



Formalisation de l'adaptation conservatrice dans les espaces métriques

Julien Cojan, Jean Lieber

► **To cite this version:**

Julien Cojan, Jean Lieber. Formalisation de l'adaptation conservatrice dans les espaces métriques. 16ème atelier de Raisonnement à Partir de Cas - RàPC 2008, Apr 2008, Nancy, France. inria-00336024

HAL Id: inria-00336024

<https://hal.inria.fr/inria-00336024>

Submitted on 31 Oct 2008

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

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

Formalisation de l'adaptation conservatrice dans les espaces métriques

Julien Cojan et Jean Lieber
Orpailleur, LORIA
CNRS, INRIA, Universités de Nancy,
BP 239, 54 506 Vandœuvre-lès-Nancy

Résumé

L'adaptation conservatrice est une approche de l'adaptation en raisonnement à partir de cas qui s'appuie sur le principe du changement minimal du contexte source vers le contexte cible. Cette notion de changement minimal du contexte se retrouve dans la notion d'opérateur de révision des connaissances, notion formalisée par la théorie AGM. Cette théorie peut alors être réutilisée pour formaliser l'adaptation conservatrice. Cette formalisation a été effectuée et étudiée dans le cadre de la logique propositionnelle lors d'un travail antérieur. Or, la logique propositionnelle n'est pas un formalisme suffisant pour la plupart des applications du RàPC, qui utilisent des formalismes « attribut-valeur simple ». Cet article étudie une extension de l'adaptation conservatrice à un cadre formel plus général : celui des espaces métriques qui inclut les formalismes « attribut-valeur simple ». Dans ce cadre, un cas s'interprète comme un sous-ensemble de l'espace. Les opérateurs de révision, et en particulier l'opérateur associé à la distance de l'espace métrique, sont étudiés dans ce formalisme, notamment du point de vue des postulats AGM. L'adaptation conservatrice peut alors être formalisée dans le contexte d'un espace métrique donné. Une extension vers une adaptation « moins conservatrice » et vers une adaptation conservatrice floue seront également étudiés.

Mots clés : Adaptation, adaptation conservatrice, révision, espaces métriques.

1 Introduction

Le raisonnement à partir de cas (RàPC [6]) est un modèle de résolution de problèmes à partir d'une base de cas, un cas étant la représentation d'un épisode de résolution de problème. Le cœur de ce raisonnement consiste en deux étapes : la remémoration, où l'on sélectionne dans la base un cas proche du problème à résoudre et l'adaptation, où l'on construit une solution au problème à partir de celle du cas remémoré. On s'intéressera dans cet article à la phase d'adaptation. Dans [5], une méthode d'adaptation reposant sur la théorie de la révision, l'adaptation conservatrice, a été proposée. Dans cet article les cas étaient représentés en logique propositionnelle, ce qui a un intérêt théorique pour examiner la faisabilité de l'approche, mais reste limité vis-à-vis de la représentation des cas utilisée dans la plupart des systèmes de RàPC. On se propose ici d'étendre l'adaptation conservatrice à un cadre formel plus général et, partant, applicable à une plus grande classe d'applications.

L'article est organisé comme suit. La section 2 est consacrée à des rappels sur le RàPC, la section 3 à la représentation des cas dans un espace métrique. Dans la section 4, l'adaptation conservatrice sera formalisée dans le cadre formel de ces espaces. Une extension sera nécessaire, on introduira dans la section 5 l'adaptation conservatrice floue. Un exemple suivi sera présenté tout au long de l'article pour illustrer de façon concrète les notions introduites.

2 Rappels sur le RàPC

De façon générale, un cas est la représentation partielle ou complète d'un épisode de résolution de problème. Un cas source est un cas complètement résolu, noté *cas-source*. Une base de cas est un ensemble fini de cas sources. Un cas cible, noté *cas-cible*, est un cas non entièrement résolu. Reasonner à partir de cas consiste

à compléter cas-cible en cas-cible-complété à l'aide d'une base de cas. Cela se fait en deux étapes principales :

- La remémoration, qui consiste à sélectionner dans la base de cas un cas cas-srce proche de cas-cible.
- L'adaptation, qui consiste à compléter cas-cible en cas-cible-complété à partir de cas-srce.

cas-cible-complété peut ne pas être entièrement résolu et consister en une étape intermédiaire de résolution en prenant le rôle de cas-cible dans le cycle précédent.

On notera CD les connaissances du domaine : CD exprime des connaissances supposées correctes mais pas nécessairement complètes. En particulier, CD donne des conditions nécessaires pour qu'un cas soit licite.

En RàPC, les cas sont généralement représentés par des couples problème-solution, un cas source sera noté cas-srce = (srce, Sol(srce)) où srce est sa partie problème et Sol(srce) sa partie solution. Du cas cible cas-cible seule la partie problème cible est connue — cas-cible = (cible, ?) — l'adaptation consiste alors à déterminer une solution Sol(cible) de cible à partir de cas-srce pour obtenir cas-cible-complété = (cible, Sol(cible)). Cette représentation repose sur l'hypothèse que la représentation d'un cas source en une partie problème et une partie solution peut se faire de façon unique, indépendante du cas cible. Cette hypothèse est restrictive comme le montre l'exemple suivant, ce qui motive la formulation plus générale du raisonnement à partir de cas utilisée dans cet article.

Exemple 2.1. *Un livre de recettes peut-être vu comme une base de cas. Considérons pour simplifier qu'une recette consiste en un titre, une liste d'ingrédients et une liste d'instructions. Ce livre de cuisine est souvent utilisé pour répondre à deux types de questions :*

- *Quels ingrédients sont nécessaires pour préparer un plat donné ? On demande de fournir la liste des ingrédients à partir du nom d'une recette.*
- *Quel plat peut-être préparé avec les ingrédients dont on dispose ? On demande de fournir une recette dont la liste des ingrédients est incluse dans la liste des ingrédients disponibles.*

Ces deux requêtes s'appliquent bien à la base de cas mais le découpage problème-solution est de nature différente, en particulier la liste d'ingrédients fait partie de la solution dans la première requête et du problème dans la seconde.

3 Représentation des cas dans un espace métrique

3.1 Rappels sur les espaces métriques

Définition 3.1. *Une distance sur un ensemble \mathcal{U} est une application $d : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}_+$ vérifiant pour tous $x, y, z \in \mathcal{U}$:*

- (i) $d(x, y) = 0$ ssi $x = y$ (séparation)
- (ii) $d(x, y) = d(y, x)$ (symétrie)
- (iii) $d(x, z) \leq d(x, y) + d(y, z)$ (inégalité triangulaire)

Un espace métrique est un couple (\mathcal{U}, d) où d est une distance sur \mathcal{U} .

Pour $x \in \mathcal{U}$ et $A, B \in 2^{\mathcal{U}}$ ($2^{\mathcal{U}}$ est l'ensemble des parties de \mathcal{U}), on note :

$$d(A, y) = \inf_{x \in A} d(x, y)$$

$$d(A, B) = \inf_{y \in B} d(A, y) = \inf_{x \in A, y \in B} d(x, y)$$

On adopte la convention $\inf \emptyset = +\infty$ dans \mathbb{R} , ce qui implique :

$$d(\emptyset, x) = d(A, \emptyset) = d(\emptyset, A) = +\infty \quad \text{pour } x \in \mathcal{U} \text{ et } A \in 2^{\mathcal{U}} \quad (3.1)$$

Définition 3.2. *Une mesure de similarité est une application $S : \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$ telle que*

$$\text{pour tous } x, y \in \mathcal{U} \quad S(x, y) = 1 \quad \text{ssi } x = y \quad (3.2)$$

On définit une mesure de similarité S associée à une distance d par¹ :

$$\text{pour tous } x, y \in \mathcal{U} \quad S(x, y) = e^{-d(x, y)} \quad (3.3)$$

On peut vérifier que (3.3) et la propriété (i) de la distance d entraînent (3.2). À l'inverse, à une mesure de similarité S on peut associer une fonction d selon (3.2) — $d(x, y) = -\ln S(x, y)$ — qui vérifie l'axiome (i) des distances mais pas nécessairement les axiomes (ii) et (iii).

Étant donné $x \in \mathcal{U}$, $A, B \in 2^{\mathcal{U}}$ et d associée à S selon (3.3), on note :

$$\begin{aligned} S(A, y) &= e^{-d(A, y)} = \sup_{x \in A} S(x, y) \\ S(A, B) &= e^{-d(A, B)} = \sup_{y \in B} S(A, y) = \sup_{x \in A, y \in B} S(x, y) \end{aligned} \quad (3.4)$$

Définition 3.3. Étant donné un espace métrique (\mathcal{U}, d) , l'adhérence d'un sous-ensemble A de \mathcal{U} est le plus petit fermé contenant A , c'est l'ensemble $\text{adh}(A) = \{x \in \mathcal{U} \mid d(A, x) = 0\}$. Un sous-ensemble A de \mathcal{U} est dit

- Fermé si $A = \text{adh}A$.
- Ouvert si son complémentaire est fermé, c'est-à-dire que pour tout $x \in A$ il existe $\delta > 0$ tel que $\{y \in \mathcal{U} \mid d(x, y) < \delta\} \subseteq A$.

Définition 3.4. Un espace métrique (\mathcal{U}, d) est discret si pour tout $x \in \mathcal{U}$ il existe $\delta > 0$ tel que $\{y \in \mathcal{U} \mid d(x, y) < \delta\} = \{x\}$.

