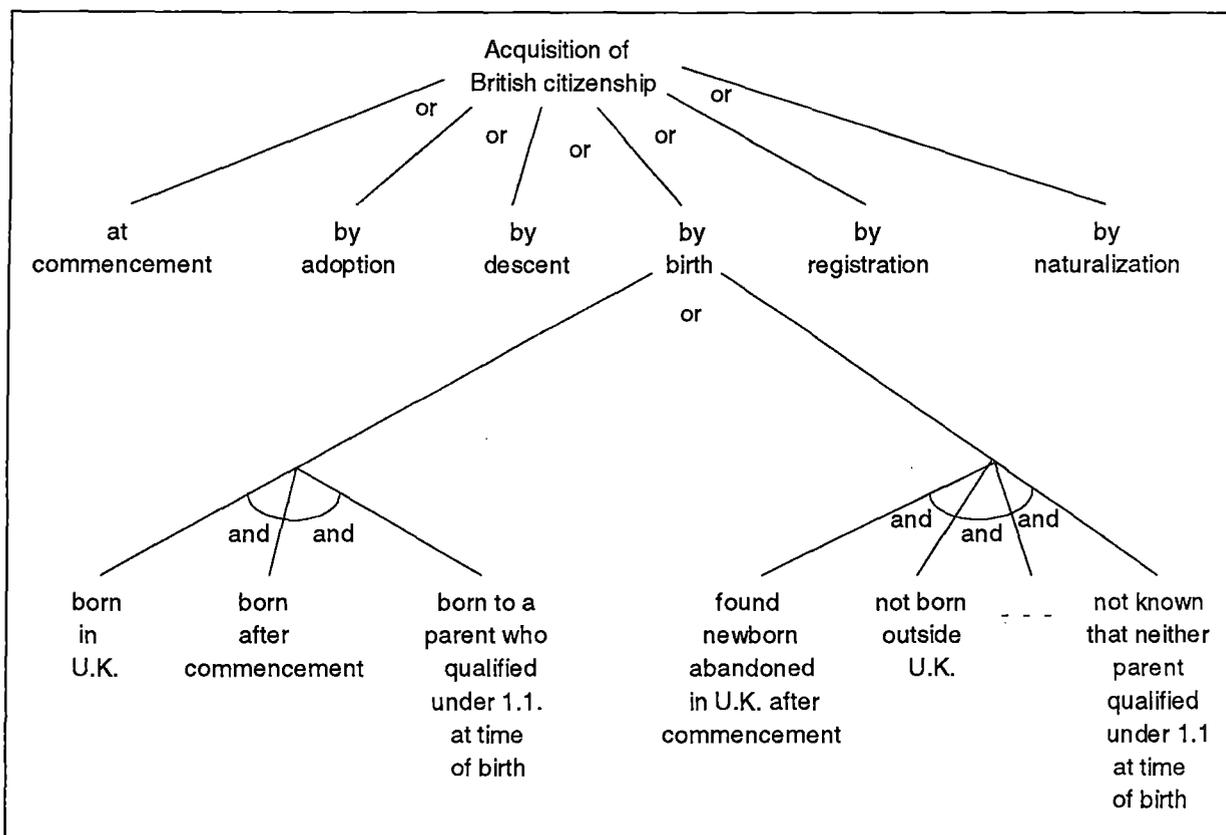


Figure 2

Au travers de type de formalisation, à cause du domaine d'application, les auteurs ont dû abondamment faire usage de la « négation par échec »²¹ pour formaliser la négation,

²⁰Ce sont des clauses ayant la forme suivante: A si B₁ et B₂ et... et B_n :

où A est un atome, c'est-à-dire un symbole qui représente une phrase déclarative qui peut être vraie ou fausse, mais pas les deux simultanément;

et où B_i est un littéral, c'est-à-dire un atome ou sa négation),

c'est-à-dire des clauses de Horn mais où les conditions ne sont pas seulement positives.

ce qui est une forme de raisonnement non-monotone. Ce type de négation est utilisé pour formaliser des phrases dans lesquelles on retrouve des expressions comme : « à moins que l'on ne prouve le contraire » ou « sous réserve de la section... ».

Exemple 3

Dans le « British Nationality Act », les enfants abandonnés sont considérés comme étant citoyens anglais (selon la section 1-(2)) si les deux parents sont inconnus. Mais supposons que plus tard on apprenne qu'aucun des parents n'est de nationalité anglaise ni né en Angleterre. Sous de telles conditions, la conclusion précédente tombe (bien que cette disposition n'ait pas été envisagée explicitement par le législateur).

La méthodologie utilisée par Sergot et son équipe procède par essais/erreurs: en effet, lors de la formalisation, ils sont souvent amenés, face à de nouvelles dispositions de la loi, à revoir le formalisme utilisé jusque là.

Exemple 4

Tout d'abord, lorsqu'un certain nombre de conditions étaient remplies, ils concluaient que la personne devenait anglaise. Mais par la suite, ils se sont rendu compte que cela ne suffisait pas. Certaines dispositions de la loi envisageaient les cas où la nationalité anglaise avait été obtenue par les dispositions du point 1.-(2), ou ... Il fallait donc modifier le formalisme pour ajouter la façon dont on obtenait la nationalité anglaise.

Notons qu'un changement de formalisation peut devenir très lourd quand on a déjà introduit un certain nombre de règles dans le système.

L'avantage du SEJ obtenu par Sergot, mais c'est également le cas pour la plupart des SEJs, est qu'il permet non seulement « d'appliquer » la loi, mais aussi de la développer, car il peut mettre en évidence - avant ou après mise en application de la loi - des ambiguïtés et des imprécisions indésirables.

La démarche entreprise par Sergot et son équipe a bien fonctionné dans ce cas précis parce qu'elle s'est appliquée à un domaine, basé sur des règles (*rule-based*), qui se prêtait bien à cette formalisation. Mais son projet n'aurait probablement pas abouti dans

²¹ Une conclusion est tirée par défaut en l'absence d'information prouvant le contraire, mais si cette information devient disponible, la conclusion tirée précédemment est remplacée par la nouvelle conclusion.

un autre domaine du droit, à cause des problèmes liés à la texture ouverte et à la gestion des exceptions dont nous avons déjà parlé.

4.2 The Pension Rules of the Indian Central Civil Service

Ici aussi, Sergot, Kamble et Bajaj ont implémenté *the Pension Rules of the Indian Central Civil Service*²² en programme logique et de par la nature de cette loi, cela c'est ramené à traiter le calcul du temps de service pris en compte pour l'évaluation du montant des pensions des employés d'une entreprise.

Ce qui est intéressant dans ce programme, c'est l'architecture utilisée. Sergot, Kamble et Bajaj y ont séparé les données sur le monde réel, des règles de traitement de ces données. On peut dire qu'on retrouve ici, ce que nous avons appelé une formalisation isomorphique.

Un premier programme (en prolog) est chargé de récolter les données se trouvant sur le livre de service²³, d'en vérifier la cohérence et ensuite, de les transformer en base de données Prolog. Le second - qui est un ensemble de règles définissant les périodes de temps à prendre en compte pour le calcul du montant des pensions - modélise le guide administratif des fonctionnaires. Il utilise les données du livre de service, plus éventuellement des compléments d'information qu'il demande à l'utilisateur du système, afin de fournir le montant de la pension auquel à droit un employé (ou ouvrier) particulier.

5. Différentes techniques pour traiter la texture ouverte

5.1 Pondérations

Pour Bench-Capon et Sergot²⁴, la texture ouverte permet aux tribunaux d'appliquer la loi à des cas concrets et de construire ainsi des précédents non exhaustifs. Pour ces auteurs, cette texture ouverte est divisée en trois zones, une zone supérieure à l'intérieur de laquelle on applique clairement la loi; une zone inférieure dans laquelle on n'applique pas la loi, et une zone intermédiaire ambiguë.

²²[SER1991]

²³Le livre de service est le carnet dans lequel on tient à jour des informations sur l'état de service, sur la carrière des employés (ou ouvriers). On y inscrit son salaire, ses promotions éventuelles, ses périodes d'absences, de congés ...

²⁴[BEN1988]

Ces deux auteurs proposent différentes façons de traiter la texture ouverte selon le type de SEJ que l'on veut élaborer.

Si on envisage qu'un SEJ prenne des décisions, une des solutions pourrait être de remplacer les concepts vagues par une approximation au travers de concepts non vagues qui se substitueraient bien aux concepts vagues dans la majorité des cas. Ces approximations pourraient être arrêtées en interviewant des experts et en consultant la doctrine et les précédents. Il serait également possible d'associer un facteur de vérité à chaque fait et à chaque règle, et ainsi, à l'aide de la théorie des probabilités ou des logiques floues, on arriverait à donner des solutions pondérées par un degré de vérité. Mais le problème de cette technique, c'est qu'en fait, on déplace le problème d'interprétation, il faudrait maintenant interpréter des affirmations pondérées comme : "il est vrai, à un degré de vérité de 0.537, que M. X était de mauvaise foi".

Si on envisage un système d'aide à la décision, on pourrait avoir recours à un système à base de cas (où les cas de la base pourraient être stéréotypés et/ou réels), comprenant également la décision rendue pour ces cas. Pour l'examen d'un cas donné, le système chercherait des cas similaires et les proposerait à l'utilisateur, qui n'aurait plus qu'à décider lequel suivre. Mais cette technique poserait des problèmes pour le stockage des exemples (critères de stockage) et pour la définition d'un algorithme de recherche de cas similaires²⁵.

Comme disent Bench-Capon et Sergot²⁶ :

... We may teach the use of a concept by showing examples where it applies and where it does not apply. The concept has been successfully taught, however, only when the trainee is able to go beyond the training examples and apply the concept to new examples. This would suggest that he has formulated a rule that generalises the examples he was given.

Si on arrivait à formuler ces règles générales, les cas n'auraient plus d'utilité que pour une justification ou pour une explication. Mais, dans l'état actuel des choses, des systèmes capables d'induire des règles à partir de cas ne sont pas prêts d'exister dans le domaine juridique. En attendant, le concepteur de la base de cas pourrait ajouter les règles nécessaires à chaque nouveau cas. Mais le problème n'est pas tant de transformer ces règles en un code exécutable par la machine, mais bien de découvrir et de

²⁵Des systèmes à base de cas seront étudiés au point 6.1., et nous verrons comment ces difficultés ont été surmontées.

²⁶[BEN1988] page 50.

comprendre, en terme de règles, la portée des cas particuliers. Une règle générale ne peut s'obtenir qu'à partir du moment où il y a une certaine tradition dans la façon de juger certains cas.

Mais dans la zone intermédiaire d'un concept flou, la détermination d'une règle pourrait être sujette à controverse, alors plutôt que d'avoir à choisir un point de vue, on pourrait retenir les deux dans le système. Dès lors, à une question particulière, le système donnerait plusieurs solutions avec des justifications différentes. L'utilisateur n'aurait alors plus qu'à choisir la justification qui lui paraît la plus convaincante, ou s'il a une solution à atteindre, savoir sur quelle interprétation il devra la défendre. Le rôle du système dans ce dernier cas, serait donc de faire faire à l'utilisateur un choix éclairé.

Mais ici, disent Berman et Hafner²⁷, Bench-Capon et Sergot n'ont pas pris en considération le fait que certains concepts peuvent être revus au cours du temps, et qu'en droit, il est couramment fait usage d'analogies ou "fictions juridiques"²⁸ dans lesquelles une violation des catégories normales (définies par les précédents sur lesquels on s'appuie) est explicitement adoptée. La nature évolutive de la loi fait qu'elle crée des précédents conflictuels ou des versions conflictuelles d'une règle qui coexistent dans la base de connaissance.

5.2 Résoudre uniquement les cas simples

Gardner²⁹ s'attaque, quant à elle, à un autre problème, celui de faire distinguer à son système les problèmes simples des questions juridiques difficiles, et ensuite de lui faire résoudre uniquement les problèmes simples. Pour elle, on se trouve en face d'un problème difficile quand il existe des règles concurrentes dans le système, ou des prédicats non résolus, ou encore des cas pratiques antagonistes.

La technique utilisée est d'abord de permettre à tout prédicat indéfini d'être potentiellement la cause d'un cas complexe, et après, de fournir des moyens de conclure que certains d'entre eux ne le sont pas. Ceci évite de faire des erreurs fatales en considérant simple un cas complexe, mais cela peut parfois entraîner une perte de ressources quand on en arrive à considérer complexe un cas simple.

Ce système contient trois types de connaissances juridiques: premièrement des règles juridiques générales dérivées de la jurisprudence; deuxièmement, des règles de sens

²⁷[BER1988]

²⁸Exemple de fiction juridique: la personnalité morale permet de considérer des groupements de personnes et/ou de biens comme des personnes et ces entités sont traitées par référence au traitement réservé aux personnes physiques.

²⁹[RIS1988]

commun à propos du monde auquel le système est dédié; et troisièmement, des cas de jurisprudence. A partir de ces connaissances, trois heuristiques sont proposées pour tenter de ramener des cas considérés comme complexes dans des cas simples. Par la première, on essaie de donner une réponse en faisant appel aux règles de sens commun. Si on n'y arrive pas, l'heuristique suivante essaie de trouver des cas similaires. Et la dernière cherche des cas avec des solutions contradictoires aux solutions potentielles proposées par une des deux heuristiques précédentes. S'il n'en existe pas, on se retrouve devant un cas simple dont la solution est celle donnée par le système.

Le programme de Gardner est bien de type non-monotone puisque avant de considérer un cas comme difficile, il compare des alternatives conflictuelles pour voir laquelle - s'il y en a une - est préférée aux autres.

Le problème de cette démarche, c'est qu'en droit, il est presque toujours possible pour un cas donné d'en trouver un autre qui lui soit opposable. La solution serait de dire que tous les cas ne sont pas égaux (on pourrait établir des règles de priorité) ou que tous les arguments d'un cas n'ont pas la même robustesse.

Malgré ses inconvénients, l'approche par exemples paraît intéressante; c'est elle qui semble le mieux prendre en charge la texture ouverte des concepts juridiques et elle reflète la façon dont on aimerait que cela se passe idéalement dans ce type de situation.

6. Modélisation de l'argumentation juridique

L'argumentation est un élément essentiel dans la plus part des procédures juridiques, il faut convaincre un juge, un jury, des collègues, ... que la position que l'on adopte est la bonne, ou meilleure qu'une autre. Ici aussi, l'I.A. essaie de s'emparer du type de raisonnement qu'adoptent les juristes afin de leur fournir des outils d'aide à l'argumentation. Nous parlerons d'un système basé sur l'argumentation basée sur des cas similaires et d'un système plus dynamique basé sur les faits du cas et sur les règles que l'on peut lui appliquer.

6.1 Modélisation à base de cas et de règles

Créer de « bons » arguments juridiques est difficile pour les hommes, comme pour les machines. La difficulté réside en partie - du moins dans le contexte du raisonnement à base de cas - dans le contrôle des tâches, c'est-à-dire, savoir que faire et quand le faire. Il faut, en particulier, choisir entre une stratégie guidée par la spécification

(*specification-driven*³⁰) ou guidée par les cas (*case-driven*³¹). Trouver un cas disponible qui soit similaire au cas présent et qui aille dans le sens désiré, c'est trouver un bon argument. On est donc contraint, dans tout ce processus de détermination, par les cas jurisprudentiels que l'on peut mettre à sa disposition, que l'on peut découvrir.

CABARET³², le système développé par Skalac et Rissland, est une combinaison de deux systèmes :

- ∞ un S.E. à base de règles (*rule-based*) traditionnel;
- ∞ et un S.E. à base de cas (*case-based*) construit sur le modèle d'HYPO. Ce système à base de cas peut prendre le relais lorsque les règles n'ont pas abouti³³, ou pour s'opposer aux résultats obtenus par le système à base de règles³⁴;
- + pour chacun de ces systèmes, il y a un **processus d'observation** qui prend des notes sur le comportement du système. Ensuite, grâce à des heuristiques de contrôle, un **module de contrôle** (le *control module*) détermine quelle est la prochaine tâche à mettre dans l'**agenda** du système au complet. Cela peut être une tâche pour le système à base des cas ou celui à base de règles.

6.1.1 Le système HYPO

Le système HYPO de Rissland et Ashley³⁵ analyse les cas de sa base de cas et le cas qui lui est soumis à l'aide de dimensions. Les dimensions sont une formulation abstraite d'aspects d'un cas, elles sont identifiées par un ingénieur des connaissances comme pertinentes pour ou contre une certaine affirmation juridique. La tâche d'HYPO n'est pas de fournir une solution, mais de donner aux deux parties d'une

³⁰ Aussi appelée approche descendante (*top-down*) ou indépendante des cas (*case-base-independent*). On spécifie quel serait le cas idéal que l'on cherche pour la situation envisagée, et ce indépendamment des cas existants.

³¹ Aussi appelé approche montante (*bottom-up*) ou dépendante des cas (*case-base-dependent*). On crée et définit des arguments en fonction des cas qui existent et qui sont disponibles; on fait un examen exhaustif de ces cas.

³² [SKA1992]

³³ Exemple: le cas en question ne satisfait pas toutes les condition d'une règle, on peut alors chercher et trouver un cas qui a les mêmes caractéristiques et auquel la règle s'est appliquée.

³⁴ Exemple: Pour dire que la règle utilisée par la partie adverse pour aboutir à une conclusion n'est pas applicable, on peut chercher un cas dans lequel toutes les conditions de la règle étaient satisfaites, mais auquel la règle ne s'est pas appliquée.

affaire la possibilité d'appuyer « raisonnablement » leur point de vue sur des cas de jurisprudence. Il permet également d'évaluer la valeur relative de ces cas.

HYPO propose un cas, quand il est similaire aux faits courants, et qu'il contient l'enjeu du débat. La notion de similitude est implémentée comme suit : tous les cas dans la base qui ont au moins une dimension en commun avec le cas considéré sont ordonnés sur base de l'ensemble d'inclusion de facteurs qu'ils partagent avec le cas en question. Si l'ensemble des dimensions qu'un cas partage avec le cas courant contient celui d'un autre cas, le premier est plus similaire que le dernier. Mais HYPO ne dit pas si un cas est suffisamment similaire pour entraîner la même conclusion.

CABARET utilise le raisonnement non-monotone en ce qu'il permet au système à base des cas de tirer des conclusions différentes de celle du système à base de règles, alors que les conditions de cette règle sont remplies. On modélise ici, le caractère d'invalidation des règles. Ce système permet également de tirer des conclusions différentes, à partir des mêmes règles et de la même base de cas, selon point de vue où l'on se place.

Exemple 5

Un avocat, ayant un client qui satisfait à toute les préconditions d'une règle en sa défaveur, pourra chercher des cas jurisprudentiels qui ressemblent à celui de son client et dans lesquels toutes les préconditions de la même règle étaient remplies, mais où la règle n'a pas été appliquée. De cette façon, il disposera de bons arguments pour défendre son client.

Dans CABARET, l'approche montante et l'approche descendante sont combinées pour générer des arguments juridiques, comme c'est d'ailleurs le cas en droit³⁶. CABARET utilise une structure à étages composée, au plus haut niveau, de stratégies d'arguments, ensuite, de mouvements pour terminer avec des primitives. Cette structure permet de déterminer une spécification de classes de cas qui peuvent, potentiellement, être utiles ainsi que différentes façons de les utiliser dans l'argumentation.

³⁵ [RIS1989]

³⁶ Avec la doctrine du *Stare Decisis* les avocats doivent au moins être guidés par les cas puisqu'ils sont liés aux précédents. D'autre part, celui qui plaide en justice est guidé par la nécessité d'atteindre certains buts (il doit savoir où il veut aller!). Ce sont les besoins d'arguments, pour atteindre ce but, qui définissent les desiderata du cas parfait.

Après avoir trouvé les cas qui rencontrent les caractéristiques désirées, le module de raisonnement à base de cas trie ces cas, les compare et choisit ceux qui seront utilisés pour l'argumentation.

CABARET n'est pas un système destiné à fournir des solutions, son but est plutôt de présenter, selon les points de vue de l'utilisateur, plusieurs façons d'argumenter en faveur d'une conclusion, de mettre en doute les conclusions de la partie adverse, de suggérer, en faisant des analogies, les informations qui seraient nécessaires, ...

Skalac et Rissland présentent deux cadres conceptuels pour réduire les difficultés de l'argumentation. Le premier provient de l'observation que les arguments juridiques sont souvent présentés sous des formes stéréotypées qui peuvent être classifiées (il y a les arguments doublement négatifs, les arguments directs, les arguments gratuits, les arguments bateau, les arguments pari, ...³⁷). L'autre cadre conceptuel provient du modèle, utilisé dans CABARET, indiquant comment utiliser les cas pour interpréter les termes d'une règle de droit. Ce modèle organise les éléments de l'argument, en contexte statutaire, en trois parties (trois étages) :

- adopter une stratégie d'arguments : il y a quatre stratégies possibles qui dépendent des caractéristiques du cas (est-ce qu'il tombe ou non sous la portée de la loi?) et du point de vue que l'on veut défendre (*pro* ou *con* la loi?³⁸). Le "choix" posé dans ces deux dimensions détermine la stratégie comme on peut le voir dans le tableau qui suit.

³⁷ Pour plus de détails, voir [SKA1992] page 15→27.

³⁸ Respectivement dans le sens de « en faveur » ou « en défaveur » de la loi.

	point de vue pro la loi	point de vue con la loi
Cond. de la loi rencontrées	Confirmer que la loi s'applique bien au cas en question, comme le prétend sa stricte interprétation.	Prétendre que la loi, dans ce cas précis, ne s'applique pas, bien que sa stricte interprétation dise le contraire.
Cond. de la loi non rencontrées	Prétendre que la loi s'applique au cas, alors que sa stricte interprétation dit le contraire.	Dire que comme la stricte interprétation de la loi le dit, la loi ne s'applique pas à ce cas.

Tableau 1

- faire des mouvements d'arguments : les mouvements possibles dépendent des dispositions des précédents (sont-elles celles désirées ou non?), de l'applicabilité de la loi dans ces cas (est-ce que la loi s'est appliquée ou non?) et de la stratégie que l'on a précédemment « choisie ». Voici la table des mouvements possibles si on se place dans une stratégie de qui prend un point de vue *con* de la loi :

	dispositions désirées	dispositions non désirées
La loi s'est appliquée	Utiliser l'analogie pour affirmer que la loi ne s'applique pas au cas présent.	Faire de la différenciation pour affirmer que la loi s'applique au cas en question.
La loi ne s'est pas appliquée	Utiliser l'analogie pour affirmer que la loi s'applique au cas présent.	Faire de la différenciation pour affirmer que la loi ne s'applique pas au cas en question.

Tableau 2

- et implémenter des techniques d'arguments primitifs : c'est donc ce qui permet d'accomplir les mouvements d'arguments.

La connexion entre les cadres conceptuels est que ce modèle contient le vocabulaire et les méthodes nécessaires pour implémenter des arguments très similaires aux arguments juridiques. Ces formes implémentées pourraient être appliquées comme objet de contrôle pour combiner les facettes montantes et descendantes de la formation d'arguments.

6.2 Un modèle computationnel d'argumentation dialectique

Farley et Freeman³⁹ présentent un modèle computationnel d'argumentation dialectique pouvant servir de base à des prises de décisions dans des domaines, comme le droit, où la connaissance est, comme nous l'avons déjà dit, incomplète, incertaine, et incohérente. Dans ces domaines, l'argumentation dialectique, avec ses arguments *pro* et *con* l'affirmation discutée, permet de donner une nouvelle définition de la preuve.

Dans ce modèle, un **argument** est une donnée qui, grâce à une « **autorisation** » (c'est une règle d'inférence spécifique au domaine d'argumentation), **soutient** une **affirmation** (conclusion), mais cette « autorisation » doit elle même être **justifiée** pour être valide. Cette représentation est une extension de celle de Toulmin dans « *The Uses of Argument* »⁴⁰ à laquelle Farley et Freeman ajoutent que les justifications et les autorisations sont elles aussi des affirmations, et qu'elles sont donc aussi sujettes à des exceptions. Il s'ensuit que toutes les affirmations doivent être soutenues. Il existe dans ce schéma de représentation, deux types de soutien : l'atomique qui est de l'information provenant de l'extérieur du domaine d'argumentation; la TAU (« Toulmin Argument Unit ») où la conclusion est soutenue par une donnée et une autorisation. Cela se présente comme indiqué à la Figure 3.

³⁹ [FAR199?]

⁴⁰ [TOU1958]

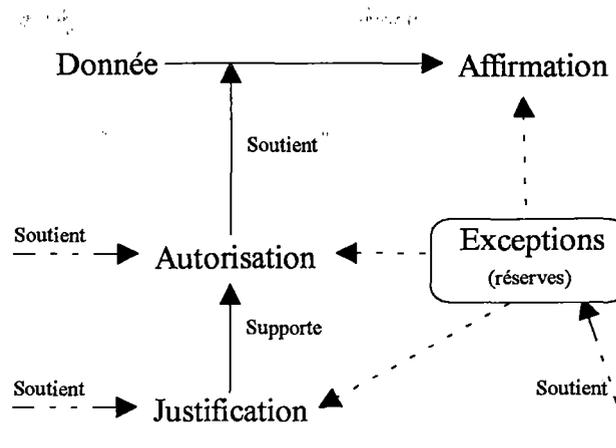


Figure 3

L'autorisation possède deux champs propositionnels, l'antécédent et la conséquence, ainsi que deux types ou attributs :

- ☞ type 1 : indique le type de lien entre l'antécédent et la conséquence : c'est une relation d'explication comme un lien causal [ex] (*ex : le feu provoque de la fumée*); ou c'est un lien de corrélation comme un indice, un signe [si] (*ex : la plupart des crashes d'avions ont lieu le mardi*)
- ☞ type 2 : indique la robustesse de la déduction d'une conséquence à partir d'un antécédent donné. Dans l'ordre décroissant de la force du lien (c'est-à-dire de la « vérité » de ce lien), ce champ peut prendre les valeurs suivantes : suffisant [s] (*ex : l'homme est mortel*), par défaut [df] (*ex : les oiseaux volent*) ou indicateur [ev] (*ex : les habitants des Bermudes sont souvent des Anglais*).⁴¹

Dans ce modèle, on modélise les deux parties (opposées) d'un conflit juridique. Chaque partie cherche, au travers d'un processus guidé par le but à atteindre, à soutenir une affirmation. Le processus de recherche s'arrête lorsqu'un soutien est trouvé dans la B.D., ou qu'il n'en existe pas. Dans ce processus de preuve dialectique, affirmations et « autorisations » sont sans cesse mises en doute par apport de réserves, elles sont même parfois réfutées. Et c'est la notion de charge de la preuve qui arbitre et qui détermine les résultats de cette dialectique.

