Une procedure de decision en logique non-monotone

Philippe Besnard

CENTRE DE RENNES
IRISA

Rapports de Recherche

N° 217

UNE PROCÉDURE DE DÉCISION EN LOGIQUE NON-MONOTONE

Philippe BESNARD

Juillet 1983

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(3) 954 90 20

INRIA

CENTRE DE RENNES

IRISA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(3) 954 90 20

Rapports de Recherche

N° 217

UNE PROCÉDURE DE DÉCISION EN LOGIQUE NON-MONOTONE

Philippe BESNARD

Juillet 1983

UNE PROCEDURE DE DECISION EN LOGIQUE NON-MONOTONE

Philippe BESNARD

Abstract

From default logic, a complete non-monotonic logic with a first-order based proof theory formulation, a decidable logical subsystem is proposed. It relies on a first-order solvable class, for which a decidable resolution strategy is given. The non-monotonic proof-procedure we present uses this strategy, so that it is proved to be complete and decidable when applied to the considered logical subsystem.

Résumé

Dans ce document, nous nous intéressons à la logique des défauts, une logique non-monotone complète dont la théorie de la démonstration se réfère à la théorie de la démonstration du calcul des prédicats du premier ordre. A partir d'une classe solvable du premier ordre, pour laquelle nous développons une procédure de décision, nous déterminons un fragment décidable de la logique des défauts. Finalement, nous présentons une procédure de preuve, que nous démontrons complète et décidable pour le fragment considéré.

INTRODUCTION

La logique constitue une des principales approches du problème de la représentation des connaissances. Elle permet en particulier d'employer des techniques de démonstration développées en déduction automatique. Historiquement, l'évolution de la recherche en déduction automatique s'est faite dans le sens d'un accroissement non seulement de l'efficacité des procédures de preuve, mais aussi de la richesse des formalismes traités. Ainsi les premiers programmes de démonstration automatique ([New 56]) étaient restreints à la logique propositionnelle, un système simple et décidable, mais évidemment limité dans sa puissance d'expression. Pour ce qui est du calcul des prédicats du premier ordre, les premières procédures (par exemple [Gil 60]), étaient plus ou moins directement inspirées des travaux de Herbrand, jusqu'à ce que Robinson propose le principe de résolution [Rob 65]. Un certain nombre de recherches ultérieures ont eu pour but d'affiner la résolution, conduisant à diverses restrictions, telles que le A-ordering [Kow 69].

Cependant aucune de ces procédures ne parvient à rendre compte de certains schémas inférentiels qui interviennent pourtant de facon courante dans le raisonnement humain. McCarthy fut le premier à s'en préoccuper, en plaidant en faveur de l'introduction d'une certaine connaissance de "sens commun" dans les programmes [McC 68].

Peu après, Hayes et McCarthy [Hay 69] proposaient d'employer le calcul situationnel pour incorporer cette connaissance dans les systèmes de déduction automatique. L'expérimentation menée par Green [Gre 69] montrait rapidement les limitations de cette approche, en particulier le problème du cadre en résolution de problèmes [Rap 71].

De ce fait, Robinson [Rob 69] préconisait l'emploi de logiques d'ordre supérieur. Mais en contrepartie de leur plus grande puissance d'expression, ces formalismes posent des problèmes beaucoup plus profonds (par exemple, à partir de l'ordre 2, l'unification est indécidable [Gol 81] [Hue 76]).

De ce point de vue, la logique modale est moins sévèrement pénalisée. Il existe cependant certaines difficultés (Seules les formules qui ne comportent pas de quantificateur dans le champ d'un opérateur modal peuvent se mettre sous la forme prénexe requise par une extension de la résolution dûe à Farinas del Cerro [Far 81]).

Récemment, en vue de résoudre ce problème du raisonnement de "sens commun", les recherches se sont dirigées vers l'étude de logiques non-monotones [AI 80]. De telles logiques s'opposent à la logique classique du premier ordre qui est dite monotone car l'adjonction de nouveaux axiomes ne fait jamais décroitre l'ensemble des théorèmes d'un système axiomatique.

La circonscription, une proposition de McCarthy [McC 80], est certainement la solution la plus séduisante en raison de sa théorie des modèles qui est particulièrement élégante. La circonscription est une logique non-monotone qui ne prend en compte que les modèles minimaux (Un modèle M est inférieur à un modèle M', relativement au prédicat P, si l'extension de P dans M est incluse dans l'extension de P dans M'). La théorie de la démonstration associée comporte un "schéma de circonscription" qui peut être comparé aux schémas d'axiomes apparaissant dans certaines formulations de la théorie de la démonstration du calcul des prédicats du premier ordre (Par exemple [Kle 67]). Malheureusement, la question est alors de découvrir une "substitution du schéma de circonscription" qui permettra d'obtenir un axiome pertinent vis-à-vis du problème considéré.

Bossu et Siegel [Bos 81] ont mené une étude traitant d'une approche sémantique essentiellement identique. Ils ont développé une méthode de démonstration basée sur l'évaluation des littéraux qui sont caractéristiques des différents modèles minimaux (Ce sont notamment les littéraux positifs des clauses non régulières).

McDermott et Doyle [McD 80] ont proposé une logique non-monotone complète, construite autour d'un opérateur modal de possibilité M. La procédure de preuve qu'ils donnent pour le cas propositionnel fait partie des méthodes de tableaux, peu efficaces. En ce qui concerne la théorie des modèles développée par McDermott et Doyle, les mondes possibles se définissent comme les points fixes pour la déduction qui sont consistants. Un théorème est une formule qui appartient à chaque monde possible. Cependant, la notion de consistance développée pose quelques difficultés. Ainsi, ~P et ~MP ("il est impossible que P soit vrai") ne sont pas équivalents. Cette observation a suscité 2 études qui reprennent la même démarche que celle de McDermott et Doyle, non plus sur la base de la logique classique du premier ordre, mais à partir de la logique modale [McD 81] et de la logique intuitionniste [Gab 82]. L'efficacité des procédures de preuve respectives est à peu près identique (Pour la logique intuitionniste, Fitting donne comme procédure de preuve une variante de la méthode des tableaux [Fit 71]).

Dans la logique des défauts [Rei 80], une formule est un théorème si elle appartient à un quelconque monde possible. Un monde possible, appelé par Reiter extension, est consistant et déductivement clos. Les extensions sont obtenues par l'application de défauts, à partir d'un monde initial, commun à toutes les extensions. Un défaut représente une formule qu'il est possible de dériver, sous réserve de consistance. Reiter a prouvé que la logique des défauts est complète, mais qu'elle n'est même pas semi-décidable. Il n'existe donc pas de procédure qui soit capable, pour une quelconque formule valide (au sens de la logique des défauts), de démontrer cette formule en un temps fini. Le résultat de complétude n'est donc pas exploitable dans une mise en oeuvre. Pour cette raison, il nous a paru intéressant d'essayer de délimiter un sous-ensemble décidable de la logique des défauts et de développer une procédure de preuve adaptée.

La propriété de consistance imposée aux extensions se retrouve dans la définition d'une preuve par défaut. Cette définition fait appel à la notion de satisfiabilité d'un ensemble de formules du calcul des prédicats du premier ordre. Seule une procédure de preuve décidable permettra de toujours effectuer les tests de consistance en un temps fini. Un sous-ensemble décidable de la logique des défauts s'appuiera donc sur une classe solvable de formules du premier ordre. Dans la première partie, nous présentons une telle classe, constituée par les clauses à variables prédéfinies. Nous y décrirons également une procédure de preuve décidable pour les clauses à variables prédéfinies. La deuxième partie sera consacrée à la logique des défauts. Nous introduirons une catégorie particulière de défauts, dits sans prérequis. Nous discuterons brièvement de l'intérêt de ces défauts. Finalement, à partir des clauses à variables prédéfinies et des défauts sans prérequis, nous déterminerons un sous-ensemble de la logique des défauts. Dans la dernière partie, nous étudierons une procédure de preuve décidable pour ce sous-ensemble.

I LA SATURATION PAR ENSEMBLES

I 1 Introduction

La semi-décidabilité du calcul des prédicats du premier ordre signifie que les procédures de preuve opérant sur tout le langage sont, au mieux, des procédures de semi-décision. Elles se terminent toujours lorsqu'on leur soumet un ensemble insatisfiable de formules. En revanche, il se peut qu'elles ne s'arrètent jamais si on leur soumet un ensemble satisfiable de formules. Mais il existe des classes de formules, spécifiées par leurs propriétés syntaxiques, pour lesquelles le problème de décision admet une solution. Pour chacune de ces classes, dites solvables, il est possible de construire une procédure de preuve décidable, même s'il n'existe pas de procédé systématique pour cela. Il convient de noter qu'exhiber une procédure de preuve décidable pour une certaine classe de formules démontre par la même que cette classe est solvable. C'est la démarche qu'ont suivi Bossu et Siégel pour montrer que les clauses à variables prédéfinies forment une classe solvable [Bos 81].

Dans cette première partie, nous établirons la décidabilité de la saturation par ensembles pour les clauses à variables prédéfinies. Nous définirons tout d'abord ce qu'est une clause à variables prédéfinies. Nous discuterons ensuite de la mise sous forme bien ordonnée des clauses à variables prédéfinies. Pour les clauses à variables prédéfinies mises sous forme bien ordonnée nous pourrons alors définir la résolution sur le littéral de tête. Nous indiquerons comment, en appliquant sous certaines conditions (de subsomption) la résolution sur le littéral de tête à un ensemble de clauses à variables prédéfinies, on obtient toujours un ensemble de clauses saturé, en un nombre fini de pas. Sachant qu'un ensemble saturé est inconsistant si et seulement si il contient la clause vide, nous disposerons ainsi d'une procédure de preuve décidable pour les clauses à variables prédéfinies. Cette procédure, qui sera appelée saturation par ensembles, est une généralisation de la saturation [Bos 81] de Bossu et Siégel, auxquels la plupart des résultats de cette première partie sont dûs.

I 2 Ordonnancement des clauses à variables prédéfinies

Nous commencerons par donner la syntaxe des clauses à variables prédéfinies. Puis nous définirons ce qu'est un ordre atomique, avant de construire celui qui nous servira par la suite dans la définition d'une clause bien ordonnée. Enfin, nous montrerons comment, pour les clauses à variables prédéfinies mises sous forme bien ordonnée, la résolution sur le littéral de tête permet de diminuer le nombre des résolvantes.

Définition:

Une clause est à variables prédéfinies si

Ses symboles fonctionnels sont des constantes

-Toute variable qui apparait dans un littéral positif apparait également dans un littéral négatif

Exemple:

$P(x) \vee \bar{~}Q(x,y)$

$P(x,b,y) \vee \bar{~}Q(x) \vee \bar{~}R(y)$

$P(a,c)$

Contre-exemple: Les 2 clauses ci-dessous ne sont pas à variables prédéfinies

$P(x)$

$\bar{~}P(a,x) \vee Q(y)$

Comme son nom l'indique, le principe de la saturation par ensembles est de générer toutes les clauses possibles jusqu'à l'obtention d'un état stable pour la résolution. Eu égard au nombre excessivement élevé de résolvantes potentielles, cette optique exige donc une restriction efficace de la résolution. La technique adoptée s'inspire du A-ordering [Kow 69], stratégie de résolution qui distingue un littéral particulier sur lequel et sur lequel seulement, la résolution pourra s'effectuer. Le choix de ce littéral se fait en fonction d'un ordre atomique, qui est une relation d'ordre sur l'ensemble des formules atomiques.

Nous allons maintenant présenter l'ordre atomique qui sera appliqué aux clauses à variables prédéfinies.

Construction d'un ordre atomique particulier

L'ordre atomique, noté $<$, considéré par la suite, est défini par

1_ Un ordre total strict sur les symboles relationnels

$P <_r Q <_r S$ (l'ordre alphabétique)

2_ Un ordre total strict sur les constantes

$a <_c b <_c e$ (l'ordre alphabétique)

3_ Soient $P(t_1,\ldots,t_n)$ et $P'(t'_1,\ldots,t'_m)$ 2 formules atomiques.

cas où $P=P'$: $P(t_1,\ldots,t_n) < P'(t'_1,\ldots,t'_m)$ s'il existe $k \in \{1,\ldots,n=m\}$ tel que

$-t_i=t'_i$ pour $0\leqslant i<k$

$-t_k <_c t'_k$

cas où $P\neq P'$: $P(t_1,\ldots,t_n) < P'(t'_1,\ldots,t'_m)$ si $P <_r P'$

4_ Le signe du littéral est inopérant

Exemple:

$Q(a) < \bar{} R$

$\bar{} Q(a) < \bar{} Q(b)$

$\bar{} P(x) < Q(c)$

Pour les ordres notés $<_c$ et $<_r$ qui servent à l'élaboration de l'ordre atomique $<$, n'importe quel ordre total strict convient tout autant que l'ordre alphabétique.

L'ordre $<$ est partiel car les variables ne sont comparables ni entre elles ni avec les constantes. Ainsi $\bar{} Q(x)$ et $Q(a)$ ne sont pas ordonnables.

Cet ordre sur les littéraux est employé dans la définition d'une clause bien ordonnée. Dans une telle clause, la position des littéraux est caractéristique et contrainte par des règles faisant appel à cet ordre atomique.

Définition:

Une clause à variables prédéfinies, sans littéraux égaux, est sous forme bien ordonnée si son littéral de tête est

.soit un littéral négatif

.soit un littéral positif inférieur à tous les autres littéraux de la clause et tel que toute variable qui apparait dans un autre littéral apparait aussi dans ce littéral de tête

Exemple:

$P(x) \vee \tilde{}Q(x,y)$ est une clause à variables prédéfinies dont la seule forme bien ordonnée est $\tilde{}Q(x,y) \vee P(x)$

Après fusion de ses littéraux égaux, une clause à variables prédéfinies peut toujours se mettre sous forme bien ordonnée.

L'ordonnancement des littéraux à l'intérieur d'une clause, grâce à la distinction d'un littéral de tête, permet de restreindre la résolution.

Définition:

Soient l et l' 2 formules atomiques unifiables par le pgu σ

Soient 2 clauses c=lr et c'=l'r' où r et r' sont des clauses éventuellement vides.

La résolvante sur le littéral de tête de c et c' est la clause $(rr')_{\sigma}$

Exemple: c_3 est la résolvante sur le littéral de tête de c_1 et c_2 :

$c_1 = P(x) \vee \tilde{}Q(x,y)$

$c_2 = \tilde{}P(a)$

$c_3 = \tilde{}Q(a,y)$

Contre-exemple: $c_1 = P \vee R$ et $c_2 = Q \vee \tilde{}R$

Ces 2 clauses n'ont pas de résolvante sur le littéral de tête. La résolution sur le littéral de tête diminue le nombre de résolvantes qu'il est possible de générer.

I 3 La résolution sur le littéral de tête

Dans cette section, nous allons tout d'abord montrer que la résolvante sur le littéral de tête de 2 clauses à variables prédéfinies mises sous forme bien ordonnée est elle-même une clause à variables prédéfinies. Or nous avons vu qu'une clause à variables prédéfinies peut toujours se mettre sous forme bien ordonnée. Nous en conclurons que l'application répétée de la résolution sur le littéral de tête transforme un ensemble de clauses à variables prédéfinies en un autre ensemble de clauses à variables prédéfinies. Enfin nous montrerons que cette application ne peut se répéter qu'un nombre fini de fois avant que chaque résolvante sur le littéral de tête ne soit logiquement équivalente à une clause déjà existante.

Considérons une variable x qui apparait dans un littéral positif de la résolvante sur le littéral de tête de 2 clauses à variables prédéfinies. Dans le cas où x apparait dans un littéral négatif de la clause parent dont le littéral de tête est positif, alors ce littéral négatif, qui contient x, appartient à la résolvante. Dans le cas contraire, x n'apparait dans aucun littéral de la clause parent dont le littéral de tête est positif car c'est une clause à variables prédéfinies. Mais puisque x est une variable de la résolvante, x apparait donc dans un littéral positif de l'autre clause parent. Celle-ci étant à variables prédéfinies, possède un littéral négatif contenant x. Or ce littéral négatif appartient à la résolvante car il ne peut s'agir du littéral de tête (sinon x apparaitrait dans le littéral de tête positif de l'autre clause parent). Finalement, toute variable qui apparait dans un littéral positif de la résolvante apparait aussi dans un littéral négatif. Par ailleurs, si les seuls termes des clauses parents sont des constantes et des variables, la résolvante n'aura que des constantes pour symboles fonctionnels. La résolvante est donc à variables prédéfinies, ainsi que l'énonce le résultat ci-dessous:

Proposition 1.1:

La résolvante sur le littéral de tête de 2 clauses à variables prédéfinies est une clause à variables prédéfinies.