Définition 3.5. Une relation binaire \leq sur un ensemble E est un pré-ordre si

- Elle est réflexive : pour tout $x \in E$ on a $x \leq x$.
- Elle est transitive : pour tous $x, y, z \in E$, si $x \leq y$ et $y \leq z$, alors $x \leq z$.

Un pré-ordre \leq est un ordre s'il est antisymétrique : pour tous $x, y \in E$, si $x \leq y$ et $y \leq x$, alors $x = y$. Un (pré-)ordre \leq est complet si pour tous $x, y \in E$ on a soit $x \leq y$, soit $y \leq x$.

Soit \leq un pré-ordre sur E et $F \subseteq E$. $x \in F$ est minimal dans F pour \leq s'il n'existe pas de $y \in F$ tel que $y \leq x$ et $x \not\leq y$.

3.2 Représentation des cas et des connaissances du domaine

Représentation des cas. Dans la suite, on considère que les cas peuvent être caractérisés par des sous-ensembles d'un espace métrique (\mathcal{U}, d) . Une représentation de cas dans cet espace consiste en un langage de concepts \mathcal{L}_C , un cas étant représenté par un concept. On suppose que \mathcal{L}_C est clos pour la négation et la conjonction de concepts :

$$\text{si } C, D \in \mathcal{L}_C \quad \text{alors} \quad \neg C \in \mathcal{L}_C \quad \text{et} \quad C \wedge D \in \mathcal{L}_C$$

$$C \vee D \quad \text{est définie par} \quad \neg(\neg C \wedge \neg D)$$

Par ailleurs, \mathcal{L}_C contient les éléments \top et \perp . Enfin, pour $\delta \in \mathbb{R}_+$, on introduit le constructeur G^δ qui à $C \in \mathcal{L}_C$ associe $G^\delta(C) \in \mathcal{L}_C$ (2).

La sémantique du langage des concepts est donnée par la fonction Ext qui à $C \in \mathcal{L}_C$ associe $\text{Ext}(C) \in 2^{\mathcal{U}}$ et qui vérifie les propriétés suivantes, pour $C, D \in \mathcal{L}_C$ et $\delta \in \mathbb{R}_+$:

$$\begin{aligned} \text{Ext}(\top) &= \mathcal{U} & \text{Ext}(\perp) &= \emptyset \\ \text{Ext}(\neg C) &= \mathcal{U} \setminus \text{Ext}(C) & \text{Ext}(C \wedge D) &= \text{Ext}(C) \cap \text{Ext}(D) \\ \text{Ext}(C \vee D) &= \text{Ext}(\neg(\neg C \wedge \neg D)) = \text{Ext}(C) \cup \text{Ext}(D) & \text{Ext}(G^\delta(C)) &= \{x \in \mathcal{U} \mid d(C, x) \leq \delta\} \end{aligned}$$

¹N'importe quelle fonction $f : \mathbb{R}_+ \rightarrow [0, 1]$ continue, strictement décroissante et telle que $f(0) = 1$ et $\lim_{x \rightarrow +\infty} f(x) = 0$ aurait convenu à la place de $x \mapsto e^{-x}$. Par exemple, le choix de $f(x) = \frac{1}{1+x}$ est souvent fait en RàPC. Nous avons fait ce choix pour des raisons de simplification qui apparaîtront à la section 5. Mais comme nous ne nous intéressons aux valeurs de d et S que pour des comparaisons par \leq , ce choix n'a pas de conséquences en dehors de la simplification qu'il entraîne.

²Ce constructeur est inspiré de [1].

Notons que $G^0(C)$ n'est pas nécessairement équivalent à C , mais qu'on a $\text{Ext}(G^0(C)) = \text{adh}(\text{Ext}(C))$. On supposera en outre que \mathcal{L}_C contient des concepts représentant les singletons, c'est-à-dire que pour $x \in \mathcal{U}$ il existe $C \in \mathcal{L}_C$ tel que $\text{Ext}(C) = \{x\}$.

Définition 3.6. Un modèle de $C \in \mathcal{L}_C$ est, par définition, un élément de $\text{Ext}(C)$, les relations de conséquence \vDash et d'équivalence \equiv sur \mathcal{L}_C sont définies par :

$$\begin{aligned} C \vDash D & \text{ si } \text{Ext}(C) \subseteq \text{Ext}(D) \\ C \equiv D & \text{ si } \text{Ext}(C) = \text{Ext}(D) \end{aligned}$$

Un concept $C \in \mathcal{L}_C$ est satisfiable si $\text{Ext}(C) \neq \emptyset$, autrement dit, si $C \not\vDash \perp$. Si $A \in 2^{\mathcal{L}_C}$ et $C \in \mathcal{L}_C$, $A \vDash C$ signifie qu'il existe un nombre fini de d'éléments de A , $D_1, \dots, D_n \in A$ tels que $(D_1 \wedge \dots \wedge D_n) \vDash C$.

Afin d'alléger les notations, pour $C, D \in \mathcal{L}_C$ et $x \in \mathcal{U}$:

$$\begin{aligned} d(C, x) &= d(\text{Ext}(C), x) & S(C, x) &= S(\text{Ext}(C), x) \\ d(C, D) &= d(\text{Ext}(C), \text{Ext}(D)) & S(C, D) &= S(\text{Ext}(C), \text{Ext}(D)) \end{aligned}$$

On notera \mathcal{E} l'ensemble des extensions de concepts : $\mathcal{E} = \{\text{Ext}(C) \mid C \in \mathcal{L}_C\} \subseteq 2^{\mathcal{U}}$.

Remarque 3.1. L'hypothèse mentionnée dans la section 2 de la décomposition unique des cas en une partie problème et une partie solution consiste à décomposer \mathcal{U} en deux composantes $\mathcal{U} = \mathcal{U}_{\text{pb}} \times \mathcal{U}_{\text{sol}}$ où \mathcal{U}_{pb} (resp., \mathcal{U}_{sol}) est un ensemble d'instances de problèmes (resp., de solutions). Elle suppose de plus que, si $C \in \mathcal{L}_C$, alors $\text{Ext}(C)$ est de la forme $\text{Ext}_{\text{pb}}(C_{\text{pb}}) \times \text{Ext}_{\text{sol}}(C_{\text{sol}})$ où $C_{\text{pb}} \in \mathcal{L}_{C_{\text{pb}}}$ et $C_{\text{sol}} \in \mathcal{L}_{C_{\text{sol}}}$, $\mathcal{L}_{C_{\text{pb}}}$ et $\mathcal{L}_{C_{\text{sol}}}$ étant des langages de représentation respectivement de problèmes et de solutions et Ext_{pb} et Ext_{sol} les fonctions extension associées.

Cependant, cette représentation n'est pas stable par négation et disjonction, par exemple dans $\mathcal{U} = \{0, 1\}^2$ avec $\mathcal{U}_{\text{pb}} = \mathcal{U}_{\text{sol}} = \{0, 1\}$, soit $C \in \mathcal{L}_C$ d'extension $\text{Ext}(C) = \{(1, 1)\} = \{1\} \times \{1\}$, dans le formalisme général $\neg C$ a une extension $\{(0, 0), (0, 1), (1, 0)\}$ or celle-ci ne peut se mettre sous la forme $A_{\text{pb}} \times A_{\text{sol}}$ avec $A_{\text{pb}}, A_{\text{sol}} \subseteq \{0, 1\}$. De même, si $D \in \mathcal{L}_C$ a pour extension $\text{Ext}(D) = \{(0, 0)\} = \{0\} \times \{0\}$, $\text{Ext}(C \vee D)$ ne peut être exprimé comme produit de sous-ensembles de \mathcal{U}_{pb} et \mathcal{U}_{sol} .

Autrement dit, il faut choisir entre le fait que, si C est un cas, $\neg C$ l'est également, et la décomposabilité entre partie problème et partie solution.

Représentation des connaissances du domaine. Le seul rôle de CD dans cet article est de représenter des conditions nécessaires pour qu'un cas C soit licite, i.e. que tout $x \in \text{Ext}(C)$ soit licite³. On peut modéliser CD par un sous-ensemble X de \mathcal{U} tel que, pour $x \in \mathcal{U}$, si $x \notin X$ alors x est illicite. Si $x \in X$ cela signifie en outre que les connaissances du domaine n'excluent pas que x puisse être licite (si on suppose que CD est complet, alors x est licite dans ce cas). On suppose que CD peut être exprimé dans le langage des concepts \mathcal{L}_C , autrement dit : $\text{CD} \in \mathcal{L}_C$ et $\text{Ext}(\text{CD}) = X$.

Pour que l'approche présentée dans cet article soit opérationnelle au sein d'un système informatique, le choix de \mathcal{L}_C n'est pas indifférent. Le langage doit être récursif (le test $C \in \mathcal{L}_C$ doit être décidable) et les inférences utilisées doivent être décidables. Nous ne reviendrons pas sur ce point dans cet article.

3.3 Exemple : les formalismes « attributs-valeurs simples »

On suppose que \mathcal{U} est un produit cartésien :

$$\mathcal{U} = U_1 \times U_2 \times \dots \times U_n$$

³En pratique on considérera le contexte d'un cas C , c'est à dire la conjonction $\text{CD} \wedge C$, qui est nécessairement licite. La question qu'on pourra se poser au sujet du contexte $\text{CD} \wedge C$ sera plutôt sa satisfiabilité.

où les U_i sont soit \mathbb{R} , soit \mathbb{Z} , soit l'ensemble des booléens $\mathbb{B} = \{V, F\}$ (V dénote « vrai » et F , « faux »), soit un autre ensemble fini donné en extension (« type énuméré ») : ce que nous appelons ici « valeur simple », c'est donc une valeur numérique ou une valeur d'un type énuméré.