Ce modèle d'argumentation fait coexister les raisonnements déductifs et plausibles. Donc, pour éviter les combinaisons inappropriées et permettre uniquement la génération

⁴¹ cf. [FAR199?] p4 pour l'application de l'autorisation dans différents sens.

de conclusions acceptables, il faut consulter la table des combinaisons permises⁴² pour combiner de façon correcte le type de raisonnement et les attributs de l'« autorisation ».

Exemple 6

La combinaison « modus ponens/abduction⁴³ » n'est pas permise entre deux autorisations explicatives, sauf si elles sont toutes deux de type indicateur car dans ce cas, la force de la conclusion que l'on tirera sera tellement faible qu'on peut se le permettre.

Voici un exemple de combinaison « modus ponens/abduction » inappropriée conduisant à une conclusion inacceptable : on sait que la pluie mouille l'herbe :

$r_1 : \text{pluie} \rightarrow \text{herbe_mouillée}$

et que arroser mouille aussi l'herbe :

$r_2 : \text{arroser} \rightarrow \text{herbe_mouillée}$

Sachant qu'il a plu par modus ponens on peut en conclure que l'herbe est mouillée

$\text{pluie} \rightarrow_{\text{modus ponens}} \text{herbe_mouillée}$

et de là, par abduction on peut conclure que l'herbe a été arrosée

$\text{herbe_mouillée} \rightarrow_{\text{abduction}} \text{arroser}$

En raison de l'utilisation d'arguments basés sur des connaissances incertaines, et sur du raisonnement plausible, il faut donner différents qualificatifs aux soutiens apportés à une affirmation (par ordre décroissant de soutien, on a : fort [s], ordinaire-habituel [!], crédible [+] et inconnu [?]). Il s'ensuit quelques règles logiques⁴⁴ qui permettent de déterminer le qualificatif des différents éléments du processus d'argumentation.

⁴² cf. [FAR199?] table 2

⁴³ Si on a la règle « $p \rightarrow q$ » et qu'on a « q », par abduction on peut conclure qu'on a « p ».

⁴⁴ Le qualificatif d'une affirmation est celui associé à l'argument qui apporte le plus fort soutien. Pour les données et les « autorisations », ils sont donnés explicitement au moment où ils sont rentrés dans le système, et reste inchangés jusqu'à ce qu'on trouve un meilleur soutien. Toute affirmation résultant d'un support TAU possède au moins la qualification associée à l'« autorisation ».

Dans ce processus dialectique d'argumentation, la réfutation peut porter sur une affirmation ou sur un argument qui la soutient. Les mouvements qui peuvent mener à une réfutation sont de deux types⁴⁵ :

- ↻ court-circuitage (*undercutting*) : c'est trouver de la faiblesse dans ce qui soutient l'affirmation (données ou liens). Si ce mouvement est mené à bien, on a la suppression de l'argument en question, ou le soutien une affirmation rivale (en fait on laisse l'affirmation originale inchangée, mais on montre son imperfection⁴⁶);
- ↻ objection (*rebutting*) : c'est s'attaquer à une affirmation, sans regarder ses soutiens, et la réfuter. Le succès de ce mouvement apporte un soutien à une affirmation rivale.

Pour mener au mieux cette dialectique, le système utilise des heuristiques qui lui permettent de choisir le mouvement à appliquer pour contrer au mieux la partie adverse, tout en essayant également d'anticiper les mouvements qu'elle pourrait faire. Deux groupes d'heuristiques sont utilisés, un pour les mouvements (on préfère un mouvement qui contre un TAU aux autres qui le rendent seulement controversé, les mouvements spécifiques sont préférés aux généraux, les mouvements qui attaquent un TAU plus près de la racine sont préférés, les mouvements court-circuitage sont préférés à ceux d'objection), l'autre pour les « autorisations » (les raisonnements forts sont préférés, les « autorisations » fortes sont préférées, les « autorisations » où des données soutiennent des parties du raisonnement qui ont déjà un soutien cohérent sont préférées).

L'argumentation dialectique commence avec la partie-1 qui essaie de trouver un soutien à son affirmation, si elle n'en trouve pas, le processus se termine, et la partie-1 abandonne son affirmation. Si elle trouve un soutien, le contrôle est passé à la partie-2 qui essaie de réfuter les affirmations (au sens large) de la partie-1. Les deux parties tentent, chacune à leur tour, des mouvements d'arguments jusqu'à en réussir un, ou qu'il n'y ait plus d'argument de disponible. Quand un mouvement est réussi, le qualificatif de l'affirmation en entrée devient une vérification pour la partie courante, comme aux échecs, quand on analyse ce qu'un coup va rapporter. La partie courante donne alors le contrôle à l'adversaire pour qu'il tente de réfuter la réfutation, etc. Le processus s'arrête quand une partie ne trouve plus de contre-argument, et c'est l'autre qui l'emporte.

⁴⁵ cf. [FAR1997] table 3

⁴⁶ Par exemple, si on a l'affirmation « $X \rightarrow_{af} C$ » et « X », si on arrive à trouver une affirmation « $X \wedge Y \rightarrow_s \neg C$ » et « Y », alors on a court-circuité la première affirmation en trouvant un support à une affirmation rivale.

Si on fournit au système qui vient d'être décrit un ensemble d'« autorisations », de données, et de conclusions, il peut générer une argumentation dialectique. Le passage d'un coté à l'autre comme vérification des qualificatifs est réalisé par un mouvement réussi. Et le fait de décider quel mouvement est suffisant pour générer la vérification des qualificatifs d'un coté et décider quand le processus d'argumentation est fini, dépend de la charge de la preuve qui détermine qui doit faire une preuve, et à quel niveau elle doit se situer.

Nous allons maintenant voir à l'Exemple 7 un pas d'argumentation.

Exemple 7

Reprenons un exemple donné par Toulmin⁴⁷ disant que quiconque est né aux Bermudes peut être présumé sujet britannique (W2), à moins que ses deux parents ne soient étrangers (W5). Les statistiques montrent que la plupart des gens qui parlent anglais et qui ont un numéro d'identification des Bermudes sont nés aux Bermudes (W3). Disposer d'un numéro d'identification des Bermudes, ou avoir des compétences de travail particulières permet d'obtenir un permis de travail (respectivement W4 et W6), à moins qu'il n'y ait déjà assez de travailleurs dans ce métier (W7). La plupart des gens qui ont un passeport britannique, ou qui payent des taxes aux Bermudes, ou qui habitent les Bermudes sont des sujets britanniques (respectivement W1, W8 et W9).

Ce qui donne les règles suivantes (W pour autorisation) :

- W1 : (passeport britannique) → ex ev (sujet britannique)
- W2 : (né aux Bermudes) → ex df (sujet britannique)
- W3 : (parle anglais) (id# des Bermudes) → si ev (né aux Bermudes)
- W4 : (id# des Bermudes) → ex df (permis de travail)
- W5 : (né aux Bermudes) (parents étrangers) → ex df (pas sujet britannique)
- W6 : (compétences de trav. particulières) → ex df (permis de travail)
- W7 : (compétences de trav. particulières)(atteint le quota) → ex df (pas de permis de travail)
- W8 : (paye les taxes aux Bermudes) → ex ev (sujet britannique)
- W9 : (habite aux Bermudes) → ex ev (sujet britannique)

Pour Harry, nous avons les faits suivants (d pour donnée):

- d1 : parle anglais (!? Observé)
- d2 : passeport britannique (?! Donné)
- d3 : métier particulier (!? Donné)
- d4 : atteint le quota (!? Donné)
- d5 : paye les taxes aux Bermudes (!? Donné)
- d6 : habite aux Bermudes (!? Donné)

⁴⁷ [TOU1958]

d7 : permis de travail (!? Donné)

Avec cela, on aimerait savoir si Harry est un sujet britannique ce qui s'exprime comme suit :

affirmation : sujet britannique (?? Nil)

Le processus dialectique commence avec la partie-1 qui essaye de soutenir son affirmation. Comme il n'y a pas de fait qui soutient directement l'affirmation, la partie-1 va chercher la « meilleure » autorisation comme soutien. Elle prend W2, mais il lui faut alors trouver du soutien pour né aux Bermudes (puisque cette affirmation n'est pas présente dans les faits), ce qu'elle trouve avec W3, reste encore à trouver du soutien pour id# des Bermudes. Ce soutien est trouvé au travers de W4 par abduction. On retrouve cette chaîne de TAU à la Figure 4.

*Légende : t_i pour identifier une TAU,
mp: pour modus ponens
abd: pour abduction directe*

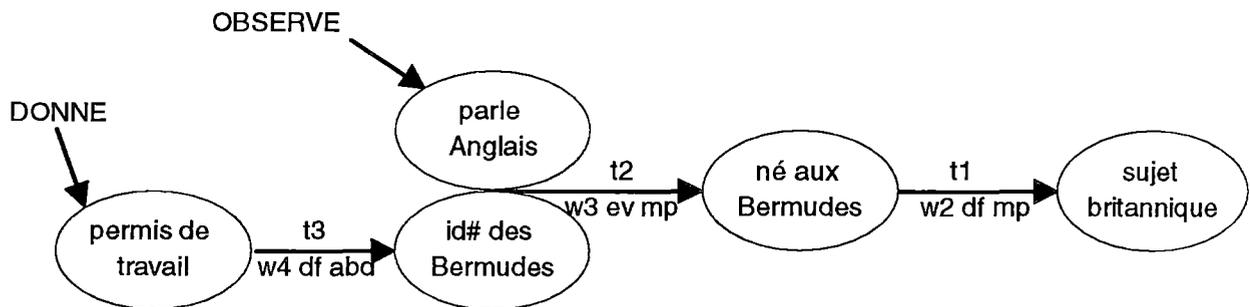


Figure 4

Après cela, la main est passée à la partie-2 dont le but est d'essayer de trouver une faille dans le raisonnement de la partie adverse. Elle essaie tour à tour de détruire un des TAU de la chaîne, mais pour cela, il faut qu'elle trouve une TAU qui ait au moins le même niveau de preuve que celui de la TAU qu'elle attaque. La partie-2 parvient à ses fins grâce à qui fournit une autre explication pour « permis de travail », ce qui permet de mettre en doute t₃ et en conséquence, toute la chaîne de TAU. (Cf. Figure).

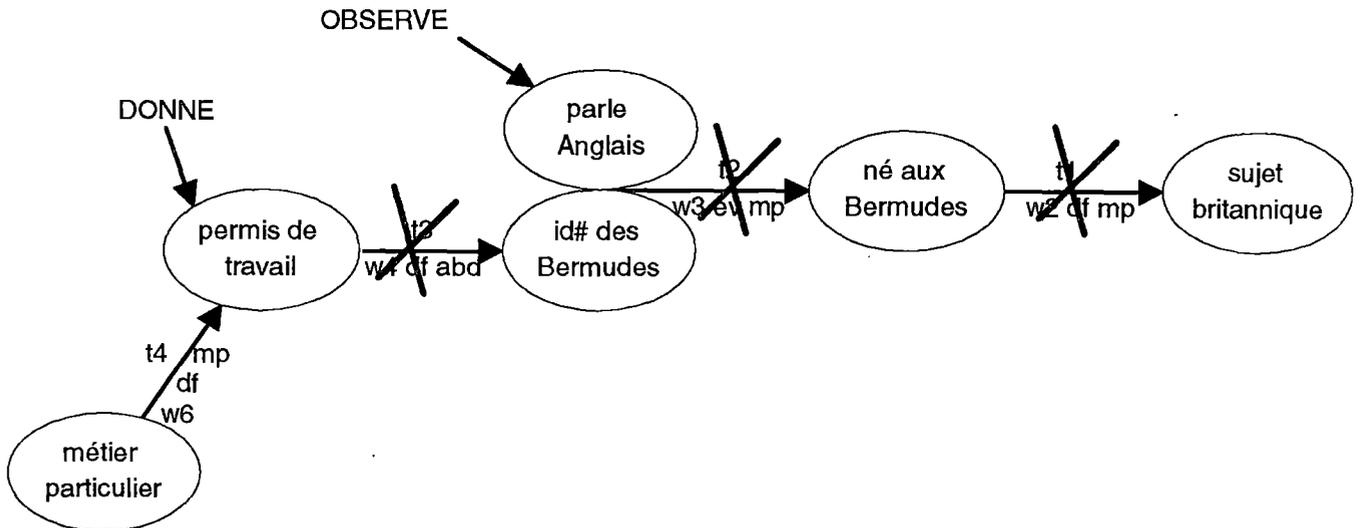


Figure 5

On voit clairement le caractère non-monotone de cette argumentation dialectique, puisque les conclusions sont sans cesse remises en questions. On assiste à chaque pas de la dialectique, à un va-et-vient entre une conclusion et son opposée: elles sont chacune à leur tour considérées comme étant « vraies ».

7. Représentation des connaissances juridiques

7.1 La DataLex Legal Workstation

La station de travail juridique *DataLex*⁴⁸ de G. Greenleaf, E. Mowbay et A. Tyree, met à la disposition des juristes plusieurs techniques informatiques travaillant sur une même source de droit. Ces auteurs ont appliqué trois techniques différentes à l'*Austalian Privacy Act* à savoir, la technique d'hypertexte, celle de repérage d'information et celle de système expert.

⁴⁸ [GRE1991]

L'**hypertexte** permet aux juristes de pouvoir visualiser rapidement toutes les références faites dans un texte, mais aussi les références faites dans ces textes référencés et ainsi de suite⁴⁹.

La technique de **repérage d'information**, quant à elle, permet aux juristes de retrouver rapidement un chapitre, un article, une section, un paragraphe ou un mot quelconque dans les textes contenus dans le système.

Le juriste a également à sa disposition un **SEJ** qui lui permet, à partir de règles tirées automatiquement de l'*Australian Privacy Act* et exprimées dans un langage « proche » de l'anglais⁵⁰, de faire du chaînage avant ou arrière, ainsi que de l'inférence procédurale⁵¹. A ce point cependant, il semble inévitable de s'interroger sur ce SEJ. Il fonctionne sur des règles obtenues mécaniquement à partir du texte original grâce à un analyseur « fonctionnant » on ne sait trop comment.

Ce qui est toutefois remarquable dans ce système, c'est qu'il peut être vu comme une interaction homme/machine dans laquelle chacun fait ce qu'il sait le mieux faire. La machine contrôle les aspects qui peuvent être « aisément » formalisés, alors que l'utilisateur contrôle ceux qui demandent des aptitudes qui sont difficilement formalisables sous cette forme (incluant l'interprétation des textures ouvertes).

7.2 Prolex

En raison de son caractère vague et sa texture ouverte, la connaissance juridique est un type de connaissance qui ne peut être formalisé de manière simple, ni rigoureuse. Oskamp⁵² propose plusieurs manières de la représenter, elle distingue plusieurs groupes de connaissances qui auront chacun leurs propres termes de représentation et leurs propres mécanismes de raisonnement.

⁴⁹ Afin d'éviter le plat de spaghettis dans ces références hypertextes, ils utilisent un index qui reprend pour chaque texte important une liste de liens vers toutes les autres sources citées dans le texte en question. De plus, afin d'automatiser la création des liens hypertextes, un script spécifique est écrit pour chaque catégorie standard de documents et pour chaque document important non standard. Ainsi, lorsqu'un changement se produit dans un texte, il suffit de le repasser au travers du script pour avoir un document hypertexte à jour.

⁵⁰ Ce langage comprend la plupart des structures trouvées dans les langages procéduraux conventionnels (par exemple, l'assignation, des boucles, l'« IF-THEN-ELSE », ...).

⁵¹ Comme le langage utilisé est relativement « complet », il permet à une règle d'en appeler une autre, c'est ce qu'on appelle l'inférence procédurale.

⁵²[OSK1988] et [OSK1989]

Le but d'Oskamp est de construire un noyau pour réaliser des SEJs et d'offrir par la même occasion, une méthodologie de développement de SEJs. Comme exemple d'application elle développe un SEJ, PROLEX, sur la loi néerlandaise des baux à ferme.

Dans les pratiques juridiques, Oskamp distingue l'aide de première urgence, qui consiste à déterminer si un cas relève du droit et si oui, à expliquer l'applicabilité de la loi. Quant au deuxième niveau, il s'agit des cas qui sont présentés devant les tribunaux, et où il faut convaincre le juge que la vision des choses que l'on a est la bonne. Ici, l'avocat(e) ne fait pas uniquement appel à des connaissances juridiques, mais aussi à d'autres éléments comme les récents développements sociaux, la personnalité du juge, etc.

C'est au premier niveau que le système d'Oskamp apporte un support, parce que les juristes qui travaillent à ce niveau ne sont souvent pas des spécialistes du domaine juridique auquel ils sont confrontés.

Oskamp distingue 4 groupes de connaissances dans son système qui sont en relation avec les différentes sources du droit dont elles proviennent :

- 1) **Le groupe législatif** : qui reprend les connaissances liées à la législation au sens large, c'est-à-dire les lois, arrêtés et règlements.
- 2) **Le groupe jurisprudentiel** : qui comprend les connaissances liées à la jurisprudence.
- 3) **Le groupe de la doctrine** : qui comprend ce qu'on appelle la doctrine (c'est une source de droit écrite et informelle - en Belgique - comprenant diverses opinions de différents experts juridiques à propos d'une matière particulière).⁵³
- 4) **Le groupe connaissances d'expert** : qui comprend les connaissances liées à l'expérience des experts.

L'idée de ce groupement de connaissances selon leur origine, c'est de permettre de les représenter par des techniques appropriées⁵⁴ et de ne représenter chaque connaissance qu'une seule fois. En plus de ces quatre groupes, le système a besoin de méta-

⁵³Le point commun entre le groupe 2 et 3 est qu'ils expliquent tous deux les buts de la législation au sens large, ainsi que la façon dont elle doit être comprise; mais d'un côté, les connaissances proviennent des tribunaux (groupe 2) et de l'autre, elles proviennent de sources très variées.

⁵⁴Trouver une technique appropriée à ces connaissances est toujours un sujet de recherche, on peut citer les travaux de l'équipe de Sergot sur *the British Nationality Act* essayant de formaliser la législation, ou, parmi les divers travaux menés sur le raisonnement à base de cas, ceux de Rissland et Ashley qui ont donné lieu au projet HYPO.

connaissances pour remplir au mieux sa tâche, c'est elle qui rend possible l'expression de l'importance des différentes parties de connaissances en interrelations. On trouvera cette méta-connaissances dans le mécanisme de combinaison.

Le mécanisme de combinaison ne contient donc pas la connaissance, mais s'y rapporte par des références. Pour gérer l'interrelation des connaissances, le mécanisme de combinaison doit « décider » quelles connaissances consulter et dans quel ordre; pour cela, il a besoin entre autres de la connaissance des experts juridiques.

Dans PROLEX, la connaissance est représentée à différents niveaux dans une certaine hiérarchie, chaque niveau traite d'un aspect différent du domaine de connaissances.

La première couche comprend les **groupes de connaissances** qui ont chacun leur propre langage de représentation, et leur propre technique de raisonnement. Par exemple, Le groupe législatif est représenté par un système de production avec un mécanisme de chaînage avant et arrière; le groupe jurisprudence utilise des frames et un mécanisme d'inférence (*abstraction finder*) qui permet de trouver des analogies et le groupe de connaissances d'experts ne contient que des heuristiques de bas niveau.

La deuxième couche est constituée de **groupes logiques** qui servent à structurer les groupes de connaissances. Chaque groupe logique comprend des connaissances, provenant de divers groupes de connaissances, considérées comme liées par les experts. On appelle unité de connaissance logique une portion de connaissance qui est exprimée en un seul langage de représentation de connaissance quelconque. Si un groupe logique active une unité logique, le mécanisme de raisonnement qui lui est associé est immédiatement déclenché. Quand un groupe logique est actif, toutes ses unités le sont, et seul les unités actives peuvent être utilisées par les mécanismes de raisonnement. Cette façon de fonctionner permet au système de se focaliser sur une matière particulière.

La troisième couche est un **réseau de classification** des groupes logiques. Pour chaque domaine, un réseau distinct est construit (ses noeuds sont constitués de groupes logiques). Ce réseau représente la méta-connaissance à proprement parler, il a deux rôles. Le premier est de classer les groupes logiques en fonction de leur pertinence par rapport aux éléments du problème, et de représenter le flux d'activation⁵⁵ des groupes logiques. Le second rôle est de fournir un graphe du flux de connaissances au

⁵⁵Le flux d'activation circule des faits initiaux vers les unités logiques et enfin vers les groupes logiques, ce flux est qualifié de flux avant.

gestionnaire d'agenda ⁵⁶ qui décide sur quelle matière il faut poursuivre le raisonnement.

Voici comment tout cela fonctionne (voir Figure 6) : le gestionnaire d'agenda travaille à partir de la méta-connaissance, tirée du réseau de classification, et du feed-back des mécanismes de raisonnement. Il met à jour les connaissances actives (en activant les unités logiques d'un groupe logique, et donc les différents mécanismes de raisonnement). Ensuite, il choisit la technique de raisonnement la plus prometteuse. Ce mécanisme fait une certaine partie du travail, émet un nouveau rapport (c'est le feed-back) et ensuite rend la main à l'agenda. Le mécanisme de raisonnement écrit ses conclusions sur un tableau noir (*black board*) afin d'être lues par les autres mécanismes de raisonnement. Deux autres modules travaillent sur ce tableau, *l'explicateur* qui permet de répondre à certaines questions sur le contenu du tableau noir, et un système de maintien de la cohérence (*thruth-maintenance system*) pour ce tableau noir.

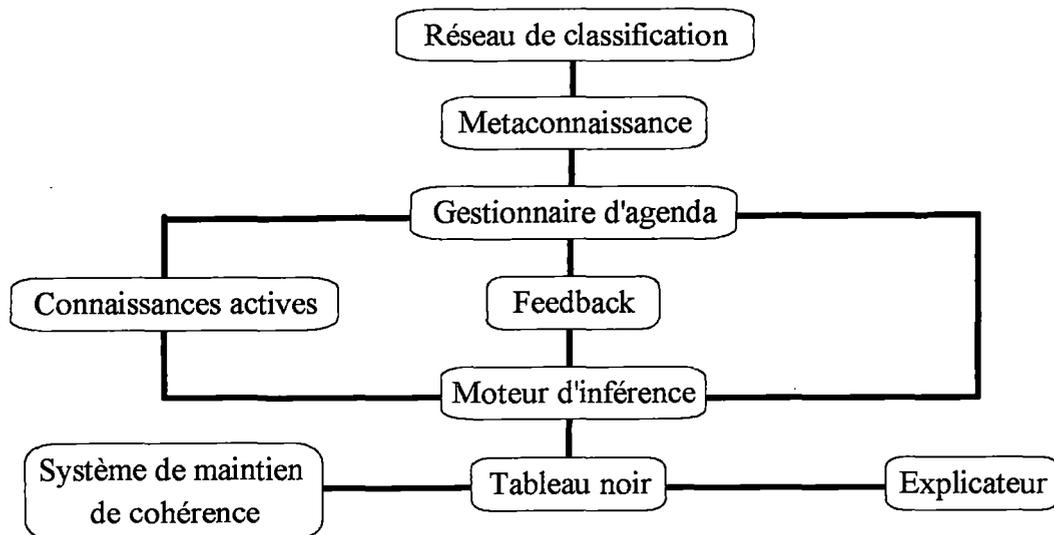


Figure 6

Cette théorie, bien que plus ou moins implémentée au travers du système PROLEX, doit, d'après Oskamp, être plutôt vue comme une situation idéale, comme le but à atteindre par les futurs systèmes.

⁵⁶C'est le mécanisme de contrôle, de combinaison à proprement parler.

Nous venons de présenter dans cette section un ensemble de systèmes pouvant fournir, et même parfois comparer, des solutions conflictuelles et alternatives à un problème juridique. Tous ces systèmes utilisent des méthodes particulières de modélisation - du raisonnement juridique - qui n'ont que peu de bases formelles explicites. On pourrait, sans aucune connotation péjorative, qualifier ces méthodes de stratagèmes fonctionnant plus ou moins bien selon les cas. Dans la suite de ce mémoire, nous allons nous placer à un niveau beaucoup plus formel et voir comment la logique peut être un outil de modélisation du raisonnement non-monotone; et plus particulièrement, du raisonnement juridique.

Chapitre 2 : Formalisations du raisonnement non-monotone

Dans beaucoup de situations, nous sommes amenés à tirer des conclusions par défaut¹ sur base d'une information incomplète. Le résultat d'un tel raisonnement est généralement discutable, c'est-à-dire qu'il n'est pas nécessairement vrai même si l'information sur laquelle il est basé est vraie. Néanmoins, ce type de raisonnement - appelé « non-monotone » pour la première fois par Minsky² en 1975 - est couramment utilisé dans ce qui est appelé le raisonnement de sens commun.

Comme cela a été dit au chapitre précédent, il y a principalement deux types de raisons d'utiliser le raisonnement non-monotone lorsque l'on veut appliquer l'IA au droit. Le premier type de raisons se situe au niveau de la représentation des connaissances. En effet, pour des raisons de validation et de mises à jour, on désire avoir ce qui fut appelé - au point 3.2 *Les exceptions aux règles* - une formalisation isomorphique, c'est-à-dire conserver la séparation d'origine entre les règles et leurs exceptions.