De cette proposition il suit que, partant d'un ensemble de clauses à variables prédéfinies, la production des résolvantes sur le littéral de tête mène toujours à un autre ensemble de clauses à variables prédéfinies.

Exemple:

$c_1 = \sim P(a)$

$c_2 = P(x) \vee \sim Q(x,b)$

$c_3 = Q(a,b) \vee R$

Les clauses c_1 et c_2 ont pour résolvante sur le littéral de tête la clause

$c_4 = \tilde{}Q(a,b)$

Cette résolvante peut elle-même se résoudre avec la clause c_3, d'où

$c_5 = R$

Cependant, la proposition 1.1 ne garantit pas que cette production se termine: le nombre de résolvantes sur le littéral de tête n'est peut-être pas toujours fini, même si l'ensemble de départ est fini. Les 2 propositions suivantes contribuent à déterminer sous quelles conditions la production de résolvantes sur le littéral de tête est un processus fini pour les clauses à variables prédéfinies.

Proposition 1.2:

La résolvante sur le littéral de tête de 2 clauses bien ordonnées ne contient pas plus de variables que son parent dont le littéral de tête est négatif.

En effet, toutes les variables de la clause parent dont le littéral de tête est positif apparaissent dans ce littéral de tête (car la clause est bien ordonnée). La résolution va unifier ce littéral de tête avec celui de l'autre clause parent. Cette dernière, dont le littéral de tête est négatif, contient donc toutes les variables de la résolvante.

Soit un ensemble fini de clauses bien ordonnées. Il existe un entier n tel que toute clause admet au plus n variables. La proposition 1.2 permet de conclure qu'aucune résolvante ne comportera plus de n variables. Il est alors possible de renommer les variables de toute clause à partir d'un ensemble fini $\{x_1,...,x_n\}$ de variables. L'ensemble des variables étant fini, de même que celui des constantes et des symboles relationnels, il n'est possible de construire qu'un ensemble fini de littéraux et donc de clauses bien ordonnées. D'où le prochain résultat:

Proposition 1.3:

1_ Si E est un ensemble fini de clauses bien ordonnées,

2_ Si $s=c_1...c_n...$ est une suite de clauses bien ordonnées dont tout élément c est

.soit un élément de E

.soit la résolvante sur le littéral de tête de 2 clauses qui précèdent c_i dans la suite

3_ Si aucune clause de s n'est logiquement équivalente à une clause qui la précède dans la suite

alors s est finie.

Etant donné un ensemble C de clauses à variables prédéfinies, si l'on ne produit pas les résolvantes logiquement équivalentes à une clause déjà existante, on aboutit donc toujours, après un nombre fini de résolutions sur le littéral de tête, à un ensemble C' de clauses qui n'augmente plus par résolution sur le littéral de tête.

L'ensemble C est satisfiable si et seulement si l'ensemble C' l'est lui-même. En effet, les 2 ensembles sont logiquement équivalents puisque la résolution constitue une procédure de preuve qui est saine [Rob 65].

Il manque néanmoins un critère permettant de tester la satisfiabilité de l'ensemble C'. Le prochain paragraphe fournit un tel critère, en montrant que l'ensemble C' est saturé et qu'un ensemble saturé est inconsistant si et seulement si la clause vide y appartient.

I 4 Ensemble saturé

Définition:

Un ensemble C de clauses est dit saturé pour la résolution sur le littéral de tête si toute résolvante sur le littéral de tête de 2 clauses de C, qui n'est pas une tautologie, est subsumée par une clause de C.

Le théorème 1.4 détermine, relativement aux définitions précédentes, un cas où il existe un critère pour juger de la satisfiabilité d'un ensemble de clauses à variables prédéfinies.

Théorème 1.4:

Un ensemble saturé de clauses bien ordonnées est inconsistant ssi il contient la clause vide.

La démonstration prenant plusieurs pages, n'est pas reprise ici. Il importe toutefois de signaler que le cas terminal est prouvé d'abord. Pour le passage au niveau général, il convient de remarquer que toute instance d'une clause bien ordonnée est bien ordonnée. Un ensemble de clauses bien ordonnées est donc logiquement équivalent à l'ensemble des instances terminales bien ordonnées de ses clauses, ensemble auquel le résultat du cas terminal peut s'appliquer.

Le théorème 1.4 précise que l'ensemble sur lequel a lieu le test d'appartenance de la clause vide doit être saturé. Or d'après la définition, il n'est pas nécessaire qu'un ensemble saturé contienne une clause subsumée par une autre. Cette observation conduit dans la pratique à renforcer, dans la proposition 1.3, la troisième condition qui devient:

3_ Aucune clause de s n'est subsumée par une clause qui la précède dans la suite

Cette modification évite de générer les clauses subsumées par une clause déjà produite. En particulier, les tautologies ne viennent plus grossir l'ensemble de clauses existant.

Au vu des propositions 1.1 et 1.3, il est clair que, partant d'un ensemble quelconque de clauses à variables bien ordonnées, on peut toujours obtenir un ensemble saturé en un nombre fini de résolutions sur le littéral de tête. Aussi le théorème 1.4, joint à ces 2 résultats, décrit-il la décidabilité des clauses à variables prédéfinies, sans toutefois donner une procédure de preuve à proprement parler. Ce chapitre se terminera donc par la donnée d'une procédure de preuve effective, décidable pour les clauses à variables prédéfinies.

I 5 Saturation par ensembles

Dans cette dernière partie du premier chapitre, nous donnons une procédure de preuve appelée saturation par ensembles. Nous basant sur les résultats précédents, nous démontrerons d'abord que cette procédure se termine toujours. Nous montrerons ensuite que la saturation par ensembles produit un ensemble saturé, donc auquel le théorème 1.4 peut s'appliquer. Nous aurons ainsi prouvé que la saturation par ensembles est décidable pour les clauses à variables prédéfinies. Nous terminerons en montrant que la saturation par ensembles reste décidable pour une restriction particulière du test de subsomption.

Définition:

Soient E et F 2 ensembles de clauses bien ordonnées.

Soit f une fonction de sélection d'une clause parmi un ensemble de clauses.

La saturation de F sur E, notée S(E,F), se définit récursivement par

-$S(E,\emptyset)=E$

-$S(E,F)=S(E \cup \{c\}, F' - \{c\})$ où

.c=f(F) est appelée clause génératrice.

.F' est l'ensemble F augmenté de toute résolvante sur le littéral de tête de c et d'une clause de E, qui n'est pas subsumée par une clause de E.

Notation:

Considérons les suites (E_i) et (F_i) définies par

$$E_1 = E$$

$$F_1 = F$$

$$E_{i+1} = E_i \cup \{f(F_i)\}$$

$$F_{i+1} = R_i \cup F_i - \{f(F_i)\}$$

Si $c_i = f(F_i)$ désigne la clause génératrice, R_i est l'ensemble des résolvantes sur le littéral de tête de c_i et d'une clause de E_i, qui ne sont pas subsumées par une clause de E_i.

La définition se réécrit alors:

-$S(E_i,\emptyset) = E_i$

-$S(E_i,F_i) = S(E_i \cup f(F_i), R_i \cup F_i - f(F_i)) = S(E_{i+1},F_{i+1})$

Exemple:

$E = \{ A \vee B \}$ et $F = \{ \tilde{}B \vee \tilde{}C \, , \tilde{}A \}$

$S(E_1,F_1) = S(E_2,F_2)$

Si $f(F_1) = \tilde{}B \vee \tilde{}C$ alors

$E_2 = \{ A \vee B \, , \tilde{}B \vee \tilde{}C \}$

$F_2 = \{ \tilde{}A \}$

$S(E_2,F_2) = S(E_3,F_3)$

$f(F_2) = \tilde{}A$ donc

$E_3 = \{ A \vee B \, , \tilde{}B \vee \tilde{}C \, , \tilde{}A \}$

$F_3 = \{ B \}$

$S(E_3,F_3) = S(E_4,F_4)$

$f(F_3) = B$ donc

$E_4 = \{ A \vee B \, , \tilde{}B \vee \tilde{}C \, , \tilde{}A \, , B \}$

$F_4 = \{ \tilde{}C \}$

$S(E_4,F_4) = S(E_5,F_5)$

$f(F_4) = \tilde{}C$ donc

$E_5 = \{ A \vee B \, , \tilde{}B \vee \tilde{}C \, , \tilde{}A \, , B \, , \tilde{}C \}$

$F_5 = \emptyset$ car aucune clause n'a été produite

$S(E_5,F_5) = E_5$

$S(E,F) = E_5$

Nous allons maintenant montrer que, selon le procédé itératif décrit ci-dessus, la saturation par ensembles se termine toujours si les 2 ensembles d'entrée E et F sont finis.

Proposition 1.5:

Si E et F sont finis, alors S(E,F) est fini et finiment calculable.

Preuve:

Tout d'abord, il est clair que E_i et F_i sont finis.

En effet, E_1 et F_1 sont finis par hypothèse.

Par ailleurs, si l'on suppose E_i et F_i finis, $E_{i+1} = E_i \cup \{f(F_i)\}$ est aussi fini.

Sous les mêmes hypothèses, $F_{i+1} \subset F_i \cup R_i$ est fini, puisqu'il n'est possible de produire qu'un nombre fini de résolvantes issues de la clause génératrice $c_i = f(F_i)$ et d'une clause de E_i

Il reste à montrer qu'il existe n tel que $F_n = \emptyset$

Construisons une suite $(s_i)_{i \in N}$ telle que:

-s_i est une suite de clauses de E_i

-toute clause de E_i est subsumée par une clause de s_i

-aucune clause de s_i n'est subsumée par une clause qui la précède.

De plus, pour tout $i>1$, s_i doit avoir s_{i-1} pour sous-suite initiale.

Hormis s_1, chaque suite s_i est donc ainsi déterminée de façon unique.

La proposition 1.3 appliquée à $E \cup F$, s'écrit:

1_ $E \cup F$ est un ensemble fini de clauses bien ordonnées

2_ $s_i = c_1 ... c_m ...$ est une suite de clauses bien ordonnées dont tout élément c_j est

-soit un élément de $E \cup F$

-soit la résolvante sur le littéral de tête de 2 clauses qui précèdent c_j dans la suite

3_ Aucune clause de s_i n'est subsumée par une clause qui la précède dans la suite

alors s_i est finie.

Donc il n'existe pas de suite infinie de clauses de S(E,F) telle qu'aucune clause ne soit subsumée par une clause qui la précède.

Ce qui veut dire qu'il existe k tel que pour tout $i \geq k$, $s_i = s_k$

F_i est formé de

-FS_i les clauses subsumées par une clause de s_i

-FNS_i les clauses qui ne sont subsumées par aucune clause de s_i

Pour tout $i>k$, la clause génératrice c_i appartient à FS_i

(Sinon $E_{i+1} = E_i \cup \{c_i\}$ avec $c_i \in FNS_i$

donc c_i n'est subsumée par aucune clause de $s_i = s_k$

donc s_{i+1} est formée de s_i augmentée de c_i

ce qui est en contradiction avec $s_{i+1} = s_i$)

La clause génératrice c_i ne produit que des clauses de FNS_{i+1}, pour $i>k$

(La définition de F_{i+1} ignore toute résolvante r de c_i et d'une clause de E_i si elle est subsumée par une clause de E_i.

Une clause de $F_{i+1} - F_i$ ne peut donc être subsumée par un élément de s_{i+1} puisque $s_{i+1} = s_i$ est une suite d'éléments de E_i)

De ces 2 points, il vient $FS_{i+1} \subset FS_i$ pour $i>k$.

Comme FS_k est fini, $FS_{k+card(FS_k)} = \emptyset$

Comme la clause génératrice, lorsqu'elle existe, ne peut appartenir qu'à $FS_{k+card(FS_k)}$, il en résulte l'égalité:

$F_n = \emptyset$ pour $n = k+card(FS_k)$

CQFD

Dans la proposition suivante, nous indiquons dans quel cas la saturation par ensembles produit un ensemble saturé.

Proposition 1.6:

S(E,F) est saturé si E est saturé

Preuve:

-Si c_1 et c_2 sont 2 clauses de E

Alors l'hypothèse E saturé implique que la résolvante sur le littéral de tête de c_1 et c_2 est subsumée par une clause de E, donc de S(E,F).

-Si c_1 est une clause de E et c_2 une clause de S(E,F) - E

Par la définition et la proposition 1.5, la saturation de F sur E comporte la saturation de F_i contenant c_2 sur E_i contenant c_1 avec $f(F_i) = c_2$

Soit r la résolvante sur le littéral de tête de c_1 et c_2

-cas où r appartient à R_i:

$S(E,F) = S(E_i \cup \{f(F_i)\}, R_i \cup F_i - \{f(F_i)\})$

donc r appartient à S(E,F).

-cas où r n'appartient pas à R_i:

C'est que, par définition de $S(E_i,F_i)$, il existe une clause c de E_i qui subsume r.

Mais $S(E,F) = S(E_i,F_i)$, donc c appartient à S(E,F).

-Si c_1 et c_2 sont 2 clauses de S(E,F) - E

La démonstration se ramène à celle du cas précédent.

Si par exemple c_1 devient clause génératrice avant c_2 (supposition qui a un sens en vertu de la définition et de la proposition 1.5), $S(E,F) = S(E_i,F_i)$ où $f(F_i) = c_1$ et $c_2 \in F_i$

Il suffit d'appliquer le résultat du cas précédent à $S(E_{i+1},F_{i+1})$

CQFD

Contre-exemple de la réciproque

$E = \{ A \vee B , \sim A \vee C \}$ et $F = \{ A \vee C \}$

$S(E,F) = \{ A \vee B , \sim A \vee C , A \vee C , C \}$

S(E,F) est saturé, mais pas E.

Corollaire 1.7:

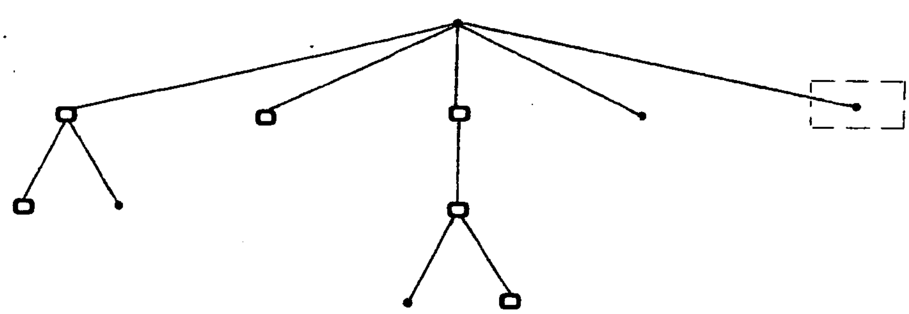
S(∅,C) est un ensemble saturé logiquement équivalent à C.

La saturation [Bos 81] constitue un cas particulier de la saturation par ensembles. Pour leur algorithme, Bossu et Siégel ont exhibé une propriété qui minimise le nombre de tests de subsomption à effectuer. Le résultat qu'ils énoncent se généralise aisément. La démonstration qui suit est entièrement calquée sur la leur, à ceci près qu'ici la fonction de sélection de la clause génératrice est quelconque.

Terminologie

Pour tout élément r de R_i, résolvante sur le littéral de tête d'une clause c de E_i et de la clause génératrice c_i, c_i est le père de r et c la mère de r.

Les clauses produites à la nième étape de saturation peuvent se représenter comme les noeuds d'un arbre $A^n(E)$:



La racine de l'arbre est une clause fictive.

Les noeuds du deuxième niveau sont les clauses de F.

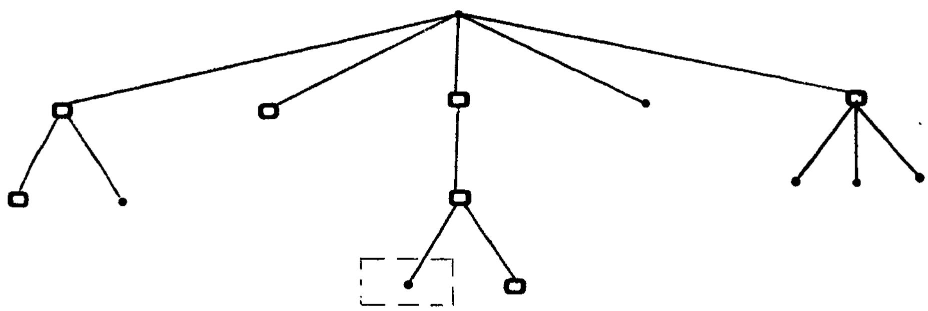
Les arcs représentent la relation "est le père de".