Pour $i \in \{1, \dots, n\}$, l'attribut a_i est la projection sur la $i^{\text{ème}}$ coordonnée :

$$a_i(x_1, x_2, \dots, x_i, \dots, x_n) = x_i$$

Le langage \mathcal{L}_C est formé d'expressions à valeurs booléennes sur les paramètres formels a_1, a_2, \dots, a_n : $C = P(a_1, a_2, \dots, a_n)$. L'extension d'un tel concept C est :

$$\begin{aligned} \text{Ext}(C) &= \{x \in \mathcal{U} \mid P(a_1(x), a_2(x), \dots, a_n(x)) = V\} \\ &= \{(x_1, x_2, \dots, x_n) \in \mathcal{U} \mid P(x_1, x_2, \dots, x_n) = V\} \end{aligned}$$

\mathcal{L}_C est évidemment toujours considéré comme clos pour la négation et la conjonction.

3.3.1 Exemple culinaire

Reprenons l'exemple des recettes de cuisine⁴. Supposons qu'on veuille préparer une tarte aux poires pour quatre personnes mais que l'on n'ait trouvé qu'une recette de tarte aux pommes pour 6 personnes dans le livre de recettes. Pour simplifier, nous n'allons pas considérer les instructions mais seulement les ingrédients intervenant dans cet exemple, cela donne la liste suivante d'attributs :

- $a_1 = \text{nb-parts}$ désignant le nombre de personnes pour lequel cette recette est prévue, $U_1 = \mathbb{N} \setminus \{0\}$.
- $a_2 = \text{type-plat}$ désigne le type de plat, $U_2 = \{\text{entrée, résistance, dessert}\}$.
- $a_3 = \text{sucré}$ désigne la quantité de sucre contenue dans la recette, (en grammes équivalent saccharose), $U_3 = \mathbb{R}_+$.
- $a_4 = \text{saccharose}$ désigne la quantité de saccharose, en grammes, $U_4 = \mathbb{R}_+$.
- $a_5 = \text{masse-pâte}$ désigne la quantité de pâte brisée, en grammes, $U_5 = \mathbb{R}_+$.
- $a_6 = \text{masse-fruit}$ désigne la quantité de fruit, en grammes, $U_6 = \mathbb{R}_+$.
- $a_7 = \text{nb-pommes}$ désigne le nombre de pommes, $U_7 = \mathbb{N}$.
- $a_8 = \text{nb-poire}$ désigne le nombre de poires, $U_8 = \mathbb{N}$.

L'espace est donc $\mathcal{U} = (\mathbb{N} \setminus \{0\}) \times \{\text{entrée, résistance, dessert}\} \times \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{N} \times \mathbb{N}$. On définit une distance d sur \mathcal{U} telle que pour $x, y \in \mathcal{U}$:

$$d(x, y) = \sum_{i=1}^8 w_i d_i(x, y)$$

$$\text{avec : } d_1(x, y) = |a_1(y) - a_1(x)|, \quad d_2(x, y) = \begin{cases} 0 & \text{si } x = y \\ 1 & \text{sinon} \end{cases}$$

$$\text{et pour } i \in \{3, \dots, 8\} \quad d_i(x, y) = \left| \frac{a_i(y)}{a_1(y)} - \frac{a_i(x)}{a_1(x)} \right|$$

Les coefficients w_i sont choisis à la fois pour normaliser (une différence de 1g de masse-fruit est moins importante qu'une différence de type-plat ou de masse-fruit) et pour pondérer les différents attributs (on pondérera plus lourdement type-plat que nb-parts car passer de 5 à 6 personnes devrait constituer une adaptation plus facile que passer d'un dessert à un plat de résistance). Nous avons choisi :

$$\begin{array}{cccc} w_1 = 100 & w_2 = 1000 & w_3 = 1 & w_4 = 0, 1 \\ w_5 = 5 & w_6 = 2 & w_7 = 10 & w_8 = 10 \end{array}$$

⁴Les valeurs numériques introduites dans cet exemple sont « bidon ». Il est évident que les auteurs disposent des « véritables » valeurs, qui leur ont été confiées par un grand cuisinier peu avant son décès dans les conditions tragiques que l'on sait dans les milieux autorisés. À sa demande, nous attendrons quelques mois avant de les dévoiler⁵.

⁵Ou pas.

Les connaissances du domaine CD expriment que le sucre contenu dans la recette est la somme de saccharose et du sucre contenu dans les nb-pommes (en moyenne 13g par pomme) et dans les nb-poires (en moyenne 10g par poire), de même, la quantité de fruit est la somme de celle de nb-pommes (120g en moyenne) et de nb-poires (90g en moyenne) :

$$\begin{aligned} \text{CD} = & (\text{sucre} = \text{saccharose} + 13 \times \text{nb-pommes} + 10 \times \text{nb-poires}) \\ & \wedge (\text{masse-fruit} = 120 \times \text{nb-pommes} + 90 \times \text{nb-poires}) \end{aligned}$$

La recette de tarte aux pommes pour quatre personnes dont les ingrédients sont : une pâte brisée de 150g, 3 pommes et 60g de sucre sera représentée par le concept $\text{tarte-pommes-6p} = (\text{nb-parts} = 6) \wedge (\text{type-plat} = \text{dessert}) \wedge (\text{masse-p\^ate} = 150) \wedge (\text{saccharose} = 60) \wedge (\text{nb-pommes} = 3) \wedge (\text{nb-poires} = 0)$. On peut généraliser cette recette à un nombre quelconque de personnes en conservant les proportions, par le concept $\text{cas-srce} = \text{tarte-pommes}$:

$$\begin{aligned} \text{tarte-pommes} = & (\text{type-plat} = \text{dessert}) \wedge (\text{masse-p\^ate} = 25 \times \text{nb-parts}) \\ & \wedge (\text{saccharose} = 10 \times \text{nb-parts}) \wedge (2 \times \text{nb-pommes} = \text{nb-parts}) \wedge (\text{nb-poires} = 0) \end{aligned}$$

Le cas cible à résoudre est

$$\text{tarte-poires-4p} = (\text{nb-parts} = 4) \wedge (\text{type-plat} = \text{dessert}) \wedge (\text{nb-pommes} = 0)$$

3.3.2 Logique propositionnelle avec un nombre fini de variables

L'ensemble des formules sur les variables propositionnelles p_1, \dots, p_n ($n \in \mathbb{N}$) correspond au cas particulier de la représentation par attributs-valeurs où $\mathcal{U} = \mathbb{B}^n$. En effet, si f est une formule logique propositionnelle sur p_1, \dots, p_n , on peut lui associer la fonction $P_f : \mathbb{B}^n \rightarrow \mathbb{B}$ telle que si I est une interprétation des variables p_1, \dots, p_n , alors I est un modèle de f ssi $P_f(I(p_1), I(p_2), \dots, I(p_n)) = V$. Inversement, à une fonction $P : \mathbb{B}^n \rightarrow \mathbb{B}$ on peut associer une formule f unique à l'équivalence logique près telle que $P = P_f$.

Exemple 3.1. À $f = a \wedge \neg(b \vee \neg c)$ on peut associer $P_f : (x, y, z) \in \mathbb{B}^3 \mapsto P(x, y, z) = \text{et}(x, \text{non}(\text{ou}(y, \text{non}(z))))$.

Pour $I \in \mathcal{U}$, $i \in \{1, \dots, n\}$ et f une formule propositionnelle sur p_1, \dots, p_n , on note $a_i(I) = I(p_i)$ et $\text{Ext}(f) = \{x \in \mathcal{U} \mid P_f(a_1(x), a_2(x), \dots, a_n(x)) = V\}$. On obtient l'équivalence suivante qui identifie la sémantique obtenue à celle de la logique propositionnelle : I est un modèle de f ssi $I \in \text{Ext}(f)$. Cela justifie l'utilisation du formalisme décrit dans la section 3.2 pour la généralisation de formules de logique propositionnelle comme il sera fait dans la section suivante.

4 Formalisation de l'adaptation conservatrice dans un espace métrique

4.1 Révision dans $(\mathcal{U}, d, \mathcal{L}_C)$

Dans [4], Katsuno et Mendelzon proposent des postulats que devrait vérifier un opérateur de révision \circ sur des formules propositionnelles. Étant donné les formules C, C', D, D' et F :

$$\begin{array}{l} \text{Postulats de base} \left\{ \begin{array}{l} \text{(R1)} \quad C \circ D \vDash D \\ \text{(R2)} \quad \text{si } C \wedge D \text{ est satisfiable alors } C \circ D \equiv C \wedge D \\ \text{(R3)} \quad \text{si } D \text{ est satisfiable alors } C \circ D \text{ l'est aussi} \\ \text{(R4)} \quad \text{si } C \equiv C' \text{ et } D \equiv D' \text{ alors } C \circ D \equiv C' \circ D' \end{array} \right. \\ \\ \text{Postulats de minimalité} \left\{ \begin{array}{l} \text{(R5)} \quad (C \circ D) \wedge F \vDash C \circ (D \wedge F) \\ \text{(R6)} \quad \text{si } (C \circ D) \wedge F \text{ est satisfiable alors } C \circ (D \wedge F) \vDash (C \circ D) \wedge F \end{array} \right. \end{array}$$

Katsuno et Mendelzon montrent, également dans [4], qu'en logique propositionnelle leurs postulats sont équivalents aux postulats AGM [3], cette démonstration s'applique aussi à notre formalisme (dans lequel on parle de classes à la place de formules).