Le deuxième type de raisons se place au niveau de la nature du raisonnement qu'utilisent les juristes; ces raisons sont les suivantes :

- Premièrement, les juristes doivent souvent raisonner sur base d'une information incomplète, car trouver une information complète est souvent quelque chose de très coûteux et même parfois impossible. Dès lors, ils supposent les choses normales jusqu'à preuve du contraire afin de pouvoir appliquer des règles générales invalidables. De cette façon, il leur est possible d'arriver rationnellement à une conclusion.
- Deuxièmement, les juristes doivent utiliser une information incohérente pour arriver à leurs conclusions. Le fait que les lois évoluent sous la tutelle de législateurs différents dans le temps et l'espace peut mener à des incohérences. Afin d'éviter de légiférer pour supprimer l'incohérence en cas de conflit, le droit recourt à des méta-règles de résolution de conflits. Trois grands principes de résolution sont utilisés en droit : *Lex Superior* et *Lex Specialis* sont souvent utilisés pour faire exception à certaines règles; *Lex Posterior* est souvent utilisé afin de pouvoir édicter de

¹ Ces conclusions sont tirées en l'absence d'informations contradictoires.

² [MIN1975]

nouvelles « lois » sans avoir à supprimer d'anciennes lois couvrant, dans certains cas, le même domaine. Ces méta-règles sont, bien entendu, générales et leur issue peut changer avec l'apport de nouvelles informations.

L'incohérence provient aussi de l'incomplétude, de l'ambiguïté et du vague de beaucoup de règles juridiques et de cas de jurisprudence. Cette ouverture du droit fait que les juristes sont souvent en désaccord sur des questions d'interprétations juridiques ayant pour objet la classification d'un cas comme instance d'un concept de droit. Donc, si nous voulons modéliser la réalité juridique, il faut prendre en compte un système contenant différentes vues - incohérentes entre elles - de la « loi ».

Parmi les langages de représentation de connaissances qui permettent le raisonnement non-monotone nous distinguons deux groupes. D'un côté nous avons les systèmes ayant évolué de travaux en IA, ils renferment un composant non-monotone qui résulte d'algorithmes astucieux adaptés au langage particulier utilisé et à la tâche qui est dédiée au système. Ce composant peut être un réseau d'héritage, un système de maintenance de cohérence ou un programme logique à négation par échec. De l'autre côté, nous avons les formalismes dérivés de la logique classique dont le but est de modéliser l'inférence par défaut comme la logique classique modélise l'inférence déductive.

Le premier groupe de formalismes est celui qu'utilisaient les premiers systèmes experts juridiques, comme ceux que nous avons présentés au premier chapitre. Nous n'approfondirons pas ce sujet, nous reviendrons toutefois sur les systèmes de maintenance de cohérence qui seront présentés aux points 9 et 10 comme une implémentation possible d'un système non-monotone.

Nous allons consacrer la majeure partie de ce chapitre à ce qu'on peut appeler « les logiques non-monotones » dont le but est de fournir des méthodes de preuve générales pour le raisonnement par défaut que le raisonnement juridique utilise.

Concepts de base

Le raisonnement déductif standard procède de la façon suivante: si on sait que tous les hommes sont mortels et que Socrate est un homme, alors on peut déduire - par syllogisme - que Socrate est mortel. Par la suite, rien de ce que l'on ajoutera à notre connaissance ne pourra changer ce fait. On dit du raisonnement déductif standard qu'il est monotone, c'est-à-dire qu'il possède la propriété suivante:

$$\text{Si } A \mid - p \text{ alors } A, B \mid - p$$

Comme son nom l'indique, le raisonnement non-monotone ne partage pas cette propriété. L'ensemble des conclusions possibles n'augmente pas de façon monotone avec l'acquisition de nouvelles informations. Par exemple, si la seule chose que l'on sait à propos de Tweety est qu'il est un oiseau, on pourra en conclure qu'il peut voler; mais en apprenant par la suite que Tweety est un pingouin, cette conclusion tombe à l'eau.

1. Propriétés méta-théoriques

Les propriétés structurelles méta-théoriques d'une logique peuvent être étudiées de manière abstraite en considérant ses relations d'inférence comme des opérateurs travaillant sur des phrases du langage. Par exemple, si $X \vdash a$ signifie « la phrase a est prouvable (selon la logique du premier ordre) à partir de l'ensemble fini de phrases X », alors nous avons les propriétés suivantes pour la plupart des logiques classiques et plus particulièrement pour la logique du premier ordre :

$$X, a \vdash a$$

$$\text{si } X \vdash a, \text{ alors } X, y \vdash a$$

$$\text{si } X \vdash y \text{ et } X, y \vdash a, \text{ alors } X \vdash a$$

où X, a est une notation abrégée pour l'ensemble $X \cup \{a\}$.

De la même manière, on peut étudier les propriétés structurelles méta-théoriques que devrait avoir une logique non-monotone. Si $X \sim a$ signifie que « la phrase a est non-monotoniquement déductible à partir de l'ensemble de phrases X », alors :

$$\textit{Inclusion} \quad X, a \sim a \quad (1)$$

$$\textit{Supraclassicalité} \quad \text{si } X \vdash a \text{ alors } X \sim a \quad (2)$$

$$\textit{Monotonie prudente} \quad \text{si } X \sim y \text{ et } X \sim a \text{ alors } X, y \sim a \quad (3)^3$$

$$\textit{Raccourci} \quad \text{si } X \sim y \text{ et } X, y \sim a \text{ alors } X \sim a \quad (4)$$

Illustrons ces propriétés au travers d'exemples:

³ Dans cette proposition, on interdit que y contienne quelque chose qui contredise a puisque toutes les deux sont tirées du même ensemble de prémisses; sous cette condition, on a la propriété de monotonie.

Exemple 8

Supposons que quand il y a une soirée, normalement, Pierre y assiste. Si X est la condition qu'il y a une soirée, et a le fait que, normalement, Pierre va à la soirée:

- (1) si on apprend que, normalement, Pierre va à la soirée, alors on garde notre conclusion antérieure qui dit que, normalement, Pierre ira à la soirée;
- (2) si on est certain que Pierre ira à la soirée quoi qu'il arrive, alors, normalement, il ira à la soirée;
- (3) si, normalement, Pierre va à la soirée, et que, normalement, Marie y va aussi; alors, savoir que, normalement, Marie y va ne devrait pas nous faire perdre la conclusion qui dit que, normalement, Pierre ira à cette soirée;
- (4) si, normalement, Marie va à la soirée et que, normalement, quand Marie va à une soirée, Pierre y va aussi; alors, normalement, Pierre ira à la soirée.

En plus de ces propriétés structurelles, on peut se demander comment des opérateurs de phrases propositionnelles et du premier ordre interagissent avec la dérivation non-monotone. Voici par exemple les règles pour la conjonction, la disjonction et la négation:

si $X \vdash a$ et $X \vdash b$, alors $X \vdash a \wedge b$ (ET)

si $X, y \vdash a$ et $X, z \vdash a$, alors $X, y \vee z \vdash a$ (OU)

si $X \vdash a$ et non $X \vdash \neg y$, alors $X, y \vdash a$ (Rationalité)

Le *ET* est un principe familier qui semble raisonnable pour toute logique non-monotone:

Si, normalement, Pierre va à la soirée et que, normalement, Julie va aussi à cette soirée, alors on peut dire que, normalement, Pierre et Julie vont à la soirée.

Le *OU* supporte, quant à lui, un type de raisonnement par cas, mais reste naturel comme le montre l'exemple suivant:

Si, normalement, Pierre va à une soirée les mardis, et que, normalement, il va aussi à une soirée les jeudis, alors il devrait, normalement, aller à une soirée si on est mardi ou jeudi.

La *rationalité* est une propriété très forte, elle signifie que toute proposition qui n'est pas explicitement contredite par un ensemble de prémisses peut être ajoutée à cet ensemble sans en changer ses conclusions. Pour l'exemple de la soirée, cela pourrait donner :

Si, normalement, Pierre va à une soirée, apprendre que Socrate est mortel ne changera pas la conclusion. Il en est de même si on apprend que Pierre est en vacances. Cela pourrait mettre en doute le fait que Pierre ira à la soirée, mais ce n'est pas le cas, car cette phrase ne contredit pas explicitement le fait que Pierre ira à la soirée.

Les tentatives de formalisation du raisonnement non-monotone ont débouché au début des années 1980 sur différentes « familles » de logiques qu'il est possible de classer de différentes façons. Nous allons choisir le même classement que celui utilisé par Brewka, Dix et Konolige dans [BRE1991]. Ces auteurs distinguent deux classes de logiques non-monotones : d'un côté la classe des logiques qui possèdent la propriété méta-théorique de monotonie prudente et qui définissent une notion sémantique de préférence d'un ensemble de modèles d'une théorie (les logiques à préférence de modèles); d'un autre côté celle qui ne possède pas ces deux caractéristiques (les logiques à base de cohérence ou aussi appelées à point fixe + les méthodes abductives que nous n'aborderons pas ici).

Logiques à préférence de modèles

Les logiques à préférence de modèles - d'une théorie - sont basées sur des notions sémantiques de choix d'un ensemble de modèles préférés. Shoham⁴ définit de manière générale une logique à préférence de modèles en utilisant un ordre arbitraire \prec sur les interprétations d'un langage. Soient \mathcal{L} un langage et \prec un ordre stricte bien fondé⁵ de l'interprétation de \mathcal{L} . On définit alors une logique à préférence \mathcal{L}_\prec comme suit :

⁴ [SHO1987]

⁵ Par ordre bien fondé on veut dire qu'il n'y a pas de chaîne descendante infinie ... $i_3 \prec i_2 \prec i_1$.

Définition 1 : (logique à préférence)

Une interprétation i satisfait de manière préférentielle une phrase A (noté $i \models_{\prec} A$) ssi i satisfait A et il n'existe pas de $i' \prec i$ satisfaisant A . i est un modèle préféré de A .

A implique de manière préférentielle B (noté $A \models_{\prec} B$) ssi tous les modèles préférés de A satisfont B .

La formulation générale de la préférence de modèles est utile en tant qu'outil analytique, il permet d'isoler les propriétés des logiques à préférence de modèles.

Il y existe différentes façons de spécifier ce que signifient des modèles préférés, nous en verrons deux dans cette section :

- l'hypothèse de monde clos, qui est une méthode simple, basée sur l'idée que tous les termes qui ne sont pas prouvés vrais, sont considérés comme faux;
- la circonscription, qui est exprimée sous la forme d'une formule du second ordre minimisant les extensions d'un ensemble de prédicats.

2. L'hypothèse de monde clos

L'hypothèse de monde clos (HMC) utilise, dans le contexte de bases de données déductives, une forme simple de préférence pour l'information négative. Dans un tel contexte, ne sont considérées comme vraies que les instances positives des prédicats appartenant à la base de données, tout le reste est supposé faux.

Si la base de données est un simple ensemble de faits, l'HMC peut être simplement implémentée par une recherche dans la base de données. Si par contre, la base de données comprend un composant déductif, alors l'HMC peut être définie comme le fait Reiter⁶ par :

Définition 2 : (HMC)

$$\text{Fermeture (BD)} = \text{BD} \cup \{\neg P(t) \mid \text{BD} \not\models P(t)\}$$

Où $P(t)$ est une instance d'un prédicat P .

⁶ Cité dans [BRE1991].

L'HMC n'est toutefois applicable qu'à des bases de données ne contenant pas de disjonctions entre deux atomes ou plus. Dans de tels cas, l'HMC peut produire des résultats incohérents, comme :

$$DB = \{A \vee B\} \quad \Rightarrow \quad \text{Fermeture}(BD) = \{A \vee B, \neg A, \neg B\}$$

L'idée de base de l'HMC est donc de minimiser le nombre de prédicats dans la base de données à ceux nécessaires, à savoir les faits positifs. De cette façon, on évite d'avoir à enregistrer tout ce qui est faux (ce qui est, par ailleurs, impossible à faire).

3. La circonscription

La circonscription est une méthode qui préfère les modèles d'une théorie⁷ qui ont des extensions minimales pour quelques prédicats et fonctions représentant des connaissances exceptionnelles. La circonscription consiste donc à faire en sorte que dans les modèles préférés, un minimum des prédicats exceptionnels soient vrais. Nous allons examiner la circonscription sous deux angles : syntaxique et sémantique.

3.1 Forme syntaxique

L'idée est que si on veut poser une hypothèse sur le domaine d'une théorie, il faut ajouter à la théorie une formule qui exprime explicitement cette hypothèse : si on fait cela, on peut raisonner de façon monotone sur la théorie ainsi étendue. Plus formellement, si T est la théorie initiale, $C(T)$ la formule exprimant notre hypothèse et \vdash_{circ} la conséquence de la circonscription, alors :

$$T \vdash_{\text{circ}} \phi \text{ ssi } T + C(T) \vdash \phi$$

Le caractère non-monotone de cette idée réside dans le fait que le contenu de la formule de circonscription dépend entièrement de la théorie : ainsi si la théorie est étendue à T' , la formule de circonscription doit être changée en $C(T')$, et les conséquences déductives de $T + C(T)$ ne sont plus valables pour $T' + C(T')$.

Cela fonctionne comme ceci : si certains prédicats d'une théorie sont considérés comme exprimant des propriétés ou relations exceptionnelles, alors la formule de circonscription circonscrit ces prédicats. C'est-à-dire qu'elle imposera que les seuls objets pouvant posséder ces propriétés ou relations exceptionnelles sont ceux qui sont nécessaires à la lumière de la théorie.

⁷ Par théorie, nous entendons une collection de phrases implicitement liées par la conjonction.

Exemple 9

Supposons une base de données contenant le fait $Q(a)$, la circonscription de Q fait que la formule $\forall x. Q(x) \leftrightarrow x=a$. Alors, si b est un objet différent de a , la circonscription de Q implique la conclusion $\neg Q(b)$. Mais si on apprend plus tard que $Q(b)$ est vrai, la conclusion $\neg Q(b)$ tombe et les nouvelles conclusions sont déduites de la formule $\forall x. Q(x) \leftrightarrow x=a \vee x=b$.

Exemple 10

Si on a une théorie qui porte sur la loi criminelle, et qu'il est vu comme exceptionnel qu'un criminel soit un malade mental, alors le prédicat « malade mental » peut être circonscrit : dans ce cas, si rien ne peut être prouvé à propos de la mauvaise santé mentale du criminel, il (ou elle) sera présumé(e) en bonne santé mentale.

On voit que cette forme de circonscription se comporte comme une hypothèse de monde clos capable de traiter des théories plus riches que celles exprimables dans les bases de données. En fait, la circonscription fait en sorte que les prédicats circonscrits renvoient une valeur fausse pour toutes leurs instances dont la théorie ne dit pas qu'ils doivent renvoyer la valeur vraie (cf. Exemple 9). Ce type simple de circonscription est défini par:

Définition 3 : (circonscription de prédicats)

Soit T une formule du premier ordre (la conjonction de prémisses de la théorie) comprenant un prédicat n -aire P . Soit $T(P')$ le résultat du remplacement de toutes les occurrences de P dans T par une variable de prédicat P' . Alors la circonscription prédicat de P dans T est la formule du second ordre suivante :

$$T(P) \wedge \forall P' [(T(P') \wedge \forall x (P'x \rightarrow Px)) \rightarrow \forall x (Px \rightarrow P'x)]$$

où $x = x_1, \dots, x_n$.

Cela veut dire que parmi les prédicats P' satisfaisant à $T(P')$, P est parmi les plus « forts » c'est-à-dire que ces P' ne peuvent être strictement plus forts que P . Ainsi :

Si $T(Q)$ est le fait $Q(a)$, par exemple, si $P'x \equiv x=a$ on obtient la formule : $Q(a) \wedge [a=a \wedge \forall x (x=a \rightarrow Q(x)) \rightarrow \forall x (Q(x) \rightarrow x=a)]$.

Formule qui peut être simplifiée en « $\forall x. Q(x) \leftrightarrow x=a$ » qui dit que seul a possède la propriété Q .

3.2 Forme sémantique

L'idée qui se trouve derrière la forme sémantique de la circonscription est qu'au lieu de vérifier tous les modèles des prémisses pour savoir si une conclusion est valide, seuls les modèles dont les extensions sont minimales seront vérifiés. Pour obtenir ces modèles minimaux, seules les extensions des prédicats circonscrits devront être minimisées, alors que d'autres devront être fixes et d'autres encore pourront varier.

Ce traitement différent des extensions est dû à la nature du raisonnement dans lequel on distingue trois types d'informations :

- les faits en entrée;
- l'information exceptionnelle qui doit être, si possible, supposée fausse (minimisée);
- et les faits en sortie, c'est-à-dire les conclusions qui nous intéressent.

Nous avons donc que :

- les prédicats d'entrée devront rester fixes puis qu'ils décrivent simplement les faits;
- les prédicats exceptionnels devront être minimisés;
- et les prédicats de sortie devront pouvoir varier.

Ce qui nous amène à la définition généralisée suivante d'un modèle minimal (qui correspond en fait à la *circonscription variable*) :

Définition 4 : (modèle minimal)

Soit T une théorie du premier ordre, P , Q et Z une partition de l'ensemble de tous les prédicats du langage de T . P représente l'ensemble de prédicats circonscrits (minimisés), Q l'ensemble de prédicats fixes, et Z contient les prédicats variables. Soit M et M' deux modèles de T . On dira que $M \leq_{P,Q,Z} M'$ ssi

- M et M' ont le même domaine;
- $\forall p \in P : I_M(p) \subseteq I_{M'}(p)$;
- $\forall q \in Q : I_M(q) = I_{M'}(q)$.

où I_M = l'interprétation du symbole p dans le modèle M .

M est un *modèle minimal* de $T_{(P,Q,Z)}$ ssi M est minimal dans le préordre $\leq_{P,Q,Z}$ c'est-à-dire ssi il n'y a pas de modèle M' de T tel que $M' \leq_{(P,Q,Z)} M$ et pas $M \leq_{(P,Q,Z)} M'$.

Définition 5 : (la conséquence minimale pour la circonscription variable)

$T_{(P, Q, Z)} \models_{\min} \phi$ ssi ϕ est vraie dans tous les modèles minimaux de $T_{(P, Q, Z)}$ (par rapport aux définitions habituelles de vrai)

L'exemple qui suit montre comment cette définition de conséquence minimale peut être utilisée pour modéliser le raisonnement par défaut.

Exemple 11.

1. $\forall x(\text{Commis-crime}(x) \wedge \neg \text{Malade-mental}(x)) \rightarrow \text{Punir}(x)$
2. $\text{Commis-crime}(\text{Alfred})$

Dans cet exemple, nous voulons savoir si une personne ayant commis un crime doit être punie ou non selon qu'elle est considérée, ou non, comme mentalement malade. Nous avons donc : $Q = \{\text{Commis-crime}\}$, $P = \{\text{Malade-mentale}\}$, $Z = \{\text{Punir}\}$. Les modèles minimaux de $\{1, 2\}_{(P, Q, Z)}$ sont ceux où « $\text{Malade-mental}(\text{Alfred})$ » est faux, et comme « $\text{Commis-crime}(\text{Alfred})$ » est vrai dans tous les modèles, on a que l'antécédent de (1) est vrai dans tous les modèles minimaux. Il en résulte que « $\text{Punir}(\text{Alfred})$ » est vrai dans ces modèles minimaux. Donc, avec P , Q et Z , $\{1, 2\}$ a comme conséquence minimale « $\text{Punir}(\text{Alfred})$ ».

Le caractère non-monotone de la circonscription apparaît lorsque l'on ajoute aux prémisses de l'Exemple 11 la règle suivante :

3. $\text{Malade-mental}(\text{Alfred})$

Ainsi, l'antécédent de (1) devient faux dans tous les modèles minimaux. Dès lors, la conclusion déclarant que « $\text{Punir}(\text{Alfred})$ » est vrai n'est plus valide.

Un technique souvent utilisée pour pouvoir appliquer la circonscription consiste à utiliser des prédicats d'anormalité. Un défaut⁸ peut alors être représenté comme suit :

⁸ Tout au long de ce mémoire, le terme « défaut » sera utilisé dans le sens de dispositions qui doivent être considérées comme vraies en l'absence d'information contradictoire.

$$1. \forall x \text{ Oiseau}(x) \wedge \neg an_1(x) \rightarrow \text{Vole}(x)$$

La signification de $an_1(x)$ est ici : « x est anormal par rapport à la règle 1, c'est-à-dire par rapport aux oiseaux qui volent ».

Ainsi, les prédicats à minimiser sont ceux qui expriment l'anormalité : dans notre exemple c'est « $an_1(x)$ ».

Mais cette technique souffre de plusieurs problèmes lorsque l'on veut l'appliquer au raisonnement de sens commun. L'un de ces problèmes, la spécificité, se produit quand il y a un conflit de défauts comme nous pouvons le voir dans l'exemple suivant :

Exemple 12.

1. $\forall x \text{ Oiseau}(x) \wedge \neg an_1(x) \rightarrow \text{Vole}(x)$
2. $\forall x \text{ Pingouin}(x) \wedge \neg an_2(x) \rightarrow \neg \text{Vole}(x)$
3. $\forall x \text{ Oiseau}(x) \rightarrow \text{Pingouin}(x)$
4. $\text{Pingouin}(\text{Tweety})$

Il y a deux modèles $\leq_{(an_1, an_2; \text{Oiseau}, \text{Pingouin}; \text{Vole})}$ minimaux. Un dans lequel « $an_1(\text{Tweety})$ » est vrai et donc dans lequel Tweety vole; un autre où « $an_2(\text{Tweety})$ » est vrai et où Tweety ne vole pas. Nous aimerions que seul le deuxième modèle soit pris en compte puisqu'il utilise une information plus spécifique : « $\text{Pingouin}(\text{Tweety})$ ».

Une façon d'éliminer le premier modèle serait d'ajouter la règle « $\forall x \text{ Pingouin}(x) \rightarrow an_1(x)$ » à la théorie; mais cette technique exige que l'on connaisse toutes les relations de spécificité entre les prédicats de la théorie, ce qui peut être très contraignant. De plus, si nous nous retrouvions face à un pingouin anormal pouvant voler, nous aimerions que la règle (1) soit d'application, ce qui ne serait pas le cas. Il faut alors procéder autrement.

D'une manière tout à fait générale, il est possible de résoudre ce problème en définissant un ordre parmi les prédicats à minimiser. On partitionne donc les prédicats P en P^1, P^2, \dots, P^n . P^1 contenant les prédicats à minimiser ayant la plus haute priorité; si deux modèles ont les mêmes extensions dans P^1 , alors on compare les extensions de ces modèles dans P^2 , etc.

L'exemple précédent se résout donc en partitionnant P en $P^1 = \{an_2(x)\}$ et $P^2 = \{an_1(x)\}$, ce qui nous donne bien le résultat désiré, à savoir : « $\neg \text{Vole}(\text{Tweety})$ ».

Comme la plupart des formalismes non-monotones, la circonscription a quelques inconvénients. Premièrement, il n'y a aucune garantie que toutes les théories aient un modèle minimal.

Deuxièmement, comme nous avons pu le voir, la circonscription est un formalisme bien plus expressif et plus puissant que la logique du premier ordre. Mais cette expressivité a pour contrepartie qu'il est plus difficile de prouver un théorème dans la circonscription; de plus, toutes les théories de preuves de la circonscription sont incomplètes⁹ (cela est dû au fait que, généralement, la formule de circonscription est du deuxième ordre).

Troisièmement, et peut-être le plus important, le choix du prédicat à substituer par P' dans la formule de circonscription (cf. Définition 3) demande une « réelle » intelligence, non « encore » mécanisable.

Logiques à base de cohérence

Les logiques qui vont être présentées sous ce point utilisent explicitement la notion de cohérence dans leur définition fondamentale. Ces logiques peuvent être divisées en:

- *logiques à point fixe* (logique de défauts, logiques non-monotones modales) où l'on définit les phrases prouvables au travers d'un opérateur à point fixe. Les points fixes de cet opérateur ont la propriété d'avoir été générés par l'application d'autant de défauts cohérents que possible;
- *logiques à cohérence maximale* où les défauts sont représentés par des formules du premier ordre moins fiables¹⁰. Les conclusions sont obtenues par défaut en considérant un sous-ensemble cohérent maximal des prémisses (incluant les prémisses qui représentent des défauts).

Le point commun de ces logiques est qu'elles génèrent parfois plusieurs ensembles de formules cohérentes. Ces ensembles représentent les différents ensembles de croyances qu'un raisonneur peut adopter sur base des faits et défauts mis à sa disposition. Ils sont appelés *extensions* (ou *expansion* pour la logique autoépistémique).

Exemple 13.

⁹ Pour plus d'informations à ce sujet, consulter [SCH1986]

¹⁰ Elles sont moins fiables parce qu'elles sont utilisées dans le cadre de la formalisation du raisonnement non-monotone et perdent donc la propriété de monotonie.

1. Normalement, quand il fait chaud, il fait beau;
2. Normalement, quand il pleut, il fait mauvais;
3. Le 12-08-1993 il a plu mais il faisait chaud.

Sur base de ces règles, nous voulons savoir quel temps il faisait le 12-08-1993. Dans un tel cas, les logiques à base de cohérence génèrent deux extensions opposées : une (avec 1 & 2) concluant qu'il faisait beau; et une autre (avec 2 & 3) concluant qu'il faisait mauvais.

Se pose alors la question de savoir comment on peut dériver des formules quand on est face à plusieurs extensions. Une possibilité est de ne dériver des formules que de l'intersection des extensions, ce qui est une vision pessimiste puisqu'on ne permet aucun conflit. Mais il y a des cas où chaque extension a son intérêt, comme lorsque l'on modélise le raisonnement de diagnostic où chaque extension correspond à un diagnostic. Il se peut aussi que les extensions n'aient pas la même importance. Il faut alors trouver des critères de préférence permettant de faire un choix, cela pourrait être par exemple, la spécificité.

Comme il semble ne pas y avoir de façon correcte d'utiliser les extensions, les logiques à base de cohérence ne font que les définir, laissant libre leur utilisation.