Les noeuds entourés d'un carré sont des clauses de E_n.

Les autres noeuds sont les clauses de F_n.

Les pointillés signalent la nième clause génératrice.

A l'étape n+1, les fils de la clause génératrice sont produits:



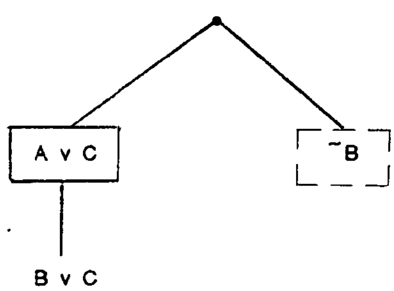
Exemple:

$E = \{ \sim A \vee B \}$ et $F = \{ A \vee C , \sim B \}$

Pour $f(F_1) = A \vee C$, les ensembles obtenus sont

$E_2 = \{ \sim A \vee B , A \vee C \}$

$F_2 = \{ \sim B , B \vee C \}$

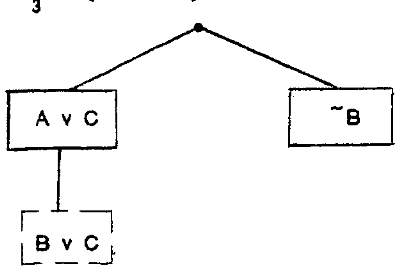


$A^2(F)$

Pour $f(F_2) = \sim B$,

$E_3 = \{ \sim A \vee B , A \vee C , \sim B \}$

$F_3 = \{ B \vee C \}$

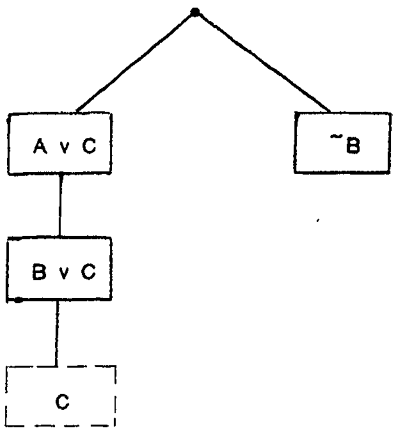


$A^3(F)$

Pour $f(F_3) = B \vee C$,

$E_4 = \{ \sim A \vee B , A \vee C , \sim B , B \vee C \}$

$F_4 = \{ C \}$



$A^4(F)$

L'arbre $A^*(F)$ est défini comme la limite pour n tendant vers l'infini des arbres A. Pour que la saturation par ensembles soit décidable, il suffit que pour tout ensemble F fini de clauses bien ordonnées, le nombre de noeuds de $A^*(F)$ soit fini. Or $A^*(F)$ est un arbre tel que, de chaque noeud ne part qu'un nombre fini de flèches (le nombre de fils de chaque clause génératrice est fini). Donc, d'après le théorème de König, le nombre de noeuds de $A^*(F)$ est fini ssi toute branche est finie.

Proposition 1.8:

La saturation par ensembles reste décidable lorsque seules les résolvantes subsumées par un de leurs "ancêtres mâles" ne sont pas produites.

Preuve:

Si ces résolvantes ne sont pas produites et si $c_1 \to c_2 \to \dots \to c_n \to \dots$ est une branche de $A^*(F)$, aucune clause de cette branche n'est subsumée par une clause qui la précède. Comme toutes les clauses de cette branche sont bien ordonnées et ont été produites à partir d'un ensemble fini de clauses, la branche est finie (proposition 1.3).

CQFD

II LA LOGIQUE DES DEFAUTS

II 1 Introduction

Le calcul des prédicats du premier ordre possède la propriété de monotonie que l'on définit habituellement de la manière suivante:

Soient S et S' 2 systèmes axiomatiques tels que $S \subset S'$

Pour toute formule F, si $S \models F$ alors $S' \models F$

Dans une logique monotone, l'ensemble des théorèmes ne décroit jamais lorsque l'ensemble des axiomes augmente.

La logique des défauts [Rei 80], qui a pour objet l'établissement de suppositions, est non-monotone. Il est facile de s'en rendre compte, puisqu'une supposition, considérée comme un théorème tant qu'elle n'est pas démentie par les axiomes, peut toujours être réfutée par l'ajout d'un axiome contradictoire.

Ce deuxième chapitre est consacré à l'étude de la logique des défauts. Nous commencerons par présenter, sur un exemple, la fonction des défauts. Nous introduirons ensuite les théories avec défauts, ce qui nous conduira à définir la notion d'extensions qui s'apparente à celle des mondes possibles. Après avoir vu qu'une formule est dite "valide par défaut" s'il existe une quelconque extension où elle est vraie, nous donnerons la définition d'une preuve par défaut. Nous présenterons alors le théorème de complétude de la logique des défauts. Nous définirons la classe des défauts sans prérequis, avant d'exposer rapidement ses avantages sur le plan de la représentation des connaissances. Nous expliquerons également pourquoi la logique des défauts, restreinte aux clauses à variables prédéfinies, est décidable.

II 2 Les défauts

La logique des défauts permet effectuer des inférences malgré l'absence de certaines prémisses, ce qui peut s'appeler faire des suppositions. L'exemple suivant concernant des oiseaux illustre parfaitement la finalité et l'intérêt de cette logique.

Le propre des oiseaux est de voler, bien qu'il existe certaines exceptions, comme l'autruche et le pingouin. En logique du premier ordre, cette connaissance se représente par l'implication

$$(x)\ \text{oiseau}(x)\ \&\ \tilde{}\,\text{autruche}(x)\ \&\ \tilde{}\,\text{pingouin}(x)... \Rightarrow \text{vole}(x)$$

Une formule de ce genre a 2 inconvénients:

-La "découverte" d'une exception supplémentaire rend cette formule caduque. En d'autre termes, une telle formule ne supporte pas l'apparition de contre-exemples.

-Pour prouver qu'un oiseau vole, il est indispensable d'établir qu'il ne compte pas parmi les exceptions. En conséquence, il est impossible de prouver qu'un oiseau, sur lequel rien n'est précisé, vole. Ceci parce qu'il est impossible de démontrer qu'il n'est ni une autruche, ni un pingouin,... Et pourtant, intuitivement, un oiseau n'est ni une autruche, ni un pingouin, à moins que cela ne soit explicitement spécifié.

D'une manière générale, la logique du premier ordre ne permet pas d'inférer une information incertaine, aussi plausible qu'elle puisse être.

En vue de résoudre ces problèmes, Reiter a envisagé la contruction de règles (les défauts [Rei 80]) comportant une conclusion sujette à caution. Dans ce formalisme, la loi régissant l'aptitude des oiseaux à voler s'exprime par un défaut:

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

qui signifie "Les oiseaux volent", mais qui s'interprète "Si x est un oiseau et s'il est consistant de supposer que x peut voler, alors inférer que x peut voler".

Quant aux exceptions, ce sont autant de formules:

$$(x)\ \text{pingouin}(x) \Rightarrow \tilde{}\,\text{vole}(x)$$

$$(x)\ \text{autruche}(x) \Rightarrow \tilde{}\,\text{vole}(x)$$

Avec cette représentation, les 2 inconvénients évoqués plus haut disparaissent. En effet, l'ajout d'une exception se résume alors à l'ajout d'une formule, tandis que le défaut ainsi mis en cause ne change ni de signification, ni d'interprétation. Enfin, il n'est plus nécessaire de prouver que l'oiseau en question n'appartient pas aux espèces qui ne volent pas.

Syntaxe d'un défaut

Un défaut se note $$\frac{P(x)\ :MQ_1(x),...,Q_n(x)}{R(x)}$$

$P,Q_1,...,Q_n,R$ sont des formules du premier ordre dont $x=x_1,...,x_p$ sont les seules variables libres.

$P(x)$ est le prérequis du défaut.

$R(x)$ est le conséquent du défaut.

La présence de l'opérateur modal de possibilité M traduit seulement le rôle joué par les formules $Q_1,...,Q_n$

Si $n=1$ et si $Q(x)=R(x)$, le défaut est dit normal.

Un défaut est clos s'il ne contient pas d'occurence libre de variables, sinon il est ouvert.

II 3 Les théories avec défauts clos normaux

Dans ce paragraphe nous allons définir les théories avec défauts et leurs extensions. Rappelons qu'une extension correspond, en un certain sens, à un monde possible. Nous nous limiterons à l'étude des théories avec défauts normaux car elles sont les seules à toujours posséder une extension. Nous verrons que la logique des défauts est semi-monotone, c'est-à-dire que l'ajout de défauts ne remet pas en cause la validité par défaut des formules.

Définition:

Une théorie avec défauts est un couple (Δ,W) où W est un ensemble consistant de formules closes du premier ordre et Δ un ensemble de défauts.

Une théorie avec défauts joue le rôle de plusieurs théories du premier ordre. En effet, les défauts sont des règles qui étendent la théorie initiale sous-jacente W. Ils permettent d'ajouter à W un certain nombre de formules qu'il est possible de "croire". Le fait que ces formules ne soit pas forcément toutes compatibles détermine les différentes théories, appelées extensions, finalement obtenues.

Dans un premier temps, l'étude des théories avec défauts se bornera à celle des théories avec défauts clos (dans lesquelles tout défaut est clos). Au paragraphe II 5, tous les résultats obtenus seront généralisés pour inclure le cas des défauts ouverts (Un défaut ouvert étant alors considéré comme l'ensemble de ses instances). Dans cette section, les preuves, relativement techniques, sont omises (Se reporter à [Rei 80]).

Définition:

Soit une théorie (Δ,W) avec défauts clos et E un ensemble de formules closes.

Soit une suite définie par

$-E_0 = W$

$-E_{i+1} = Th(E_i) \cup \{R \,/\, \frac{P : MQ_1,\dots,Q_n}{R} \in \Delta \;\&\; P \in E_i \;\&\; \sim Q_1,\dots,\sim Q_n \notin E\}$

alors E est une extension de (Δ,W) si $E = \bigcup_{i=0}^{\infty} E_i$

Le lemme ci-dessous, qui nous servira ultérieurement, confirme ce qui a été dit dans l'introduction générale, à savoir que les extensions sont déductivement closes.

Lemme 2.1:

Etant donné une théorie (Δ,W) avec défauts clos, soit une suite définie par

$-E_0 = W$

$-E_{i+1} = Th(E_i) \cup \{R \,/\, \frac{P :MQ_1,...,Q_n}{R} \in \Delta \;\&\; P \in E_i \;\&\; \sim Q_1,...,\sim Q_n \notin E\}$

si $E = \bigcup_{i=0}^{\infty} E_i$ alors $E = Th(E)$

Preuve:

$-E \subseteq Th(E)$ est évident

$-Th(E) \subseteq E$

Soit F une formule de Th(E).

D'après le théorème de compacité, il existe un sous-ensemble fini S de E tel que $S \models F$

Or $E = \bigcup_{i=0}^{\infty} E_i$ et pour tout i $E_i \subseteq E_{i+1}$

Donc, comme $S \subseteq E$ est fini, il existe k tel que $S \subseteq E_k$

Puisque $F \in Th(S)$ alors $F \in Th(E_k)$ d'où $F \in E_k$ et $F \in E$

CQFD

Exemples:

$W = \emptyset$ et $\Delta = \{ \frac{:MC}{\sim B}, \frac{:MB}{\sim C} \}$

Cette théorie avec défauts a 2 extensions, $E_1 = \{\sim B\}$ et $E_2 = \{\sim C\}$ selon le défaut activé (par exemple, si le premier défaut est appliqué, $\sim B$ devient vrai, donc B n'est plus consistant et il n'est alors pas possible d'appliquer le second défaut).

$W' = \{ B \Rightarrow \sim A \;\&\; \sim C \}$ et $\Delta' = \{ \frac{:MA}{A}, \frac{:MB}{B}, \frac{:MC}{C} \}$

Là aussi il y a 2 extensions: $E'_1 = Th(W' \cup \{A,C\})$ et $E'_2 = Th(W' \cup \{B\})$

La théorie avec défauts $(\{:MA/\sim A\},\emptyset)$ n'a aucune extension (à ne pas confondre avec le cas d'une extension unique égale à W), ce qui amène à considérer les théories avec défauts normaux.

Proposition 2.2

Toute théorie avec défauts clos normaux a une extension.

A la suite de cette proposition, Reiter montre que si une théorie avec défauts clos normaux possède une extension inconsistante, alors c'est sa seule extension. Il montre également que si une théorie (Δ,W) n'a pas d'extension consistante, alors W est inconsistant. Comme par définition W est consistant, il vient:

Proposition 23:

Toute extension d'une théorie avec défauts clos normaux est consistante

Nous allons introduire une nouvelle notion avant de donner la proposition 24, qui comme la précédente, nous servira pour la démonstration de complétude de notre procédure de preuve, au chapitre III.

Définition:

Soit E une extension d'une théorie avec défauts clos normaux (Δ,W)

L'ensemble des défauts générateurs de E est $DG(E)=\{ \frac{P :MR}{R} \in \Delta \ / \ P \in E \ \& \ \sim R \notin E \}$

Proposition 24:

Si E est une extension d'une théorie avec défauts normaux (Δ,W) alors $E=Th(W \cup Consequents(DG(E)))$

Le théorème 25 énonce que les défauts ont un comportement monotone. Il garantit que si un ensemble de formules $P_1,...,P_n$ appartient à une même extension de (Δ,W), alors il existe une extension de $(\Delta \cup \Delta',W)$ contenant $P_1,...,P_n$. Dans la pratique, l'adjonction de nouveaux défauts pourra donc s'effectuer directement, sans précaution particulière.

Théorème 25: Semi-monotonie des défauts

Soient Δ et Δ' 2 ensembles de défauts clos normaux tels que $\Delta' \subseteq \Delta$.

Si E' est une extension de (Δ',W) alors (Δ,W) a une extension E telle que $E' \subseteq E$

Exemple:

$W = \emptyset$ et $\Delta = \{ \frac{:MA \vee B}{A \vee B} , \frac{:M\sim A \ \& \sim B}{\sim A \ \& \sim B} \}$

Cette théorie a 2 extensions: $E_1 = \{ \sim A , \sim B \}$ et $E_2 = \{ A \vee B \}$

La formule A qui est inconsistante avec l'extension E_1, peut être introduite par l'intermédiaire du défaut (:MA/A) sans que soit invalidé aucun théorème de la théorie (Δ,W).

Les extensions de la nouvelle théorie avec défauts sont $E'_1 = E_1$ et $E'_2 = \{ A \vee B , A \}$ $(E_2 \subseteq E'_2)$

Toutes les formules qui étaient valides par défaut dans la première théorie le sont donc encore dans la seconde.

II 4 Preuves par défaut

Une théorie avec défauts peut s'assimiler à plusieurs théories du premier ordre qui décrivent les situations vraisemblables. Un enchainement logique voudrait qu'une formule soit vraie "par défaut" si elle est valide dans une certaine extension.

Etant donné une théorie avec défauts clos normaux, il s'agit donc de déterminer s'il existe une extension contenant la formule à prouver par défaut.

Définition:

Soit une theorie (Δ,W) avec défauts clos normaux et soit P une formule close du premier ordre.