Ces postulats expriment des caractéristiques désirées pour un opérateur de révision mais ils ne sont pas constructifs et ne garantissent pas l'existence ni l'unicité d'un tel opérateur. Dans le même article il est donné une caractérisation des opérateurs de révision vérifiant ces postulats par leur extension. Elle repose sur la notion d'*assignement fidèle*.

Définition 4.1. *Un assignement fidèle est une application qui à une formule C associe un pré-ordre \leq_C sur les interprétations vérifiant les propriétés suivantes :*

- (1) *Si $x, y \in \text{Ext}(C)$, alors $x <_C y$ est faux.*
- (2) *Si $x \in \text{Ext}(C)$ et $y \notin \text{Ext}(C)$, alors $x <_C y$.*
- (3) *Si $C \equiv D$, alors $\leq_C = \leq_D$.*

où $x <_C y$ ssi $x \leq_C y$ et non $y \leq_C x$.

Étant donné une partie \mathcal{M} de \mathcal{U} et un pré-ordre \leq sur \mathcal{M} , on note :

$$\text{Min}(\mathcal{M}, \leq) = \{x \in \mathcal{M} \mid x \text{ est minimal pour } \leq \text{ dans } \mathcal{M}\}$$

Dans le cadre de la logique propositionnelle finie, les opérateurs vérifiant (R1)–(R6) sont liés aux assignements fidèles par le théorème suivant :

Théorème 3.3 de [4] : Un opérateur de révision en logique propositionnelle satisfait les postulats (R1) – (R6) ssi il existe un assignement fidèle qui associe à chaque formule C un pré-ordre complet \leq_C tel que

$$\text{Ext}(C \circ D) = \text{Min}(\text{Ext}(D), \leq_C) \quad (4.1)$$

Dans notre formalisme plus général, la condition (4.2) d'existence de minimas pour chaque \leq_C est nécessaire :

Proposition 4.1. *Un opérateur de révision \circ sur \mathcal{L}_C satisfait les postulats (R1)–(R6) ssi il existe un assignement fidèle qui associe à chaque formule C un pré-ordre complet \leq_C tel que pour tout $D \in \mathcal{L}_C$*

$$\begin{aligned} \text{si } D \text{ est satisfiable alors } \text{Min}(\text{Ext}(D), \leq_C) \neq \emptyset \\ \text{Ext}(C \circ D) = \text{Min}(\text{Ext}(D), \leq_C) \end{aligned} \quad (4.2)$$

On s'intéresse au cas particulier de l'opérateur de révision \circ^d défini à partir d'une distance d sur \mathcal{U} :

Définition 4.2. *Étant donné une distance d sur \mathcal{U} , l'opérateur \circ^d est défini par l'extension des concepts $C \circ^d D$ pour $C, D \in \mathcal{L}_C$:*

$$\text{Ext}(C \circ^d D) = \{x \in \text{Ext}(D) \mid d(C, x) = \Delta\} = \{x \in \text{Ext}(D) \mid d(C, x) \leq \Delta\} \quad \text{où } \Delta = d(C, D)$$

Cette définition est donnée modulo l'équivalence logique, ce qui correspond au principe de non pertinence de la syntaxe formalisée dans (R4). Intuitivement, \circ^d sélectionne les interprétations de D qui sont les plus proches de celles de C pour la distance d . Par ailleurs, en s'inspirant de [1], on peut donner la définition équivalente de \circ^d :

$$C \circ^d D = G^\Delta(C) \wedge D \quad \text{avec } \Delta = d(C, D)$$

On notera d'une part que $\Delta = \inf\{\delta \in \mathbb{R}_+ \mid G^\delta(C) \wedge D \text{ est satisfiable}\}$ et, d'autre part que cette définition équivalente prouve l'existence du concept $C \circ^d D$ et justifie *a posteriori* l'introduction du constructeur de concepts G^δ . De plus, \circ^d peut être mis sous la forme (4.1) en notant pour $C \in \mathcal{L}_C$ et $x, y \in \mathcal{U}$:

$$x \leq_C^d y \quad \text{ssi} \quad d(C, x) \leq d(C, y) \quad (4.3)$$

Mais comme on le verra dans la section 4.3.2, l'application qui associe à un concept C le pré-ordre \leq_C^d n'est pas toujours un assignement fidèle, elle peut ne pas respecter la condition (2), elle peut aussi ne pas respecter la condition (4.2) sur l'existence de minimas. On verra qu'il y a équivalence entre la satisfaction des postulats AGM par \circ^d et le fait que $C \mapsto \leq_C^d$ soit un assignement fidèle vérifiant la condition (4.2).

4.2 L'adaptation conservatrice dans $(\mathcal{U}, d, \mathcal{L}_C)$

4.2.1 Définition

L'adaptation conservatrice consiste à compléter cas-cible par une adaptation de cas-srce, cette adaptation consistant en une modification minimale. L'idée sous-jacente étant que deux cas proches ont souvent des solutions proches.

Dans [5], l'adaptation conservatrice est définie en logique propositionnelle, étant donné un problème cible, un cas source cas-srce, des connaissances du domaine CD et un opérateur de révision \circ , l'adaptation conservatrice donne cas-cible-complété tel que :

$$(CD \wedge \text{cas-srce}) \circ (CD \wedge \text{cas-cible}) \equiv_{\text{CD}} \text{cas-cible-complété} \quad (4.4)$$

où \equiv_{CD} est l'équivalence modulo CD, c'est-à-dire que pour deux concepts C et D , $C \equiv_{\text{CD}} D$ si $CD \wedge C \equiv CD \wedge D$.

Remarque 4.1. *CD doit cependant être conservé dans (4.4) car $(CD \wedge C) \circ (CD \wedge D)$ n'est pas nécessairement équivalent modulo CD à $C \circ D$. En particulier $CD \wedge D$ et donc $(CD \wedge C) \circ (CD \wedge D)$ peuvent-être satisfiable mais $CD \wedge (C \circ D)$ ne pas l'être. Par exemple avec $\mathcal{U} = \mathbb{Z}$, $d(x, y) = |y - x|$, $\circ = \circ^d$ et C, D, CD tels que $\text{Ext}(C) = \{0\}$, $\text{Ext}(D) = \{1, 2\}$, $\text{Ext}(CD) = \{0, 2\}$, $\text{Ext}(C \circ^d D) = \{1\}$ donc $\text{Ext}(CD \wedge (C \circ^d D)) = \emptyset$, alors que $\text{Ext}((CD \wedge C) \circ^d (CD \wedge D)) = \{2\}$.*

Exemple 4.1. *En reprenant l'exemple 3.3.1,*

$$\begin{aligned} CD \wedge \text{tarte-pommes} &\equiv (\text{type-plat} = \text{dessert}) \wedge (\text{sucres} = 16, 5 \times \text{nb-parts}) \\ &\quad \wedge (\text{saccharose} = 10 \times \text{nb-parts}) \wedge (\text{masse-p\^ate} = 25 \times \text{nb-parts}) \\ &\quad \wedge (\text{masse-fruit} = 60 \times \text{nb-parts}) \wedge (2 \times \text{nb-pommes} = \text{nb-parts}) \\ &\quad \wedge (\text{nb-poires} = 0) \\ CD \wedge \text{tarte-poires-4p} &\equiv (\text{nb-parts} = 4) \wedge (\text{type-plat} = \text{dessert}) \\ &\quad \wedge (\text{sucres} = \text{saccharose} + 10 \times \text{nb-poires}) \\ &\quad \wedge (\text{masse-fruit} = 90 \times \text{nb-poires}) \end{aligned}$$

La distance entre ces deux extensions est $\Delta = 27, 6$ et l'unique couple $(x, y) \in \mathcal{U} \times \mathcal{U}$ l'atteignant est le suivant :

$$\begin{aligned} x &= (4, \text{dessert}, 66, 40, 100, 240, 2, 0) \in \text{Ext}(CD \wedge \text{tarte-pommes}) \\ y &= (4, \text{dessert}, 66, 36, 100, 270, 0, 3) \in \text{Ext}(CD \wedge \text{tarte-poires-4p}) \end{aligned}$$

Par définition de l'adaptation conservatrice on a alors $\text{Ext}(\text{tarte-poires-4p-complétée}) = \{y\}$, c'est à dire que pour notre tarte aux poires pour quatre personnes, on prendra 3 poires, 36g de sucre et autant de pâte brisée : 100g.

En comparant les recettes x et y , on peut remarquer que le poids moindre des nb-poires par rapport aux nb-pommes a été compensé par leur nombre et que la plus grande quantité de sucre qu'elles contiennent a été compensé par une réduction de la quantité de saccharose ajouté. Ceci est la conséquence d'un choix judicieux de facteurs dans la définition de d .

L'adaptation conservatrice peut parfois être trop restrictive et générer un cas-cible-complété qui n'est qu'une résolution partielle de cas-cible car trop précis « dans sa partie problème ».