4. Logique de défauts

Une des logiques non-monotones les plus citées avec la circonscription est la logique de défauts de Reiter¹¹. Elle suppose que la connaissance soit représentée sous la forme d'une paire (F, δ) appelée théorie de défauts. F est un ensemble de formules représentant l'ensemble des faits considérés comme vrais et δ est un ensemble de défauts de la forme :

$$\frac{\alpha(x) : \beta(x)}{\gamma(x)}$$

Où $\alpha(x)$, $\beta(x)$ et $\gamma(x)$ sont des formules appelées respectivement précondition, test (justification ou condition de cohérence) et conséquence du défaut. Pour un tuple a de termes, un tel défaut permet de dériver $\gamma(a)$ à partir de $\alpha(a)$ si et seulement si $\beta(a)$ n'est pas dérivable. On dira d'un défaut qu'il est fermé s'il ne contient pas de variable libre;

¹¹ [REI1980]

dans le cas contraire, il est ouvert et peut être vu comme le patron¹² de toutes ses instances. On définit alors une théorie de défauts fermée comme étant une théorie de défauts où les variables libres de tous les défauts ont été instanciées; une théorie de défauts contenant des défauts ouverts est donc comprise comme une théorie fermée habituellement infinie.

Exemple 14

$$\frac{\text{oiseau}(x) : \text{vole}(x)}{\text{vole}(x)}$$

Ce défaut signifie que « par défaut, les oiseaux peuvent voler » c'est-à-dire que si on ne sait pas qu'un oiseau particulier ne peut pas voler, alors, on suppose qu'il le peut. On peut déduire à partir de « oiseau(Tim) » que Tim peut voler, mais si on apprend par la suite que « $\neg\text{vole}(\text{Tim})$ » devient vrai, le défaut est bloqué, et la conclusion précédemment tirée devient invalide.

Dans certains cas, il est possible de rencontrer des conflits entre défauts. C'est le cas si on ajoute le défaut suivant à notre exemple.

$$\frac{\text{blessé}(x) : \neg\text{vole}(x)}{\neg\text{vole}(x)}$$

Ce défaut exprime le fait que « par défaut, si x est blessé, alors il ne peut pas voler ». Donc, si on apprend que Tim est blessé, les deux défauts deviennent applicables. Mais si le premier s'applique, le second est bloqué, et inversement.

Pour apporter une solution à ce problème, il faut d'abord définir ce qu'est l'extension d'une théorie de défauts.

Une nouvelle hypothèse (ou croyance) peut être dérivée en utilisant les faits et en ajoutant autant de défauts que l'on veut, pour autant que la cohérence soit préservée. Si on utilise autant de défauts que possible (c'est-à-dire si l'utilisation d'un autre défaut entraînerait une incohérence), l'ensemble résultat est appelé « *extension* ».

¹² Au sens de modèle.

Définition 6 : (extension)

Soit une théorie de défauts fermés (F, D) , E un ensemble de formules, et $\Gamma(E)$ le plus petit ensemble de formules tel que:

1. $F \subseteq \Gamma(E)$,
il comprend les faits;
2. $\text{Th}(\Gamma(E)) = \Gamma(E)$,
où $\text{Th}(X)$ = toutes les conséquences monotones de X ,
il est déductivement fermé;
3. Si $\alpha(x) : \beta(x) / \gamma(x) \in D$, $\alpha(x) \in \Gamma(E)$ et $\neg\beta(x) \notin E$, alors $\gamma(x) \in \Gamma(E)$,
il doit contenir les conséquences des défauts applicables.

E est une extension de (F, D) ssi $E = \Gamma(E)$, c'est-à-dire ssi E est un point fixe de Γ .

Puisque les défauts d'une théorie peuvent être en conflit, une théorie de défauts peut avoir 0, 1 ou N extensions (mutuellement incohérentes).

Exemple 15

Soient deux règles générales disant que si un meurtre a été prémédité, la peine maximum est de 15 ans; alors que si un meurtre a été commis lors d'un duel, la peine maximum est de 12 ans. Ce qui nous donne :

$$d1 = \text{Meurtre} \wedge \text{Prémédité} : \text{Maximum} = 15 / \text{Maximum} = 15$$

$$d2 = \text{Meurtre} \wedge \text{Duel} : \text{Maximum} = 12 / \text{Maximum} = 12$$

$$F = \{\text{Duel} \rightarrow \text{Prémédité},$$

$$\text{Meurtre},$$

$$\text{Duel},$$

$$\neg(\text{Maximum} = 12 \wedge \text{Maximum} = 15)\}$$

$$\delta = \{d1, d2\}$$

Il faut savoir quel défaut appliquer sur base des faits que l'on possède. Si nous appliquons $d1$, alors « $\text{Maximum} = 12$ » n'est plus cohérent avec F et $d2$ est bloqué; par contre si nous appliquons $d2$ en premier, c'est $d1$ qui est bloqué. Cette théorie de défauts a donc deux extensions possibles:

$$E1 = F \cup \{\text{Maximum} = 15\}$$

$$E2 = F \cup \{\text{Maximum} = 12\}$$

Elles sont toutes deux raisonnables sur base de (F, δ) , mais elles sont incohérentes entre elles, il faut faire un choix.

Pour permettre de faire un choix parmi plusieurs extensions, deux notions alternatives de conséquence pour la logique de défauts doivent être définies.

Définition 7 : (conséquence crédule et sceptique)

Soit (F, δ) une théorie de défauts et A un ensemble de formules du premier ordre. Alors,

- $A, (F, \delta) \Vdash^s \phi$ ssi ϕ est présente dans toutes les extensions de $(F \cup A, \delta)$
(conséquence sceptique)

c'est une vue sceptique des choses puisqu'on ne veut tirer de conclusions que de choses indiscutables

- $A, (F, \delta) \Vdash^c \phi$ ssi ϕ est présente dans au moins une extension de $(F \cup A, \delta)$
(conséquence crédule)

ici, on est crédule parce qu'on est très permissif

Mais dans certains cas d'extensions multiples, une seule d'entre elles était voulue. Ce qui donna l'idée à Reiter de n'utiliser que des défauts normaux qui sont de la forme :

$$\alpha(x) : \omega(x) / \omega(x)$$

ou des défauts semi-normaux qui eux sont de la forme :

$$\alpha(x) : \beta(x) \wedge \omega(x) / \omega(x)$$

où $\beta(x)$ sera typiquement la négation de la précondition du défaut avec lequel il est en conflit et qui aura la priorité. Ce qui pourrait donner pour l'Exemple 15:

Si nous voulons que $d2$ ait priorité sur $d1$, nous transformerons $d1$ comme ceci :

$$d1' = \frac{\text{Meutre} \wedge \text{Prémédité} : \neg \text{Duel} \wedge \text{Maximum} = 15}{\text{Maximum} = 15}$$

alors, la théorie $(F, \{d1', d2\})$ n'a qu'une seule extension qui est $E2$ puisque la condition de cohérence de $d1'$ n'est plus cohérente avec les faits car $F \ni \{\text{Duel}\}$.

Il est à remarquer que $\beta(x)$ est ajouté au test (ou condition de cohérence) du défaut et non à sa précondition, de cette façon, l'absence de la circonstance exceptionnelle n'a pas besoin d'être prouvée pour que la règle générale soit utilisée.

D'une part, les avantages de cette technique font que cette représentation, sous forme de règles d'inférence comprenant un test de cohérence, est simple et assez intuitive.

D'autre part, son pouvoir d'expression est supérieur à celui de ses concurrents comme la circonscription. Ce pouvoir d'expression est dû en grande partie à la représentation des défauts sous forme de règles d'inférence qui permettent en plus d'éviter des problèmes liés la contraposition. En logique classique $A \supset B \Leftrightarrow \neg B \supset \neg A$; mais pour les défauts, cette contraposition est parfois, et même souvent, indésirable. Par exemple, supposons qu'on ait un défaut disant que, normalement, les informaticiens n'ont que peu de connaissances sur le raisonnement non-monotone. De cette règle, on ne doit pas pouvoir dériver qu'en général, les gens qui ont de bonnes connaissances à propos du raisonnement non-monotone ne sont pas des informaticiens. La représentation adoptée par la logique des défauts permet d'éviter ce problème.

L'utilisation de ces règles d'inférence a toutefois ses inconvénients. Premièrement, l'existence d'extensions n'est pas toujours garantie comme le montre le petit exemple suivant.

Exemple 16

$$(\{\text{vrai} : \neg A / A\}, \{\})$$

Si E ne comprend pas A alors E ne peut pas être une extension puisque le défaut n'a pas été appliqué. Par contre si E comprend A alors $\Gamma(E)$ produit un ensemble ne comprenant pas A puisque le défaut est inapplicable par rapport à E. Donc, E n'est de nouveau pas une extension.

Deuxièmement, la logique des défauts ne permet pas de raisonner par cas. Considérons l'exemple suivant :

Exemple 17

$$d1 = \text{Mineur}(x) : \text{Incapable}(x) / \text{Incapable}(x)$$

$$d2 = \text{Handicapé-mental}(x) : \text{Incapable}(x) / \text{Incapable}(x)$$

$$F = \{\text{Mineur}(\text{John}) \vee \text{Handicapé-mental}(\text{John})\}$$

$$\delta = \{d1, d2\}$$

Intuitivement on devrait pouvoir déduire « Incapable(John) » puisque peu importe que John soit mineur ou souffre d'un handicap mental, un des deux défauts devrait s'appliquer. Mais en logique de défauts, un défaut ne s'applique que quand ses préconditions ont été dérivées, or de « Mineur(John) \vee Handicapé-mental(John) » ni la précondition du défaut (d1) ni celle de (d2) ne peuvent être dérivées. Le problème vient du fait que de (d1) et de (d2) on ne puisse déduire le défaut suivant :

$$d3 = \text{Mineur}(x) \vee \text{Handicapé-mental}(x) : \text{Incapable}(x) / \text{Incapable}(x)$$

On voit qu'en terme des propositions méta-théoriques des relations d'inférence non-monotone (Cf. Propriétés méta-théoriques) nos définitions d'inférence - crédule et sceptique - violent le OU.

Le troisième inconvénient se situe au niveau de la complexité computationnelle due principalement à une définition non constructive de l'extension : en effectuant le test de cohérence afin de savoir si un défaut peut être utilisé pour élargir une extension, il ne suffit pas d'inspecter l'extension construite jusque là; il faut envisager, donc deviner, le contenu de l'extension tout entière, ce qui est difficilement « mécanisable ». Plus précisément, on sait que les extensions de la logique des défauts ne sont pas « récursivement énumérables »; le problème de savoir si une formule donnée appartient à une extension d'une théorie de défauts est semi-décidable¹³.

5. Logique modale non-monotone

Une autre façon de tirer des conclusions de connaissances incomplètes est d'essayer d'avoir une théorie qui décrive « complètement » ce qui est connu. Pour Moore¹⁴, une théorie peut être vue comme une base de données (BD) de telle façon que si une affirmation n'est pas dans la BD et ne peut en être déduite, alors on sait que l'affirmation n'est pas connue. Les logiques modales non-monotones utilisent un opérateur modal pour exprimer le fait que l'on sait, que l'on connaît ou ne connaît pas quelque chose. Plusieurs formalismes ont été proposés pour ce type logique, nous n'allons examiner que l'un des plus utilisés : la logique autoépistémique de Moore (LAE)¹⁵.

L'idée de la LAE est de modéliser un agent introspectif idéal raisonnant à propos de ses propres connaissances, c'est-à-dire que cet agent sait les connaissances qu'il possède, mais aussi celles qu'il ne possède pas. Ce qui nous donne d'une manière plus formelle (si l'opérateur modal est « L » et signifie « on sait que », et que T est l'ensemble de toutes les connaissances introspectives de l'agent) :

¹³ Cela veut dire que le mieux que pourra faire un algorithme est de trouver la réponse correcte si elle est « oui »; mais si la réponse est « non », rien ne garantit que l'algorithme ne bouclera pas indéfiniment.

¹⁴ [MOO1985]

¹⁵ [MOO1985]

- si $p \in T$, alors $Lp \in T$;
 - si $p \notin T$, alors $\neg Lp \in T$.
- + En plus de cela, T doit être déductivement fermé puisqu'un « raisonneur » idéal a non seulement des connaissances explicites, mais aussi des connaissances qui découlent logiquement de ses connaissances explicites.

Un ensemble T possédant ces 3 propriétés est appelé un *ensemble stable*.

D'observations introspectives comme celles-là on peut tirer des conclusions comme le montre l'exemple suivant.

Exemple 18

$(Oiseau(x) \wedge \neg LPingouin(x)) \rightarrow Vole(x);$
 $Oiseau(Tweety).$

Ne pas savoir que Tweety est un pingouin est donc exprimé par « $\neg LPingouin(Tweety)$ », on peut dès lors inférer très facilement que Tweety peut voler.

Comme pour la logique des défauts, la LAE peut conduire à ce qu'un « raisonneur » se retrouve face à plusieurs ensembles stables exclusifs de croyances qu'il peut adopter, c'est-à-dire des expansions stables (appelées extensions par Konolige¹⁶).

Définition 8 : (expansion stable)

Soit A un ensemble de formules de la LAE. T est une expansion stable de A ssi

$$T = \text{Th}(A \cup \{Lp \mid p \in T\} \cup \{\neg Lp \mid p \notin T\})$$

¹⁶ [KON1988]

Exemple 19.

$$A = \{(Quaker \wedge \neg L\neg Pacifiste) \rightarrow Pacifiste; \\ (R\acute{e}publicain \wedge \neg LPacifiste) \rightarrow Pacifiste; \\ Quaker, Pacifiste\}$$

Comme ni « Pacifiste » ni « \neg Pacifiste » ne sont dans A , les expansions sont:

$$T_1 = A \cup \{\neg L\neg Pacifiste, Pacifiste, LPacifiste\}; \\ T_2 = A \cup \{\neg LPacifiste, \neg Pacifiste, L\neg Pacifiste\}.$$

Mais, il n'y a aucune expansion contenant à la fois « Pacifiste » et « \neg Pacifiste » puisque si le premier défaut est appliqué, le deuxième devient inapplicable à cause de « $Lpacifiste$ ». Il en est de même si on commence par appliquer le deuxième défaut.

Voyons maintenant comment se manifeste le caractère non-monotone de la LAE au travers de l'exemple du pingouin Tweety.

Exemple 20.

$$\text{Soit } A = \{ (Oiseau(x) \wedge \neg LPingouin(x)) \rightarrow Vole(x); \\ Oiseau(Tweety)\}$$

Dès lors, « $\neg LPingouin(Tweety)$ » est dans T puisque « $Pingouin(Tweety)$ » n'y est pas. En plus, puisque A est inclus dans T (Cf. Définition 8) et que T est déductivement fermé, par simple modus ponens « $Vole(Tweety)$ » est aussi dans T . Mais une fois que l'on ajoute « $Pingouin(Tweety)$ » à l'ensemble A , T contient alors « $LPingouin(Tweety)$ » au lieu de « $\neg LPingouin(Tweety)$ » et « $Vole(Tweety)$ » n'appartient plus à T .

Ce type de logique partage les mêmes désavantages que la logique des défauts à savoir que la définition de l'expansion n'est pas constructive et comme la logique des défauts, la LAE est computationnellement peu intéressante.

6. Logique à cohérence maximale

Dans le cadre des logiques à cohérence maximale, le raisonnement non-monotone est vu comme la prise en charge de l'incohérence d'un ensemble de formules de la logique

classique. Il n'est alors pas nécessaire de développer une nouvelle logique. Cette approche, bien que moins expressive que les précédentes, a un intérêt pratique qui réside à la fois dans une définition de l'extension qui est plus simple et dans des facilités d'implémentation.

En représentant les défauts par des formules d'implications classiques comme :

$$\text{Oiseau}(\text{Tweety}) \supset \text{Vole}(\text{Tweety})$$

les prémisses seront, en général, incohérentes entre elles. Il faut donc des mécanismes qui permettent de gérer cette incohérence (si elle est présente) d'une façon qui reflète notre idée du comportement des défauts.

Une façon standard de faire cela est de considérer des sous-ensembles cohérents maximaux de prémisses (ou sous-théories) et de les ordonner en introduisant une relation de préférence entre eux. En général ce sera le sous-ensemble contenant l'information la plus spécifique qui sera choisi comme nous pouvons le voir en reformulant l'exemple de l'oiseau Tweety.

Exemple 21.

1. $\text{Oiseau}(\text{Tweety}) \supset \text{Vole}(\text{Tweety})$
2. $\text{Pingouin}(\text{Tweety}) \supset \neg \text{Vole}(\text{Tweety})$
3. $\text{Pingouin}(\text{Tweety}) \wedge \text{Oiseau}(\text{Tweety})$

On obtient trois sous-théories (1) & (2), (2) & (3), (1) & (3). Le fait que (1) contienne une information par défaut peut être pris en compte en préférant les sous-théories qui ne comprennent pas (1). Ce qui fait que la sous-théorie préférée serait celle composée de (2) & (3).

Un système basé sur ce principe est appelé *logique à cohérence maximale*. Une théorie de défauts dans cette logique est constituée d'une paire $(T, <)$ où T est un ensemble (cohérent ou non) de prémisses et $<$ un ordre de cohérence maximale sur les sous-ensembles de T . On peut alors définir, en n'utilisant que les sous-ensembles préférés, la notion de conséquence comme suit :

Définition 9 : (conséquence faible et forte)

Soit T un ensemble - qui peut être incohérent - de prémisses, $SUB(T)$ l'ensemble des sous-ensembles à cohérence maximale de T , et $<$ un ordre strict partiel sur $SUB(T)$.

- La formule p est faiblement prouvée par la théorie $(T, <)$ ssi $\exists S$, un élément $<$ -minimal de $SUB(T)$, tels que $S \vdash p$;
- La formule p est fortement prouvée par la théorie $(T, <)$ ssi $\forall S$ étant un élément $<$ -minimal $SUB(T) : S \vdash p$.

On appelle E une extension de T ssi $E = Th(S)$ où S est un élément $<$ -minimal de $SUB(T)$.

Reprenons l'Exemple 21 :

Si on utilise un ordre qui préfère la sous-théorie composée de (2) & (3), alors « $\neg Vole(Tweety)$ » sera fortement prouvé par les prémisses.

6.1 Théorie de Poole

Une façon très pratique de définir un ordre $<$ sur les sous-théories est d'introduire une structure sur T et d'en dériver l'ordre. Par exemple, Poole¹⁷ divise T en :

- un ensemble cohérent F de formules du premier ordre fermées représentant les faits connus;
- un ensemble δ de formules du premier ordre, éventuellement contradictoires, représentant les défauts.

Dans un tel cas, une sous-théorie est préférée si elle contient les faits.

Définition 10 : (scénario)

un scénario d'une théorie de défauts (F, δ) est un ensemble cohérent $F \cup D$ où D est un sous-ensemble de δ où les variables sont instanciées.

Définition 11 : (explication)

l'explication d'une formule fermée ϕ à partir de (F, δ) est un scénario de (F, δ) qui implique ϕ .

Par ces définitions, on voit que pour construire une explication, il faut respecter les faits (F) et utiliser autant de défauts que l'on veut pour autant qu'ils soient cohérents avec

¹⁷ [POO1988]

les faits et entre eux. Ici aussi, la notion d'extension d'une théorie existe, elle est définie par Poole comme suit :

Définition 12 : (extension)

une extension de (F, δ) est un ensemble de conséquences logiques d'un scénario maximal (par rapport à l'ensemble d'inclusion) de (F, δ) .

Illustrons cette théorie par un exemple.

Exemple 22

$$1. A\text{-un-motif}(x) \rightarrow \text{Suspect}(x)$$

$$2. (A\text{-un-motif}(x) \wedge A\text{-un-alibi}(x)) \rightarrow \neg \text{Suspect}(x)$$

$$F = \{A\text{-un-motif}(\text{John}), A\text{-un-motif}(\text{Bill}), A\text{-un-alibi}(\text{Bill})\}$$

$$\delta = \{1, 2\}$$

Avec cette théorie (F, δ) « $\text{Suspect}(\text{John})$ », « $\text{Suspect}(\text{Bill})$ » et « $\neg \text{Suspect}(\text{Bill})$ » peuvent être expliqués avec les scénarios suivants :

$$E1 = F \cup \{A\text{-un-motif}(\text{John}) \rightarrow \text{Suspect}(\text{John})\}$$

$$E2 = F \cup \{A\text{-un-motif}(\text{Bill}) \rightarrow \text{Suspect}(\text{Bill})\}$$

$$E3 = F \cup \{(A\text{-un-motif}(\text{Bill}) \wedge A\text{-un-alibi}(\text{Bill})) \rightarrow \neg \text{Suspect}(\text{Bill})\}$$

Si la notion de préférence est définie sur base du principe de spécificité, alors d'après la Définition 9 « $\text{Suspect}(\text{John})$ » est fortement dérivable de (F, δ) puisqu'il n'y a pas d'explication pour le contraire, et « $\neg \text{Suspect}(\text{Bill})$ » est aussi fortement dérivable parce que $E3$ est basé sur une information plus spécifique que $E2$.

Il est à remarquer que les défauts de la théorie de Poole ne sont pas implicitement quantifiés, ils servent uniquement de schémas pour leurs instances, comme c'est le cas pour la logique des défauts. La raison en est que si on avait utilisé la fermeture universelle des défauts dans la définition d'un scénario, un défaut ne pourrait plus être appliqué à des cas normaux dès qu'il y aurait une exception comme le montre l'exemple suivant :

$$F = \{Oiseau(Tweety), Oiseau(Jumbo), \neg Vole(Tweety)\}$$

$$\delta = \{\forall x (Oiseau(x) \rightarrow Vole(x))\}$$

Le défaut quantifié universellement ne peut être utilisé pour expliquer « $Vole(Jumbo)$ » puisqu'il implique également « $Vole(Tweety)$ » qui est opposé aux faits.

6.2 Théorie de Brewka

Brewka va généraliser la théorie de Poole en utilisant une *logique de défauts hiérarchisée* où l'ensemble des prémisses T est partitionné en niveaux (T_1, T_2, \dots, T_n) . Cette hiérarchie de prémisses fonctionne comme suit : une formule p appartenant à T_k est d'une priorité supérieure à (plus fiable que) une formule $q \in$ appartenant à T_j si $k < j$. Puisque dans toutes logiques non-monotones les faits d'une théorie doivent être vrais dans tous les modèles de cette théorie, il semble raisonnable de placer les faits au plus haut niveau, c'est-à-dire dans T_1 .

Cette hiérarchie sur les prémisses nous permet de construire un ensemble cohérent de prémisses en prenant d'abord un sous-ensemble cohérent maximal dans T_1 , en ajoutant ensuite autant de prémisses que possible de T_2 à condition de préserver la cohérence, etc. Au cours de ce processus, il est courant de rencontrer des formules en conflit au sein d'un même niveau, ce qui donne lieu des ensembles cohérents distincts, mutuellement incohérents (ces ensembles sont analogues aux extensions de la logique des défauts). Formellement, cela nous donne :

Définition 13 : (sous-théorie préférée de Brewka)

Soit $T = (T_1, T_2, \dots, T_n)$ une théorie de défauts hiérarchisée. $S = S_1 \cup S_2 \cup \dots \cup S_n$ est une sous-théorie préférée de T ssi pour tous k ($1 \leq k \leq n$) : $S_1 \cup \dots \cup S_k$ est un sous-ensemble de cohérence maximale de $T_1 \cup \dots \cup T_k$.

Illustrons cela en reprenant la théorie de l'Exemple 22 :

Nous allons répercuter la spécificité du défaut (2) en le mettant à un niveau supérieur au défaut (1):

$$T_1 = \{A\text{-un-motif}(John), A\text{-un-motif}(Bill), A\text{-un-alibi}(Bill)\} \quad (F)$$

$$T_2 = \{(A\text{-un-motif}(x) \wedge A\text{-un-alibi}(x)) \rightarrow \neg Suspect(x)\} \quad (2)$$

$$T_3 = \{A\text{-un-motif}(x) \rightarrow Suspect(x)\} \quad (1)$$

Nous allons essayer de construire une sous-théorie préférée.

Comme les faits sont cohérents, nous avons:

$$S_1 = \{A\text{-un-motif}(John), A\text{-un-motif}(Bill), A\text{-un-alibi}(Bill)\}$$

Remarque : ici aussi (1) et (2) sont des patrons pour leurs instances et non des formules implicitement universellement quantifiées.

Nous pouvons ensuite ajouter pour Bill une instance de (2) :

$$S_2 = \{ (A\text{-un-motif}(\text{Bill}) \wedge A\text{-un-alibi}(\text{Bill})) \rightarrow \neg \text{Suspect}(\text{Bill}) \}$$

Ensuite, comme $S_1 \cup S_2$ implique « $\neg \text{Suspect}(\text{Bill})$ » nous ne pouvons ajouter que le défaut (1) instancié par John :

$$S_3 = \{ A\text{-un-motif}(\text{John}) \rightarrow \text{Suspect}(\text{John}) \}$$

Ce qui nous donne comme sous-théorie préférée $S_1 \cup S_2 \cup S_3$ qui implique « $\text{Suspect}(\text{John})$ » et « $\neg \text{Suspect}(\text{Bill})$ ».

Problèmes des logiques non-monotones

7. Effort d'unification

Comme nous venons de le voir, il existe de nombreuses façons - concurrentes - de formaliser le raisonnement non-monotone. Ces dernières années, certains efforts d'unification ont été faits. Plusieurs chercheurs¹⁸ ont tenté de définir les contraintes minimales que devrait remplir une fonction - envoyant un ensemble de formules sur une formule - pour être appelée conséquence non-monotone. Mais jusqu'à présent, il n'y a encore aucune définition qui puisse s'appliquer à l'ensemble des définitions de conséquence des différentes logiques non-monotones.

Un autre effort d'unification s'est porté sur la recherche de relations formelles entre les différentes théories. Par exemple, Konolige¹⁹ a établi une traduction entre la logique des défauts et la logique autoépistémique afin d'obtenir des résultats équivalents des deux cotés. Konolige a montré que chaque théorie de la LAE possède une forme normale dans laquelle chaque phrase est de la forme :

¹⁸ [SHO1988] & [GAB1985] & [KRA1990]

¹⁹ [KON1988]

$$(L\phi \wedge \neg L\psi_1 \wedge \dots \wedge \neg L\psi_n) \rightarrow \chi$$

et peut être transformée en un défaut :

$$\phi : \neg\psi_1, \dots, \psi_n / \chi$$

de telle façon que pour toute *extension* de la théorie de défauts ainsi obtenue, il existe une *expansion* dont la partie non-modale est la même que l'*extension*. D'autres chercheurs ont trouvé des relations similaires entre différents formalismes du raisonnement non-monotone²⁰.