Une suite finie $\Delta_0,\ldots,\Delta_k$ de sous-ensembles finis de Δ est une preuve par défaut de P ssi

1_ $W \cup Consequents(\Delta_0) \vdash P$

2_ Pour $i=0\ldots k$, $W \cup Consequents(\Delta_i) \vdash Prerequis(\Delta_{i-1})$

3_ $\Delta_k = \emptyset$

4_ $W \cup (\bigcup_{i=0}^{k} Consequents(\Delta_i))$ est satisfiable

Exemple:

$\delta_1 = \dfrac{E \vee F :MA \,\&\, F}{A \,\&\, F}$ $\delta_2 = \dfrac{A :MB}{B}$ $\delta_3 = \dfrac{A \,\&\, E :MC}{C}$ $\delta_4 = \dfrac{:M\,\tilde{}E}{\tilde{}E}$

$W = \{ C \Rightarrow G , A \,\&\, B \Rightarrow E , E \vee G , G \Rightarrow F \}$

La théorie $(\{\delta_1, \delta_2, \delta_3, \delta_4\}, W)$ a 2 extensions $E_1 = Th(W \cup \{ A \,\&\, F , B , C \})$ et $E_2 = Th(W \cup \{ A \,\&\, F , \tilde{}E \})$

La preuve par défaut de G qui exprime son appartenance à E_1 est $\{\delta_3\} , \{\delta_1, \delta_2\} , \{\delta_1\} , \{ \}$ car

$W \cup Consequents(\{\delta_3\}) \vdash G$

(en effet $C , C \Rightarrow G \vdash G$ où C est le consequent de δ_3)

$W \cup Consequents(\{\delta_1, \delta_2\}) \vdash Prerequis(\{\delta_3\})$

(en effet $A , B , A \,\&\, B \Rightarrow E \vdash A \,\&\, E$

où $A \,\&\, E$ est le prerequis de δ_3

A est le consequent de δ_1

B est le consequent de δ_2);

$W \cup Consequents(\{\delta_1\}) \vdash Prerequis(\{\delta_1, \delta_2\})$

(en effet A , E v G , G => F |- A & (E v F)

où E v F est le prerequis de δ_1

A est le prerequis de δ_2

A est le consequent de δ_1)

W U Consequents(Ø) |- Prerequis({ δ_1 })

(en effet E v G , G => F |- E v F où E v F est le prerequis de δ_1)

Et W U Consequents({ δ_1 ,δ_2 ,δ_3 }) est consistant.

G appartient aussi à E_2, c'est pourquoi il admet une seconde preuve par défaut qui est { δ_4 },{ }

(E v G , ~E |- G et le prerequis de δ_4 est vrai).

Aucune extension ne contient B & ~E. Cependant { δ_2 ,δ_4 },{ δ_1 },{ } vérifie les conditions 1 à 3 imposées pour être une preuve par défaut de B & ~E. Ceci illustre le fait que la condition 4 correspond à la vérification de l'existence d'une extension construite à partir des défaut employés dans la preuve éventuelle.

L'étude des théories avec défauts clos normaux s'achève par un résultat de complétude.

Théorème 26:

Etant donné P une formule close du premier ordre, une théorie (Δ,W) avec défauts clos normaux a une extension E telle que P appartient à E ssi P a une preuve par défaut relativement à (Δ,W).

II 5 Théories avec défauts normaux

Tous les résultats vus jusqu'ici s'appliquent également au cas des théories avec défauts normaux (clos et/ou ouverts) pourvu que chaque défaut soit considéré comme l'ensemble de ses instances qui sont des défauts clos.

Voici le détail des transformations qui réalisent cette idée.

Soit une théorie (D,W) avec défauts normaux.

Toutes les formules, tant celles de W que celles intervenant dans les défauts, sont mises sous forme skolémisée.

Soit F l'ensemble des symboles fonctionnels apparaissant dans la théorie avec défauts normaux résultant de cette opération.

Si H(F) est l'univers de Herbrand alors DEFAUTS-CLOS(Δ)= $\{ \delta(g) \ / \ \delta(x) \in \Delta$ avec $g \in H(F) \}$

Ainsi DEFAUTS-CLOS(Δ) est l'ensemble des instances des défauts de Δ obtenues en parcourant l'univers de Herbrand, incluant les termes dérivés des fonctions de Skolem précédemment introduites. L'instanciation s'est faite sur les variables libres, mais n'a pas affecté les autres variables du défaut.

Exemple:

$$W = \{ \ (Ex)\ P(x) \ \} \text{ et } \Delta = \left\{ \frac{:MP(b)}{P(b)} , \frac{:M(x)\ Q(x,y)}{(x)\ Q(x,y)} \right\}$$

$$W = \{ \ P(a) \ \}$$

La variable x quantifiée existantiellement dans W a été transformée en a, une fonction de Skolem d'arité nulle.

$$H(F) = \{ \ a \ , \ b \ \} \text{ d'où DEFAUTS-CLOS}(\Delta) = \left\{ \frac{:MP(b)}{P(b)} , \frac{:M(x)\ Q(x,a)}{(x)\ Q(x,a)} , \frac{:M(x)\ Q(x,b)}{(x)\ Q(x,b)} \right\}$$

Définition:

E est une extension pour la théorie avec défauts normaux (Δ,W) ssi E est une extension pour la théorie avec défauts clos normaux (DEFAUTS-CLOS(Δ),W).

Tous les résultats restent valables puisque, pour les établir, aucune supposition n'a été faite sur le caractère fini ou infini de l'ensemble des défauts d'une théorie avec défauts clos.

II 6 Défauts et clauses à variables prédéfinies

La définition d'une preuve par défaut fait clairement ressortir que déterminer si une formule P est vraie par défaut ou pas, nécessite que soient résolus 2 problèmes de satisfiabilité. Le premier est de savoir si P est conséquence sémantique d'un certain ensemble de formules. Le second est de savoir si ce même ensemble de formules est consistant. Ces déductions s'effectuent dans le cadre de théories du premier ordre. Afin que ces problèmes soient décidables, il convient de se restreindre à une classe solvable de formules. Celle des clauses à variables prédéfinies présente l'avantage d'être accompagnée d'une procédure de preuve appropriée, la saturation par ensembles.

Cependant, cet aspect déductif d'une procédure de preuve par défaut n'est pas seul en cause en regard de la décidabilité de la logique des défauts.

Le choix des clauses à variables prédéfinies est dicté par une contrainte supplémentaire, relative à la finitude de l'ensemble des conséquents des défauts. D'après ce qui a été exposé à la fin du paragraphe précédent, à une théorie avec défauts normaux (Δ,W) est associée la théorie avec défauts clos normaux (DEFAUTS-CLOS(Δ),W). DEFAUTS-CLOS(Δ) est inclus dans l'ensemble des instances des défauts de Δ. Les éléments de DEFAUTS-CLOS(Δ) sont les défauts de Δ dont les variables libres ont été instanciées par les termes de l'univers de Herbrand de la théorie Prerequis(Δ) U Consequents(Δ) U W. Quand cette théorie peut se mettre sous la forme d'un ensemble fini de clauses à variables prédéfinies, elle ne comporte aucun symbole fonctionnel d'arité non nulle. Les seuls termes concrets sont des constantes, lesquelles n'apparaissent qu'en nombre fini. Puisqu'un univers de Herbrand n'est composé que de termes concrets, celui-ci est fini. Donc, si Δ est fini, DEFAUTS-CLOS(Δ) le sera aussi.

Ceci garantit que le nombre d'ensembles de formules candidats pour constituer une extension est fini. "L'ensemble de formules considéré forme-t-il une extension?" est donc une question qui ne peut se répéter indéfiniment s'il s'agit à chaque fois d'un ensemble différent (égal à W U Consequents(Δ'), sachant que Δ' décrit l'ensemble des parties de DEFAUTS-CLOS(Δ)). Dès lors, les 2 problèmes de satisfiabilité exposés au début de ce paragraphe ne se poseront qu'un nombre fini de fois.

Il existe donc une procédure de preuve décidable pour toutes les théories (Δ,W) telles que chaque formule de W, de Consequents(Δ) et de Prerequis(Δ) puisse se mettre sous forme de clauses à variables prédéfinies.

II 7 Défauts sans prérequis

Le présent paragraphe introduit la classe des défauts normaux sans prérequis. A tout défaut normal, il est possible d'associer de facon naturelle un défaut normal sans prérequis. Appliquée à une théorie avec défauts normaux, cette relation conserve la validité par défaut. Cette propriété fait l'objet d'une étude formelle en II 7 1. Sur le plan pratique, les répercussions de l'emploi de cette relation sont analysées en II 7 2.

II 7 1 Théories avec défauts sans prérequis

Un défaut normal est sans prérequis s'il s'écrit $\frac{:MR(x)}{R(x)}$

A tout défaut normal δ est associé un défaut normal sans prérequis, noté $sp(\delta)$, par la transformation:

$$\delta = \frac{P(x)\ :MR(x)}{R(x)} \qquad \vdash\!\!-\!\!-> \qquad sp(\delta) = \frac{:MP(x) => R(x)}{P(x) => R(x)}$$

Par abus de notation, $sp(\Delta)$ désignera l'ensemble des défauts sans prérequis obtenu en appliquant la transformation ci-dessus à chaque défaut de Δ. Le résultat suivant exprime que toute extension de la théorie (Δ,W) est contenue dans une extension de $(sp(\Delta),W)$. (En particulier, toute formule valide par défaut relativement à (Δ,W) l'est aussi par rapport à $(sp(\Delta),W)$).

Proposition 2.7:

Si E est une extension d'une théorie (Δ,W) avec défauts normaux, alors il existe une extension E' de $(sp(\Delta),W)$ telle que toute formule valide dans E est valide dans E'

Preuve:

Le résultat va d'abord être établi pour le cas d'une théorie avec défauts clos normaux, avant d'être étendu aux théories avec défauts normaux (clos ou ouverts).

Par ailleurs, il est clair qu'il suffit de montrer que pour toute extension E d'une théorie (Δ,W), il existe une extension E', contenant E, de la théorie (Δ',W) obtenue en remplacant un seul défaut par sa forme sans prérequis.

Soit E une extension d'une théorie $(\Delta \cup \{\delta\},W)$.

Considérons une extension E' de $(\Delta \cup \{sp(\delta)\},E)$.

Il existe au moins une telle extension E' car toute théorie avec défauts clos normaux a une extension. (Proposition 2.2)

Montrons que:

(1) E' est aussi une extension de $(\Delta \cup \{sp(d)\},W)$

(2) E' contient E (immédiat)

Par définition d'une extension, $E = \bigcup_{i=0}^{\infty} E_i$

$-E_0 = W$

$-E_{i+1} = Th(E_i) \cup \{R \,/\, \frac{P:MR}{R} \in \Delta \;\&\; P \in E_i \;\&\; \neg R \notin E\}$

Par définition d'une extension, $E' = \bigcup_{i=0}^{\infty} E'_i$

$-E'_0 = W$

$-E'_{i+1} = Th(E_i) \cup \{R \,/\, \frac{P:MR}{R} \in \Delta \cup \{sp(\delta)\} \;\&\; P \in E'_i \;\&\; \neg R \notin E'\}$

Il reste à prouver que E' est également une extension de $(\Delta \cup \{sp(\delta)\}, W)$ (point (1)).

Considérons la suite (F_i) définie par

$-F_0 = W$

$-F_{i+1} = Th(F_i) \cup \{R \,/\, \frac{P:MR}{R} \in \Delta \cup \{sp(\delta)\} \;\&\; P \in F_i \;\&\; \neg R \notin E'\}$

Il suffit de montrer que $E' = \bigcup_{j=0}^{\infty} F_j$

Pour cela, nous établirons au préalable le résultat auxiliaire $E \subseteq \bigcup_{j=0}^{\infty} F_j$

Il s'obtient par récurrence sur i pour $E_i \subseteq \bigcup_{j=0}^{\infty} F_j$

-pour le rang i=0, $E_0 \subseteq F_0$ car $E_0 = W = F_0$

-pour le rang i+1.

Soit $R \in E_{i+1}$

.Si $R \in Th(E_i)$ alors $R \in \bigcup_{i=0}^{\infty} F_i$ car $E_i \subseteq \bigcup_{i=0}^{\infty} F_i$ et $Th(\bigcup_{i=0}^{\infty} F_i) = \bigcup_{i=0}^{\infty} F_i$

.Si R est conséquent d'un défaut $\frac{P:MR}{R}$ de Δ

Par définition, $P \in E_i$ et $\neg R \notin E$

$\neg R$ n'appartient pas à E' sinon E' serait inconsistant puisque $R \in E_{i+1} \subseteq E \subseteq E'$

Mais puisque $P \in E_i$ et $E \subset \bigcup_{j=0}^{\infty} F_j$ il suit $P \in \bigcup_{j=0}^{\infty} F_j$

Donc pour un certain k, $P \in F_k$, qui entraine $R \in F_{k+1}$, soit $R \in \bigcup_{j=0}^{\infty} F_j$

.Si R est conséquent du défaut $\delta = \frac{P:MR}{R}$

Pour les mêmes raisons qu'au cas précédent, $\neg R$ n'appartient pas à E', pas plus que $P \;\&\; \neg R$

Comme $P \;\&\; \neg R \notin E'$ et $\frac{:MP \Rightarrow R}{P \Rightarrow R} = sp(\delta) \in \Delta \cup \{sp(\delta)\}$,

par définition de la suite (F_i), $P \Rightarrow R \in F_1$ donc $P \Rightarrow R \in \bigcup_{j=0}^{\infty} F_j$

Puisque $P \in E_i$, d'après l'hypothèse de récurrence, $P \in \bigcup_{j=0}^{\infty} F_j$

Et comme $P \Rightarrow R \in \bigcup_{j=0}^{\infty} F_j$ et $P \in \bigcup_{j=0}^{\infty} F_j$ par le modus ponens, $R \in Th(\bigcup_{j=0}^{\infty} F_j)$

Or $\bigcup_{j=0}^{\infty} F_j$ est déductivement clos, d'où $R \in \bigcup_{j=0}^{\infty} F_j$

CQFD

Montrons, par double inclusion, que $E' = \bigcup_{j=0}^{\infty} F_j$

$E' \subset \bigcup_{j=0}^{\infty} F_j$

Récurrence sur i pour $E'_i \subset \bigcup_{j=0}^{\infty} F_j$

-pour i=0, $E'_0 = E$ par définition de E'

D'après le résultat auxiliaire, $E \subseteq \bigcup_{j=0}^{\infty} F_j$ donc $E'_0 \subseteq \bigcup_{j=0}^{\infty} F_j$

-au rang i+1, l'hypothèse de récurrence est $E'_i \subseteq \bigcup_{j=0}^{\infty} F_j$

Soit $R \in E'_{i+1}$

.Si $R \in Th(E'_i)$

Par l'hypothèse de récurrence, $E'_i \in \bigcup_{j=0}^{\infty} F_j$

Comme $\bigcup_{j=0}^{\infty} F_j$ est déductivement clos, $Th(E'_i) \subseteq \bigcup_{j=0}^{\infty} F_j$ et donc $R \in \bigcup_{j=0}^{\infty} F_j$

.Si R est conséquent d'un défaut $\frac{P :MR}{R} \in \Delta \cup \{sp(\delta)\}$

Par définition, $P \in E'_i$ et $\neg R \notin E'$, d'où par hypothèse de récurrence $P \in \bigcup_{j=0}^{\infty} F_j$

Il existe donc k tel que $P \in F_k$

Puisque $P \in F_k$ et $\neg R \notin E'$, alors par définition de (F_i), $R \in F_{k+1}$ d'où $R \in \bigcup_{j=0}^{\infty} F_j$

$\bigcup_{j=0}^{\infty} F_j \subseteq E'$

Récurrence sur j pour $F_j \subseteq \bigcup_{i=0}^{\infty} E'_i$

-pour j=0, $F_0 = W$ par définition de la suite (F_i)

Par définition de E et E', $W \subseteq E = E'_0$, d'où $F_0 \subseteq E'_0$ et $F_0 \subseteq \bigcup_{i=0}^{\infty} E'_i$

-au rang j+1, l'hypothèse de récurrence est $F_j \subseteq \bigcup_{i=0}^{\infty} E'_i$

Soit $R \in F_{j+1}$

.Si $R \in Th(F_j)$

$F_j \subseteq \bigcup_{i=0}^{\infty} E'_i$ (hypothèse de récurrence) et $\bigcup_{i=0}^{\infty} E'_i$ est déductivement clos, d'où $Th(F_j) \subseteq \bigcup_{i=0}^{\infty} E'_i$

.Si R est conséquent d'un défaut $\frac{P :MR}{R}$ de $\Delta \cup \{sp(\delta)\}$

Par définition, $P \in F_j$ et $\neg R \notin E'$, d'où par hypothèse de récurrence $P \in \bigcup_{i=0}^{\infty} E'_i$

Il existe donc k tel que $P \in E'_k$

Comme $P \in E'_k$ et $\neg R \notin E'$, par définition de E', $R \in E'_{k+1}$ d'où $R \in \bigcup_{i=0}^{\infty} E'_i$

Finalement, les 2 inclusions donnent $E' = \bigcup_{j=0}^{\infty} F_j$

(1) et (2) ayant été démontrés, il vient: Si δ est un défaut clos normal, alors pour toute extension E de $(\Delta \cup \{\delta\}, W)$, il existe une extension E' de $(\Delta \cup \{sp(\delta)\}, W)$ telle que E soit incluse dans E'

Les considérations exposées au paragraphe II 5 concernant la correspondance entre un ensemble de défauts normaux et l'ensemble de leurs instances closes permettent d'étendre la propriété à tout défaut normal (clos ou ouvert).