Exemple 4.2. *Considérons $\mathcal{U} = \mathbb{R}^3$ et $CD = \top$, le but étant de déterminer l'aire d'ellipses, c'est-à-dire de déterminer les triplets $(a, b, s) \in \mathcal{U}$ tels que s soit la surface de l'ellipse de paramètres a et b . Pour connaître cela on considère cas-cible = \top , supposons que l'on connaisse la surface d'un cercle, on a alors cas-srce d'extension $\text{Ext}(\text{cas-srce}) = \{(a, b, s) \mid a = b \text{ et } s = \pi a^2\}$. cas-srce \models cas-cible donc $(CD \wedge \text{cas-srce}) \wedge (CD \wedge \text{cas-cible}) \equiv CD \wedge \text{cas-srce}$ qui est satisfiable, le postulat (R2) implique alors que cas-cible-complété $\equiv_{\text{CD}} \text{cas-srce}$. On n'obtient alors que la réponse partielle correspondant aux cercles, cas-cible-complété est trop spécifique.*

Les postulats de la révision et l'adaptation conservatrice. Nous pouvons considérer la signification des postulats (R1) – (R6) dans le cadre de l'adaptation conservatrice.

- (R1) appliqué à l'adaptation conservatrice donne $CD \wedge \text{cas-cible-complété} \models CD \wedge \text{cas-cible}$ (car $\text{cas-cible-complété}$ est défini modulo CD), ce qui se entraîne que $\text{cas-cible-complété} \models_{CD} \text{cas-cible}$. Si cette assertion est violée, alors $\text{cas-cible-complété}$ ne spécialise pas cas-cible modulo CD , c'est-à-dire qu'il ne remplit pas son objectif.
- (R2) signifie, dans le cadre de l'adaptation conservatrice, que si cas-srce n'est pas incompatible avec cas-cible modulo les connaissances du domaine, i.e. $CD \wedge \text{cas-srce} \wedge \text{cas-cible}$ est satisfiable, alors cas-srce complète cas-cible correctement et donc $\text{cas-cible-complété} \equiv_{CD} \text{cas-srce} \wedge \text{cas-cible}$.
- (R3) est une obligation de résultat, si cas-cible est consistant modulo CD , alors l'adaptation conservatrice donne un $\text{cas-cible-complété}$ compatible avec CD .
- (R4) signifie simplement que l'adaptation conservatrice respecte le principe de non pertinence de la syntaxe.
- (R5) et (R6) traduisent, selon [4], l'idée de changement minimal et, dans le contexte de l'adaptation conservatrice, le fait que celle-ci fait un changement minimal sur le contexte source pour être en accord avec le contexte cible.

4.3 Problèmes de satisfaction des postulats AGM

4.3.1 Le Problème de la conformité AGM

En reprenant les notations de [2], pour une logique (\mathcal{L}, Cn) où \mathcal{L} est un langage et $Cn : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}$ un opérateur de conséquences satisfaisant les axiomes de Tarsky (voir plus bas), on définit les notions d'AGM-conformité et de décomposabilité.

Définition 4.3. (\mathcal{L}, Cn) est AGM-conforme s'il existe un opérateur de révision vérifiant les postulats (R1) – (R6).

(\mathcal{L}, Cn) est décomposable si pour tout $A, B \in 2^{\mathcal{L}}$ tels que $Cn(\emptyset) \subsetneq Cn(B) \subsetneq Cn(A)$, il existe $C \in 2^{\mathcal{L}}$ tel que $Cn(C) \subsetneq Cn(A)$ et $Cn(A) = Cn(B \cup C)$.

Soit $\mathcal{L} = \mathcal{L}_C$ et $Cn : 2^{\mathcal{L}_C} \rightarrow 2^{\mathcal{L}_C}$ telle que pour $A \in 2^{\mathcal{L}_C}$, $Cn(A) = \{C \in \mathcal{L}_C \mid A \models C\}$. (\mathcal{L}_C, Cn) vérifie bien les axiomes de Tarsky, pour $A, B \in 2^{\mathcal{L}_C}$:

- $Cn(Cn(A)) = Cn(A)$ (itération).
- $A \subseteq Cn(A)$ (inclusion).
- si $A \subseteq B$ alors $Cn(A) \subseteq Cn(B)$ (monotonie).

Théorème 2 de [2] : Une logique (\mathcal{L}, Cn) est AGM-conforme ssi elle est décomposable.

Notre formalisme est assez expressif pour satisfaire cette condition :

Proposition 4.2. (\mathcal{L}_C, Cn) est décomposable, et elle est donc AGM-conforme.

4.3.2 Satisfaction des postulats AGM par \circ^d

On s'intéressera dans cette partie à la satisfaction des postulats AGM par un opérateur \circ^d .

Proposition 4.3. \circ^d vérifie les postulats (R1) et (R4) – (R6).

Proposition 4.4. En revanche, pour certaines représentations $(\mathcal{U}, d, \mathcal{L}_C)$, \circ^d ne vérifie pas (R2) et (R3).

Proposition 4.5. \circ^d vérifie (R2) ssi $(\mathcal{U}, d, \mathcal{L}_C)$ vérifie la propriété (4.5) :

$$\text{pour tous } A \in \mathcal{E}, x \in \mathcal{U} \quad d(A, x) = 0 \text{ implique } x \in A \quad (4.5)$$

Ce qui est équivalent à :

$$C \mapsto \overset{d}{\llcorner}_C \text{ défini par (4.3) est un assignement fidèle} \quad (4.6)$$

Remarque 4.2. La proposition 4.5 implique que pour toute distance d , \circ^d satisfait le postulat modifié

$$(R2') \text{ si } C \wedge D \text{ est satisfiable alors } \text{Ext}(C \circ D) = \text{adh}(\text{Ext}(C)) \cap \text{Ext}(D)$$

Cette modification est légère : en particulier dans le cas discret, et donc en logique propositionnelle finie, (R2) et (R2') sont équivalents. Nous ne poursuivrons pas l'étude du postulat (R2') dans cet article.

Proposition 4.6. \circ^d vérifie (R3) ssi $(\mathcal{U}, d, \mathcal{L}_C)$ vérifie la propriété (4.7) :

$$\text{pour tous } A, B \in \mathcal{E}, B \neq \emptyset \quad d(A, B) = \Delta \text{ implique qu'il existe } x \in B \text{ tel que } d(A, x) = \Delta \quad (4.7)$$

Ce qui est équivalent à la condition sur l'assignement fidèle défini par (4.3) :

$$C \mapsto \leq_C^d \text{ vérifie (4.2)} \quad (4.8)$$

Remarque 4.3. (4.5) consiste à ne considérer que des ensembles fermés : $\mathcal{E} \subseteq \mathcal{F}$ où \mathcal{F} désigne l'ensemble des parties fermées de \mathcal{U} , ce qui implique entre autre que les ouverts de \mathcal{U} sont des fermés et réciproquement. En particulier si (\mathcal{U}, d) est discret alors (4.5) est vérifié pour tout \mathcal{E} , et \circ^d vérifie (R2).

Étant donné une distance d , si on définit une distance \tilde{d} par :

$$\text{pour } x, y \in \mathcal{U} \quad \tilde{d}(x, y) = \begin{cases} 0 & \text{si } x = y \\ 1 + d(x, y) & \text{sinon} \end{cases}$$

alors (\mathcal{U}, \tilde{d}) est un espace métrique discret et vérifie (4.5), donc d'après la propriété 4.5, $\circ^{\tilde{d}}$ vérifie (R2). De plus pour $C, D \in \mathcal{L}_C$ si $C \wedge D$ est insatisfiable, alors $C \circ^{\tilde{d}} D \equiv C \circ^d D$.

Remarque 4.4. (4.7) implique (4.5). En effet si $d(A, x) = 0$, par symétrie de d , $d(\{x\}, A) = 0$ et (4.7) donne l'existence d'un $y \in A$ tel que $d(\{x\}, y) = d(x, y) = 0$ donc $x = y \in A$.

La proposition 4.6 pose le problème de ce que doit être la révision lorsque la plus courte distance n'est pas réalisée. c'est-à-dire lorsqu'il existe deux concepts C et D tels que pour $\Delta = d(C, D)$:

$$\text{Ext}(C \circ^d D) = \{x \in \text{Ext}(D) \mid d(C, x) = \Delta\} = \emptyset$$

mais que pour tout $\varepsilon > 0 \quad \{x \in \text{Ext}(D) \mid d(C, x) = \Delta + \varepsilon\} \neq \emptyset$

Exemple : Soit $\mathcal{U} = (\mathbb{R}^2, d_2)$ où d_2 est la distance euclidienne ($d_2((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$), et les parties $A = \mathbb{R} \times \mathbb{R}_-$ et $B = \{(x, y) \in \mathbb{R}^2 \mid x \geq 1 \text{ et } y \geq \frac{1}{x}\}$. A et $B \neq \emptyset$ sont fermés, $d_2(A, B) = 0$ et pourtant $A \cap B = \emptyset$. La distance \tilde{d}_2 ne fait pas mieux (A et B sont des extensions de concepts C et D tels que $C \wedge D$ est insatisfiable), $\tilde{d}_2(A, B) = 1$ mais $\{x \in B \mid \tilde{d}_2(A, x) = 1\} = \emptyset$.

L'introduction d'une marge $\varepsilon > 0$ permet de résoudre le problème de la satisfaction de (R3), ceci donne un opérateur de révision paramétré par ε , dans la section suivante on considérera cette révision sous un formalisme de révision floue.

5 Formalisation de l'adaptation conservatrice floue dans un espace métrique

L'adaptation conservatrice est construite à partir d'un opérateur de révision. On veut étendre l'adaptation conservatrice par une adaptation conservatrice floue pour éliminer le problème de l'existence de minimas pour la distance associée. Pour cela il sera d'abord nécessaire de considérer la révision floue et l' α -révision lien entre les révisions classiques et floues.