Mais malgré ces *efforts* d'unifications, on ne sait toujours pas si ces différentes logiques modélisent des aspects différents du raisonnement ou si elles sont simplement des rivales dans la formalisation d'un même raisonnement.

8. Problèmes d'implémentations

L'inconvénient majeur des logiques non-monotones est que, sans de sévères restrictions, elles ne peuvent être implémentées efficacement.

Une cause première de cette inefficacité d'implémentation réside dans le fait que les méthodes de preuves non-monotones²¹ sont globales; par opposition aux méthodes de preuves standards qui, elles, sont locales. Locales en ce sens que, si on trouve une preuve monotone pour une formule ϕ en consultant une partie de la BD, cette preuve est considérée comme émanant de la BD tout entière. Ce qui est normal, puisque aucune information supplémentaire ne pourra invalider cette preuve. Quant aux méthodes de preuves non-monotones, si elles trouvent une preuve pour ϕ , elles devront continuer à parcourir toute la BD puisqu'une nouvelle information peut invalider cette preuve. Donc, les méthodes de preuves globales doivent inspecter la BD tout entière pour chaque conclusion, ce qui les rend relativement lentes.

En plus de cela, les logiques non-monotones englobent généralement la logique des prédicats du premier ordre qui n'est que semi-décidable; autrement dit, l'indérivabilité d'une formule est indécidable. Or en logiques non-monotones, comme la validité d'une conclusion dépend de l'indérivabilité d'autres formules, même si la bonne réponse

²⁰ [LIF1987], [GEL1989], [GEL1990] & [POO1988]

²¹ Seul la circonscription n'utilise pas de test de cohérence, mais son problème est qu'elle est basée sur la logique du second ordre qui a un système de preuve incomplet.

attendue est « oui », l'algorithme ne peut garantir qu'il la trouvera puisque pour cela, il devra répondre à des questions dont la réponse est « non »! Donc, les logiques non-monotones ne sont même pas semi-décidables.

Cela veut donc dire que, si on veut implémenter des systèmes raisonnant de façon non-monotone, il faudra soit restreindre les logiques pour qu'il existe un système de preuve décidable; soit permettre des approximations des résultats. Nous verrons un exemple de chacune de ces approches.

Il faut savoir que la plupart des méthodes d'implémentations d'inférences non-monotones furent d'abord développées sans aucune référence à une logique particulière. Ce n'est que plus tard, que leurs relations avec les formalismes non-monotones existants furent constatées et analysées.

9. Systèmes à processus non-monotone

Dans ces systèmes, la notion de cohérence est approximée par un résolveur de problèmes qui considère une formule comme cohérente quand sa négation n'appartient pas à l'ensemble courant de croyances. Cet ensemble courant de croyances est déterminé par un sous-système : un système de maintenance de cohérence.

Un système de maintenance de cohérence enregistre toutes les dépendances logiques entre les formules. Ainsi, lorsqu'une conclusion non-monotone est invalidée, le système peut retirer, de l'ensemble courant de croyances, non seulement cette conclusion, mais aussi toutes celles qui en ont été dérivées. En dehors de cela, ce type de systèmes est également utile pour d'autres tâches liées au raisonnement classique ou non-monotone. Il peut servir à faire du raisonnement hypothétique (c'est-à-dire observer ce qui se passe lorsque l'on change les prémisses) ou détecter les sources d'incohérences, ce qui est indispensable pour les systèmes devant gérer une information incohérente.

Concrètement, le résolveur de problèmes travaille sur une base de connaissances composée de défauts de Reiter propositionnels. Ces défauts de Reiter sont en fait présentés sous la variante syntaxique suivante :

$$B_1 \wedge \dots \wedge B_N : \neg E_1, \dots, \neg E_N / C$$

devient

$$\textit{si } B_1, \dots, B_N \textit{ alors } C \textit{ sauf si } E_1, \dots, E_N$$

Le résolveur n'applique une règle que quand ses conditions-*si* ont été établies et qu'aucune de ses conditions-*sauf si* n'ait été dérivée jusque là. Ceci est plus faible que

ce qui est exigé pour les logiques non-monotones, c'est pourquoi le système de maintenance de cohérence est nécessaire.

Le résolveur crée une justification à chaque fois qu'une règle est appliquée. Une justification est juste un ensemble noeuds, c'est-à-dire de noms représentant une formule particulière. L'ensemble courant de croyances est alors déterminé en étiquetant les noeuds du réseau de justifications avec IN ou OUT.

La notation utilisée pour les justifications est la suivante :

$$\langle A_1, \dots, A_N / B_1, \dots, B_N \rightarrow C \rangle$$

où A_i , B_j et C sont des noeuds. Si le réseau contient cette justification et que tous les A_i sont dans IN et tous les B_j sont dans OUT, alors C est étiqueté IN. En plus, un noeud ne peut être étiqueté IN que s'il a une dérivation valide, non-circulaire basée sur les justifications disponibles. Si ces deux conditions sont remplies, on dit que l'étiquetage est admissible et les noeuds qui y sont étiquetés correspondent à l'extension de la théorie de défauts correspondante.

Exemple 23.

Supposons les justifications suivantes :

1. $\langle / \rightarrow \text{Oiseau-Tw} \rangle$
2. $\langle \text{Oiseau-Tw} / \text{Non-Vole-Tw} \rightarrow \text{Vole-Tw} \rangle$
3. $\langle \text{Pingouin-Tw} / \rightarrow \text{Non-Vole-Tw} \rangle$

Il n'y a qu'un seul étiquetage admissible, celui où Oiseau-Tw et Vole-Tw sont IN et les autres OUT. Supposons que pendant l'exploration de la base de connaissances le résolveur de problèmes tombe sur une règle disant que Tweety vole, alors la justification suivante est ajoutée:

4. $\langle / \rightarrow \text{Pingouin-Tw} \rangle$

Le système de maintenance de cohérence doit recalculer l'étiquetage admissible; ce qui donne : Pingouin-Tw, Non-Vole-Tw et Oiseau-Tw dans IN et le reste dans OUT.

En fait, nous voyons que - si les théories logiques sont finies et décidables - le comportement d'un système composé d'un résolveur de problèmes et d'un système de maintenance de cohérence peut être considéré comme une approximation incrémentale d'une « extension » de la base de connaissances du résolveur.

10. Système prouveur pour la logique de défauts

A la seule condition que les théories logiques soient finies et décidables, le système de maintenance de cohérence peut être utilisé comme un prouveur de théorèmes pour la logique des défauts (et donc la LAE). Dans ce qui suit, nous n'allons aborder que le cas de la logique des défauts.

Supposons une théorie de défauts (F, D) . On peut transformer les défauts de D en justifications; le défaut :

$$A : B_1, \dots, B_n / C$$

devient

$$\langle A / \neg B_1, \dots, \neg B_n \rightarrow C \rangle$$

Mais il faut encore exprimer les relations logiques entre prérequis, conditions de cohérence et conséquence des défauts, étant donné les faits F . Pour cela, il suffit de chercher des ensembles minimaux de conséquences des défauts de D qui, avec F , peut être utilisés pour dériver des préconditions ou des négations de conditions de cohérence de défauts de D . Pour chacun de ces ensembles de conséquences, une justification correspondante peut être créée. Ce qui veut dire que pour (F, D) :

$$J_1 := \{ \langle A / \neg B_1, \dots, \neg B_n \rightarrow C \rangle / A : B_1, \dots, B_n / C \in D \}$$

et

$$J_2 := \{ \langle A_1, \dots, A_m / \rightarrow C \rangle / A_i = \text{conséquence d'un défaut de } D;$$

$C = \text{prérequis ou condition de cohérence de } D;$

$$F \cup \{A_1, \dots, A_m\} \vdash C$$

$$S \subset \{A_1, \dots, A_m\} \Rightarrow W \cup S \perp C$$

J_2 n'est calculable avec un prouveur de théorèmes que si l'ensemble des formules sur lequel on travaille est un sous-ensemble décidable de la logique du premier ordre. Sous ces conditions, Junker et Konolige²² ont montré qu'il existait une bijection f entre l'ensemble des extensions de (F, D) et l'ensemble des étiquetages admissibles de $J_1 \cup J_2$ tel que pour toute extension E :

$$E = Th (W \cup \{p / f(E)(p) = IN\})$$

²² [JUN1989]

11. Conclusion

Comme nous avons pu le voir, il n'y a pas que les types de raisonnements traditionnels qui peuvent être étudiés en logique. Dire que les raisonnements portant sur des règles sujettes à des exceptions ou sur de l'information incohérente et/ou incomplète sortent du cadre de la logique n'est plus pensable. Et puisque - à cause de la nature imprévisible et ouverte du monde auquel la loi s'applique, des intérêts contradictoires et de l'évolution des opinions - ce type d'informations est abondant dans le domaine juridique, une analyse logique du raisonnement juridique ne peut plus être considérée comme d'importance mineure.

Margés le gros problème d'efficacité dont souffrent les logiques non-monotones, comme le dit Etherington²³, les théories formelles sur le raisonnement non-monotone restent intéressantes en ce qu'elles servent de norme de bon comportement pour les systèmes informatiques implémentant du raisonnement non-monotone. De tels systèmes peuvent résoudre le problème d'efficacité en restreignant le langage pour qu'il ait un système de preuve efficace ou en implémentant une approximation de la logique qui occasionnera des erreurs au sein du système. Les théories formelles peuvent donc servir de mesure pour l'évaluation et la critique du comportement de tels systèmes grâce à leur définition d'inférence valide.

²³ [ETH1988]

Chapitre 3 : Théorie de Prakken et améliorations

Un résultat de l'étude du raisonnement de sens commun (c'est-à-dire raisonner sur une information incomplète ou incohérente) par l'IA fut l'émergence de la logique non-monotone. Une caractéristique de ce raisonnement est qu'il faut souvent faire un choix entre différentes façons incohérentes de compléter l'information, ou de la rendre cohérente. Afin de permettre un choix, beaucoup de logiques non-monotones ont été étendues avec de l'information sur des priorités. Mais on a souvent voulu croire¹ que ces priorités étaient générales et d'une nature indépendante du domaine étudié. Dès lors, les questions de priorités se plaçaient à un niveau métalogique.

Pour Prakken² cette « croyance » est irréaliste, dans la plupart des domaines de raisonnement de sens commun, comme le droit ou la bureaucratie, les principes de résolution de conflits font eux-mêmes partie du domaine de la théorie. En plus, non seulement ils varient de domaine en domaine, mais ils peuvent être incomplets et incohérents comme les informations du domaine peuvent l'être. C'est-à-dire que le raisonnement sur les priorités est lui-aussi non-monotone.

Dans un premier temps, nous allons présenter la théorie de Prakken à propos du raisonnement avec et sur les priorités; par après, nous tenterons de pousser plus loin cette théorie.

Théorie de Prakken

Nous allons nous intéresser dans ce point à l'étude d'un raisonnement portant sur des relations de préférence. Nous verrons la méthode Prakken permettant d'étendre certaines logiques en exprimant sous forme de prémisses de l'information à propos de priorités, en dérivant des priorités en conclusions et en utilisant ces conclusions pour tirer d'autres conclusions.

¹ [BRE1989], [GEF1992] & [SIM1992]

Dans ces travaux, les informations sur les priorités sont exprimées séparément du reste des prémisses et ne sont sujettes qu'à une forme de raisonnement très limité (les nouvelles priorités ne peuvent être dérivées que de manière déductive à partir des axiomes d'ordre uniquement; par exemple les axiomes de transitivité et de réflexivité dans le cas d'un préordre partiel).

² [PRA1995]

1. Approche basée sur l'extension

Le type de raisonnement dont nous parlons est souvent utilisé en droit, notamment quand il s'agit d'utiliser des règles de résolution de conflits entre certaines règles juridiques. Voyons cela au travers d'un exemple; dans le système juridique hollandais on trouve par exemple dans la section 5 de la loi sur les provisions générales (AB), entre autres, le principe disant : « la présente règle invalide les règles antérieures »; on trouve ensuite dans la section 1-(2) du code pénal (Sr) un principe disant que si des changements surviennent lors du traitement d'une affaire criminelle, il faut appliquer la loi qui est la plus favorable au suspect; on trouve aussi dans la section 1637c du code civil (CW) le principe disant que les lois sur les contrats de travail ont priorité sur les lois portant sur n'importe quel autre type de contrat.

Pour appliquer ces principes il faut avoir une information autre que celle des priorités. Pour appliquer 1637c CW il faut avoir des informations sur le type de contrat dont il est question; pour utiliser 1-(2) Sr il faut savoir quelle est la règle la plus favorable au suspect. En plus de cela, ces règles de priorité peuvent être en conflit : supposons deux règles, l'une, la plus ancienne, portant sur n'importe quel type de contrats (sauf les contrats de travail) et l'autre, la plus récente, portant sur les contrats de travail, dans ce cas, 1637c BW et 5 AB sont en conflit. Pour résoudre un tel type de conflit, les juristes font appel à d'autres règles de résolution de conflit basées sur la structure hiérarchique du système juridique ou sur la spécificité. On voit donc que des règles de résolution de conflits peuvent aussi bien être appliquées à des règles de droit « normales » qu'à d'autres règles de résolutions de conflits.

Nous allons alors voir comment exprimer les informations sur les priorités de la même façon que les informations du domaine que l'on traite; comment exprimer les différentes relations possibles entre les différentes règles de résolution de conflits et comment utiliser les priorités pour résoudre des conflits à propos des priorités.

1.1 Méthode générale

Les logiques qui raisonnent sur de l'information sur laquelle porte des relations de priorité (appelées LIP) reçoivent en entrée une paire $(P, (O, \leq))$, où P comprend les prémisses³ du problème et O est un ensemble ordonné par la relation d'ordre \leq ⁴. Cet

³ Un ensemble de prémisses sera un ensemble de formules exprimées dans un langage comprenant la logique du premier ordre.

⁴ $x \leq y$ signifie « y est préféré à x ».

ordre est censé résoudre les conflits éventuels survenant entre les prémisses, il est d'ailleurs souvent défini sur une partie ou sur la totalité des prémisses, c'est à dire que $O \subseteq P$; nous dirons dorénavant que les LIP reçoivent en entrée une paire (P, \leq) . A partir de ces entrées, les LIP fournissent en résultat un ensemble d'extensions (déductivement fermées) où chaque extension représente une image cohérente du problème où les conflits qui existaient au sein des prémisses ont été résolus de manières différentes.

Dans beaucoup de LIP les prémisses P sont partitionnées en un sous-ensemble contenant les faits; c'est-à-dire l'information certaine, stricte (ce sont les prémisses présentes dans toutes les extensions, elles seront nommées par f_1, f_2, \dots, f_m) et en un sous-ensemble contenant les défauts; c'est-à-dire l'information invalidable (nommées par d_1, d_2, \dots, d_n). Il faut également que le langage des LIP soit assez expressif que pour traiter l'information à propos des priorités; remarquons que sera souvent le cas puisque la plupart des LIP comprennent la logique du premier ordre.

Il y a **deux approches** pour enrichir une LIP de façon à ce qu'elle puisse raisonner à propos de priorités. La **première** est de changer la métathéorie de la LIP de telle façon que les priorités exigées par la métathéorie soient une conséquence de la LIP elle-même. Si la métathéorie d'une LIP veut que pour deux objets o_1 et o_2 on ait l'information : $o_1 \leq o_2$, elle dira que $o_1 \leq o_2$ ssi $t_1 \preceq t_2$ est une conséquence des prémisses (où \preceq est le nom de \leq et t_1 et t_2 sont les noms de o_1 et o_2). Mais il faut aussi que la LIP enrichie fasse en sorte que les axiomes correspondant à l'ordre soient présents dans toutes les extensions : si la LIP d'origine exige que \leq^5 soit un préordre partiel, les axiomes suivants devront faire partie de toutes les extensions de la LIP enrichie :

$$f_1 : \forall x \forall y \forall z ((x \preceq y \wedge y \preceq x) \rightarrow x \preceq z) \quad (\text{transitivité})$$

$$f_2 : \forall x (x \preceq x) \quad (\text{réflexivité})$$

Par défaut, nous supposerons dans le reste de ce chapitre que les axiomes d'ordre appropriés, ainsi que les définitions habituelles de \prec^6 et de $\not\prec^7$ sont implicitement contenus dans toutes les extensions.

⁵ Si la LIP en utilisée partitionne P en prémisses strictes et invalidables, il faut alors que ces propriétés soient ajoutées aux prémisses strictes.

⁶ $x \prec y = (x \preceq y \wedge \neg(y \preceq x))$

⁷ $x \not\prec y = y \prec x$

La seconde approche, celle proposée par Prakken dans [PRA1995], se veut plus générale. Son but est de trouver un critère pour tester si les conséquences logiques de la LIP enrichie contiennent vraiment l'ordre approprié. Il va développer une notion indépendante spécifiant dans quelles conditions une extension est fidèle à ce qu'elle dit à propos l'ordre.

Soit un ensemble de prémisses P , on dira qu'une extension est fidèle par rapport à un ordre \leq ssi ce qu'elle dit à propos de \leq est vrai, c'est-à-dire ssi l'extension est créée avec \leq ou encore ssi l'extension est une extension de (P, \leq) .

Exemple 24

Soit les prémisses invalidables suivantes⁸ :

d_1 : si a alors b

d_2 : si c alors $\neg b$

et les prémisses strictes :

f_1 : $a \wedge c$

f_2 : $d_1 \prec d_2$

Sans relation d'ordre entre d_1 et d_2 , nous aurions les extensions suivantes :

E_1 : $Th\{a, c, b, d_1 \prec d_2\}$

E_2 : $Th\{a, c, \neg b, d_1 \prec d_2\}$

Mais comme nous voulons que $d_1 \prec d_2$ signifie que « d_1 est inférieur à d_2 », il y a un problème avec E_1 qui dit que d_1 est inférieur à d_2 tout en contenant la conséquence de d_1 sans contenir celle de d_2 . Elle a donc été « créée » en préférant d_2 à d_1 . Comme E_1 est en contradiction avec l'ordre qui lui a donné le jour, nous dirons que E_1 n'est pas fidèle. Par contre E_2 est bien une extension fidèle.

Nous allons maintenant formaliser cette notion d'extension fidèle d'une LIP, elle est basée sur la notion de description d'un ordre.

⁸ Dans cet exemple, les lettres a , b et c sont des metavariables pour des formules de la LIP utilisée. Ce qui se trouve à gauche des « : » est le nom de la formule qui se trouve à droite.

Définition 14 : (description d'un ordre)⁹

Soit L le langage d'une LIP. Pour tout ordre \leq sur des objets o_1, \dots, o_n appelés t_1, \dots, t_n , dans L , la fonction $f(\leq)$ est définie comme :

$$\{t_i \preceq t_j \mid (o_i, o_j) \in \leq\} \cup \{t_i \not\preceq t_j \mid (o_i, o_j) \notin \leq\}.$$

On définit alors la notion d'extension fidèle : une extension est fidèle si elle est cohérente avec la description de l'ordre avec lequel elle peut être construite.

Définition 15 : (extension fidèle - noté E.F. -)⁹

Soit L une LIP. E est une L-E.F. d'un ensemble P de formules de L ssi pour un ordre \leq :

1. E est une L-extension de (P, \leq) ; et
2. \leq est compatible avec E , c'est-à-dire $f(\leq) \cup E$ est cohérent (ou encore $\leq \models E$).

Mais les extensions doivent posséder une propriété supplémentaire, elles ne doivent pas contenir d'information infinie à propos des priorités.

Définition 16 : (extension définie)⁹

Une extension est définie sur des priorités ssi l'ensemble des ordres compatibles avec l'extension a un plus petit élément (par rapport à l'inclusion de relations);

Cette méthode d'enrichissement de LIP peut s'appliquer à différentes LIP existantes, par exemple à une logique à préférence de modèles - comme la circonscription que nous ne verrons pas ici¹⁰ - ou à une logique à base de cohérence comme la logique de défauts que nous allons examiner au point suivant.

1.2 Extension de la logique de défauts avec priorités de Brewka**1.2.1 Théorie de défauts avec priorités**

La logique de défauts avec priorités de Brewka (LDP) est une version de sa logique de défauts à laquelle des priorités ont été ajoutées. Les défauts normaux comme $a : b / b$ seront, dans le reste du texte, représentés par $a \Rightarrow b$.

⁹ Tiré de [PRA1995]

¹⁰ Pour plus d'information, voir [PRA1995].

Définition 17 : (théorie de défauts avec priorités)¹¹

L'entrée d'une LDP est une théorie de défauts avec priorités : c'est-à-dire un triplet $(F, D, <)$, où F est un ensemble de formules du premier ordre; D est un ensemble fini de défauts normaux et $<$ est un ordre strict sur D .

Définition 18 : (défaut actif)⁸

Soit E un ensemble de formules. Un défaut $a \Rightarrow b$ est actif dans E ssi $a \in E$, $b \notin E$ et $\neg b \notin E$.

Définition 19 : (extension « prioritisée »)⁸

Soit $\Delta = (F, D, <)$ une théorie de défauts avec priorités et \ll un ordre total strict contenant $<$. E est l'extension « prioritisée » de Δ , générée par \ll ssi $E = \cup E_i$, où

$E_0 := \text{Th}(F)$ et

$E_{i+1} := E_i$

si aucun défaut n'est actif dans E_i ;

$\text{Th}(E_i \cup \{b\})$

dans le cas contraire, où b est le conséquent du défaut \ll -maximal actif dans E_i .

Définition 20 : (LDP-extension)⁸

E est une LDP-extension de $\Delta = (D, F, <)$ ssi il existe un ordre total strict contenant $<$ qui génère E .

1.2.2 Extension d'une théorie de défauts avec priorités

Prakken veut étendre une LDP ainsi définie pour qu'elle puisse raisonner sur des priorités. L'idée est que les priorités doivent être dérivées dans la logique, à partir des faits et des défauts qui sont fournis en entrée. Il n'est alors pas nécessaire de faire référence à un ordre lors de la définition des entrées d'une LDP étendue (noté LDPE) : l'entrée est une théorie de défauts c'est-à-dire une paire (F, D) (où F et D sont définis comme dans une LDP). $<$ sera appelé par \prec et les défauts par les termes d_1, \dots, d_n . On supposera enfin que F contient les axiomes de transitivité et d'irréflexivité pour \prec ; cela garantit que $<$ est un ordre partiel strict.

$$\forall x \forall y \forall z (x \prec y \wedge y \prec z) \rightarrow x \prec z \quad (\textit{transitivité})$$

$$\forall x (x \not\prec x) \quad (\textit{irréflexivité})$$

Les versions de LDP qui contiennent ces axiomes dans F seront appelées LPD^+ . On étend alors le système de Brewka en modifiant la définition de LPD^+ comme suit :

¹¹ Tirée de [BRE1994]

Définition 21 : (LDPE)

Une LDPE est une LDP⁺ aux modifications suivantes près : dans la Définition 19, la seconde condition de E_{i+1} est éclatée en deux :

- $E_{i+1} = E_i$ si aucun défaut n'est actif dans E_i ;
- $E_{i+1} = \text{Th}(E_i \cup \{b\})$ dans le cas contraire, où b est le conséquent d'un défaut d actif dans E_i et tel que pour tout d' actif dans E_i : E ne comprend pas $d \ll d'$.

Définition 22 : (extension-LDPE)

Un ensemble de formules est une extension-LDPE de (F, D) ssi pour un certain ordre $<$ cette ensemble est une extension-PDL fidèle de $(F, D, <)$.

La Définition 21 n'exige pas que $<$ soit un ordre partiel strict, dès lors, si on admet le symbole d'ordre \preceq dans le langage et que l'on remplace les axiomes d'un ordre partiel strict dans F par ceux d'un préordre partiel, la LDPE peut aussi traiter les préordres partiels.

Si on reformule la Définition 19 comme ceci :

Définition 23 : ()

Soit $\Delta = (F, D, <)$ une théorie de défauts avec priorités et \ll un ordre total strict contenant $<$. E est l'extension « prioritisée » de Δ générée par \ll ssi $E = \cup E_i$, où

$E_0 := \text{Th}(F)$ et

$E_{i+1} = E_i$ si aucun défaut n'est actif dans E_i ;

$\text{Th}(E_i \cup \{b\})$ dans le cas contraire, où b est le conséquent d'un défaut d actif dans E_i et tel que pour tout d' actif dans E_i : $\neg(d \ll d')$.

la LDP peut aussi traiter les préordres partiels.

Exemple 25.

Soit $\Delta = (F, D)$ où $F = \emptyset$, $D = \{d_1, d_2, d_3\}$ et

$d_1 : T \Rightarrow a$

$d_2 : T \Rightarrow \neg a$

$d_3 : T \Rightarrow d_1 \not\ll d_2$

Appliquons la Définition 23. $E_0 = \text{Th}(\emptyset)$. Supposons que l'on ajoute d'abord la conséquence de d_3 : on obtient $E_1 = \text{Th}(\{d_1 \not\ll d_2\})$. On a alors le choix entre la conséquence de d_1 ou de d_2 . Dans le premier cas, on obtient $E_2 = \text{Th}(\{d_1 \not\ll d_2, a\})$ et dans le

second cas, $E_2' = Th(\{d_1 \prec d_2, -a\})$. Comme à la troisième étape il n'y a plus de défauts actifs, il faut se demander si E_2 ou/et E_2' sont des LDPE-extensions de : la réponse est oui dans les deux cas.

Brewka¹² a montré que l'existence d'une extension était garantie dans une LDP, mais comme l'ordre utilisé, pour « construire » les E.F., dans les LDPE est lui-même déterminé par l'extension, l'existence d'une E.F. n'est, dans ce cas, pas garantie. Voici un exemple de théorie sans E.F..

Exemple 26.

Soit $\Delta = (F, D)$, $F = \emptyset$; $D = \{d_1, d_2\}$ et

$d_1 = T \Rightarrow d_1 \prec d_2$

$d_2 = T \Rightarrow d_2 \prec d_1$

Il y a deux candidats à l'E.F.