CQFD

II 7 2 Intérêt des défauts sans prérequis

La proposition 2.7 a mis en évidence une classe particulière de défauts. Mais les défauts sont destinés à représenter des informations. Il est donc naturel d'essayer d'identifier, dans une optique de représentation des connaissances, ce qui différencie les défauts normaux de leur forme sans prérequis. De fait, la forme sans prérequis caractérise une connaissance permanente par opposition à une connaissance "situationnelle".

Pour illustrer ce propos, reprenons l'exemple des oiseaux et considérons les 2 défauts qui permettent d'exprimer que les oiseaux volent:

$$\delta = \frac{\text{oiseau}(x) : M\text{vole}(x)}{\text{vole}(x)} \qquad sp(\delta) = \frac{: M\text{oiseau}(x) \Rightarrow \text{vole}(x)}{\text{oiseau}(x) \Rightarrow \text{vole}(x)}$$

Disposer du défaut $sp(\delta)$ équivaut bien à savoir, à tout instant, que si x est un oiseau alors il vole. En revanche, avec δ, seulement lorsqu'un certain x est un oiseau, on sait qu'il vole.

A priori, on peut penser que l'une et l'autre forme puissent être employées indifféremment. Cependant, la proposition 2.7 indique que toute déduction par défaut faite à partir de la forme avec prérequis l'est aussi à partir de la forme sans prérequis. Sans chercher à caractériser formellement les déductions supplémentaires éventuellement obtenues par la forme sans prérequis, nous donnons maintenant deux exemples où, pour cette raison, les défauts avec prérequis se révèlent contre-intuitifs, à l'inverse des défauts sans prérequis.

Exemple:

$$W = \{ \neg\text{vole}(Max) \} \qquad \text{et} \qquad \Delta = \left\{ \frac{\text{oiseau}(x) \Rightarrow \text{vole}(x)}{\text{vole}(x)} \right\}$$

La seule extension est $E = W = \{ \neg\text{vole}(Max) \}$

Il n'y a pas d'extension où $\sim$oiseau(Max) soit vrai, alors qu'il faut pourtant s'attendre à ce que Max ne soit pas un oiseau (Puisque le défaut indique que les oiseaux volent).

Le défaut sans prérequis postulant que les oiseaux volent est $sp(\Delta) = \{ \frac{:M oiseau(x) \Rightarrow vole(x)}{oiseau(x) \Rightarrow vole(x)} \}$

Ici au contraire, l'extension de $(sp(\Delta),W)$ contient $\sim$oiseau(Max): $E'=Th(\{ \sim vole(Max), oiseau(Max) \Rightarrow vole(Max) \})$

Exemple:

Soient Aude et Guy, deux étudiants, respectivement en beaux-arts et dentaire. En cas de réussite dans ses études, chacun fera carrière dans son domaine:

diplomé(Aude) => artiste(Aude)

diplomé(Guy) => dentiste(Guy)

Si au moins l'un des deux doit subir ses examens avec succès,

diplomé(Aude) v diplomé(Guy)

d'où il est possible de déduire

artiste(Aude) v dentiste(Guy)

En général, les artistes, comme les dentistes, ne sont pas salariés: $\Delta = \{ \frac{artiste(x) \vee dentiste(x) :M \sim salarie(x)}{\sim salarie(x)} \}$

Si W est l'ensemble des formules résumant la situation de Aude et Guy, alors la seule extension de (Δ,W) est $E=Th(W)$.

Il n'existe aucune extension de (Δ,W) où quelqu'un (Anne ou Guy) exercera une profession non salariée. Or, W contient la formule

artiste(Aude) v dentiste(Guy)

et donc contient

artiste(Aude) v dentiste(Aude) v artiste(Guy) v dentiste(Guy)

Donc le prérequis de l'un des deux défauts δ_1 ou δ_2 est vrai.

$$\delta_1 = \frac{artiste(Aude) \vee dentiste(Aude) :M \sim salarie(Aude)}{\sim salarie(Aude)}$$

$$\delta_2 = \frac{artiste(Guy) \vee dentiste(Guy) :M \sim salarie(Guy)}{\sim salarie(Guy)}$$

Cependant, $\sim$salarie(Aude) v $\sim$salarie(Guy) n'appartient pas à l'extension E.

La forme sans prérequis est $sp(\Delta) = \{ \frac{:M artiste(x) \vee dentiste(x) \Rightarrow \sim salarie(x)}{artiste(x) \vee dentiste(x) \Rightarrow \sim salarie(x)} \}$

L'instanciation de ce défaut ouvert donne deux défauts δ_1 et δ_2 dont les conséquents sont

artiste(Aude) v dentiste(Aude) => ~salarie(Aude)

artiste(Guy) v dentiste(Guy) => ~salarie(Guy)

Ces deux implications peuvent se combiner.

artiste(Aude) v dentiste(Aude) v artiste(Guy) v dentiste(Guy) => ~salarie(Aude) v ~salarie(Guy)

L'antécédent de cette implication est vérifié par Th(W), la conclusion n'apporte pas d'inconsistance dans E', donc ~salarie(Aude) v ~salarie(Guy) est une formule de E', l'extension de (sp(Δ),W).

Tout défaut normal possède une forme sans prérequis. Partant d'une théorie avec défauts normaux, il est donc toujours possible de se ramener à une théorie avec défauts normaux sans prérequis. La proposition 2.7 et la discussion ci-dessus semblent indiquer que cette transformation mène à une théorie qui décrit la situation considérée d'une facon plus conforme à la réalité. En tout état de cause, cette rapide discussion devrait être replacée dans le cadre plus général d'une étude de l'adéquation des théories avec défauts pour la représentation des connaissances (voir [Rei 81]).

III UNE PROCEDURE DE PREUVE DECIDABLE

Dans ce chapitre, nous considérerons uniquement les théories avec défauts (Δ,W) telles que

-Δ est un ensemble de défauts sans prérequis

-les formules de W et de Consequents(Δ) sont en nombre fini et peuvent se mettre sous forme de clauses à variables prédéfinies

Nous aurons ainsi isolé un sous-ensemble de la logique des défauts. Nous indiquerons tout d'abord comment la saturation par ensembles peut être employée dans la recherche d'une preuve par défaut d'une formule. Nous en tirerons une procédure de preuve pour ce sous-ensemble de la logique des défauts. Nous démontrerons que cette procédure est saine, complète et décidable. Nous terminerons en donnant une procédure équivalente mais beaucoup plus efficace.

III 1 Principe

Partant de la définition d'une preuve par défaut, nous examinerons les opérations que doit effectuer une procédure de preuve. Parmi ces opérations, toutes celles qui concernent la production de clauses procèdent d'un même traitement lorsque la saturation par ensembles est utilisée. Nous en exposerons brièvement les raisons.

Rappelons que E est une extension d'une théorie (Δ,W) si et seulement si E est une extension de (DEFAUTS-CLOS(Δ),W) (cf II 5). Nous nous servirons de ce résultat car la procédure que nous développerons ne s'appliquera qu'aux seuls défauts clos.

Reprenons la définition d'une preuve par défaut d'une formule Q relativement à une théorie (Δ,W). D'après la remarque précédente, nous pouvons écrire:

Une suite finie $\Delta_0,\ldots,\Delta_k$ de sous-ensembles finis de DEFAUTS-CLOS(Δ) est une preuve par défaut de P ssi

1_ $W \cup \mathrm{Consequents}(\Delta_0) \vdash P$

2_ Pour $i=0..k$, $W \cup \mathrm{Consequents}(\Delta_i) \vdash \mathrm{Prerequis}(\Delta_{i-1})$

3_ $\Delta_k = \emptyset$

4_ $W \cup (\bigcup_{i=0}^{k} \mathrm{Consequents}(\Delta_i))$ est satisfiable

Les défauts considérés étant sans prérequis, il suit que trouver une preuve par défaut d'une formule Q relativement à une théorie (Δ,W) revient à

-Isoler un sous-ensemble Δ' de DEFAUTS-CLOS(Δ) tel que $W \cup \mathrm{Consequents}(\Delta') \vdash Q$

-Démontrer la consistance de W U Consequents(Δ')

La recherche d'une preuve par défaut comporte deux volets: la découverte d'une réfutation et une vérification de satisfiabilité. Pour chaque sous-ensemble Δ' sont à tester:

-l'inconsistance de W U Consequents(Δ') U $\{\tilde{}Q\}$

-la consistance de W U Consequents(Δ')

Avec l'emploi de la saturation par ensembles, ce double test devient:

-Ayant saturé W U Consequents(Δ'), tester si la clause vide y appartient

-Ayant saturé W U Consequents(Δ') U $\{\tilde{}Q\}$, tester si la clause vide y appartient

Ce traitement est à effectuer pour tout sous-ensemble Δ', donc en particulier pour DEFAUTS-CLOS(Δ) lui-même. Or par définition d'un ensemble saturé, W U Consequents(Δ') U $\{\tilde{}Q\}$ n'est saturé que s'il en va de même pour tous ses sous-ensembles. Le calcul de S(Ø,W U Consequents(DEFAUTS-CLOS(Δ)) U $\{\tilde{}Q\}$) englobe donc tous les autres. Une fois qu'il a été réalisé, il ne reste qu'à effectuer les tests (sur la clause vide) pour tous les sous-ensembles Δ'.

Les tests sont définis par rapport à un certain sous-ensemble Δ'. Pour qu'ils puissent avoir lieu, les clauses de S(Ø,W U Consequents(DEFAUTS-CLOS(Δ)) U $\{\tilde{}Q\}$) doivent donc être identifiées relativement aux différents sous-ensembles Δ'. A cette fin, nous allons introduire la notion de support d'ancêtres.

A chaque clause sera associé son support d'ancêtres (l'union des ensembles auxquels ses ancêtres appartiennent). Ce support d'ancêtres identifiera l'origine de la clause.

Définition:

Le support d'ancêtres d'une clause c sera noté SA(c) et défini par

-SA(c) = W si $c \in W$

-SA(c) = W U Consequents(δ_i) si $c \in$ Consequents(δ_i) ($\delta_i \in \{\delta_1,\dots,\delta_n\}$ = DEFAUTS-CLOS(Δ))

-SA(c) = W U $\{\tilde{}Q\}$ si $c \in \{\tilde{}Q\}$

-SA(c) = SA(c_1) U SA(c_2) si c est la résolvante sur littéral de tête de c_1 et c_2

Le rôle privilégié tenu par W rend compte de l'inclusion de W dans chacune des extensions.

Exemple:

$W = \{\tilde{}A \vee \tilde{}B\}$ et $\Delta = \{\frac{:MA}{A}, \frac{:MB}{B}\}$ et $\{\tilde{}Q\} = \{\tilde{}A\}$

Calculons $S(\emptyset, W \cup Consequents(\delta_1, \delta_2) \cup \{\sim Q\})$

$c_1 = \sim A \vee \sim B$ et $SA(c_1)=W$

$c_2 = A$ et $SA(c_2)=W \cup Consequents(\delta_1)$

$c_3 = r(c_1, c_2) = \sim B$ et $SA(c_3)=W \cup Consequents(\delta_1)$

$c_4 = B$ et $SA(c_4)=W \cup Consequents(\delta_2)$

$c_5 = r(c_3, c_4) = \Box$ et $SA(c_5)=W \cup Consequents(\delta_1, \delta_2)$

$c_6 = \sim A$ et $SA(c_6)=W \cup \{\sim Q\}$

$c_7 = r(c_2, c_6) = \Box$ et $SA(c_7)=W \cup Consequents(\delta_1) \cup \{\sim Q\}$

$E_0 = \{c_1\}$ est saturé et $E_0 \mid = \mid W$

$E_1 = \{c_1, c_2, c_3\}$ est saturé et $E_1 \mid = \mid W \cup Consequents(\delta_1)$

$E_2 = \{c_1, c_4\}$ est saturé et $E_2 \mid = \mid W \cup Consequents(\delta_2)$

$E_3 = \{c_1, c_2, c_3, c_4, c_5\}$ est saturé et $E_3 \mid = \mid W \cup Consequents(\delta_1, \delta_2)$

E_0, E_1, E_2, E_3 sont donc des ensembles saturés logiquement équivalents à $W \cup Consequents(\Delta')$ pour Δ' valant respectivement $\emptyset$, $\{\delta_1\}$, $\{\delta_2\}$, $\{\delta_1, \delta_2\}$. Attention, E_0, E_1, E_2, E_3 ne correspondent pas tous à des extensions (E_3 ne décrit aucune extension et E_0 décrit partiellement l'extension que E_1 décrit entièrement)

$R_0 = \{c_1, c_6\}$ est saturé et logiquement équivalent à $W \cup \{\sim Q\}$

$R_1 = \{c_1, c_2, c_3, c_6, c_7\}$ est saturé et logiquement équivalent à $W \cup Consequents(\delta_1) \cup \{\sim Q\}$

$R_2 = \{c_1, c_4, c_6\}$ est saturé et logiquement équivalent à $W \cup Consequents(\delta_2) \cup \{\sim Q\}$

$R_3 = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$ est saturé et $R_3 \mid = \mid W \cup Consequents(\delta_1, \delta_2) \cup \{\sim Q\}$

R_0, R_1, R_2, R_3 sont eux des ensembles saturés logiquement équivalents à $W \cup Consequents(\Delta') \cup \{\sim Q\}$ pour Δ' valant respectivement $\emptyset$, $\{\delta_1\}$, $\{\delta_2\}$, $\{\delta_1, \delta_2\}$.

Grâce à E_1 et R_1 nous avons la preuve que A est valide par défaut (A est vraie dans l'extension décrite par E_1). Remarquons que, pour tout sous-ensemble C de $S(\emptyset, W \cup Consequents(DEFAUTS\text{-}CLOS(\Delta)) \cup \{\sim Q\})$, C est logiquement équivalent à T si C regroupe toutes les clauses c telles que $SA(c) \subseteq T$

L'exemple qui précède a esquissé la technique qui va être employée par la procédure de preuve.

III 2 Une procédure de preuve par défaut

Notation: S(∅, W U Consequents(DEFAUTS-CLOS(Δ)) U {~Q}) sera désormais abrégé en C

Une procédure de preuve par défaut doit rechercher s'il existe un sous-ensemble Δ' de DEFAUTS-CLOS(Δ) tel que

-W U Consequents(Δ') U {~Q} est inconsistant

Ce qui correspond à la présence, dans C, d'une clause c vérifiant

-c = □

-{~Q} ⊂ SA(c)

-W U Consequents(Δ') est consistant

Ce qui correspond à l'absence, dans C, d'une clause c' vérifiant

-c' = □

-SA(c') ⊂ SA(c)

-{~Q} ⊄ SA(c')

La procédure de preuve par défaut P se déduit alors immédiatement.