5.1 L' α -révision dans $(\mathcal{U}, d, \mathcal{L}_C)$

Comme on l'a vu dans la section précédente, le postulat AGM (R3) n'est pas satisfait si la distance entre deux extensions n'est pas atteinte. Pour contourner ce problème on propose de considérer un seuil flou sur la distance dans l'espace des extensions. Pour se placer dans le formalisme habituel du flou on remplacera le paramètre $\varepsilon \in [0, +\infty]$ par $\alpha \in [0, 1]$ en réécrivant l'inégalité $d(C, x) \leq \Delta + \varepsilon$ où $\Delta = d(C, D)$, avec la mesure de similarité S associée à la distance d :

$$S(C, x) \geq \Sigma \times \alpha$$

où $\Sigma = S(C, D) = e^{-\Delta}$ et $\alpha = e^{-\varepsilon} \in [0, 1]$.

Définition 5.1. On définit donc \circ_α^d tel que :

$$\text{Ext}(C \circ_\alpha^d D) = \{x \in \text{Ext}(D) \mid S(C, x) \geq \Sigma \times \alpha\}$$

De façon équivalente, on pourrait définir \circ_α^d par $C \circ_\alpha^d D = G^{\Delta+\varepsilon}(C) \wedge D$.

Proposition 5.1. Comme attendu \circ_1^d et \circ^d donnent des résultats logiquement équivalents, et pour tous $1 \geq \alpha \geq \beta \geq 0$:

$$C \circ^d D \equiv C \circ_1^d D \vDash C \circ_\alpha^d D \vDash C \circ_\beta^d D \vDash C \circ_0^d D \equiv D$$

Proposition 5.2. Pour $\alpha \in [0, 1]$, \circ_α^d vérifie les postulats (R1), (R4) et (R5).

Proposition 5.3. Pour $\alpha < 1$, \circ_α^d vérifie le postulat (R3).

En revanche dans le cadre général pour $\alpha < 1$, \circ_α^d ne vérifie pas (R2) et (R6). Pour le postulat (R2), le problème est le même que dans le cas $\alpha = 1$ (cf. proposition 4.4), l'extension retenue peut être trop grande, elle l'est même d'autant plus que α est petit.

Exemple 5.1. Dans $\mathcal{U} = \mathbb{Z}$ avec $d(x, y) = |y - x|$, soit C, D tels que $\text{Ext}(C) = \{0\}$ et $\text{Ext}(D) = \{0, 1\}$, $C \wedge D$ est satisfiable. Alors $d(C, D) = 0$ donc $S(C, D) = 1$ et

$$\text{Ext}(C \circ_{e^{-1}}^d D) = \{x \in \text{Ext}(D) \mid S(C, x) \geq e^{-1}\} = \{0, 1\} \neq \text{Ext}(C \wedge D)$$

Dans cet exemple $\circ_{e^{-1}}^d$ ne vérifie pas (R2) alors que \circ_1^d si.

Pour le postulat (R6), l'extension retenue peut aussi être trop grande. Ceci n'est pas surprenant vu qu'il exprime d'après [4] le changement minimal, or, le principe de l' α -révision est justement de ne pas se restreindre au minimum.

Exemple 5.2. Toujours avec $\mathcal{U} = \mathbb{Z}$ et $d(x, y) = |y - x|$, soit C, D, F tels que $\text{Ext}(C) = \{0\}$ et $\text{Ext}(D) = \{1, 2, 3\}$ et $\text{Ext}(F) = \{2, 3\}$. $S(C, D) = e^{-1}$ et $\text{Ext}(C \circ_{e^{-1}}^d D) = \{x \in \text{Ext}(D) \mid S(C, x) \geq e^{-2}\} = \{1, 2\}$. $(C \circ_{e^{-1}}^d D) \wedge F$ est donc satisfiable, d'extension $\{2\}$. Mais $S(C, D \wedge F) = e^{-2}$ donc

$$\text{Ext}(C \circ_{e^{-1}}^d (D \wedge F)) = \{x \in \text{Ext}(D \wedge F) \mid S(C, x) \geq e^{-3}\} = \{2, 3\} \not\subseteq \text{Ext}((C \circ_{e^{-1}}^d D) \wedge F) = \{2\}$$

5.2 L'adaptation α -conservatrice dans $(\mathcal{U}, d, \mathcal{L}_C)$

La définition de l'adaptation α -conservatrice est similaire à celle de l'adaptation conservatrice (4.4). Étant donné un problème cas-cible, un cas source cas-srce, des connaissances du domaine CD et un opérateur de révision \circ^d , l'adaptation conservatrice donne cas-cible-complété tel que :

$$(CD \wedge \text{cas-srce}) \circ_\alpha^d (CD \wedge \text{cas-cible}) \equiv_{\text{CD}} \text{cas-cible-complété}$$

L'intérêt de cette extension de l'adaptation conservatrice est d'abord de garantir un résultat, mais aussi de proposer une adaptation « moins conservatrice » : si on note cas-cible-complété $_\alpha$ le résultat de l'adaptation α -conservatrice pour $\alpha \in [0, 1]$, la proposition 5.1 indique que, pour $1 \geq \alpha \geq \beta \geq 0$:

$$\text{cas-cible-complété}_1 \vDash \text{cas-cible-complété}_\alpha \vDash \text{cas-cible-complété}_\beta \vDash CD \wedge \text{cas-cible}$$

Exemple 5.3. En reprenant l'exemple 4.1, les extensions de tarte-poires-4p-complétée croissent à mesure que α décroît. Le « gonflement » de $\text{Ext}(\text{tarte-poires-4p-complétée})$ est lié à la forme de d , pour de grandes valeurs de α ($\alpha > e^{-12,55}$), seules les composantes continues (masse-pâte, sucre et donc sucre) peuvent varier. La première variation de nb-poires est permise pour $\alpha = e^{-12,55}$. Les gros coefficients de d_1 et d_2 font que nb-parts et type-plat ne varient que pour des valeurs de α très faibles ($\alpha = e^{-100}$ et e^{-1000} respectivement), ainsi pour $\alpha = e^{-12,15} > e^{-100}$, seul $y = (4, \text{dessert}, 66, 40, 240, 100, 2, 0) \in \text{Ext}(\text{tarte-pommes})$ est à distance de moins de $\Delta - \ln \alpha = 40, 15$ de tarte-poires-4p, donc par symétrie de d :

$$\begin{aligned} \text{Ext}(\text{tarte-poires-4p-complétée}_\alpha) &= \{x \in \mathcal{U} \mid d(\text{tarte-poires-4p}, x) = d(y, x) \leq 40, 15\} \\ &= \{(4, \text{dessert}, 66, 46, 100, 180, 0, 2)\} \\ &\cup \{(4, \text{dessert}, 30 + \text{saccharose}, \text{saccharose}, \text{masse-pâte}, 270, 0, 3) \mid \\ &\quad |36 - \text{saccharose}| + 0, 1|40 - \text{saccharose}| + 5|100 - \text{masse-pâte}| \leq 50, 6\} \end{aligned}$$

Car $50, 6 = 4 \times 40, 15 - 10 \times 2 - 10 \times 3 - 2 \times 30$.

5.3 L'adaptation conservatrice floue dans $(\mathcal{U}, d, \mathcal{L}_C)$

5.3.1 Rappels sur les sous-ensembles flous

Dans [7], Lotfi Zadeh définit un sous-ensemble flou F d'un univers \mathcal{U} comme étant une application F de \mathcal{U} dans le segment $[0, 1]$ des réels. Pour $x \in \mathcal{U}$, $F(x)$ représente son degré d'appartenance à F . En particulier un sous-ensemble classique A de \mathcal{U} est un sous-ensemble flou :

$$x \in \mathcal{U} \mapsto A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{sinon} \end{cases}$$

On note $\mathcal{F}(\mathcal{U})$ l'ensemble des sous-ensembles flous de \mathcal{U} . L. Zadeh généralise l'intersection, le complémentaire et l'union aux sous-ensembles flous de \mathcal{U} par les égalités suivantes où $x \in \mathcal{U}$:

$$\begin{aligned} (F \cap G)(x) &= \min(F(x), G(x)) & \overline{F}(x) &= 1 - F(x) \\ (F \cup G)(x) &= \max(F(x), G(x)) = \overline{\overline{F} \cap \overline{G}}(x) \end{aligned}$$

Pour $\alpha \in]0, 1]$, une α -coupe de F est un sous-ensemble classique de \mathcal{U} défini par :

$$F_\alpha = \{x \in \mathcal{U} \mid F(x) \geq \alpha\}$$

Un sous-ensemble flou F définit ainsi une famille de sous-ensembles emboîtés $(F_\alpha)_{\alpha \in]0, 1]}$ (si $\beta \geq \alpha$, alors $F_\beta \subseteq F_\alpha$). Réciproquement, une famille de sous-ensembles classiques emboîtés $(F_\alpha)_{\alpha \in]0, 1]}$ définit un unique sous-ensemble flou F de \mathcal{U} par :

$$F(x) = \sup\{\alpha \in]0, 1] \mid x \in F_\alpha\} \quad (5.1)$$

avec la convention $\sup \emptyset = 0$ dans $]0, 1]$ (d'où $F(x) = 0$ si x n'appartient à aucun F_α).