$E_1 : Th(F \cup \{d_1 \prec d_2\})$

$E_2 : Th(F \cup \{d_2 \prec d_1\})$

Examinons d'abord E_1 : il faut trouver un ordre $<$ dans lequel E_1 est vrai et tel que E_1 est une extension de $(\Delta, <)$. Le seul candidat est $d_1 \prec d_2$. Toutefois, E_1 n'est pas une extension de $(\Delta, \{d_1 \prec d_2\})$; c'est plutôt une extension de $(\Delta, \{d_2 \prec d_1\})$, mais $\{d_2 \prec d_1\}$ n'est pas compatible avec E_1 . Donc, E_1 n'est pas une E.F.. On peut faire une analyse symétrique de E_2 pour en conclure que Δ n'a pas de E.F..

Nous verrons dans la section suivante de ce chapitre (Améliorations) la raison pour laquelle cet exemple n'a pas de E.F..

1.2.3 Principes juridiques de résolution de conflits

Nous allons maintenant appliquer cette théorie à des exemples juridiques pratiques, mais pour cela, il va falloir combiner plusieurs critères d'ordre.

Dans la plupart des systèmes juridiques - dans le nôtre en particulier - on trouve les trois critères généraux d'ordre suivants : « Lex Superior », basé sur la structure hiérarchique du système juridique; « Lex Posterior », basé sur la date de promulgation

¹² [BRE1994]

de la norme; et « Lex Specialis », qui prend en compte le principe de spécificité d'une norme par rapport à une autre. Il semble admis que le principe hiérarchique (H) prime, mais il n'y a pas de consensus en ce qui concerne les deux principes restants : la temporalité (T) et la spécificité (S). Dans le reste de cet exposé, nous supposerons que T prévaut sur S.

Voyons ce qui peut se passer sur base de H et T si deux règles (r_1 et r_2) sont en conflit.

H	$r_1 < r_2$			$r_1 = r_2$			$r_1 \# r_2$		
T	$r_1 < r_2$	$r_1 = r_2$	$r_1 > r_2$	$r_1 < r_2$	$r_1 = r_2$	$r_1 > r_2$	$r_1 < r_2$	$r_1 = r_2$	$r_1 > r_2$
\Rightarrow	$r_1 < r_2$	$r_1 < r_2$	$r_1 < r_2$	$r_1 < r_2$	$r_1 = r_2$	$r_1 > r_2$	$r_1 \# r_2$	$r_1 \# r_2$	$r_1 \# r_2$

Comme H l'emporte sur T, lorsque H préfère une règle, globalement cette règle est préférée à l'autre (colonnes 2, 3 et 4); mais quand deux règles sont issues de sources de même type, la décision est déléguée à T (colonnes 5, 6 et 7); il se peut également que deux règles en conflit soient issues de deux types différents de sources entre lesquels il n'existe pas de relation hiérarchique. Dans ce cas, le consensus dit que le conflit ne peut être résolu par T (colonnes 8, 9 et 10).

1.2.4 Formalisation des principes de résolution

Pour formaliser cela, Prakken¹³ se base sur deux hypothèses : premièrement, les différents critères d'ordonnancement juridique sont tous des opinions sur une même relation d'ordre; et deuxièmement, ils induisent un ordre strict. Supposons qu'on ait deux règles r_1 et r_2 issues de la même source, mais r_1 a été promulguée plus tard que r_2 ; si on formalise cela en utilisant les expressions $r_1 \approx r_2$ et $r_2 < r_1$, on aboutit à une contradiction logique : l'ordre $H > T$ produirait une E.F. avec $r_1 \approx r_2$ et T ne pourrait être utilisé pour résoudre le conflit, contrairement à ce qui devrait être intuitivement fait.

L'idée de la méthode de Prakken est que si deux règles sont intuitivement du même type hiérarchique, on ne formalise pas cette relation.

¹³ [PRA1995]

Exemple 27.

$H : \text{Inférieur}(x, y) \Rightarrow x < y$

$T : \text{Plus_tôt}(x, y) \Rightarrow x < y$

$h : \text{Loi}(x) \wedge \text{Constitution}(y) \Rightarrow \text{Inférieur}(x, y)$

$O : T < H$

Si on a les faits : $\text{Loi}(r_1) \wedge \text{Loi}(r_2) \wedge \text{Plus_tôt}(r_2, r_1)$ alors la conclusion est : $r_2 < r_1$.

L'incomparabilité de source est gérée en rendant les deux règles formellement incompatibles c'est-à-dire :

$H : \text{Incomparable}(x, y) \Rightarrow x \not< y \wedge y \not< x$

Comme cela contredit logiquement ce que dit T, l'application de T est bloquée comme cela était désiré.

Nous allons maintenant formaliser deux exemples tirés de la loi hollandaise. Mais avant, nous allons présenter la notation qui sera utilisée : chaque schéma de défaut est noté par une constante r . Les instanciations des schémas seront notées par le terme $inst(r, t_1, \dots, t_n)$ où t_1, \dots, t_n sont les termes survenant dans l'instanciation.

Exemple 28.

La section 5 de la loi sur les provisions générales (AB) exprime entre autres le principe suivant : « la présente règle invalide les règles antérieures »; alors que la section 1-(2) du code pénal (Sr) dit que si des changements surviennent lors du traitement d'une affaire criminelle, il faudra appliquer la loi qui est la plus favorable au suspect. Dans la loi hollandaise, 1 Sr est vue comme une exception à 5 AB. On formalise cela en utilisant la notation suivante : tout défaut contenant des variables libres sera noté $r(x_1, \dots, x_n)$ et chacune de ses instanciations avec les termes t_1, \dots, t_n sera noté $r(t_1, \dots, t_n)$.

c, s, r_1, r_2 sont des variables mais 5, 1, S et TS sont des symboles de fonctions.

D contient :

$5(r_1, r_2) : \text{Plus_tard}(r_1, r_2) \Rightarrow r_1 < r_2$

$1(r_1, r_2, c, s) : \text{Affaire_criminelle}(c) \wedge \text{Suspect}(c, s) \wedge \text{Changement_pendant}(r_1, c, r_2) \wedge \text{Plus_favorable_pour}(s, r_1, r_2) \Rightarrow r_2 < r_1$

$S(r_1, r_2) : \text{Plus_spécifique}(r_1, r_2) \Rightarrow r_2 \prec r_1$

$TS(r_1, r_2) : S(r_1, r_2) \prec 5(r_1, r_2)$

et F contient :

$\text{Plus_spécifique}(1(r_1, r_2, c, s), 5(r_1, r_2))$

$\forall C, r_1, r_2 (\text{Changement_pendant}(r_1, c, r_2) \rightarrow \text{Plus_tard}(r_1, r_2))$

Supposons que pensant une affaire criminelle « af_1 » avec un suspect « Tom » une section « i » Sr se change en « i' » qui est moins favorable à Tom. On ajoute donc à F :

$\text{Affaire_criminelle}(af_1)$

$\text{Suspect}(af_1, \text{Tom})$

$\text{Changement_pendant}(i, af_1, i')$

$\text{Plus_favorable_pour}(\text{tom}, i, i')$

Alors, la version instanciée de $S(r_1, r_2)$, c'est-à-dire :

$S(1(i, i', af_1, \text{Tom}), 5(i, i'))$

tire comme conclusion que :

$5(i, i') \prec S(1(i, i', af_1, \text{Tom}))$

C'est-à-dire que $1 Sr$ à priorité sur $5 AB$. Ce résultat fait que $i Sr$ est préféré à $i' Sr$.

Exemple 29.

Il existe également une section du code civil (CW), la section 1637c, qui dit que les lois sur les contrats de travail ont priorité sur les lois portant sur n'importe quel autre type de contrat. Soit la formalisation suivante de 1637c BW et 5 AB instanciées par une règle r_1 portant sur les contrats et une règle antérieure r_2 portant sur les contrats de travail (que nous noterons respectivement « Ctr » et « Travail »). La profondeur d'instanciation d'un schéma r sera notée par r, r', r'', \dots

$1637c : \text{Ctr}(r_1) \wedge \text{Travail}(r_2) \Rightarrow r_1 \prec r_2$

$5 : \text{Plus_tard}(r_1, r_2) \Rightarrow r_2 \prec r_1$

Si on suppose par la suite que 5 AB et 1637c BW sont eux-mêmes des règles sur les contrats (de travail), on peut les instancier eux-mêmes pour résoudre le conflit qui les oppose; si on suppose que 5 AB est plus ancien que 1637c BW :

$1637c' : \text{Ctr}(5) \wedge \text{Travail}(1637c) \Rightarrow 5 \prec 1637c$

5' : $Plus_tard(5, 1637c) \Rightarrow 1637c \prec 5$

Mais on peut continuer en ajoutant un « ' » à chaque terme et ainsi de suite de manière infinie. Il ne s'agit pas d'un cas d'autoréférence puisqu'une instance d'une règle se réfère à une autre de ses instances; en plus, il y a deux E.F. : l'une donnant priorité à toutes les instances de 1637c BW et une autre donnant priorité à toutes les instances de 5 AB. Le problème est de nature pratique, en effet, il semble que l'on soit entré dans une boucle infinie.

Nous venons donc de présenter la méthode générale de Prakken permettant aux logiques avec priorités de prendre en compte la formalisation d'un raisonnement portant sur des priorités. L'intérêt de cette technique est qu'elle permet de formaliser un très grand nombre d'exemples qui ne peuvent être formalisés avec les logiques existantes. Malheureusement, il reste un problème - inhérent au type de domaine étudié - d'implémentation dû aux différentes définitions de points fixes utilisées dans la méthode.

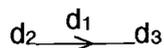
Dans la section suivante, nous allons nous placer à un niveau logique et tenter de trouver sous quelles conditions - dans cette méthode - l'existence d'une extension fidèle est garantie.

Améliorations

Nous allons nous tenter de voir, dans cette section, sous quelle(s) condition(s) l'existence d'une extension fidèle est garantie au sein d'un ensemble de défauts formant entre eux des boucles (circuits).

2. Représentation

Dans cette section, nous représenterons un défaut de type $d_1:d_2 \prec d_3$ par un graphe

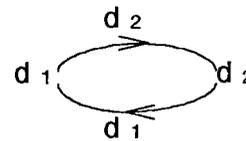


Exemple 30

$$d_1 : d_1 \prec d_2$$

$$d_2 : d_2 \prec d_1$$

\Leftrightarrow



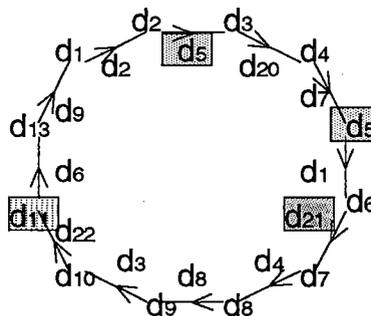
Nous aurons aussi besoin de représenter des ordres entre des défauts, nous utiliserons pour cela la convention suivante :

- « X / Y » signifie que défaut X est au-dessus du défaut Y dans l'ordre que l'on impose aux défauts.

3. Définitions

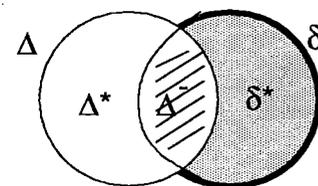
Comme le montre l'Exemple 30, un ensemble de défauts peut former graphiquement un circuit. Nous allons distinguer, au travers de l'Exemple 31, les différents types de défauts que l'on peut trouver dans un circuit. Nous avons encadré, dans cet exemple, un spécimen de chacun des types de défauts.

Exemple 31



On peut trouver des défauts qui portent sur d'autres défauts : ce sont des défauts arcs dans notre représentation (Δ dans la - Figure 7 -); on trouve aussi des défauts sur lesquels d'autres défauts portent : ce sont des défauts points dans notre représentation (δ dans la - Figure 7 -). Mais pour la suite de ce chapitre, il convient d'être plus précis. Nous distinguerons donc trois types de défauts :

- les défauts qui sont uniquement des arcs (Δ^*);
Ex : d20 , d21 et d22
- les défauts qui sont uniquement des points (δ^*);
Ex : d10 , d11 et d12



- Figure 7 -

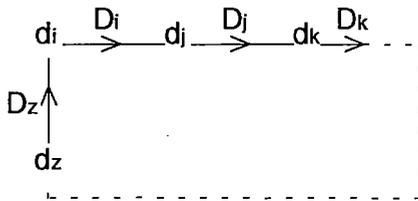
- et les défauts qui sont à la fois arcs et points (Δ).
Ex : d_1, d_2, \dots et d_9

Dans un premier temps, nous allons étudier les circuits les plus simples, à savoir ceux qui ne sont composés que de défauts arcs-points. Par la suite, nous envisagerons des circuits plus complexes, composés de défauts quelconques.

4. Circuits de défauts arcs-points

Propriété 1

Nous allons nous intéresser au cas où il y a n défauts formant un circuit entre eux, c'est-à-dire :



$$\text{Où } \{d_i \dots d_z\} = \{D_i \dots D_z\}$$

$$\text{et } \# \{d_i \dots d_z\} = n$$

Dans ce cas :

Il y a au moins une extension fidèle \Leftrightarrow 1) \exists un chemin de longueur $n-1$:

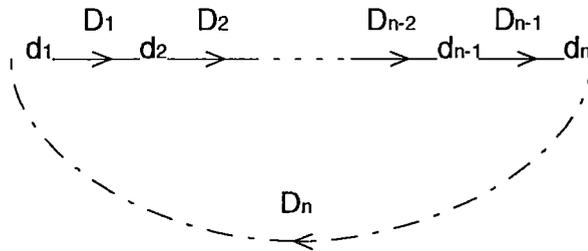


(ou encore $\Leftrightarrow \exists$ un défaut : $d_n \xrightarrow{d_n}$)

c'est-à-dire s'il existe un défaut qui prétend être supérieur.

$$2) \text{ et } \{d_1, d_2, \dots, d_{n-1}\} = \{D_1, D_2, \dots, D_{n-1}\}$$

Démonstration



On a $n-1$ défauts ($D_1 \dots D_{n-1}$) parmi n défauts formant un circuit \Rightarrow ils forment une extension et sont tel que :

$$\forall i \in [1 .. n-1] : D_i : d_{m_1} \succ d_{m_2} \text{ où } m_1 = m_2 - 1$$

$$\text{et } 1 \leq m_1 \leq n-1$$

$$\text{et } \{d_1, d_2, \dots, d_{n-1}\} = \{D_1, D_2, \dots, D_{n-1}\}$$

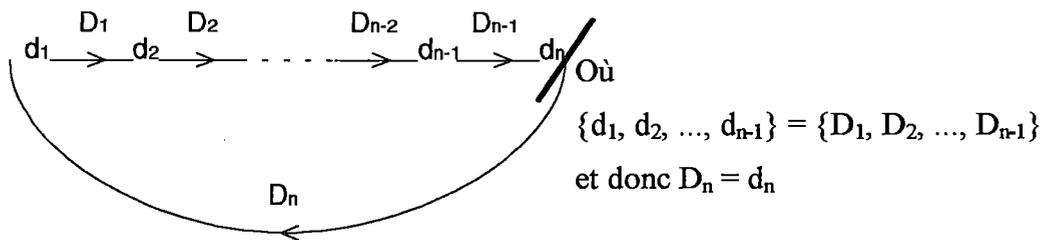
$$\text{et } d_n = D_n$$

C'est-à-dire qu'ils définissent un ordre $d_1/d_2/\dots/d_{n-1}/d_n$ où le défaut invalidé (c'est-à-dire celui qui est en contradiction avec les autres) est après tous ceux qui le contredisent, c'est-à-dire en dernière position \Rightarrow l'extension est fidèle.



On a une extension fidèle (elle est de longueur $n-1$ puisqu'elle est l'extension d'un circuit de n défauts); il existe donc un ordre $d_1/d_2/\dots/d_{n-1}/d_n$ tel que le défaut invalidé est après tous les défauts avec lesquels il est en contradiction, c'est-à-dire qu'il est en dernière position.

Cet ordre existe dans un circuit de n défauts tel que :



cqfd

Remarque : avec cette représentation et cette propriété, il est maintenant facile de voir combien ($0 \rightarrow n$) d'extensions fidèles possède un tel circuit de défauts : il suffit de

voir combien il y a de défauts prétendant être supérieurs ou autrement dit, combien il y a de défauts du type d_a : $d_a \succ d_b$.

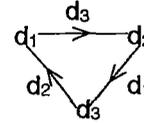
Exemple 32.

$$d_1:d_3 \prec d_2$$

$$d_2:d_1 \prec d_3$$

$$d_3:d_2 \prec d_1$$

\Leftrightarrow



D'après la propriété 1, cette théorie n'a pas d'extension fidèle.

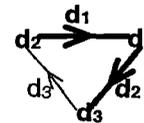
Exemple 33.

$$d_1:d_1 \prec d_2$$

$$d_2:d_3 \prec d_1$$

$$d_3:d_2 \prec d_3$$

\Leftrightarrow



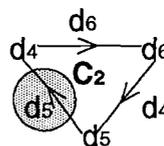
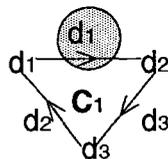
D'après la propriété 1, Le défaut d_3 est du type permettant une extension fidèle induite par l'ordre $d_2 / d_1 / d_3$, cette extension fidèle est : $\{d_3 \prec d_1, d_1 \prec d_2\}$.

Dans la suite de ce chapitre, nous appellerons les défauts qui par leur invalidation permettent d'obtenir une extension fidèle (E.F.) des *candidats* (à une E.F.).

Propriété 2

Dans le cas de N défauts formant C circuits sans aucun défaut en commun :
 chacun des C circuits a au moins une E.F. \Leftrightarrow
 l'ensemble des N défauts a au moins une E.F..

Exemple 34.



Remarque :

nous représenterons dorénavant les défauts invalidés en gras dans le texte et par un cercle grisé dans les graphes.

Dans C_1

d_1 est candidat et produit une extension $\{d_1 < d_3, d_3 < d_2\}$ fidèle à l'ordre $d_2/d_3/d_1$ ce qui sera dorénavant noté :

$$d_1 : d_2/d_3/d_1 \Rightarrow E.F. : \{d_1 < d_3, d_3 < d_2\}$$

Dans C_2

$$d_5 : d_4/d_6/d_5 \Rightarrow E.F. : \{d_5 < d_6, d_6 < d_4\}$$

L'ensemble composé des défauts de $C_1 \cup C_2$ a une E.F. : $\{d_1 < d_3, d_3 < d_2, d_5 < d_6, d_6 < d_4\}$ induite par l'ordre fidèle $d_2/d_3/d_1/d_4/d_6/d_5$ ou de tout autre ordre où les ordres $d_2/d_3/d_1$ et $d_4/d_6/d_5$ sont respectés, comme par exemple dans : $d_4/d_2/d_3/d_6/d_5/d_1$.

Démonstration



Faisons la démonstration par induction sur le nombre de circuits (C).

1. $C = 1$

Par hypothèse, ce circuit de N défauts a au moins une E.F. \Rightarrow l'ensemble des N défauts a au moins une E.F. qui est celle des N défauts.

2. $C > 1$

Par hypothèse, chaque circuit a au moins une E.F. et par hypothèse d'induction, les M ($M < N$) défauts des C-1 premiers circuits ont une E.F..

Le circuit numéro C a une E.F. \Rightarrow par la propriété 1, il existe un ordre fidèle $d_{c1}/d_{c2}/\dots/d_{ck}$ tel que $\{d_{c1}, d_{c2}, \dots, d_{ck}\}$ est l'E.F..

$\{d_i : d_i \in \text{aux C-1 premiers circuits}\} \cap \{d_{c1}, d_{c2}, \dots, d_{ck}\} = \emptyset$ car dans le type de circuits envisagé, un défaut d'un circuit ne porte que sur des défauts du circuit auquel il appartient et par hypothèse, il n'y a pas de point commun entre les circuits.

Donc, $\forall d_{ci}$ où $1 \leq i \leq k$: d_{ci} n'est contredit par aucun des défauts (ou chaîne de défauts) des C-1 premiers autres circuits.

Comme $\{d_{c1}, d_{c2}, \dots, d_{ck-1}\}$ est une E.F. : $\{d_j : d_j \in \text{à une E.F. de l'ensemble des C-1 circuits}\} \cup \{d_{c1}, d_{c2}, \dots, d_{ck-1}\}$ est une extension pour l'ensemble des C circuits car :

- si on ajoutait à $\{d_j : d_j \in \text{à une E.F. de l'ensemble des C-1 circuits}\} \cup \{d_{c1}, d_{c2}, \dots, d_{ck-1}\}$ n'importe quel défaut, appartenant à un des C-1 premiers circuits, n'appartenant pas à l'E.F. de l'ensemble des défauts des C-1 circuits - c'est-à-dire n'importe quel défaut invalidé appartenant à un des C-1 premiers circuits - on obtiendrait une incohérence au sein de $\{d_j : d_j \in \text{à}$

une E.F. de l'ensemble des C-1 circuits} \cup $d_{\text{invalidé}}$ et au sein de. $\{d_j : d_j \in \text{à une E.F. de l'ensemble des C-1 circuits} \cup \{d_{c1}, d_{c2}, \dots, d_{ck-1}\} \cup d_{\text{invalidé}};$

- si on ajoutait d_{ck} à $\{d_j : d_j \in \text{à une E.F. de l'ensemble des C-1 circuits} \cup \{d_{c1}, d_{c2}, \dots, d_{ck-1}\}$ on obtiendrait une incohérence au sein de $\{d_{c1}, d_{c2}, \dots, d_{ck-1}\} \cup d_{ck}$ et donc au sein de $\{d_j : d_j \in \text{à une E.F. de l'ensemble des C-1 circuits} \cup \{d_{c1}, d_{c2}, \dots, d_{ck-1}\} \cup d_{ck}$.

Cette extension est fidèle à un ordre : $\underbrace{d_1 / d_2 / \dots / d_f}_{\alpha} / \underbrace{d_{f+1} / \dots / d_n}_{\beta}$

où $\{\alpha\}^{14} = \{\alpha'\} \cup \{d_{c1}, d_{c2}, \dots, d_{ck-1}\}$

où α' est l'ordre auquel l'E.F. des C-1 premiers circuits est fidèle;

où α respecte les deux ordres : α' et $d_{c1} / d_{c2} / \dots / d_{ck-1}$

où $\{\beta\} = \{\beta'\} \cup d_{ck}$

où β' est l'ordre des défauts invalidés par l'E.F. des C-1 premiers circuits.



Par hypothèse, nous savons qu'il y a une E.F. sur les N défauts. Or, pour qu'un ensemble de défauts soit une E.F. il faut :

- a) qu'il soit une extension;
- b) que l'extension soit fidèle à un ordre sur les N défauts.

- a) Un ensemble des N défauts possède une extension s'il est possible de trouver un ensemble cohérent maximal de défauts parmi ces N défauts.

Or, par hypothèse, il n'y a aucun défaut en commun entre les C circuits; un défaut quelconque d'un circuit quelconque ne peut être contredit que par un défaut de ce même circuit.

Donc, en unissant les extensions des différents circuits, on obtient bien un ensemble cohérent (E) de défauts. Cet ensemble est en plus maximal puisque l'ajout de n'importe quel autre défaut (appartenant forcément aux défauts invalidés d'un des circuits) rendrait un sous-ensemble de E incohérent (le sous-ensemble composé des défauts appartenant au circuit dont provient le défaut invalidé que l'on vient d'ajouter) et par conséquent E incohérent.

- b) On a donc un ordre $O^t = d_1 / d_2 / \dots / d_N$ auquel l'extension sur les N défauts est fidèle.

¹⁴ Nous utiliserons cet abus de langage car, en fait, nous devrions dire l'ensemble des défauts composant l'ordre α .

Soient $d_1 / d_2 / \dots / d_f$ les défauts faisant partie de l'E.F.
et $d_{f+1} / \dots / d_n$ les défauts invalidés par l'E.F..

Par définition, $\forall d_i$ où $1 \leq i \leq f$, on a $d_i : A \prec B$ vrai dans l'ordre $d_1 / \dots / d_n$
où A et B sont des défauts appartenant
au même circuit que d_i

Partitionnons les défauts de cet ordre selon les circuits auxquels ils appartiennent, nous obtenons ainsi C ordres (un par circuit) : soit $O^1 = d_1^1 / d_2^1 / \dots / d_g^1$ l'ordre qui résulte de la suppression dans O^t de tous les défauts qui n'appartiennent pas à C_1 .

Comme O^t est fidèle à une extension, O^1 est composé d'une suite de défauts valides suivis d'une suite de défauts invalidés (si ce n'était pas le cas, cela signifierait que O^t n'est pas fidèle, ou que les défauts de O^1 n'appartiennent pas à un circuit) ce qui est la définition d'un ordre fidèle à une extension.

On peut effectuer le même travail pour les autres circuits, on obtient alors des ordres fidèles pour tous les circuits.

cqfd

5. Circuits de défauts quelconques

Nous commencerons, comme dans la section précédente, par étudier le cas d'un circuit isolé, ensuite nous étudierons des cas où il y a plusieurs circuits disjoints pour en arriver enfin à un ensemble de circuits liés.

5.1 Circuit quelconque isolé

Remarques :

1. les défauts arcs seront également qualifiés de défauts *déplaçables* dans un ordre, car aucun défaut ne porte sur eux, on impose donc pas d'ordre sur eux si ce n'est :
 - s'il s'agit de défauts invalidés, ils doivent être au-dessous (à droite) des défauts avec lesquels ils sont en contradiction;
 - s'il s'agit de défauts valides, ils doivent être au-dessus (à gauche) des défauts invalidés avec lesquels ils sont en contradiction.

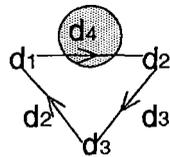
Pour les différencier des autres défauts, nous les soulignerons dans les ordres.

2. Dans un circuit quelconque isolé, le nombre de défauts points est le même que le nombre de défauts arcs. C'est très facile à voir : supposons un circuit à N

défauts (c'est-à-dire N points et N arcs), ce circuit pourrait comprendre uniquement M défauts arcs-points. Si on retire ces M points et M arcs du circuit, il reste $N-M$ points et $N-M$ arcs correspondant forcément à $N-M$ défauts arcs-points (= les $N-M$ arcs) et $N-M$ défauts points (les $N-M$ points).

Nous allons voir que dans de tels circuits, tous les défauts appartenant à D^* (D^* est l'ensemble des défauts de type Δ^* appartenant au graphe étudié) seront aussi candidats à une E.F..

Exemple 3.5.



$$d_4 : d_2 / d_3 / d_1 / \underline{d_4} \Rightarrow E.F.$$

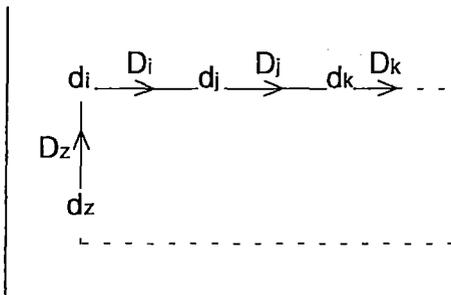
Les circuits auxquels nous nous intéressons maintenant ont M défauts $\in D^-$ (D^- est l'ensemble des défauts de type Δ^- appartenant au graphe étudié), L défauts $\in D^*$ et $L \in d^*$ (d^* est l'ensemble des défauts de type δ^* appartenant au graphe étudié).

Nous allons, dans un premier temps, étudier les cas où les défauts points ne se contredisent pas, c'est-à-dire qu'il n'existe pas d'ensemble de défauts points contredisant un autre ensemble de défauts points. Par la suite, nous envisagerons le cas général où de tels défauts peuvent être présents.

5.1.1 Circuit sans défauts points qui contredisent

Propriété 3

Nous avons un circuit :



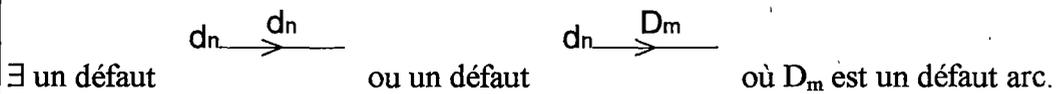
où $\{d_i \dots d_z\}$ comprend M défauts arcs-points et L défauts points (qui ne se contredisent pas);

$\{D_i \dots D_z\}$ comprend L défauts arcs et les mêmes M défauts arcs-points;

soit $n = M + L$

Dans ce cas :

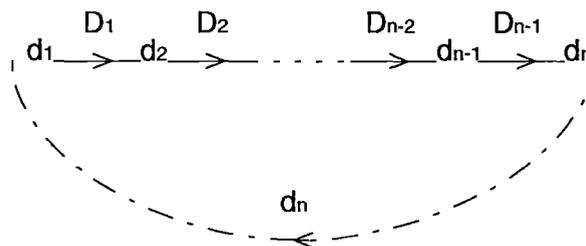
Il y a au moins une E.F. si



Démonstration

Par hypothèse, nous avons :

- un circuit :



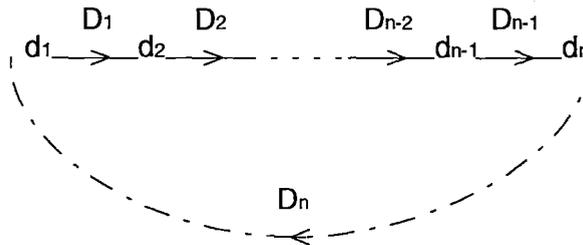
Si on prend l'ordre donné par $d_1 \xrightarrow{D_1} d_2 \xrightarrow{D_2} \dots \xrightarrow{D_{n-1}} d_n$,

on obtient un ordre, sur les défauts arcs-points et points, $O : d_1 / \dots / d_{n-1} / d_n$ où l'ensemble $\{D_i \mid 1 \leq i \leq n-1\}$ sont valides puisqu'ils ont servi à construire l'ordre. $\{d_i \mid 1 \leq i \leq n-1\}$ ne comprend lui aussi que des défauts valides puisqu'il est composé de défauts arcs-points dont on vient de dire qu'ils étaient valides et de défauts points qui par hypothèse ne peuvent être contredit par d'autres défauts points.

Donc, en ajoutant à O tous les défauts arcs de $\{D_i \mid 1 \leq i \leq n-1\}$, qui sont déplaçables, au-dessus (à gauche) de d_n avec lequel ils sont en contradiction,

on obtient un ordre O^t fidèle à l'E.F. composée de l'ensemble des défauts valides de O^t , c'est-à-dire tous les défauts sauf d_n .

- Ou un circuit :



Si on prend l'ordre donné par $d_1 \xrightarrow{D_1} d_2 \xrightarrow{D_2} \dots \xrightarrow{D_{n-2}} d_{n-1} \xrightarrow{D_{n-1}} d_n$,

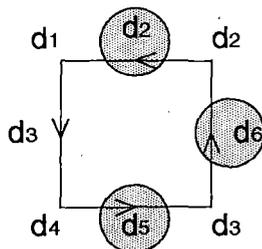
on obtient un ordre, sur les défauts arcs-points et points, $O : d_1 / \dots / d_{n-1} / d_n$ où l'ensemble $\{D_i \mid 1 \leq i \leq n-1\}$ sont valides puisqu'ils ont servi à construire l'ordre. $\{d_i \mid 1 \leq i \leq n\}$ ne comprend lui aussi que des défauts valides puisqu'il est composé de défauts arcs-points dont on vient de dire qu'ils étaient valides et de défauts points qui par hypothèse ne peuvent être contredit par d'autres défauts points.

Donc, si on place D_n au-dessous (à droite) de O et qu'ensuite on ajoute n'importe où dans O les défauts arcs de $\{D_i \mid 1 \leq i \leq n-1\}$ qui sont déplaçables. On obtient ainsi un ordre O^t où tous les défauts arcs et arcs-points valides se trouvent au-dessus de D_n avec lequel ils sont en contradiction. L'ordre O^t est donc fidèle à l'E.F. composée de l'ensemble des défauts valides de O^t , c'est-à-dire tous les défauts sauf D_n .

cqfd

Exemple 36

Soit la théorie de défaut suivante :



qui nous donne :

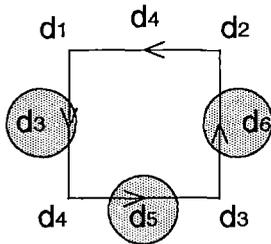
$$d_2 : d_1 / d_4 / d_3 / \underline{d_2} / \underline{d_6} / d_2 \Rightarrow E.F.$$

$$d_5 : d_3 / d_2 / d_1 / d_4 / \underline{d_6} / \underline{d_5} \Rightarrow E.F.$$

$$d_6 : d_2 / d_1 / d_4 / d_3 / \underline{d_6} / \underline{d_5} \Rightarrow E.F.$$

Exemple 37.

Soit la théorie suivante :



qui nous donne :

$$d_3 : \underline{d_5} / \underline{d_6} / d_4 / \underline{d_3} / d_2 / d_1 \Rightarrow E.F.$$

$$d_5 : d_3 / d_2 / d_1 / d_4 / \underline{d_6} / \underline{d_5} \Rightarrow E.F.$$

$$d_6 : d_2 / d_1 / d_4 / d_3 / \underline{d_6} / \underline{d_5} \Rightarrow E.F.$$

Dans cet exemple, d_3 ne correspond à aucun des deux types de défauts de la propriété que nous venons d'énoncer.

Propriété 4

Un ensemble de N défauts quelconques formant un circuit possède une E.F. si, l'ensemble des défauts arcs-points¹⁵ valides du circuit sont situés, dans l'ordre que l'on impose sur les points, au-dessus (à gauche) d'un défaut arc-point invalidé. Autrement dit : un ensemble de N défauts quelconques, formant un circuit, possède une E.F. si \exists ordre où un défaut $d_c \in D$ est tq tous les $d_v \in \{D \text{ moins } d_c\}$ sont au-dessus du défaut d_c avec lequel ils sont en contradiction.

Démonstration

La démonstration est évidente puisque d_c est au-dessous de tous les défauts arcs-points valides (c'est-à-dire en contradiction avec lui), si on place à droite tous les défauts arcs de ce circuits (ils sont déplaçables), on obtient un ordre où tous les défauts en contradiction avec d_c sont au-dessus de lui. Si on place ensuite les défauts points (qui sont valides par hypothèse) dans cet ordre - selon l'ordre défini par les défauts arcs et arcs-points valides - on obtient un ordre, sur l'ensemble des défauts du circuit, qui est fidèle à l'E.F. composée de tous les défauts du circuit sauf d_c .

cqfd

5.1.2 Les défauts points peuvent se contredire

Les défauts points, par définition, n'imposent aucun ordre sur les autres défauts, ni sur eux-mêmes. Ils ne peuvent, de plus, être invalidés que par - et seulement par - un

¹⁵ Ce sont des défauts non-déplaçables.

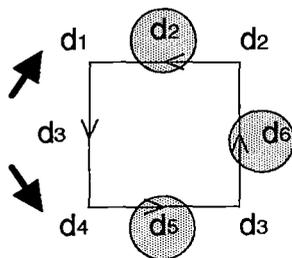
ensemble d'autres défauts points (et inversement, ils ne peuvent qu'invalider d'autres défauts points). Pour ces raisons, nous les considérerons comme passifs dans la détermination d'une E.F..

En effet, pour qu'un circuit avec de tels défauts ait la possibilité d'avoir une E.F., il faut qu'il ait une E.F. lorsque l'on considère ce même circuit sans tenir compte de la possibilité d'invalidation des défauts points (puisque les défauts points n'ont aucune influence sur les défauts arcs et arcs-points). Nous chercherons donc d'abord une E.F. dans le circuit sans tenir compte de la possible invalidation des défauts points, ensuite, nous vérifierons que l'E.F. ainsi obtenue a bien les ensembles de défauts points invalidés au-dessous des ensembles de défauts points en contradiction leur correspondant. Si c'est le cas, nous aurons bien une E.F., sinon, nous n'en aurons pas.

Voyons cela au travers d'un exemple.

Exemple 38.

Soient $d_1 : p$ et $d_4 : \neg p$.



Si on ne tient pas compte de la contradiction possible entre des défauts points :

$$d_5 : d_3 / d_2 / d_1 / d_4 / \underline{d_6} / \underline{d_5}$$

$$d_2 : d_1 / d_4 / d_3 / \underline{d_5} / \underline{d_6} / d_2$$

$$d_6 : d_2 / d_1 / d_4 / d_3 / \underline{d_5} / \underline{d_6}$$

Mais si on tient compte de la contradiction possible entre des défauts points :

$$d_5 : d_3 / d_2 / d_1 / \underline{d_4} / \underline{d_6} / \underline{d_5} \Rightarrow E.F.$$

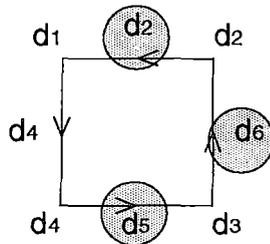
$$d_2 : d_1 / \underline{d_4} / d_3 / \underline{d_5} / \underline{d_6} / d_2 \Rightarrow E.F.$$

$$d_6 : d_2 / d_1 / \underline{d_4} / d_3 / \underline{d_5} / \underline{d_6} \Rightarrow E.F.$$

Il est évident qu'avec uniquement des paires de défauts points contradictoires, une E.F. trouvée sans en tenir compte donne une E.F. quand on prend en compte cette contradiction puisque le défaut invalidé sera forcément au-dessous de son compagnon valides (pour cela il suffit d'invalider le deuxième défaut de la paire apparaissant dans l'ordre fidèle).

Exemple 39.

Soient $d_1 : a$, $d_2 : b$, $d_3 : c$, et $d_4 : (a \vee b \vee c)$



Sans tenir compte de la contradiction entre les défauts points :

$$d_5 : d_3 / d_2 / d_1 / d_4 / \underline{d_6} / \underline{d_5}$$

$$d_6 : d_2 / d_1 / d_4 / \underline{d_5} / \underline{d_6} / d_3$$

$$d_2 : d_1 / d_4 / d_3 / \underline{d_5} / \underline{d_6} / d_2$$

En tenant compte de la contradiction entre les défauts points :

$$d_5 : d_3 / d_2 / d_1 / \underline{d_4} / \underline{d_6} / \underline{d_5} \Rightarrow E.F.$$

$$d_6 : d_2 / d_1 / \underline{d_4} / \underline{d_5} / \underline{d_6} / d_3 \Rightarrow \neg E.F. \text{ car } d_3 \text{ est au-dessous de } d_4$$

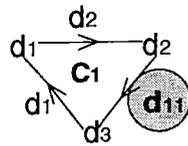
$$d_2 : d_1 / d_4 / d_3 / \underline{d_5} / \underline{d_6} / \underline{d_2} \Rightarrow E.F.$$

5.2 Ensemble de circuits quelconques disjoints¹⁶

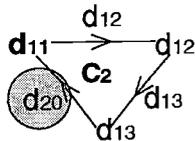
Nous allons maintenant étudier les E.F. de défauts quelconques formant plusieurs circuits sans aucun point en commun. Ici, pour que l'ensemble des défauts ait une E.F., il faut, par définition, que dans l'ordre que l'on impose sur l'ensemble des défauts, les défauts invalidés soient après tous les défauts avec lesquels ils sont en contradiction. Voyons ce que cela donne au travers de l'Exemple 40.

¹⁶ Nous n'allons pas envisager dans la suite du chapitre les cas où il y a des défauts points qui se contredisent!

Exemple 40.



$$(1) \ d_{11} : d_3 / d_1 / d_2 / \underline{d_{11}} \Rightarrow F.E.C1$$



$$(2) \ d_{20} : d_{11} / d_{12} / d_{13} / \underline{d_{20}} \Rightarrow F.E.C2$$

Il suffit donc, pour trouver une E.F. sur $C_1 \cup C_2$, de trouver un ordre respectant les deux ordres (1) et (2) puisque d_{11} n'est en contradiction qu'avec des défauts arcs de C_1 et d_{20} qu'avec des défauts arcs de C_2 . Nous obtenons donc l'ordre : $d_3 / d_1 / d_2 / \underline{d_{11}} / d_{12} / d_{13} / \underline{d_{20}}$ fidèle à l'extension : $\{d_3, d_1, d_2, d_{12}, d_{13}\}$.¹⁷

Lemme 1

Dans le cadre de circuits quelconques disjoints, un défaut arc d'un circuit quelconque, étant arc-point dans l'ensemble des défauts, ne peut être contredit par aucun ensemble de défauts appartenant à un autre circuit.

Raisonnement

Nous sommes donc en présence d'un défaut d qui est arc au sein d'un circuit C_1 et point au sein d'un autre circuit C_2 . Dans C_2 , d ne peut être contredit puisque par hypothèse les défauts points ne se contredisent pas.

Fin

Définition 24 : (ordres compatibles)

P ordres O^1, O^2, \dots, O^p sont compatibles $\Leftrightarrow \exists$ un ordre O^t dans lequel les ordres O^1, O^2, \dots et O^p sont respectés.

¹⁷ Il est à remarquer que d_{11} est déplaçable dans C_1 , mais pas dans $C_1 \cup C_2$.

Exemple 41.

$$O_1 : d_1 / d_2 / d_3$$

$$O_2 : d_6 / d_3 / d_5 / d_1$$

Dans O_1 on a d_1 avant d_3 par contre dans O_2 on a d_1 après d_3 , donc O_1 et O_2 ne peut pas être compatibles.

Propriété 5

Un ensemble de défauts quelconques formants c circuits disjoints possède au moins une E.F. \Leftrightarrow • chaque circuit pris séparément possède au moins une E.F.;

• et les ordres fidèles de ses E.F. sont compatibles.

Démonstration

Chaque circuit C_1, C_2, \dots, C_c possède au moins une E.F. qui est respectivement fidèle à O_1, O_2, \dots, O_c . Or par hypothèse, ces ordres sont compatibles c'est-à-dire qu'il existe un ordre O où les ordres fidèles O_1, O_2, \dots, O_c sont respectés; c'est-à-dire un ordre où chaque défaut invalidé est situé après tous les défauts qui lui sont opposés. L'ensemble des défauts des circuits C_1, C_2, \dots, C_c a donc une E.F. composée de l'ensemble des défauts valides dans l'ordre O .



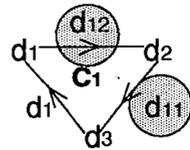
Par hypothèse, il existe un ordre O - sur l'ensemble des défauts des c circuits - impliquant une E.F. sur ces défauts (c'est-à-dire que dans cet ordre, chaque défaut invalidé est situé après l'ensemble des défauts qui lui sont opposés).

(1) Supposons qu'il existe un circuit C_j n'ayant pas d'E.F. c'est-à-dire que quel que soit l'ordre que l'on impose sur les défauts de C_j , le défaut invalidé n'est jamais après l'ensemble des défauts qui le contredisent. Donc, quel que soit l'ordre O imposé sur les défauts de C_j , le défaut invalidé de C_j ne se situe jamais au-dessous de l'ensemble des défauts avec lesquels il est en contradiction. Par conséquent, O ne peut pas être un ordre fidèle à une E.F. sur l'ensemble des défauts. Notre supposition de départ est donc irréalisable.

(2) Tous les circuits ont au moins une E.F., c'est-à-dire que chaque circuit respecte, respectivement, son ordre O_1, O_2, \dots et O_c . Mais supposons que ces ordres fidèles ne sont pas compatibles, c'est-à-dire qu'il n'existe pas d'ordre O sur les c circuits tel que chaque défaut invalidé se situe après l'ensemble des défauts qui lui sont opposés, ce qui est contraire aux hypothèses. Donc, la supposition de départ est fausse.

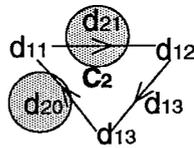
cqfd

Exemple 42



(1) $d_{11} : d_3 / d_1 / \underline{d_{12}} / \underline{d_{11}} / d_2 \Rightarrow$
E.F.C1

(2) $d_{12} : d_2 / d_3 / d_1 / \underline{d_{11}} / \underline{d_{12}} \Rightarrow$
E.F.C1



(3) $d_{20} : d_{11} / d_{12} / d_{13} / \underline{d_{21}} / \underline{d_{20}} \Rightarrow$

E.F.C2

(4) $d_{21} : d_{12} / d_{13} / d_{20} / \underline{d_{21}} / d_{11} \Rightarrow$ E.F.C2

Dans $C_1 \cup C_2$:

(1) et (3) sont incompatibles à cause de d_{11} et d_{12} .

(2) et (4) sont incompatibles à cause de d_{11} et d_{12} .

(1) et (4) donnent : $d_3 / d_1 / d_2 / d_{12} / d_{13} / \underline{d_{11}} / \underline{d_{20}} / \underline{d_{21}} \Rightarrow$ E.F.¹⁸

(2) et (3) donnent : $d_2 / d_3 / d_1 / \underline{d_{11}} / \underline{d_{12}} / d_{13} / \underline{d_{21}} / \underline{d_{20}} \Rightarrow$ E.F.

5.3 Ensemble de circuits quelconques liés

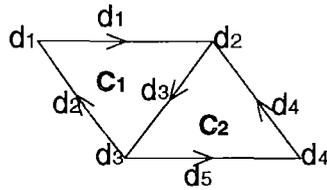
Dans le cadre de graphes de défauts quelconques composés de circuits non-disjoints, c'est-à-dire liés, nous distinguerons dans un premier temps deux types de liens : les points et les arcs¹⁹ pour étudier ensuite les graphes où ces deux types de liens apparaissent en même temps.

¹⁸ Il est à remarquer que d_{11} et d_{12} sont déplaçables dans C_1 , mais pas dans $C_1 \cup C_2$.

¹⁹ Il est bien entendu que deux circuits ayant un arc en commun ont également deux points en commun; mais le plus important dans ce cas, c'est qu'ils partagent un défaut arc ou arc-point.

5.3.1 Circuits liés par des arcs

Exemple 43.



Propriété 6

Un ensemble de N défauts formant c circuits liés par des arcs possède une E.F.

↔

- chaque circuit pris séparément possède au moins une E.F.;
- et les ordres fidèles de ces E.F. sont compatibles.

Démonstration



Par hypothèse, on a un ordre $O^t = d_1 / d_2 / \dots / d_N$ auquel l'E.F. sur les N défauts des c circuits est fidèle.

On a donc par définition que dans cet ordre, chaque défaut invalidé se situe après l'ensemble des défauts qui le contredisent.

Extrayons de cet ordre un ordre pour chaque circuit, c'est-à-dire que pour le circuit C_j on supprime de O^t tous les défauts qui n'appartiennent pas à C_j , on obtient ainsi O^j . Dans O^j , chaque défaut invalidé se situe après l'ensemble des défauts qui, dans ce circuit, le contredisent puisque dans O^t chaque défaut invalidé se situe après l'ensemble des défauts qui le contredisent. O^j est alors, par définition, fidèle à une E.F. composée de tous les défauts valides de O^j . Nous obtenons ainsi c ordres fidèles à c E.F. et compatibles entre eux puisqu'extraits d'un même ordre O^t .

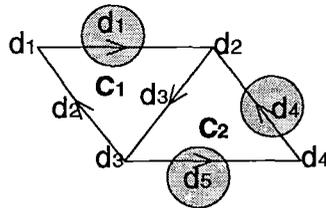


Par hypothèse, chaque circuit possède au moins une E.F. dont l'ordre qui l'induit est compatible avec les ordres qui induisent une E.F. pour les autres circuits. Par définition d'ordres compatibles, il est possible d'obtenir, par union des différents ordres des différents circuits, un ordre O^t sur l'ensemble des circuits où chaque défaut invalidé se trouve après l'ensemble des défauts avec lesquels il est en contradiction.

On obtient donc une E.F. fidèle à l'ordre O^t et composée de l'ensemble des défauts valides de O^t .

cqfd

Exemple 44.



Dans C_1 :

$$d_1 : d_2 / d_3 / \underline{d_1} \Rightarrow F.E.C1$$

Dans C_2 :

$$d_4 : d_2 / d_3 / \underline{d_5} / \underline{d_4} \Rightarrow F.E.C2$$

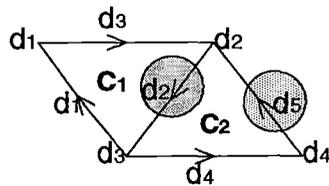
$$d_5 : d_4 / d_2 / d_3 / \underline{d_5} \Rightarrow F.E.C2$$

Dans $C_1 \cup C_2$:

$$d_1 - d_4 : d_2 / d_3 / \underline{d_1} / \underline{d_5} / \underline{d_4} \Rightarrow F.E.$$

$$d_1 - d_5 : d_4 / d_2 / d_3 / \underline{d_1} / \underline{d_5} \Rightarrow F.E.$$

Exemple 45.



Dans C_1 :

$$d_2 : d_3 / d_1 / \underline{d_2} \Rightarrow F.E.C1$$

Dans C_2 :

$$d_2 : d_3 / d_4 / \underline{d_5} / \underline{d_2} \Rightarrow F.E.C2$$

$$d_5 : d_2 / d_3 / d_4 / \underline{d_5} \Rightarrow F.E.C2$$

Dans $C_1 \cup C_2$:

$$d_2 - d_2 : d_3 / \underline{d_1} / \underline{d_4} / \underline{d_5} / \underline{d_2} \Rightarrow F.E.$$

$$d_2 - d_5 : \text{ordres incompatibles} \Rightarrow -F.E.$$

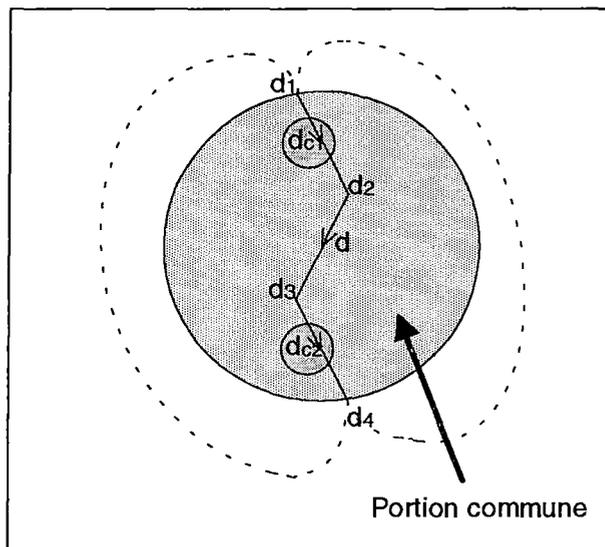
Nous remarquerons dans cet exemple que, puisque l'arc d_2 appartient à la fois à C_1 et à C_2 , l'ordre induit par le candidat d_2 et un ordre induit par tout autre candidat de C_1 ou de C_2 seront incompatibles puisque appartenant à un même circuit.

Lemme 2

N candidats différents se trouvant sur une même portion de graphe (cette portion peut appartenir à un ou plusieurs circuits) produiront des ordres incompatibles entre eux.

Raisonnement

Nous allons envisager le cas le plus simple, celui de deux candidats différents, un pour un circuit C_1 et un autre pour un circuit C_2 . Le raisonnement reste toute fois généralisable à N candidats.



Le candidat d_{c1} induira un ordre $O^{c1} : d_2 / d_3 / d_4 / \dots / d_1 / \dots$; alors que le candidat d_{c2} induira un ordre $O^{c2} : d_4 / \dots / d_1 / d_2 / d_3 / \dots$

O^{c2} place d_3 comme dernier point dans l'ordre alors que O^{c1} le place dans les premiers.

Comme on le voit ici, deux candidats différents partageant une portion de graphe ne peuvent qu'être incompatibles à cause des points du graphe qu'ils ont en commun.

Fin

Lemme 3

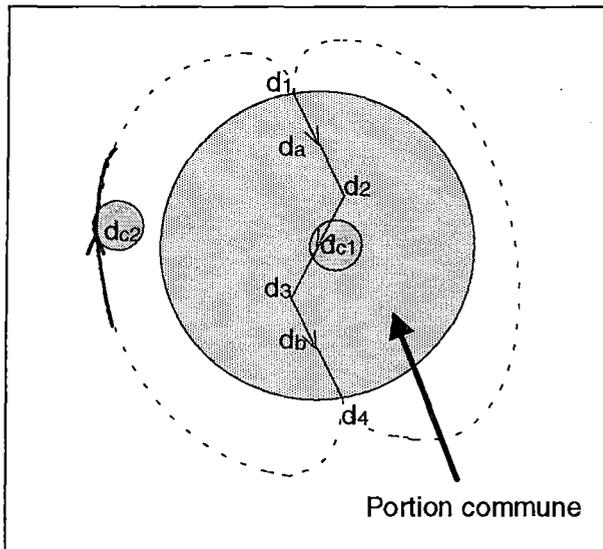
Soit deux circuits C_1 et C_2 partageant une même portion de graphe.

Si C_1 (ou C_2) a un candidat sur la portion commune et C_2 (respectivement C_1) a un candidat sur la portion non commune.

Alors, les ordres induits par ces candidats seront incompatibles.

Raisonnement

Nous retrouvons ici le même raisonnement que pour le Lemme 2.



Soit d_{C_1} le candidat du circuit C_1 et d_{C_2} le candidat de C_2 .

- Si d_{C_1} est aussi candidat pour C_2 on retombe sur le Lemme 2 où la portion commune est le circuit tout entier;
- Dans le cas contraire, d_{C_1} induira un ordre où son point origine, d_2 , sera le dernier des points; alors que d_{C_2} induira un ordre où d_2 sera quelquepart au milieu de l'ordre.

Fin

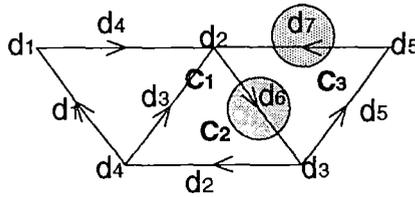
Dans l'Exemple 45, nous voyons également que des circuits ayant un candidat en commun ont une E.F. avec ce candidat, c'est le cas de d_2 .

Lemme 4

N circuits ayant un même candidat (forcément sur la portion de graphe qu'ils ont en commun) auront des ordres compatibles induits par ce candidat.

Nous n'allons pas raisonner sur ce Lemme, nous allons juste voir, dans l'exemple suivant, le cas d'un candidat commun à trois circuits.

Exemple 46.



Dans C_1 :

$$d_6 : d_3 / d_4 / d_1 / d_2 / \underline{d_6} \Rightarrow F.E.C_1$$

Dans C_2 :

$$d_6 : d_3 / d_4 / d_2 / \underline{d_6} \Rightarrow F.E.C_2$$

Dans C_3 :

$$d_6 : d_3 / d_5 / \underline{d_7} / \underline{d_6} / d_2 \Rightarrow F.E.C_3$$

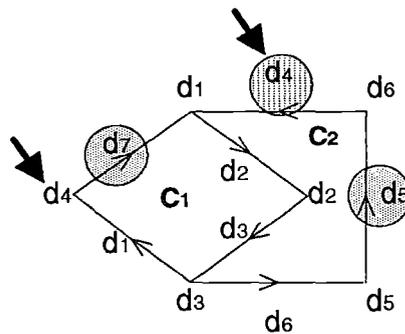
$$d_7 : d_2 / d_3 / d_5 / \underline{d_6} / \underline{d_7} \Rightarrow F.E.C_3$$

Dans $C_1 \cup C_2 \cup C_3$:

$$d_6 - d_6 - d_6 : d_3 / \begin{matrix} d_5 \\ d_4 / d_1 \end{matrix} / d_2 / \underline{d_7} / \underline{d_6} \Rightarrow F.E.$$

$$d_6 - d_6 - d_7 : \text{ordres incompatibles} \Rightarrow -F.E.$$

Exemple 47.



Dans C_1 :

$$d_7 : d_1 / d_2 / d_3 / \underline{d_7} / d_4 \Rightarrow F.E.C_1$$

Dans C_2 :

$$d_5 : d_6 / d_1 / d_2 / d_3 / \underline{d_4} / d_5 \Rightarrow F.E.C_2$$

$$d_4 : d_1 / d_2 / d_3 / d_5 / d_6 / \underline{d_4} \Rightarrow F.E.C_2$$

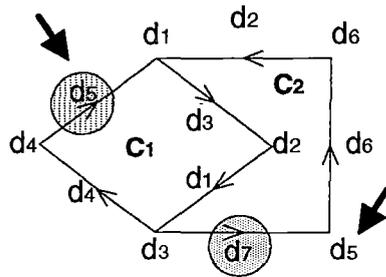
Dans $C_1 \cup C_2$:

$$d_7 - d_5 : d_6 / d_1 / d_2 / d_3 / \underline{d_7} / d_4 / d_5 \Rightarrow F.E.$$

$$d_7 - d_4 : d_1 / d_2 / d_3 / \underline{d_7} / d_5 / d_6 / d_4 \Rightarrow F.E.$$

Dans cet exemple, nous voyons qu'un défaut peut être considéré comme un candidat dans un circuit isolé du reste du graphe, alors que dans le graphe complet, il n'a pas les qualités requises pour être candidat. C'est le cas de d_4 . Mais de tels « candidats » n'offrent pas toujours un ordre compatible avec ceux des candidats des autres circuits comme le montre l'exemple suivant.

Exemple 48.



Dans C_1 :

$$d_5 : d_1 / d_2 / d_3 / d_4 / \underline{d_5} \Rightarrow F.E.C_1$$

Dans C_2 :

$$d_7 : d_5 / d_6 / d_1 / d_2 / d_3 / \underline{d_7} \Rightarrow F.E.C_2$$

Dans $C_1 \cup C_2$:

$$d_5 - d_7 : \text{incompatibles} \Rightarrow \neg F.E.$$

La raison de cette incompatibilité est que dans C_1 : l'arc d_5 doit obligatoirement se trouver après les arcs d_3 et d_1 - qui sont les arcs communs aux deux circuits -; alors que dans C_2 : le point d_5 doit se trouver avant les points d_1 et d_3 .

Définition 25 : (vrai candidat)

Un vrai candidat est un défaut qui possède les propriétés de candidat dans le graphe de défaut complet.

Propriété 7

Soit un ensemble de circuits liés par des arcs. Si chacun des C circuits a au moins un vrai candidat, alors l'ensemble des C circuits possède au moins une E.F..

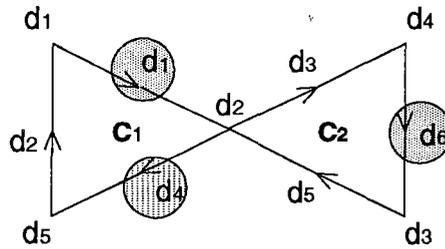
Démonstration

Reste à démontrer!

cqfd

5.3.2 Circuits liés par des points

Exemple 49.



Dans C_1 :

$$d_1 : d_2 / \underline{d_4} / \underline{d_1} / d_5 \Rightarrow F.E.C_1$$

$$d_4 : d_5 / d_1 / d_2 / \underline{d_4} \Rightarrow F.E.C_1$$

Dans C_2 :

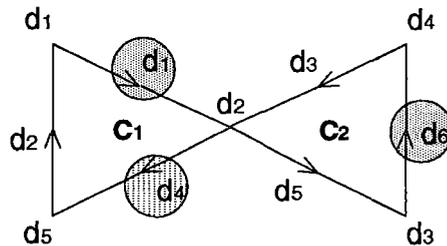
$$d_6 : d_3 / \underline{d_5} / \underline{d_6} / d_2 / d_4 \Rightarrow F.E.C_2$$

Dans $C_1 \cup C_2$:

$$d_1 - d_6 : d_3 / d_2 / \underline{d_5} / \underline{d_1} / \underline{d_6} \Rightarrow F.E.$$

$$d_4 - d_6 : \underline{d_5} / d_1 / d_2 / \underline{d_4} / \underline{d_6} \Rightarrow F.E.$$

Exemple 50.



Dans C_1 :

$$d_1 : d_2 / \underline{d_4} / \underline{d_1} / d_5 \Rightarrow F.E.C_1$$

$$d_4 : d_5 / d_1 / d_2 / \underline{d_4} \Rightarrow F.E.C_1$$

Dans C_2 :

$$d_6 : d_4 / d_2 / d_3 / \underline{d_5} / \underline{d_6} \Rightarrow F.E.C_2$$

Dans $C_1 \cup C_2$:

$$d_1 - d_6 : d_4 / d_2 / \underline{d_5} / \underline{d_1} / \underline{d_6} \Rightarrow F.E.$$

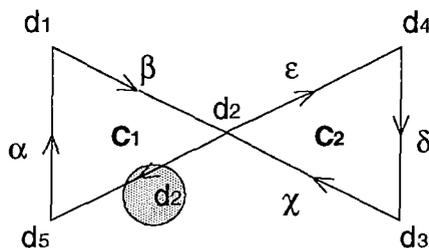
$$d_4 - d_6 : \text{ordres incompatibles à cause de } d_2 \text{ et } d_4 \Rightarrow -F.E.$$

L'Exemple 49 et l'Exemple 50 nous montrent qu'un candidat « non-vrai » (d_4), d'un circuit C , n'implique pas nécessairement un ordre qui est compatible avec les ordres d'un autre circuit C' qui lui est attaché par un point..

Nous allons maintenant nous attacher à étudier une série d'exemples qui donnent, à première vue, des résultats étonnant, mais en y regardant de plus près, cela reste logique.

Etudions maintenant le type de circuits de l'exemple suivant.

Exemple 51.



Voyons sous quelles conditions (c'est-à-dire quelles sont les valeurs que doivent prendre α , β , ϵ , δ et χ pour que) un tel graphe possède une E.F..

Dans C_1 : $d_2 : d_5 / d_1 / \beta / d_2 \Rightarrow F.E.C_1$ où α , β peuvent être des défauts arcs ou arcs-points de C_1 ou de $C_1 \cup C_2$;

Si on veut que $d_2 \cup ?$ implique un ordre fidèle, il faut qu'il y ait un candidat dans C_2 qui implique un ordre qui soit compatible avec l'ordre fidèle trouvé pour C_1 .

◇ Si c'est l'arc δ que nous invalidons pour casser le circuit C_2 , nous obtenons l'ordre $d_3 / d_2 / d_4 / \chi / \epsilon / \delta$. Cet ordre n'est compatible avec celui de C_1 que si :

- est différent de d_5 et de d_1 puisque dans C_1 ils sont avant d_2 et qu'ici d_2 doit être avant δ ;
- est différent de d_2 puisque d_2 appartient déjà à C_1 en tant qu'arc;
- χ / δ respecte $d_3 / d_2 / d_4$

Ce qui nous laisse comme possibilités les ordres :

- $d_3 / d_2 / \underline{d_6} / d_4$ la raison de cette possibilité est que d_3 (défaut arc-point) est situé, en tant que point, avant d_4 , ce qui lui permet en tant qu'arc en contradiction avec le candidat de se trouver devant lui.
- $\underline{d_7} / \underline{d_6} / d_3 / d_2 / d_4$
- $\underline{d_7} / \underline{d_6} / d_3 / d_2 / d_4$
- $d_3 / d_2 / d_4 / \underline{d_7} / \underline{d_6}$
- $d_3 / d_2 / d_4 / \underline{d_7} / \underline{d_8} / \underline{d_6}$
- $d_3 / d_2 / d_4 / \underline{d_6}$

◇ Si c'est χ que nous invalidons, nous avons l'ordre $d_2 / d_4 / d_3 / \frac{\delta}{\epsilon} / \chi$. Cet ordre n'est compatible avec celui de C_1 que si :

- est différent de d_5 et de d_1 puisque dans C_1 ils sont avant d_2 et qu'ici d_2 doit être avant χ ;
- est différent de d_2 puisque d_2 appartient déjà à C_1 en tant qu'arc;
- $\frac{\delta}{\epsilon} / \chi$ respecte $d_2 / d_4 / d_3$

Ce qui nous laisse comme possibilités les ordres :

$d_2 / d_4 / \underline{d_6} / d_3$ la raison de cette possibilité est que d_4 (défaut arc-point) est situé, en tant que point, avant d_3 , ce qui lui permet en tant qu'arc en contradiction avec le candidat de se trouver devant lui.

$\underline{d_7} / \underline{d_6} / d_2 / d_4 / d_3$

$\underline{d_7} / \underline{d_6} / d_2 / d_4 / d_3$

$d_2 / d_4 / d_3 / \underline{d_7} / \underline{d_6}$

$d_2 / d_4 / d_3 / \underline{d_7} / \underline{d_8} / \underline{d_6}$

$d_2 / d_4 / d_3 / \underline{d_6}$

◇ Si c'est ϵ qui est invalidé, nous obtenons l'ordre $d_4 / d_3 / d_2 / \frac{\delta}{\chi} / \epsilon$. Cet ordre n'est compatible avec celui de C_1 que si :

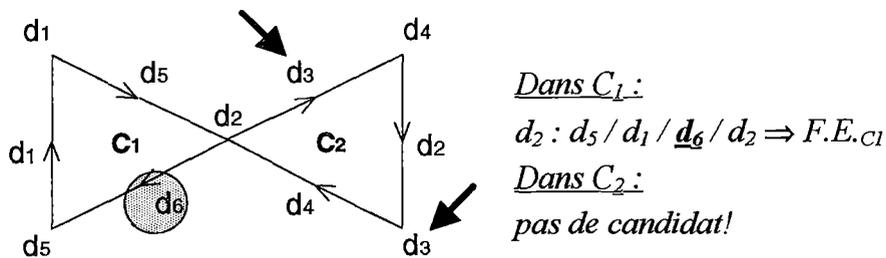
- est différent de d_5 et de d_1 puisque dans C_1 ils sont avant d_2 et qu'ici d_2 doit être avant ϵ ;

- est différent de d_2 puisque d_2 appartient déjà à C_1 en tant qu'arc;
- $\frac{\delta}{\chi}$ respecte $d_4 / d_3 / d_2$

Ce qui nous laisse comme possibilités les ordres :

- $\underline{d}_6 / d_4 / d_3 / d_2$ la raison de cette possibilité est que d_4 (défaut arc-point) est situé, en tant que point, avant d_3 , ce qui lui permet en tant qu'arc en contradiction avec le candidat de se trouver devant lui.
- $\underline{d}_7 / \underline{d}_6 / d_4 / d_3 / d_2$
- $\underline{d}_7 / \underline{d}_6 / \underline{d}_4 / d_3 / d_2$
- $d_4 / d_3 / d_2 / \underline{d}_7 / \underline{d}_6$
- $d_4 / d_3 / d_2 / \underline{d}_7 / \underline{d}_8 / \underline{d}_6$
- $d_4 / d_3 / d_2 / \underline{d}_6$

Exemple 52.



Dans $C_1 \cup C_2$:

Il n'y a pas d'E.F. car quelque soit l'ordre que l'on impose sur tous les défauts du graphe, il ne peut y en avoir de fidèle comprenant d_2, d_3 et d_4 .

Propriété 8

Un ensemble de N défauts formant c circuits liés par des points possède une E.F.

⇔

- chaque circuit pris séparément possède au moins une E.F.;
- et les ordres fidèles de ces E.F. sont compatibles.

Démonstration

Nous n'avons pas eu le temps de démontrer cette propriété qui semble toute fois vraie.

!!!!

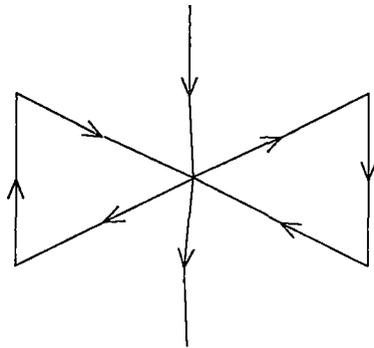
Il semble, au vues de tous les propriétés (démontrés ou pas démontrés) que nous avons présentées, qu'il existe un théorème disant plus ou moins ceci :

Théorème

Un ensemble de N défauts formants c circuits liés ou pas possède une E.F. \Leftrightarrow

- chaque circuit pris séparément possède au moins une E.F.;
- et les ordres fidèles de ces E.F. sont compatibles.

Comme on peut le voir, la théorie développée dans cette section n'est pas complète, il reste à démontrer certaines propriétés, ainsi que le théorème. Ensuite, une fois ces propriétés démontrées, il faudrait encore considérer les cas où des graphes autres que des cycles viennent se greffer sur les graphes de cycles comme :



Conclusion

Nous avons présenté dans ce mémoire les problèmes que pose le raisonnement juridique aux logiques classiques. En effet, les juristes doivent sans cesse raisonner sur des règles qui sont, implicitement ou explicitement, sujettes à des exceptions et ce souvent sur base d'une information incomplète. Ces deux caractéristiques du raisonnement juridique, nous l'avons vu, relèvent du raisonnement non-monotone.

Nous avons ensuite passé en revue un certain nombre de systèmes juridiques informatiques qui prennent en compte, plus ou moins bien, certaines de ces caractéristiques en utilisant des bases peu formelles. Nous avons vu ensuite, qu'une deuxième approche était possible : une approche où la logique (non-monotone) est utilisée. Que la logique soit un candidat possible pour l'analyse d'aspects formels du raisonnement juridique n'est pas surprenant puisque c'est le but essentiel de la logique que de systématiser les structures formelles du raisonnement.

Dans cette optique, nous avons présenté la théorie de Prakken traitant du raisonnement avec et sur les priorités. Ce type de raisonnement est fort utilisé en droit lorsqu'il s'agit de régler des conflits - d'application - entre différentes normes. Cette théorie permet de formaliser un grand nombre d'exemples qui ne peuvent être formalisés par la plupart des logiques (monotones ou non-monotones) existantes.

Afin de développer cette théorie, nous avons tenté de voir sous quelles conditions on pouvait trouver un ensemble cohérent de règles au sein d'une théorie (de défauts - comprenant les règles de priorités - dans le cas qui nous a occupé) donnée. Nous n'avons pu mener à terme cette recherche, mais nous avons, toutefois, pu peut-être montrer une voix de développement.

Au niveau applicatif de cette théorie, Prakken a planifié des tentatives de formalisation de domaines réalistes. Il reste aussi à étudier la question de l'implémentation qui, en général pour des raisons inhérentes au type de raisonnement étudié, ne sera pas une tâche aisée, il faudra adopter certaines restrictions¹.

¹ Cf. Le titre 8 : Problèmes d'implémentations.

Bibliographie

- [BEN1988] BENCH-CAPON, T.J.M. et SERGOT, M.J. 1988. Toward a Rule-Based Representation of Open Texture in Law. C. WALTER (dir.), *Computing Power and Legal Language*, 39-60. New York. Quorum Books.
- [BER1988] BERMAN, D.H. et HAFNER, C.D. 1988. Obstacles to the Development of Logic-Based Models of Legal Reasoning. C. WALTER (dir.), *Computing Power and Legal Language*, 183-214. Westport, Connecticut. Quorum Books.
- [BRE1989] BREWKA, G. 1989. Preferred subtheories : an extended logical framework for default reasoning. *Proceedings IJCAI-1989*, 1043-1048.
- [BRE1991] BREWKA, G. DIX, J. et KONOLIGE, K. 1991. A Tutorial on Nonmonotonic Reasoning. *Proceedings of the Second International Workshop on Nonmonotonic and Inductive Logic*, 1-88. Reinhardtsbrunn Castle, Germany. ACM Press.
- [BRE1994] BREWKA, G. 1994. Adding priorities and specificity to default logic. *Proceeding JELIA-94*, 247-260.
- [ETH1988] ETHERINGTON, D. 1988. *Reasoning with incomplete information*. Pitman, London.
- [FAR199?] FARLEY A.M. et FREEMAN, K. Burden of Proof in a Computational Model of Argumentation, ?
- [FUL1958] FULLER, L.L. 1958. Positivism and Fidelity to law. *Harvard Law Review* 71: 631-672.
- [GAB1985] GABBAY, D. 1985. Theoretical foundations for non-monotonic reasoning in expert systems. *Logics and Models of Concurrent Systems*, Berlin,. Springer-Verlag.
- [GAR1987] GARDNER, A. 1987. *An Artificial Intelligence approach to legal reasoning*. Cambridge, MA : MIT Press.

- [GEF1992] GEFNER, H. et PAERL, J. 1992. Conditional entailment : bridging two approaches to default reasoning. *Artificial Intelligence* 53, 209-244.
- [GEL1989] GELFOND, M. Et LIFSCHITZ, V. 1989. Compiling circumscriptive theories into logic programs. *Proceedings of the second International Workshop on Nonmonotonic Reasoning*, 74-99. Lecture notes in computer sciences 346, Springer Verlag.
- [GEL1990] GELFOND, M. Et LIFSCHITZ, V. 1990. Logic programs with classical negation. *Proceedings of the seventh Logic Programming conference*, 579-597. MIT Press.
- [GRE1991] GREENLEAF, G. MOWBRAY, A. et TYREE, A. 1991. The Datalex Legal Workstation - Integrating tools for lawyers. *Proceedings of the Third International Conference on Artificial Intelligence and Law*, 215-224. St. Catherine's College, Oxford, New York. ACM Press.
- [HAR1958] HART, H.L.A. 1958. Positivism and Separation of Law and Moral. *Harvard Law Review* 71: 593-629.
- [JUN1989] JUNKER, U. et KONOLIGE, K. 1989. Computing the Extensions of Autoepistemic and Default Logic with a TMSS. *Proceedings AAAI*.
- [KRA1990] KRAUS, S. , LEHMANN, D. et MAGIDOR M. 1990. Nonmonotonic reasoning. preferential models and cumulative logics. *Artificial Intelligence*, 44:167-207.
- [KON1988] KONOLIGE, K. 1988. On the Relation between Default an Autoepistemic Logic. *Artificial Intelligence*, 35:343-382.
- [LIF1987] LIFSCHITZ, V. 1987. On the declarative semantics of logic programs with negation. Minker, J. (Ed.) : *Foundation of deductive database and logic programming*, 177-192. Morgan Kaufman Publishers Inc., Los Altos CA.
- [LLE1950] LLEWELLYN, K.L. 1950. Remarks on the theory of appellate decisions and the rules of canons about how statutes are to be constructed. *Vand.L.Rev.*3: 395.
- [LLE1960] LLEWELLYN, K.L. 1960. The common law tradition. Little&Brown. Boston. 77-91.

- [MIN1975] MINSKY, M. 1975. A Framework for Representing Knowledge. New York. P. Winston, ed., *The Psychology of Computer Vision*, McGraw-Hill Book Co., Inc.
- [MOO1985] MOORE, R. C. 1985. Semantical Considerations on Nonmonotonic Logic. *Artificial Intelligence*, 25:75-94.
- [OSK1988] OSKAMP, A. 1988. Knowledge Representation and Legal Expert Systems. P. H. van den BERG (dir.), *Advanced Topics of Law and Information Technology*, Deventer, The Netherlands, Kluwer, Computer/Law Series vol. 3: 195-211.
- [OSK1989] OSKAMP, A. , WALKER, R.F. , SCHRICKX, J.A. et VAN DEN BERG, P.H. 1989. Prolexs, divide and rule: a legal application. *Proceeding of the second international conference on artificial intelligence and law*, 54-62. Vancouver, BC Canada. ACM Press.
- [POO1985] POOLE, D.L. 1985. On the Comparison of Theories: Preferring the Most Specific Explanation. *Proceeding of the Ninth International Joint Conference on Artificial Intelligence 1*, 18-23. Los Angeles, California. AAAI press.
- [POO1988] POOLE, D. L. 1988. A Logical Framework for Default Reasoning. *Artificial Intelligence*, 36:27-47.
- [PRA1991] PRAKKEN, H. 1991. A tool in modelling disagreement in law: preferring the most specific argument. *Proceedings of the Third International Conference on Artificial Intelligence and Law*, 165-174. Oxford, New York. ACM Press.
- [PRA1993] PRAKKEN, H. Logical Tools for Modelling Legal Argument. Thesis. Vrije Universiteit, Amsterdam.
- [REI1978] REITER, R. 1978. On Reasoning by Default. *proc. THINLAP-2, Theoretical Issues in Natural Languages Processing-2*, University of Illinois at Urbana-Champaign. 210-218.
- [REI1980] REITER, R. 1980. A Logic for Default Reasoning. *Artificial Intelligence*, 13:81-132.
- [RIS1988] RISSLAND, E.L. 1988. Artificial Intelligence and Legal Reasoning - A Discussion of the Field & Garner's Book. *AI Magazine* 9: 45-55.

- [RIS1989] RISSLAND, E.L. et ASHLEY, K.D. 1989. HYPO : A precedent-based legal reasoner. G.P.V. Vandenberghe (ed.) : *Advanced topic in law and information technology*, 213-234. Kluwer, Deventer.
- [SAL1893] SALMOND. 1893. *The First Principles of Jurisprudence* 97-98.
- [SCH1986] SCHLIPF, J. S.. 1986. How uncomputable is general circumscription. *Proceedings of the Symposium on Logic in Computer Science*, 92-95. Cambridge, MA
- [SER1986] SERGOT, M.J. SADRI, F. KAWALSKI, R.A. KRIWACZEK, F. HAMMOND, P. et CORY, H.T. 1986. The British Nationality Act as Logic Program. *Communication of ACM* 29: 370-386.
- [SER1991] SERGOT, M.J. KAMBLE, A.S. et BAJAJ, K.K. 1991. Indian Central Civil Service Pensions Rules: A case study in applying logic programming to regulations. *Proceedings of the Third International Conference on Artificial Intelligence and Law*, 118-127. St Catherine's College, Oxford, New York. ACM Press.
- [SHO1987] SHOHAM, Y. 1987. Nonmonotonic logics: Meaning and utility. *Proceedings of the International Joint Conference on Artificial Intelligence*, 23-28. Milan.
- [SHO1988] SHOHAM, Y. 1988. *Reasoning about changes. Time and causation from the standpoint of Artificial Intelligence*. MIT Press.
- [SIM1992] SIMARI, G.R. et LOUI, R.P. 1992. A mathematical treatment of defeasible argumentation and its implementation. *Artificial Intelligence* 53, 125-157.
- [SKA1992] SKALAK, D.B. et RISSLAND, E.L. 1992. Arguments and Cases: An Inevitable Intertwining. *Artificial Intelligence and Law* 1: 3-44.
- [SMI1987] SMITH, J.C. et DEEDMAN, C. 1987. The Application of Expert Systems Technology to Case-Based Law. *Proceedings of the First International Conference on Artificial Intelligence and Law*, 85-93. Boston. ACM Press.
- [TOU1958] TOULMIN, S. 1958. *The uses of argument*. Cambridge: Cambridge University Press.