Procédure de preuve P

-Calculer C

-Si C contient une clause c telle que

1) c = □

2) {~Q} ⊂ SA(c)

3) pour toute clause c' de C

.ou bien SA(c') ⊄ SA(c)

.ou bien {~Q} ⊂ SA(c')

.ou bien c' ≠ □

alors Q est vraie sinon Q est fausse

Exemple:

$$W=\{\ \tilde{}G\ ,\ \tilde{}H\ \} \quad \text{et} \quad \Delta=\left\{ \frac{:MG \vee K}{G \vee K}\ ,\ \frac{:MH \vee \tilde{}K}{H \vee \tilde{}K} \right\} \quad \text{et} \quad \{\tilde{}Q\}=\{\ K\ \}$$

$c_1 = \tilde{}G$ et $SA(c_1)=W$

$c_2 = \tilde{}H$ et $SA(c_2)=W$

$c_3 = G \vee K$ et $SA(c_3)=W \cup Consequents(\delta_1)$

$c_4 = r(c_1,c_3) = K$ et $SA(c_4)=W \cup Consequents(\delta_1)$

$c_5 = H \vee \tilde{}K$ et $SA(c_5)=W \cup Consequents(\delta_2)$

$c_6 = r(c_2,c_5) = \tilde{}K$ et $SA(c_6)=W \cup Consequents(\delta_2)$

$c_7 = r(c_4,c_6) = \square$ et $SA(c_7)=W \cup Consequents(\delta_1,\delta_2)$

$c_8 = K$ et $SA(c_8)=W \cup \{\tilde{}Q\}$

$c_9 = r(c_6,c_8) = \square$ et $SA(c_9)=W \cup Consequents(\delta_2) \cup \{\tilde{}Q\}$

Vérifions les conditions 1) à 3):

$-c_9 = \square$

$-\{\tilde{}Q\} \subset SA(c_9)$

-pour $c' = c_7$ et $c' = c_9$, on a $SA(c') \not\subset SA(c_9)$

pour toutes les autres clauses c', $c' \neq \square$

$\tilde{}K$ est donc vrai par défaut (Sa preuve par défaut étant $\{\delta_2\},\{\ \}$)

La procédure P a en effet déterminé que $W \cup Consequents(\delta_2) \cup \{K\}$ est inconsistant (grâce à l'existence de la clause c_9) et que $W \cup Consequents(\delta_2)$ est consistant (par l'absence de la clause vide parmi les clauses c_1,c_2,c_5,c_6)

La démonstration de complétude de P requiert les 2 lemmes ci-dessous.

Lemme 3.1:

Pour tout sous-ensemble Δ' de DEFAUTS-CLOS(Δ) il existe un sous-ensemble F de C tel que

i) F est saturé

ii) $F \mid = \mid W \cup Consequents(\Delta')$

iii) pour toute clause c de F, $SA(c) \subsetneq W \cup Consequents(\Delta')$

Preuve:

Il existe au moins un ensemble F satisfaisant ii) et iii) (par exemple $W \cup Consequents(\Delta')$)

Supposons que parmi les ensembles vérifiant ii) et iii) aucun ne satisfasse i)

Il existe donc au moins 2 clauses c_1 et c_2 telles que

-$SA(c_1) \subseteq W \cup Consequents(\Delta')$

-$SA(c_2) \subseteq W \cup Consequents(\Delta')$

-il existe une résolvante r sur le littéral de tête de c_1 et c_2 et elle n'est subsumée par aucune clause c tq

$SA(c) \subseteq W \cup Consequents(\Delta')$

En particulier, r n'est subsumée par aucun de ses ancêtres mâles.

Or c_1 et c_2 sont 2 clauses de C. D'après la définition de la saturation par ensembles et la propriété 1.5, $C = S(E_i,F_i)$ où E_i contient c_1 et F_i contient c_2 (ou l'inverse).

La résolvante r sur le littéral de tête n'étant subsumée par aucun de ses ancêtres mâles, est produite et ajoutée à F_i

Il existe donc r appartenant à C telle que $SA(r) = SA(c_1) \cup SA(c_2)$

Donc $SA(r) \subseteq W \cup Consequents(\Delta')$. Contradiction.

CQFD

La preuve du second lemme s'inspire d'une démonstration extraite de [Rei 80].

Lemme 3.2

Etant donné Δ' un sous-ensemble de DEFAUTS-CLOS(Δ), si $W \cup Consequents(\Delta')$ est consistant, alors il existe une extension de (DEFAUTS-CLOS(Δ),W) contenant $W \cup Consequents(\Delta')$

Preuve:

Construisons une telle extension.

Considérons la suite définie par

$E_0 = W$

$E_{i+1} = Th(E_i) \cup T_i$ où T_i est un ensemble maximal tel que

-$E_i \cup T_i$ est consistant

-si $R_i \in T_i$ alors DEFAUTS-CLOS(Δ) contient le défaut $\frac{:MR}{R}$

Notations: $E = \bigcup_{i=0}^{\infty} E_i$ et $DG_i = \{ R \mid \frac{:MR}{R} \in \text{DEFAUTS-CLOS}(\Delta)\ \&\ \sim R \notin E \}$

Pour montrer que E est une extension de (DEFAUTS-CLOS(Δ),W), il suffit de montrer que $T_i = DG_i$

Il est clair que $T_i \subseteq DG_i$ (Lemme 2.1)

Il reste à démontrer l'inclusion inverse.

Prenons l'hypothèse $T_i \neq DG_i$

Il existe donc une formule $R_i \in DG_i - T_i$

Par maximalité de T_i, $E_i \cup T_i \cup \{R\}$ est inconsistant, donc $T_i h(E_i) \cup T_i \cup \{R\}$ aussi ce qui entraine que $E_{i+1} \cup \{R\}$ est inconsistant.

Et puisque $E_{i+1} \subseteq E$, $E \cup \{R\}$ est inconsistant.

Or comme Th(E) = E, il faut donc que $\neg R \in E$ ce qui contredit l'hypothèse $R \in DG_i$

E est donc une extension de DEFAUTS-CLOS(Δ),W).

W U Consequents(Δ') étant consistant, il est possible de choisir T_0 tel que Consequents(Δ') $\subseteq T_0$

CQFD

A l'aide de ces lemmes, nous allons démontrer que la procédure P est saine et complète.

Proposition 3.3

La procédure P, appliquée à une théorie (Δ,W), évalue une formule Q à vrai ssi Q est valide dans une extension de (Δ,W)

Preuve:

=>

La procédure P ayant évalué Q à vrai, les points 1) à 3) sont satisfaits.

D'après 1) et 2), il existe une partie Δ' de DEFAUTS-CLOS(Δ) tel que W U Consequents(Δ') U $\{\neg Q\}$ est inconsistant.

(car il existe une clause c telle que c = □ et SA(c) = W U Consequents(Δ') U $\{\neg Q\}$)

Mais selon le point 3) la clause vide ne peut avoir une partie de W U Consequents(Δ') pour support d'ancêtres. Comme d'après le lemme 3.1 il existe un ensemble F saturé, logiquement équivalent à W U Consequents(Δ') et formé de clauses de C ayant une partie de W U Consequents(Δ') pour support d'ancêtres, alors W U Consequents(Δ') est consistant.

D'où W U Consequents(Δ') $\models$ Q

Et comme W U consequents(Δ') est consistant, d'après le lemme 3.2, il existe une extension de (DEFAUTS-CLOS(Δ),W), et donc de (Δ,W), qui contient W U Consequents(Δ')

<=

Soit E une extension de (Δ,W) dans laquelle Q est vraie

E est aussi une extension de (DEFAUTS-CLOS(Δ),W) (§ II 5)

Selon les propositions 2.3 et 2.4, E est consistant et logiquement équivalent à W U Consequents(DG(E))

Donc pour toute clause c', si SA(c') $\subseteq$ W U Consequents(DG(E)) alors c' est différente de la clause vide.

Le point 3) est alors établi pour le cas d'une clause c tq

-c est égale à la clause vide

-$\{ \tilde{}Q \} \subset SA(c) \subseteq W \cup Consequents(DG(E)) \cup \{ \tilde{}Q \}$

Il reste à montrer que les points 1) et 2) sont également satisfaits par une telle clause c.

Par le lemme 3.1, il existe un sous-ensemble F de C tq

-F est saturé

-$F \subseteq C$

-$F \; |=| \; W \cup Consequents(DG(E)) \cup \{ \tilde{}Q \}$

-pour toute clause c de F, $SA(c) \subseteq W \cup Consequents(DG(E)) \cup \{ \tilde{}Q \}$

Puisque Q est valide dans E, F contient la clause vide c et $SA(c) \subseteq W \cup Consequents(DG(E)) \cup \{ \tilde{}Q \}$

Or nous venons de voir que pour toute clause c, si $SA(c) \subseteq W \cup Consequents(DG(E))$ alors $c \neq \square$

Donc $\{ \tilde{}Q \} \subset SA(c)$ c'est-à-dire que les points 1) et 2) sont également vérifiés.

La procédure P évalue Q à vrai.

CQFD

Selon la proposition 1.6, l'ensemble C, de même que son calcul, est fini. (Dans l'en-tête de ce chapitre, nous nous sommes restreints à l'étude des théories (Δ,W) où Δ et W sont finis). Corollairement, le nombre de tests effectués par la procédure P est fini, ce qui entraine qu'elle se termine toujours.

Proposition 3.4:

La procédure P est décidable

III 3 Optimisation

Dans cette troisième section, nous nous intéresserons à l'efficacité de la procédure que nous venons de construire. Nous donnerons tout d'abord une autre procédure, qui minimise le nombre des clauses mises en jeu. Nous montrerons que cette procédure est équivalente à la première, ce qui lui permet de posséder la propriété de complétude. Nous conclurons également que cette procédure est elle-même décidable. Enfin nous discuterons des avantages de cette nouvelle procédure.

La procédure P est assez irréaliste. Son seul mérite est de procéder globalement, évitant ainsi dans le calcul des résolvantes toute redondance dûe à l'examen des différentes parties Δ' de DEFAUTS-CLOS(Δ). Cependant, C contient un certain nombre de clauses inutiles Considérons en effet 2 sous-ensembles Δ' et Δ'' de DEFAUTS-CLOS(Δ) tels que Δ'' soit une partie propre de Δ'. Lorsque la saturation de W U Consequents(Δ'') provoque l'apparition de la clause vide, il est inutile de continuer à produire les clauses provenant de la saturation de W U Consequents(Δ'). Ces clauses ne peuvent évidemment pas servir à prouver par défaut une formule étant donné l'insatisfiabilité de W U Consequents(Δ'). C'est sur cette base que nous allons restreindre l'ensemble des clauses utilisées par la procédure P.

Définition:

C* est l'ensemble des clauses c de C telles qu'il n'existe pas de clause c' de C vérifiant

-c' = □

-SA(c') ⊂ SA(c)

Cette définition exprime bien que de telles clauses c ne se déduisent pas d'un ensemble W U Consequents(Δ') qui contient W U Consequents(Δ'') duquel se dérive la clause vide c'.

Exemple:

$$W = \{\ \tilde{}G \vee \tilde{}H\ \} \qquad \text{et} \qquad \Delta = \left\{ \frac{:MF\ \&\ H}{F\ \&\ H}, \frac{:M\tilde{}F \vee G}{\tilde{}F \vee G}, \frac{:MG \vee K}{G \vee K} \right\} \qquad \text{et} \qquad \{\tilde{}Q\} = \{\tilde{}K\}$$

C se compose des 11 clauses ci-dessous:

C se compose des 11 clauses ci-dessous:

$c_1 = \neg G \vee \neg H$ avec $SA(c_1) = W$

$c_2 = F$ avec $SA(c_2) = W \cup Consequents(\delta_1)$

$c_3 = H$ avec $SA(c_3) = W \cup Consequents(\delta_1)$

$c_4 = \neg F \vee G$ avec $SA(c_4) = W \cup Consequents(\delta_2)$

$c_5 = r(c_2,c_4) = G$ avec $SA(c_5) = W \cup Consequents(\delta_1,\delta_2)$

$c_6 = r(c_1,c_5) = \neg H$ avec $SA(c_6) = W \cup Consequents(\delta_1,\delta_2)$

$c_7 = r(c_3,c_6) = \square$ avec $SA(c_7) = W \cup Consequents(\delta_1,\delta_2)$

$c_8 = \neg G \vee K$ avec $SA(c_8) = W \cup Consequents(\delta_3)$

$c_9 = r(c_5,c_8) = K$ avec $SA(c_9) = W \cup Consequents(\delta_1,\delta_2,\delta_3)$

$c_{10} = \neg K$ avec $SA(c_{10}) = W \cup \{\neg Q\}$

$c_{11} = r(c_9,c_{10}) = \square$ avec $SA(c_{11}) = W \cup Consequents(\delta_1,\delta_2,\delta_3) \cup \{\neg Q\}$

En revanche, C* n'a que 9 clauses: $C^* = C-\{c_9,c_{11}\}$

c_9 se déduit de $W \cup Consequents(\delta_1,\delta_2,\delta_3)$ qui ne peut décrire une extension puisque $W \cup Consequents(\delta_1,\delta_2)$ est inconsistant.

Puisque C-C* constitue l'ensemble des clauses qui ne peuvent pas intervenir dans une preuve par défaut, il parait naturel de construire une procédure P* en remplaçant, à l'intérieur de P, C par C*.

Procédure P*

-Calculer C*

-Si C* contient une clause c telle que

1*) $c = \square$

2*) $\{\neg Q\} \subset SA(c)$

3*) pour toute clause c' de C*

.soit $SA(c') \not\subset SA(c)$

.soit $\{\neg Q\} \subset SA(c')$

.soit $c' \neq \square$

alors Q est vraie sinon Q est fausse

La proposition suivante établit l'équivalence des procédures P et P*.

Proposition 3.5:

Etant donné une théorie (Δ,W), P évalue une formule Q à vrai ssi P* l'évalue à vrai

Preuve:

=>

Si P a évalué Q à vrai, alors il existe une clause c de C tq

$-c = \Box$

$-\{\neg Q\} \subset SA(c)$

-pour toute clause c' de C,

.soit $SA(c') \not\subset SA(c)$

.soit $\{\neg Q\} \subset SA(c')$

.soit $c' \neq \Box$

C'est-à-dire qu'il n'existe pas de clause c' de C telle que $SA(c') \subset SA(c)$ et $c' = \Box$

Donc c appartient à C*, et les points 1*) et 2*) se trouvent vérifiés.

3*) est immédiat car ce qui est vrai pour toute clause de C l'est à fortiori pour toute clause de C*

<=

Si P* a évalué Q à vrai alors C*, et donc C, contient une clause c tq $c = \Box$ et $\{\neg Q\} \subset SA(c)$

Les points 1) et 2) sont établis.

Le point 3) l'est aussi pour les clauses c' de C*

Il reste à établir le point 3) pour les clauses c' de C-C*

Vérifions tout d'abord rapidement que pour toute clause c' de C-C* il existe une clause c'' de C* tq

$-c'' = \Box$

$-SA(c'') \subset SA(c')$

Ceci est immédiat car sinon, par définition de C*, il existerait une suite infinie $c_1,\ldots,c_n\ldots$ de clauses de C-C* tq

$$\ldots \subset SA(c_n) \subset \ldots \subset SA(c_1) \quad \text{où } c_1 = c'$$

Supposons que pour une clause c' de C-C* on ait $SA(c') \subset SA(c)$

D'après ce qu'on vient de voir, il existe une clause c'' de C* telle que $c'' = \Box$ et $SA(c'') \subset SA(c')$

On aurait alors $SA(c'') \subset SA(c)$ avec $c'' = \Box$ ce qui contredit l'appartenance de c à C*

D'où pour toute clause c' de C-C*, $SA(c') \not\subset SA(c)$ ce qui établit définitivement le point 3*)

CQFD

La procédure P* ne présenterait pas vraiment d'intérêt s'il fallait calculer C pour obtenir C*. La fin de ce paragraphe décrit une méthode permettant un calcul direct de C*. Le principe consiste à faire de C* un sous-produit initial de la saturation par ensembles de W U Consequents(DEFAUTS-CLOS(Δ)) U { ~Q } (autrement dit le calcul de C), en imposant certaines contraintes à la fonction de sélection (Dans la procédure P, n'importe quelle fonction de sélection convient pour le calcul de C).

Nous allons reprendre la définition de la saturation par ensembles:

$C = S(E_i, F_i)$ avec $E_1 = \emptyset$ et $F_1 = W \cup \text{Consequents(DEFAUTS-CLOS}(\Delta)) \cup \{\sim Q\}$

c_i désigne la ième clause génératrice.

Avec les hypothèses que nous avons prises en début de chapitre, F_1 est fini et donc E_i et F_i aussi, d'après ce que nous avons vu au premier chapitre.