5.3.2 Révision floue avec des concepts classiques

L' α -révision vue en section 5.1 est définie par des extensions emboîtées paramétrées par $\alpha \in]0, 1]$, ce qui motive la définition de révision floue \square^d telle que l'extension de la révision floue d'un concept classique $C \in \mathcal{L}_C$ par un autre concept classique $D \in \mathcal{L}_C$ donne un concept flou caractérisé par une extension floue $\text{Ext}(C \square^d D)$ telle que pour $\alpha \in]0, 1]$, l' α -coupe $(\text{Ext}(C \square^d D))_\alpha = \text{Ext}(C \circ_\alpha^d D)$, soit, en reprenant la formule (5.1) pour tout $x \in \mathcal{U}$:

$$\text{Ext}(C \square^d D)(x) = \sup\{\alpha \in]0, 1] \mid x \in \text{Ext}(C \circ_\alpha^d D)\}$$

5.3.3 Adaptation conservatrice floue de cas classiques

On définit l'adaptation conservatrice floue de concepts classiques à partir de l'adaptation α -conservatrice en adoptant la notion de révision floue vue dans la section précédente. Ainsi si $\text{cas-cible} \in \mathcal{L}_C$ et $\text{cas-srce} \in \mathcal{L}_C$ sont des concepts classiques, l'adaptation conservatrice floue donne le concept flou $\text{cas-cible-complété}$ tel que :

$$(\text{CD} \wedge \text{cas-srce}) \sqcap^d (\text{CD} \wedge \text{cas-cible}) \equiv_{\text{CD}} \text{cas-cible-complété}$$

Cette définition a l'inconvénient de mêler des concepts classiques et des concepts flous. En particulier comme nous l'avons mentionné dans la section 2, $\text{cas-cible-complété}$ peut ne pas être complètement résolu et devoir être complété par d'autres cas sources, ce qui nécessite la définition d'une adaptation qui s'applique sur les concepts flous. Cela motive l'étude de la révision floue que nous ne traiterons pas dans cet article.

5.3.4 Exemple

En reprenant l'exemple 5.3, le résultat de l'adaptation conservatrice de tarte-pommes par tarte-poires-4p sera le concept flou $\text{tarte-poires-4p-complétée}$ donné par :

$$(\text{CD} \wedge \text{tarte-pommes}) \sqcap^d (\text{CD} \wedge \text{tarte-poires-4p}) \equiv_{\text{CD}} \text{tarte-poires-4p-complétée}$$

Et le degré d'appartenance de la recette de tarte aux poires faite avec 2 poires est $\text{Ext}(\text{tarte-poires-4p-complétée})(4, \text{dessert}, 66, 46, 100, 180, 0, 2) = e^{-12.55}$.

6 Conclusion et perspectives

Dans cet article, nous avons proposé une extension de l'adaptation conservatrice, précédemment définie en logique propositionnelle finie, à un formalisme généralisant le formalisme « attribut-valeur simple » habituellement utilisé en raisonnement à partir de cas. La formalisme proposé repose sur un espace métrique, les interprétations étant des éléments de cet espace.

À la base de l'adaptation conservatrice, les postulats de Katsuno et Mendelzon pour un opérateur de révision des connaissances — équivalent ici aux postulats AGM — ont été généralisés à notre formalisme de représentation des cas. L'extension de l'adaptation conservatrice se faisant alors naturellement à partir d'un tel opérateur de révision, nous nous sommes intéressés à son existence et à sa satisfaction aux postulats. Nous avons étudié en particulier le cas d'un opérateur de révision défini à partir d'une distance sur l'espace des extensions.

Si dans un espace métrique discret cet opérateur satisfait bien les postulats AGM, dans le cadre général ceci n'est vrai que sous certaines conditions sur la distance. En particulier la consistance de la révision, et donc la satisfiabilité du résultat de l'adaptation conservatrice, est liée à l'existence de minimas pour la distance entre deux extensions. Cela a motivé l'étude d'une adaptation conservatrice floue et donc d'un révision floue afin d'assouplir le conservatisme de l'adaptation et de garantir la satisfiabilité de son résultat.

Cependant, l'étude de la révision floue est encore incomplète. Il serait intéressant de la prolonger aux concepts flous afin de lui donner un domaine plus cohérent, notamment pour la rendre itérable. Il faut pour cela définir la révision sur les concepts flous.

L'adaptation conservatrice s'appuie sur deux sources de connaissances : les connaissances du domaine CD et les connaissances correspondant au choix de l'opérateur de révision \circ , i.e., quand $\circ = \circ^d$, au choix de la distance d sur \mathcal{U} . CD représente des connaissances *statiques* au sens où elles donnent des conditions nécessaires pour que $x \in \mathcal{U}$ — et, partant, $C \in \mathcal{L}_C$ — soit licite. En revanche, CD ne permet pas d'exprimer des connaissances *dynamiques* qui concernent la variation d'un $x \in \mathcal{U}$ à un $x' \in \mathcal{U}$. Pour reprendre l'exemple de la tarte aux poires, on aimerait pouvoir exprimer que, tous les autres attributs étant fixés, la quantité de pâte augmente strictement avec la masse de fruits ($\frac{\partial \text{masse-p\^ate}}{\partial \text{masse-fruit}} > 0$ si cet abus de notation nous est permis). Une perspective de recherche consisterait à « intégrer » dans d ce type de connaissances dynamiques ; pour l'exemple, cela devra empêcher que l'adaptation conservatrice viole la contrainte $\frac{\partial \text{masse-p\^ate}}{\partial \text{masse-fruit}} > 0$.

Dans cet article, seule la condition (i) de la définition d'une distance nous a été nécessaire, des fonctions d ne vérifiant pas les autres conditions peuvent être utilisées. Pour être plus général, il faudrait étendre ce qui

a été obtenu pour un opérateur de révision défini à partir d'une distance aux opérateurs de révision définis à partir d'assignements fidèles, et en particulier, de définir une révision floue associée. En parallèle, on pourrait envisager l'extension à d'autres espaces topologiques.

Remerciements

Les auteurs remercient les deux relecteurs pour leurs remarques et les prient de les excuser de n'en avoir que partiellement tenu compte pour cet article. Qu'ils soient assurés, néanmoins, que les remarques qui ont été faites serviront de base de réflexion pour les travaux futurs.

Annexes

Preuve de la proposition 4.1.

La satisfaction de (R1) – (R6) implique l'existence d'un assignement fidèle vérifiant (4.2) : Étant donné un opérateur de révision \circ vérifiant (R1) – (R6) et un concept $C \in \mathcal{L}_C$ on définit comme dans [4] une relation \leq_C telle que pour $x, y \in \mathcal{U}$:

$$x \leq_C y \quad \text{si} \quad (x \in \text{Ext}(C) \text{ ou } x \in \text{Ext}(C \circ D)) \quad \text{où } D \in \mathcal{L}_C \text{ et } \text{Ext}(D) = \{x, y\}$$

La démonstration de Katsuno et Mendelzon s'applique à notre formalisme et montre que l'application $C \mapsto \leq_C$ est un assignement fidèle et que $\text{Ext}(C \circ D) = \text{Min}(\text{Ext}(D), \leq_C)$. En particulier si $D \in \mathcal{L}_C$ est satisfiable, (R3) implique que $\text{Ext}(C \circ D) \neq \emptyset$, \leq_C vérifie donc (4.2).

Réciproque : La démonstration de Katsuno et Mendelzon s'applique encore ici sauf pour le prédicat (R3) qui est une conséquence de la condition 4.2. \square

Preuve de la proposition 4.2. Soit $A, B \in 2^{\mathcal{L}_C}$ tels que $Cn(\emptyset) \subsetneq Cn(B) \subsetneq Cn(A)$, soit $D \in Cn(B) \setminus Cn(\emptyset) \subseteq Cn(A) \setminus Cn(\emptyset)$. Posons $C = \{F \vee \neg D \mid F \in A\} \in 2^{\mathcal{L}_C}$, montrons que $Cn(C) \subsetneq Cn(A)$:

- Pour $F \in A$, $F \vDash F \vee \neg D$ donc $C \subseteq Cn(A)$ et par monotonie et itération de Cn , $Cn(C) \subseteq Cn(A)$.
- De même, $C \subseteq Cn(\neg D)$ donc, si $Cn(A) \subseteq Cn(C)$, alors on aurait $D \in Cn(A) \subseteq Cn(\neg D)$, ce qui est absurde puisque $D \notin Cn(\emptyset)$. D'où $Cn(A) \not\subseteq Cn(C)$.

On a donc $Cn(C) \subsetneq Cn(A)$. Considérons maintenant $Cn(B \cup C)$, par monotonie et itération de Cn , $Cn(B \cup C) \subseteq Cn(A)$. Réciproquement pour $F \in A$, $(F \vee \neg D) \in C$ et $D \in Cn(B)$ donc $F \in Cn(\{D, (F \vee \neg D)\}) \subseteq Cn(Cn(B) \cup Cn(C)) \subseteq Cn(Cn(B \cup C)) = Cn(B \cup C)$, on en déduit que $A \subseteq Cn(B \cup C)$ et $Cn(A) = Cn(B \cup C)$. \square

Preuve de la proposition 4.3.

- (R1) est vérifié par construction de \circ^d : $\text{Ext}(C \circ^d D) \subseteq \text{Ext}(D)$.
- (R4) : Il s'agit de la non pertinence de la syntaxe.
Si $C \equiv C'$ et $D \equiv D'$ alors $\text{Ext}(C) = \text{Ext}(C')$ et $\text{Ext}(D) = \text{Ext}(D')$ et donc $\text{Ext}(C \circ^d D) = \text{Ext}(C' \circ^d D')$.
- Pour (R5) et (R6), il y a deux cas à considérer :

Premier cas : $\text{Ext}((C \circ^d D) \wedge F) = \emptyset$. Alors $(C \circ^d D) \wedge F$ est insatisfiable et (R5) et (R6) sont automatiquement vérifiés.

Deuxième cas : $\text{Ext}((C \circ^d D) \wedge F) \neq \emptyset$, soit $x \in \text{Ext}((C \circ^d D) \wedge F)$, d'après la définition de \circ^d , comme $x \in \text{Ext}(C \circ^d D)$:

$$d(C, x) = d(C, D) = \inf_{u \in \text{Ext}(D)} d(C, u) \leq \inf_{u \in \text{Ext}(D) \cap \text{Ext}(F)} d(C, u) \leq d(C, D \wedge F)$$

Or d'après (R1), $\text{Ext}(C \circ^d D) \subseteq \text{Ext}(D)$ et $x \in \text{Ext}(D \wedge F)$, donc $d(C, D \wedge F) \leq d(C, x)$ et :

$$d(C, D) = d(C, D \wedge F)$$

Par conséquent :

$$\begin{aligned}\text{Ext}(C \circ^d (D \wedge F)) &= \{x \in \text{Ext}(D \wedge F) \mid d(C, x) = d(C, D \wedge F)\} \\ &= \{x \in \text{Ext}(D \wedge F) \mid d(C, x) = d(C, D)\} \\ &= \{x \in \text{Ext}(D) \mid d(C, x) = d(C, D)\} \cap \text{Ext}(F) = \text{Ext}((C \circ^d D) \wedge F)\end{aligned}$$

Ce qui montre (R5) et (R6). □

Preuve de la proposition 4.4. On considère $\mathcal{U} = \mathbb{R}$, $d : (x, y) \mapsto |y - x|$ et \mathcal{E} l'ensemble des unions finies d'intervalles. Soit $C \in \mathcal{L}_C$ tels que $\text{Ext}(C) = [0, 1[$, on a $d(C, C) = 0$ et donc $\text{Ext}(C \circ^d C) = \{x \in \mathcal{U} \mid d(C, x) = 0\} = [0, 1]$. Cela contredit (R2) puisque $\text{Ext}(C \circ^d C) \neq \text{Ext}(C \wedge C)$ alors que $C \wedge C$ est satisfiable.

Soit $D \in \mathcal{L}_C$ tel que $\text{Ext}(D) =]1, 2]$, $d(C, D) = 0$ alors que cette distance n'est pas atteinte, ce qui donne $\text{Ext}(C \circ^d D) = \{x \in]1, 2] \mid d([0, 1[, x) = 0\} = \emptyset$. On a ici une contradiction avec (R3) puisque D est satisfiable mais $C \circ^d D$ ne l'est pas. □

Preuve de la proposition 4.5.

(4.5) entraîne (R2) : Supposons (4.5), si $C \wedge D$ est satisfiable alors $\text{Ext}(C) \cap \text{Ext}(D) = \text{Ext}(C \wedge D) \neq \emptyset$ et donc $d(C, D) = 0$ ($\Delta = 0$).

$$\begin{aligned}\text{Ext}(C \wedge D) &= \{x \in D \mid x \in C\} = \{x \in D \mid d(C, x) = 0\} \quad \text{d'après (4.5)} \\ &= \text{Ext}(C \circ^d D)\end{aligned}$$

(R2) entraîne (4.5) : Supposons (R2), soit $A \in \mathcal{E}$, $x \in \mathcal{U}$ et $C \in \mathcal{L}_C$ tels que $\text{Ext}(C) = A$. Si $d(A, x) = 0 < +\infty$ alors d'après (3.1) $A \neq \emptyset$, or $A = \text{Ext}(C) = \text{Ext}(C) \cap \mathcal{U} = \text{Ext}(C) \cap \text{Ext}(\top) = \text{Ext}(C \wedge \top) \neq \emptyset$. (R2) entraîne $C \circ^d \top \equiv C \wedge \top \equiv C$, d'où : $x \in \text{Ext}(C \circ^d \top) = \text{Ext}(C) = A$. Donc $x \in A$ et (4.5) est vérifié.

(4.5) est équivalent à (4.6) : En effet, $C \mapsto \leq_C^d$ vérifie les conditions (1) et (3) de la définition d'assignement fidèle 4.1, il s'agit donc d'un assignement fidèle ssi elle vérifie la condition (2), c'est-à-dire si pour tous $C \in \mathcal{L}_C$, $x \in \text{Ext}(C)$, $y \notin \text{Ext}(C)$, $d(C, x) < d(C, y)$. Or, si $x \in \text{Ext}(C)$ alors $d(C, x) = 0$ et donc $C \mapsto \leq_C^d$ est un assignement fidèle ssi pour tout $y \notin \text{Ext}(C)$, $d(C, y) > 0$, ou, en prenant la contraposée, si pour tout $y \in \mathcal{U}$, $d(C, y) = 0$ implique que $y \in \text{Ext}(C)$, ce qui, d'après la définition de \mathcal{E} est équivalent à (4.5). □

Preuve de la proposition 4.6.

(4.7) entraîne (R3) : Supposons (4.7), si D est satisfiable et $\Delta = d(C, D)$ alors (4.7) implique qu'il existe $x \in \text{Ext}(D)$ tel que $d(C, x) = \Delta$. Et donc $\text{Ext}(C \circ^d D) \neq \emptyset$ et $C \circ^d D$ est satisfiable.

(R3) entraîne (4.7) : Supposons (R3), soient $A, B \in \mathcal{E}$ avec $B \neq \emptyset$ et $\Delta = d(A, B)$. On considère C et D caractérisés par leur extensions respectives A et B . D est satisfiable donc d'après (R3) $C \circ^d D$ l'est aussi. Or $\text{Ext}(C \circ^d D) = \{x \in B \mid d(A, x) = \Delta\}$ donc il existe $x \in B$ tel que $d(A, x) = \Delta$. □

(4.7) est équivalent à (4.8) : En effet, pour $C, D \in \mathcal{L}_C$ si $\Delta = d(C, D) = \inf_{x \in \text{Ext}(D)} d(C, x)$ pour tout $\varepsilon > 0$ il existe $x \in \text{Ext}(D)$ tel que $d(C, x) < \Delta + \varepsilon$. Donc pour tout $x \in \text{Ext}(D)$ si $d(C, x) > \Delta$ alors il existe $y \in \text{Ext}(D)$ tel que $d(C, x) > d(C, y)$, et $x \in \text{Min}(\text{Ext}(D), \leq_C^d)$ ssi $d(C, x) = \Delta$. Ainsi $\text{Min}(\text{Ext}(D), \leq_C^d) \neq \emptyset$ ssi il existe $x \in \text{Ext}(D)$ tel que $d(C, x) = \Delta$ ce qui est équivalent à (4.7).

Preuve de la proposition 5.2. Soit $\alpha \in [0, 1]$, \circ_α^d vérifie (R1) et (R4) par définition. Soit $C, D, F \in \mathcal{L}_C$, si $x \in \text{Ext}((C \circ_\alpha^d D) \wedge F)$ alors d'après (R1) $x \in \text{Ext}(D \wedge F)$ et

$$S(C, x) \geq S(C, D) \times \alpha = \sup_{y \in \text{Ext}(D)} S(C, y) \times \alpha \geq \sup_{y \in \text{Ext}(D) \cap \text{Ext}(F)} S(C, y) \times \alpha = S(C, D \wedge F) \times \alpha$$

donc $x \in \text{Ext}(C \circ_\alpha^d (D \wedge F))$ et \circ_α^d vérifie (R5). □

Preuve de la proposition 5.3. Soit $\alpha < 1$ et $C, D \in \mathcal{L}_C$, avec $\text{Ext}(D) \neq \emptyset$. On a $S(C, D) = \sup_{y \in \text{Ext}(D)} S(C, y)$ d'après (3.4).

- Si C est insatisfiable, alors d’après (3.1), $S(C, D) = 0$, alors pour tout $y \in \text{Ext}(D)$, $S(C, y) = 0 = S(C, D) \times \alpha$. Donc par définition de \circ_{α}^d (5.1), pour tout $\alpha \in [0, 1]$, $\text{Ext}(C \circ_{\alpha}^d D) = \text{Ext}(D) \neq \emptyset$. Et donc $C \circ_{\alpha}^d D$ est satisfiable.
- Si C est satisfiable, alors $S(C, D) > 0$. Soit $\Sigma = S(C, D)$. Par définition de la borne supérieure, et comme $\Sigma \times \alpha < \Sigma$, il existe $y \in \text{Ext}(D)$ tel que $S(C, y) \geq \Sigma \times \alpha$. Donc $y \in \text{Ext}(C \circ_{\alpha}^d D)$. Par conséquent $C \circ_{\alpha}^d D$ est satisfiable.

Dans tous les cas, si $\alpha < 1$, alors \circ_{α}^d vérifie (R3). □

Références

- [1] M. Dalal. Investigations into a theory of knowledge base revision : Preliminary report. In *AAAI*, pages 475–479, 1988.
- [2] G. Antoniou G. Flouris, D. Plexousakis. Generalizing the AGM Postulates : Preliminary Results and Applications. In *Proceedings of the 10th International Workshop on Non-Monotonic Reasoning 2004 (NMR-04)*, 2004.
- [3] P. Gärdenfors, editor. *Belief Revision*. Cambridge University Press, 1992.
- [4] H. Katsuno et A. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3) :263–294, 1991.
- [5] J. Lieber. Application of the Revision Theory to Adaptation in Case-Based Reasoning : The Conservative Adaptation. In *Proceedings of the 7th ICCBR*, 2007.
- [6] C. K. Riesbeck et R. C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1989.
- [7] L. A. Zadeh. Fuzzy Sets. *Information and Control*, 8 :338–353, 1965.