Définissons les ensembles suivants:

$G_i = \{ c \in F_i \ / \ \not\exists j \text{ tq } j<i \ \& \ c_j = \square \ \& \ SA(c_j) \subset SA(c) \}$

$H_i = \{ c \in G_i \ / \ \{\sim Q\} \not\subset SA(c) \}$

$I_i = \{ c \in H_i \ / \ \exists (j,k) \text{ tq } j<i \ \& \ k<i \ \& \ SA(c) \subset (SA(c_j) \cup SA(c_k)) \}$

Dans le cadre du calcul de C, nous considérerons désormais les fonctions de sélection f telles que

a) $\not\exists c \in F_i$ tq $SA(c) \subset SA(f(F_i))$

b) si $G_i \neq \emptyset$ alors $f(F_i) \in G_i$

c) si $H_i \neq \emptyset$ alors $f(F_i) \in H_i$

d) si $I_i \neq \emptyset$ alors $f(F_i) \in I_i$

Employer une fonction de sélection satisfaisant a) entraine que pour tout couple (c_i,c_j) de clauses génératrices, si $i<j$ alors $SA(c_j) \not\subset SA(c_i)$. (Les clauses de F_j descendent de clauses de F_i et pour toute clause c de F_i $SA(c) \not\subset SA(c_i)$)

D'où $G_i = \{ c \in F_i \ / \ \not\exists j \text{ tq } c_j = \square \ \& \ SA(c_j) \not\subset SA(c) \}$

G_i est un sous-ensemble de C*.De plus, toute clause de F_i qui n'appartient pas à G_i est une clause de C-C*. D'autre part, si G_i est vide, alors pour tout j supérieur à i, G_j aussi sera vide (Car les clauses de F_j descendent de clauses de F_i). Le respect de la condition b) implique dès lors que les i-1 premières clauses génératrices seront les seules à appartenir à C*. Lorsque le calcul de C est effectué avec une fonction de sélection vérifiant a) et b), C* est donc un sous-produit initial de C. Dans la pratique, il suffira d'interrompre la saturation par ensembles lorsque G_i est vide pour ne produire que les clauses de C*.

D'après les résultats 3.3 et 3.5 et la discussion qui précède, il vient:

Proposition 3.6:

La procédure P* est saine, complète et décidable

Preuve:

Il reste á démontrer que a), b), c) et d) ne sont pas incompatibles.

Considérons une clause c de F_i qui n'appartient pas á G_i

Alors si $f(F_i)$ appartient á G_i, il est clair que $SA(c) \not\subset SA(f(F_i))$

Considérons une clause c de G_i qui n'appartient pas á H_i

Alors si $f(F_i)$ appartient á H_i, il est clair que $SA(c) \not\subset SA(f(F_i))$

Considérons une clause c de H_i qui n'appartient pas á I_i

Si $f(F_i)$ appartient á I_i, alors $SA(c) \not\subset SA(f(F_i))$

(Sinon $SA(c) \subset SA(f(F_i)) \subsetneq SA(c_j) \cup SA(c_k)$ ce qui contredit l'hypothèse $c \in I_i$)

Finalement, si $f(F_i)$ est une clause de I_i, alors pour toute clause c de $F_i - I_i$, $SA(c) \not\subset SA(f(F_i))$

$f(F_i)$ est donc une clause de I_i telle que pour toute clause c de I_i, $SA(c) \not\subset SA(f(F_i))$

Si I_i n'est pas vide, une telle clause existe toujours car I_i est un sous-ensemble de F_i, qui est fini.

Pour le cas où I_i est vide, il suffit d'effectuer un raisonnement analogue sur H_i

Donc une clause $f(F_i)$ qui ne falsifie pas a), b), c) et d) existe toujours.

CQFD

L'intérêt de la condition c) est illustré par la propriété ci-dessous:

$S(\emptyset, W \cup Consequents(DEFAUTS\text{-}CLOS(\Delta)) \cup \{\sim Q\}) \ |{=}|\ S(\ S(\emptyset, W \cup Consequents(DEFAUTS\text{-}CLOS(\Delta))\ ),\ \{\sim Q\}\ )$

La contrainte c) privilégie les clauses de C* qui ne sont pas issues (ni directement ni par suite de résolution) de la négation de la question. Ces clauses forment alors une sous-suite initiale de la suite des clauses génératrices. Puisque ces clauses ne dépendent pas de la question, elles serviront en pratique á l'évaluation de toutes les questions. Le calcul de C* débutera par la production de ces clauses, qu'il suffira d'effectuer une fois pour toutes. Pour chaque question, le calcul de C* s'appuiera sur cet ensemble de clauses et sera ainsi considérablement réduit.

La procédure P* est incrémentale lorsque la fonction de sélection employée satisfait le critère d):

$S(\ \emptyset\ ,\ W \cup Consequents(\delta_1,\delta_2)\ ) = S(\ S(\ \emptyset\ ,\ W\ )\ ,\ Consequents(\delta_1,\delta_2)\ )$

$S(\ S(\ \emptyset\ ,\ W\ )\ ,\ Consequents(\delta_1,\delta_2)\ ) = S(\ S(\ S(\ \emptyset\ ,\ W\ )\ ,\ Consequents(\delta_1)\ )\ ,\ Consequents(\delta_2)\ )$...

En pratique, l'adjonction d'un nouveau défaut se fera donc très économiquement par une simple "étape" de saturation.

III 4 Exemple

Nous terminerons ce chapitre en donnant un exemple d'application de la procédure P* dans un univers de cubes. Partant d'une disposition donnée des cubes A, B, C, D, il s'agira de déterminer s'il est possible d'édifier des escaliers.

Nous dirons qu'il existe un escalier menant au cube z s'il existe un escalier menant à un cube x dont la surface est dégagée et si ce cube x est placé à côté du cube y qui supporte z.

$$w_1 = \tilde{~}Escalier(x) \vee \tilde{~}Acote(x,y) \vee \tilde{~}Sur(z,y) \vee \tilde{~}Libre(x) \vee Escalier(z)$$

Lorsqu'un cube est "libre", aucun cube n'est placé dessus.

$$w_2 = \tilde{~}Libre(y) \vee \tilde{~}Sur(x,y)$$

La relation "est placé à côté de" est symétrique.

$$w_3 = \tilde{~}Acote(x,y) \vee Acote(y,x)$$

Dans la situation étudiée, les cubes A et B sont seuls sur la table, et placés côte à côte.

$$w_4 = Acote(A,B)$$

Dans ces conditions, il existe donc un escalier pour A et un pour B.

$$w_5 = Escalier(A)$$

$$w_6 = Escalier(B)$$

Ces clauses, qui constituent une base intangible pour le monde étudié, forment W.

A moins que l'inverse ne puisse être prouvé, les cubes A et B sont supposés libres.

d_1 = Libre(A) (d_1 est le consequent de $\delta_1 = \frac{:MLibre(A)}{Libre(A)}$)

d_2 = Libre(B) (d_2 est le consequent de $\delta_2 = \frac{:MLibre(B)}{Libre(B)}$)

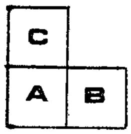
Nous allons essayer, en posant C sur A et D sur B, d'édifier des escaliers pour C et D. Les 2 seules actions autorisées correspondent aux 2 défauts ci-dessous.

d_3 = Sur(C,A) (d_3 est le consequent du défaut $\delta_3 = \frac{:MSur(C,A)}{Sur(C,A)}$)

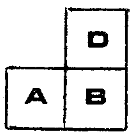
d_4 = Sur(D,B) (d_4 est le consequent du défaut $\delta_4 = \frac{:MSur(D,B)}{Sur(D,B)}$)

La théorie avec défauts $(\{\delta_1,\delta_2,\delta_3,\delta_4\},W)$ possède 4 extensions:

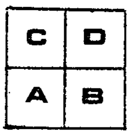
-> E_1 = Th(W U Consequents(δ_1,δ_2)) qui est représenté par les clauses dont le support d'ancêtres est inclus dans W U Consequents(δ_1,δ_2)



-> E_2 = Th(W U Consequents(δ_2,δ_3)) qui est représenté par les clauses dont le support d'ancêtres est inclus dans W U Consequents(δ_2,δ_3)



-> E_3 = Th(W U Consequents(δ_1,δ_4)) qui est représenté par les clauses dont le support d'ancêtres est inclus dans W U Consequents(δ_1,δ_4)



-> E_4 = Th(W U Consequents(δ_3,δ_4)) qui est représenté par les clauses dont le support d'ancêtres est inclus dans W U Consequents(δ_3,δ_4)

Nous employerons la procédure P* pour répondre à un certain nombre de questions concernant cet univers de cubes. Il découle des contraintes a), b), c), d) que les clauses génératrices ayant même support d'ancêtres seront consécutives, ce qui va faciliter la description du calcul de C*. E_1 étant vide, E_i est constitué des i-1 premières clauses génératrices. Le processus de saturation par ensembles consistera donc à calculer la résolvante de la clause génératrice courante avec chacune des clauses génératrices qui la précèdent.

$E_1=\emptyset$ et $F_1=G_1=H_1=\{w_1,w_2,w_3,w_4,w_5,w_6,d_1,d_2,d_3,d_4\}$ et $I_1=\emptyset$

Clauses de support d'ancêtres W

c_1 = ~Acote(x,y) v ~Escalier(x) v ~Libre(x) v ~Sur(z,y) v Escalier(z)

$I_2=\{w_2,w_3,w_4,w_5,w_6\}$

c_2 = ~Sur(x,y) v ~Libre(y)

$I_3=\{w_3,w_4,w_5,w_6\}$

c_3 = ~Acote(x,y) v Acote(y,x)

$I_4=\{w_4,w_5,w_6\}$

c_4 = Acote(A,B)

$I_5=\{r(c_1,c_4),r(c_3,c_4),w_5,w_6\}$

c_5 = ~Escalier(A) v ~Libre(A) v ~Sur(z,B) v Escalier(z) (c_5 est la résolvante de c_1 et c_4)

$I_6=\{r(c_3,c_4),w_5,w_6\}$

c_6 = Acote(B,A) (c_6 est la résolvante de c_3 et c_4)

La clause $r(c_3,c_6)$ n'est pas produite car subsumée par c_4

(Dans une résolvante, l'ancêtre mâle a l'indice le plus élevé)

$I_7=\{r(c_1,c_6),w_5,w_6\}$

c_7 = ~Escalier(B) v ~Libre(B) v ~Sur(z,A) v Escalier(z) (c_7 est la résolvante de c_1 et c_6)

$I_8=\{w_5,w_6\}$

c_8 = Escalier(A)

$I_9=\{r(c_5,c_8),w_6\}$

c_9 = ~Libre(A) v ~Sur(z,B) v Escalier(z) (c_9 est la résolvante de c_5 et c_8)

$I_{10}=\{w_6\}$

c_{10} = Escalier(B)

$I_{11}=\{r(c_7,c_{10})\}$

c_{11} = ~Libre(B) v ~Sur(z,A) v Escalier(z) (c_{11} est la résolvante de c_7 et c_{10})

$H_{12}=\{d_1,d_2,d_3,d_4\}$ et $I_{12}=\emptyset$

Clauses de support d'ancêtres W U Consequents(δ_1)

c_{12} = Libre(A)

$I_{13}=\{r(c_9,c_{12})\}$

c_{13} = ~Sur(z,B) v Escalier(z) (c_{13} est la résolvante de c_9 et c_{12})

$H_{14}=\{d_2,d_3,d_4\}$ et $I_{14}=\emptyset$

Clauses de support d'ancêtres W U Consequents(δ_2)

c_{14} = Libre(B)

$I_{15}=\{r(c_{11},c_{14})\}$

c_{15} = ~Sur(z,A) v Escalier(z) (c_{15} est la résolvante de c_{11} et c_{14})

$H_{16}=\{d_3,d_4\}$ et $I_{16}=\emptyset$

Clauses de support d'ancêtres W U Consequents(δ_3)

c_{16} = Sur(C,A)

$I_{17}=\{r(c_2,c_{16}),r(c_{15},c_{16})\}$

c_{17} = ~Libre(A) (c_{17} est la résolvante de c_2 et c_{16})

$I_{18}=\{r(c_{15},c_{16}),r(c_{12},c_{17})\}$

Clauses de support d'ancêtres W U Consequents(δ_1,δ_3)

$c_{18} = \square$ (c_{18} est la résolvante de c_{12} et c_{17})

$I_{19}=\{r(c_{15},c_{16})\}$

Clauses de support d'ancêtres W U Consequents(δ_2,δ_3)

c_{19} = Escalier(C) (c_{19} est la résolvante de c_{15} et c_{16})

$H_{20}=\{d_4\}$ et $I_{20}=\emptyset$

Clauses de support d'ancêtres W U Consequents(δ_4)

c_{20} = Sur(D,B)

$I_{21}=\{r(c_2,c_{20}),r(c_{13},c_{20})\}$

c_{21} = ~Libre(B) (c_{21} est la résolvante de c_2 et c_{20})

$I_{22}=\{r(c_{13},c_{20}),r(c_{14},c_{21})\}$

Clauses de support d'ancêtres W U Consequents(δ_1,δ_4)

c_{22} = Escalier(D) (c_{22} est la résolvante de c_{13} et c_{20})

$I_{23}=\{r(c_{14},c_{21})\}$

Clauses de support d'ancêtres W U Consequents(δ_2,δ_4)

$c_{23} = \square$ (c_{23} est la résolvante de c_{14} et c_{21})

$I_{24} = H_{24} = \emptyset$

Ces 23 premières clauses résulte de la vérification de consistance des extensions: Th(W U Consequents(Δ')) n'est pas une extension si Δ' contient δ_1 et δ_3 (cf. la clause c_{18}) ou si Δ' contient δ_2 et δ_4 (cf. la clause c_{23})

Pour répondre à des questions sur ce monde de cubes, il suffit de reprendre la saturation par ensembles là où elle a été interrompue, en introduisant les clauses issues de la négation de la question.

Question 1: Peut-il y avoir un escalier menant à C?

$G_{24}=\{q_1\}$ et $I_{24}=H_{24}=\emptyset$ avec q_1 = ~Escalier(C)

c_{24} = ~Escalier(C) (SA(c_{24})=W U {~Q_1})

$I_{25}=\{r(c_{19},c_{24})\}$

$c_{25} = \square$ (c_{25} est la résolvante de c_{19} et c_{24})

$I_{26} = H_{26} = G_{26} = \emptyset$

G_{26} est vide, toutes les clauses de C* ont donc été produites.

c_{25} est une clause de C* telle que:

-$c_{25} = \square$

-$\{ \tilde{\ }Q_1 \} \subset SA(c_{25})$

-pour toute clause c' de C*

si $c' \neq c_{18}$ et $c' \neq c_{23}$ alors $c' \neq \square$

si $c' = c_{18}$ ou $c' = c_{23}$ alors $SA(c') \not\subset SA(c_{25})$

La réponse est positive. Il est possible de construire un escalier allant à C (Si l'on pose C sur A).

Question 2: Peut-il y avoir un escalier menant à D?

$G_{24} = \{q_2\}$ et $I_{24} = H_{24} = \emptyset$ avec $q_2 = \tilde{\ }$Escalier(D)

$c_{24} = \tilde{\ }$Escalier(D) $(SA(c_{24} = W \cup \{\tilde{\ }Q_2\})$

$I_{25} = \{r(c_{22}, c_{24})\}$

$c_{25} = \square$ (c_{25} est la résolvante de c_{22} et c_{24})

$I_{26} = H_{26} = G_{26} = \emptyset$

Ici encore, la réponse est oui. Il est possible de construire un escalier allant à D (En posant D sur B).

Question 3: Peut-il y avoir en même temps un escalier pour C et un pour D?

$G_{24} = \{q_3\}$ et $I_{24} = H_{24} = \emptyset$ avec $q_3 = \tilde{\ }$Escalier(C) v $\tilde{\ }$Escalier(D)

$c_{24} = \tilde{\ }$Escalier(C) v $\tilde{\ }$Escalier(D) $(SA(c_{24}) = W \cup \{\tilde{\ }Q_3\})$

$I_{25} = \{r(c_{19}, c_{24})\}$

$c_{25} = \tilde{\ }$Escalier(D) (c_{25} est la résolvante de c_{19} et c_{24})

$F_{26} = \{r(c_{22}, c_{25})\}$ et $G_{26} = H_{26} = I_{26} = \emptyset$

$r(c_{22}, c_{25})$ n'est pas une clause de G_{26} (et donc pas de C*) car

$SA(r(c_{22}, c_{25})) = W \cup Consequents(\delta_1, \delta_2, \delta_3, \delta_4) \cup \{\tilde{\ }Q_3\}$

Or $c_{18} = \square$ et $SA(c_{18}) = W \cup Consequents(\delta_1, \delta_3) \subset SA(r(c_{22}, c_{25}))$

Il n'y a pas de réfutation dans C* pour $\tilde{\ }Q_3$: les 2 escaliers ne peuvent coexister

Finalement, il ressort de ces 3 interrogations que la construction d'un escalier pour C et celle d'un escalier pour D ne sont réalisables qu'isolément.

Admettons que l'on veuille de plus supposer que C est libre. Ceci se fait très facilement, en reprenant la saturation par ensembles à partir du point où H_i est vide.

d_5 = Libre(C) (d_5 est le consequent de $\delta_5 = \frac{:\text{MLibre(C)}}{\text{Libre(C)}}$)

$H_{24} = \{d_5\}$ et $I_{24} = \emptyset$

c_{24} = Libre(C)

$H_{25} = I_{25} = \emptyset$

La nouvelle théorie avec défauts $(\{\delta_1,\delta_2,\delta_3,\delta_4,\delta_5\},W)$ est représentée par les clauses c_1 à c_{25}

Question 4: Vérifions qu'il n'y a aucun cube sur C ((x) ~Sur(x,C) ?)

$G_{25} = \{q_4\}$ et $H_{25} = I_{25} = \emptyset$ avec q_4 = Sur(f,C) (f fonction de Skolem)

c_{25} = Sur(f,C)

$I_{26} = \{r(c_2,c_{25})\}$

c_{26} = ~Libre(C) (c_{26} est la résolvante de c_2 et c_{25})

$I_{27} = \{r(c_{24},c_{26})\}$

$c_{27} = \square$ (c_{27} est la résolvante de c_{24} et c_{26})

$I_{28} = H_{28} = G_{28} = \emptyset$

Une réfutation existe dans C*, donc aucun cube ne se trouve sur C.

L'emploi d'une théorie avec défauts nous a permis d'exprimer le problème avec peu d'axiomes, sans avoir recours à la notion d'état résultant du déplacement d'un cube. Enfin, nous avons pu déterminer que les actions de poser C sur A et D sur B contribuaient à l'édification de 2 escaliers différents, mais mutuellement exclusifs.

CONCLUSION

Dans ce document, nous avons décrit une étude consacrée à la logique des défauts. En nous appuyant sur 2 exemples, nous avons dégagé une classe particulière de défauts, dits sans prérequis, permettant selon nous une représentation plus naturelle des connaissances. Dans la perspective d'une mise en oeuvre, nous nous sommes intéressé à la décidabilité de la logique des défauts. Cette étude nous a conduit à considérer une classe solvable du premier ordre, les clauses à variables prédéfinies. Nous avons défini la saturation par ensembles, qui généralise la saturation, une procédure de preuve basée sur la résolution et décidable pour les clauses à variables prédéfinies. Nous basant sur la saturation par ensembles, nous avons élaboré une procédure de preuve décidable pour le sous-ensemble de la logique des défauts délimité par les clauses à variables prédéfinies et les défauts sans prérequis.

La propriété de semi-monotonie de la logique des défauts est pleinement exploitée car l'ajout de défauts se fait très efficacement. Ce qui laisse supposer une aptitude particulière de la procédure dans un contexte d'évolution de théories ("beliefs revision" plus précisément en représentation des connaissances).

La saturation par ensembles, et donc la procédure de preuve par défaut, est incrémentale: les clauses qui ont été produites durant la recherche d'une réfutation d'un ensemble de clauses participent à la recherche d'une réfutation d'un sur-ensemble quand l'ensemble examiné s'est révélé consistant. Cette propriété contribue pour une part importante à l'efficacité de la procédure. Considérons en effet une formule qui ne soit valide dans aucune extension. Pour le vérifier, il faudra bien, quelle que soit la stratégie de résolution retenue, évaluer la formule dans chacune des extensions (en supposant déjà que celles-ci soient connues). La saturation par ensembles effectue aussi l'évaluation de la formule dans chacune des extensions, mais détermine au passage quelles sont les extensions de la théorie. Dans ce cas de figure, la différence se situe uniquement au niveau de l'efficacité relative de la saturation par ensembles et des meilleures méthodes de résolution. Cependant, pour les questions suivantes, la saturation par ensembles n'aura plus besoin de s'exécuter entièrement. Elle pourra réutiliser, parmi les clauses qu'elle a produite précédemment, toutes celles qui ne dépendent pas de la question, sans avoir à les dériver de nouveau. Ce procédé demande évidemment une plus grande place mémoire.

La mise à jour des axiomes n'est pas très aisée, ce qui constitue un des principaux inconvénients de l'approche adoptée. Etant donné le caractère non-monotone de la logique des défauts, il parait difficile de résoudre ce problème de manière vraiment satisfaisante. Il est en tout cas exclus que les transitions d'une théorie avec défauts à l'autre puissent toujours relever d'un traitement purement incrémental.

Bien qu'elle permette de traiter les théories avec défauts ouverts, la procédure n'opère réellement que sur des défauts clos. L'instanciation des défauts ouverts de la théorie augmente donc le nombre de clauses mises en jeu, dans une proportion d'autant plus grande que le nombre de constantes apparaissant dans la théorie est important. Un prolongement naturel du travail présenté ici consisterait donc à généraliser la procédure pour qu'elle prenne effectivement en compte les défauts ouverts, mais aussi pour qu'elle admette une classe plus large de formules.

Enfin, il reste deux problèmes, que partage tout système d'interrogation basé sur le principe de résolution. Il s'agit du traitement de l'égalité (qui, bien que résolu en théorie, n'a toujours pas de solution calculatoirement applicable) et de l'extraction des réponses (qu'il serait intéressant d'étudier dans le cadre d'un système solvable -tel que celui des clauses à variables prédéfinies- car alors déterminer si une réponse est la plus générale possible est certainement décidable).

REFERENCES

[AI 80] Special issue on non-monotonic logic

Artificial Intelligence, vol. 13, 1980

[Bos 81] Bossu, G. & Siégel, P.

La saturation au secours de la non-monotonie

Thèse de 3ème cycle, Université d'Aix-Marseille 2, 1981

[Far 81] Farinas del Cerro, L.

Déduction automatique et logique modale

Thèse d'état, Université de Paris 7, 1981

[Fit 73] Fitting, M. C.

Model existence theorems for modal and intuitionistic logics

Journal of Symbolic Logic, vol. 38, pp. 613-628, 1973

[Gab 82] Gabbay, D. M.

Intuitionistic basis for non-monotonic logic

Lecture Notes in Computer Science, 138, pp. 260-273, 1982

[Gol 81] Goldfarb, D.

The undecidability of the second order unification problem

Theoretical Computer Science, vol. 13, pp. 225-230, 1981

[Gil 60] Gilmore, P. C.

A proof method for quantification theory

IBM Jl. Res. Devl., no 4, pp. 28-35, 1960

[Gre 69] Green, C. C.

Application of theorem-proving to problem solving

IJCAI 1, pp. 219-239, 1969

[Hay 69] Hayes, P. J. & McCarthy, J.

Some philosophical problems from the standpoint of Artificial Intelligence

Machine Intelligence 4, pp. 463-502, 1969

[Hue 76] Huet, G.

Résolution d'équations dans les langages d'ordre 1,2,...ω

Thèse d'état, Université de Paris 7, 1976

[Kle 67] Kleene, S. C.

Mathematical logic

John Wiley and Sons, New York, 1967

[Kow 69] Kowalski, R. & Hayes, P. J.

Semantic trees in automatic theorem-proving

Machine Intelligence 4, pp. 87-101, 1969

[McC 68] McCarthy, J.

Programs with common sense

Semantic information processing, M. Minsky (Ed.), 1968

[McC 80] McCarthy, J.

Circumscription-A form of non-monotonic reasoning

Artificial Intelligence, vol. 13, pp. 27-39, 1980

[McD 80] McDermott, D. & Doyle, J.

Non-monotonic logic I

Artificial Intelligence, vol. 13, pp. 41-72, 1980

[Mc D 82] McDermott, D.

Non-monotonic logic II

JACM, vol. 29, 1, pp. 33-57, 1982

[New 56] Newell, A. & Simon, H. A.

The logic theory machine

IRE Transactions on Information Theory, pp. 61-79, 1956

[Rap 71] Raphael, B.

The frame problem in problem-solving systems

Artificial Intelligence and Heuristic Programming

Findler & Meltzer (Eds.), American Elsevier, 1971

[Rei 80] Reiter, R.

A logic for default reasoning

Artificial Intelligence, vol. 13, pp. 81-132, 1980

[Rei 81] Reiter, R. & Criscuolo, G.

On interacting defaults

IJCAI 7, pp. 270-276, 1981

[Rob 65] Robinson, J. A.

A machine-oriented logic based on the resolution principle

JACM, vol. 12, 1, pp. 23-41, 1965

[Rob 69] Robinson, J. A.

Mechanizing higher-order logic

Machine Intelligence 4, pp. 151-170, 1969

Liste des Publications Internes IRISA

PI 150 **Construction automatique et évaluation d'un graphe d'«implication» issu de données binaires, dans le cadre de la didactique des mathématiques**
H. Rostam , 112 pages ; Juin 1981

PI 151 **Réalisation d'un outil d'évaluation de mécanismes de détection de pannes]-]Projet Pilote SURF**
B. Decouty, G. Michel, C. Wagner, Y. Crouzet , 59 pages ; Juillet 1981

PI 152 **Règle maximale**
J. Pellaumail , 18 pages ; Septembre 1981

PI 153 **Corrélation partielle dans le cas « qualitatif »**
I.C. Lerman , 125 pages ; Octobre 1981

PI 154 **Stability analysis of adaptively controlled not-necessarily minimum phase systems with disturbances**
Cl. Samson , 40 pages ; Octobre 1981

PI 155 **Analyses d'opinions d'instituteurs à l'égard de l'appropriation des nombres naturels par les élèves de cycle préparatoire**
R. Gras , 37 pages ; Octobre 1981

PI 156 **Récursion induction principle revisited**
G. Boudol, L. Kott , 49 pages ; Décembre 1981

PI 157 **Loi d'une variable aléatoire à valeur R^+ réalisant le minimum des moments d'ordre supérieur à deux lorsque les deux premiers sont fixés**
M.Kowalowka, R. Marie , 8 pages ; Décembre 1981

PI 158 **Réalisations stochastiques de signaux non stationnaires, et identification sur un seul échantillon**
A. Benveniste J.J. Fuchs , 33 pages ; Mars 1982

PI 159 **Méthode d'interprétation d'une classification hiérarchique d'attributs-modalites pour l'«explication» d'une variable ; application à la recherche de seuil critique de la tension artérielle systolique et des indicateurs de risque cardiovasculaire**
B. Tallur , 34 pages ; Janvier 1982

PI 160 **Probabilite stationnaire d'un réseau de files d'attente multiclasse à serveur central et à routages dépendant de l'état**
L.M. Le Ny , 18 pages ; Janvier 1982

PI 161 **Détection séqentielle de changements brusques des caractéristiques spectrales d'un signal numérique**
M. Basseville, A. Benveniste , pages ; Mars 1982

PI 162 **Actes regroupés des journées de Classification de Toulouse (Mai 1980), et de Nancy (Juin 1981)**
I.C. Lerman , 304 pages ;

PI 163 **Modélisation et Identification des caractéristiques d'une structure vibratoire : un problème de réalisation stochastique d'un grand système non stationnaire**
M. Prévosto, A. Benveniste, B. Barnouin , 46 pages ; Mars 182

PI 164 **An enlarged definition and complete axiomatization of observational congruence of finite processes**
Ph. Darondeau , 45 pages ; Avril 1982

PI 165 **Accès vidéotex à une banque de données médicales**
A. Chauffaut, M. Dragone, R. Rivoire, J.M. Roger , 25 pages ; Mai 1982

PI 166 **Comparaison de groupes de variables définies sur le même ensemble d'individus**
B. Escofier, J. Pages , 115 pages ; Mai 1982

PI 167 **Transport en circuits virtuels internes sur réseau local et connexion Transpac**
M. Tournois, R. Trépos , 90 pages ; Mai 1982

PI 168 **Impact de l'intégration sur le traitement automatique de la parole**
P. Quinton , 14 pages ; Mai 1982

PI 169 **A systolic algorithm for connected word recognition**
J.P. Banâtre, P. Frison, P. Quinton , 13 pages ; Mai 1982

PI 170 **A network for the detection of words in continuous speech**
J.P. Banâtre, P. Frison, P. Quinton , 24 pages ; Mai 1982

PI 171 **Le langage ADA : Etude bibliographique**
J. André, Y. Jégou, M. Raynal , 12 pages ; Juin 1982

PI 172 **Comparaison de groupes de variables : 2ème partie : un exemple d'application**
B. Escofier, J. Pajès , 37 pages ; Juillet 1982

PI 173 **Unfold-fold program transformations**
L. Kott , 29pages ; Juillet 1982

PI 174 **Remarques sur les langages de parenthèses**
J.M. Autebert, J. Beauquier, L. Boasson, G. Senizergues , 20 pages ; Juillet 1982

PI 175 **Langages de parenthèses, langages N.T.S. et homomorphismes inverses**
J.M. Autebert, L. Boasson, G. Senizergues , 26 pages ; Juillet 1982

PI 176 **Tris pour machines synchrones ou Baudet Stevenson revisited**
R. Rannou , 26 pages ; Juillet 1982

PI 177 **Un nouvel algorithme de classification hiérarchique des éléments constitutifs de tableau de contingence basé sur la corrélation**
B. Tallur , Juillet 1982 ;

PI 178 **Programmes d'analyse des re'sultats d'une classification automatique**
I.C. Lerman et collaborateurs , 79 pages ; Septembre 1982

PI 179 **Attitude à l'égard des mathématiques des élèves de sixième**
J.Degouys, R. Gras, M. Postic , 29 pages ; Septembre 1982

PI 180 **Traitements de textes et manipulations de documents : bibliographie analytique**
J. André , 20 pages ; Septembre 1982

PI 181 **Algorithme assurant l'insertion dynamique d'un processeur autour d'un réseau à diffusion et garantissant la cohérence d'un système de numérotation des paquets global et réparti**
Annick Le Coz, Hervé Le Goff, Michel Ollivier , 31 pages ; Octobre 1982

PI 182 **Interprétation non linéaire d'un coefficient d'association entre modalités d'une juxtaposition de tables de contingence**
Israël César Lerman , 34 pages ; Novembre 1982

PI 183 **L'IRISA vu à travers les stages effectués par ses étudiants de DEA (1è année de thèse)**
Daniel Herman , 41 pages ; Novembre 1982

PI 184 **Commande non linéaire robuste des robots manipulateurs**
Claude Samson , 52 pages ; Janvier 1983

PI 185 **Dialogue et représentation des informations dans un système de messagerie intelligent**
Philippe Besnard, René Quiniou, Patrice Quinton, Patrick Saint-Dizier, Jacques Siroux, Laurent Trilling , 45 pages ; Janvier 1983

PI 186 **Analyse classificatoire d'une correspondance multiple ; typologie et régression**
I.C. Lerman , 54 pages ; Janvier 1983

PI 187 **Estimation de mouvement dans une sequence d'images de télévision en vue d'un codage avec compensation de mouvement**
Claude Labit , 132 pages ; Janvier 1983

PI 188 **Conception et réalisation d'un logiciel de saisie et restitution de cartes élémentaires**
Eric Sécher , 45 pages ; Janvier 1983

PI 189 **Etude comparative d'algorithmes pour l'amélioration de dessins au trait sur surfaces point par point**
M.A. ROY , 96 pages ; Janvier 1983

PI 190 **Généralisation de l'analyse des correspondances à la comparaison de tableaux de fréquence**
Brigitte Escofier , 35 pages ; Mars 1983

PI 191 **Association entre variables qualitatives ordinales «nettes» ou «floues»**
Israël-César Lerman , 42 pages ; Mars 1983

PI 192 **Un processeur intégré pour la reconnaissance de la parole**
Patrice Frison , 80 pages ; Mars 1983

PI 193 **The Systematic Design of Systolic Arrays**
Patrice Quinton , 39 pages ; Avril 1983

PI 194 **Régime stationnaire pour une file M/H/1 avec impatience**
Raymond Marie et Jean Pellaumail , 8 pages ; Mars 1983

PI 195 **SIGNAL : un langage pour le traitement du signal**
Paul Le Guernic, Albert Benveniste, Thierry Gautier , 49 pages ; Mars 1983

PI 196 **Algorithmes systoliques : de la théorie à la pratique**
Françoise André, Patrice Frison, Patrice Quinton , 19 pages ; Mars 1983

PI 197 **HAVANE : un système de mise en relation automatique de petites annonces**
Patrick Bosc, Michèle Courant, Sophie Robin, Laurent Trilling , 79 pages ; Mai 1983

PI 198 **Une procédure de décision en logique non-monotone**
Philippe Besnard , 59 pages ; Mai 1983

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique