Actes JIAF-JFPDA 2024

Appliquer la logique des variations propositionnelles à la représentation de règles d'adaptation pour le raisonnement à partir de cas *

Nicolas François[1] **Jean Lieber**[2]
1 Lycée public Chopin, Nancy
2 Université de Lorraine, CNRS, Inria, LORIA, Nancy
`nicolas.francois@free.fr` `jean.lieber@loria.fr`

Résumé

La logique des variations propositionnelles a été étudiée dans un article précédent. Ce formalisme est issu d'une notation issue du raisonnement à partir de cas (RàPC), avec l'ajout d'une sémantique. Cet article développe cette étude avec l'objectif de formaliser à l'aide de cette logique la notion de règles d'adaptation en RàPC et leur utilisation.

Abstract

The logic of propositional variations was studied in a previous article. This formalism is based on a notation derived from case-based reasoning (CBR), with the addition of a semantics. This article develops this study with the aim of using this logic to formalize the notion of adaptation rules in CBR and their use.

1 Introduction

Le raisonnement à partir de cas (RàPC [12]) consiste à résoudre un problème en s'appuyant sur une base de cas, où un cas est la représentation d'un épisode de résolution de problème. Le modèle de processus du RàPC consiste généralement, étant donné un problème cible (problème à résoudre) en la sélection dans la base de cas d'un (parfois de plusieurs) cas jugé(s) similaire(s) au problème cible (étape de *remémoration*) et dans la modification de ce (ou ces) cas dans l'optique de la résolution du problème cible (étape d'*adaptation*).

L'adaptation peut s'appuyer sur des règles d'adaptation indiquant comment une variation entre problèmes peut se répercuter en variations entre solutions. Des notations ont été introduites pour coder ces variations, en vue de processus d'apprentissages de connaissances d'adaptation (voir, par exemple, [4]). Ces notations ont été dotées d'une sémantique pour obtenir une logique, qui a été étudiée, pour elle-même, dans [7].

L'objectif de cet article est d'étudier des liens entre cette logique et l'adaptation en RàPC, avec l'idée que les notions relatives aux variations (et aux différences, et aux (dis)similarités), fréquemment utilisées dans ce domaine, puissent être représentées explicitement, i.e. qu'on puisse les coder et raisonner avec. Plus particulièrement, elle permet d'étudier la représentation de règles d'adaptation.

La section 2 décrit la logique des variations propositionnelles (et résume ainsi [7]) et les notations et notions générales relatives au RàPC. Quelques études sur cette logique se sont avérées nécessaires pour l'étude de son application au RàPC : elles sont introduites dans la section 3. La section 4 étudie comment les logiques de variations peuvent être utilisées pour représenter des connaissances d'adaptation. La section 5 présente une discussion en lien avec d'autres travaux.

Les preuves complètes des résultats de cet article se trouvent dans le rapport [8], version étendue de cet article.

2 Préliminaires

Ces préliminaires sont en deux parties : l'une concerne la logique $(\Delta\mathcal{LP}, \models)$, l'autre, le RàPC.

*Les auteurs tiennent à remercier Tiago de Lima qui leur a conseillé des lectures sur les logiques dynamiques qui ont des liens avec la logique des variations propositionnelles, ainsi que cela est mis en évidence dans la section 5.2 de cet article. Ils veulent également exprimer leur gratitude envers Pierre Marquis pour ses conseils judicieux. Que leurs chemins soient pavés de fleurs de cerisiers. Les auteurs tiennent également à remercier les relecteurs de cet article pour leurs remarques.

2.1 La logique des variations propositionnelles

Cette section reprend les points principaux sur la logique des variations propositionnelles $(\Delta\mathcal{LP}, \models)$ introduite et décrite dans [7].

La logique propositionnelle finie $(\mathcal{LP}, \models)$ est la logique de base sur laquelle va être définie $(\Delta\mathcal{LP}, \models)$.

On se donne un ensemble fini de symboles $\mathcal{V}$: un élément de $\mathcal{V}$ est appelé *variable propositionnelle* (ou simplement *variable*). Une formule de $(\mathcal{LP}, \models)$ est soit une variable propositionnelle, soit d'une des formes suivantes : $\neg\beta$, $\beta_1 \wedge \beta_2$ et $\beta_1 \vee \beta_2$, où β, β_1 et β_2 sont des formules propositionnelles. On utilisera les abréviations usuelles suivantes (où <u>*euad*</u> se lit « est une abréviation de », avec a, une variable choisie arbitrairement et $\beta_1, \beta_2 \in \mathcal{LP}$) :

$$\top \ \underline{\textit{euad}}\ a \vee \neg a \qquad \bot \ \underline{\textit{euad}}\ a \wedge \neg a$$
$$\beta_1 \rightarrow \beta_2 \ \underline{\textit{euad}}\ \neg\beta_1 \vee \beta_2$$

Un littéral de $(\mathcal{LP}, \models)$ est une formule de la forme a ou de la forme $\neg a$ ($a \in \mathcal{V}$). Un cube de cette logique est une conjonction de littéraux dans laquelle une variable n'apparaît au plus qu'une fois. Un cube est complet si toutes les variables apparaissent dans ce cube.

L'ensemble des variables ayant des occurrences dans $\alpha \in \mathcal{LP}$ est noté $\mathcal{V}(\alpha)$.

On définit la sémantique en théorie des modèles comme suit. Une *interprétation* dans la logique propositionnelle est une fonction $\mathcal{I} : \mathcal{V} \rightarrow \{\mathbb{0}, 1\}$ où $\{\mathbb{0}, 1\}$ est l'ensemble des booléens. On note Ω l'ensemble des interprétations. Soit $\mathcal{I} \in \Omega$ et $\alpha \in \mathcal{LP}$. On définit la relation « $\mathcal{I}$ satisfait α » ($\mathcal{I} \models \alpha$, i.e. $\mathcal{I}$ est un modèle de α) de la façon suivante :

- $\mathcal{I} \models a$ si $\mathcal{I}(a) = 1$ pour une variable a ;
- $\mathcal{I} \models \neg\alpha$ si $\mathcal{I} \not\models \alpha$ pour $\alpha \in \mathcal{LP}$;
- $\mathcal{I} \models \alpha_1 \wedge \alpha_2$ si $\mathcal{I} \models \alpha_1$ et $\mathcal{I} \models \alpha_2$ pour $\alpha_1, \alpha_2 \in \mathcal{LP}$;
- $\mathcal{I} \models \alpha_1 \vee \alpha_2$ si $\mathcal{I} \models \alpha_1$ ou $\mathcal{I} \models \alpha_2$ pour $\alpha_1, \alpha_2 \in \mathcal{LP}$.

L'ensemble des modèles de α est noté $\mathcal{M}(\alpha)$. Si $\mathcal{B}$ est un ensemble fini de formules, $\mathcal{M}(\mathcal{B})$ est l'intersection des $\mathcal{M}(\alpha)$ pour $\alpha \in \mathcal{B}$, ce qui permet d'assimiler $\mathcal{B}$ à la conjonction $\bigwedge \mathcal{B}$ de ses éléments. On définit alors la relation $\mathcal{B} \models \beta$ ($\mathcal{B}$ *entraîne* β) par $\mathcal{M}(\mathcal{B}) \subseteq \mathcal{M}(\beta)$ et la relation $\alpha \models_{\mathcal{B}} \beta$ par $\mathcal{B} \cup \{\alpha\} \models \beta$. Une formule α est *satisfiable* si $\mathcal{M}(\alpha) \neq \emptyset$ (on le notera souvent $\alpha \not\models \bot$ dans l'article). Enfin, avec $\alpha, \beta \in \mathcal{LP}$, $\alpha \equiv \beta$ si $\mathcal{M}(\alpha) = \mathcal{M}(\beta)$.

La syntaxe de la logique $(\Delta\mathcal{LP}, \models)$ diffère légèrement de celle qui était donnée dans [7], au sens où certains constructeurs dans cet article sont ici des abréviations et inversement. Une formule $\varphi \in \Delta\mathcal{LP}$ est une expression d'une des formes suivantes : $\alpha \triangleright \beta$ (ce qu'on peut lire « α devient β »), $\neg\psi$, $\psi_1 \wedge \psi_2$ et $\psi_1 \vee \psi_2$, où $\alpha, \beta \in \mathcal{LP}$ et $\psi, \psi_1, \psi_2 \in \Delta\mathcal{LP}$.

On utilisera les abréviations suivantes :

$$\begin{array}{rcl}
\top_\Delta \ \underline{\textit{euad}}\ \top \triangleright \top & & \bot_\Delta \ \underline{\textit{euad}}\ \bot \triangleright \bot \\
\psi_1 \rightarrow \psi_2 \ \underline{\textit{euad}}\ \neg\psi_1 \vee \psi_2 & & \\
\alpha^{=1} \ \underline{\textit{euad}}\ \alpha \triangleright \alpha & & \alpha^{=\mathbb{0}} \ \underline{\textit{euad}}\ \neg\alpha \triangleright \neg\alpha \\
\alpha^{+} \ \underline{\textit{euad}}\ \neg\alpha \triangleright \alpha & & \alpha^{-} \ \underline{\textit{euad}}\ \alpha \triangleright \neg\alpha \\
\alpha^{=} \ \underline{\textit{euad}}\ \alpha^{=1} \vee \alpha^{=\mathbb{0}} & & \alpha^{\neq} \ \underline{\textit{euad}}\ \alpha^{+} \vee \alpha^{-} \\
\alpha^{\mathbb{0}\bullet} \ \underline{\textit{euad}}\ \alpha^{=\mathbb{0}} \vee \alpha^{+} & & \alpha^{1\bullet} \ \underline{\textit{euad}}\ \alpha^{=1} \vee \alpha^{-} \\
\alpha^{\bullet\mathbb{0}} \ \underline{\textit{euad}}\ \alpha^{=\mathbb{0}} \vee \alpha^{-} & & \alpha^{\bullet 1} \ \underline{\textit{euad}}\ \alpha^{=1} \vee \alpha^{+}
\end{array}$$

Quand le contexte ne sera pas ambigu, $\top_\Delta$ et $\bot_\Delta$ seront notés simplement $\top$ et $\bot$. On note $\mathcal{D}_0 = \{=1, =\mathbb{0}, +, -\}$ et ses éléments sont appelés *symboles de variations primitifs*. L'ensemble $\mathcal{D}_1 = \{=, \neq, \mathbb{0}\bullet, 1\bullet, \bullet\mathbb{0}, \bullet 1\}$ est celui des *symboles de variation non primitifs*. $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1$ est donc l'ensemble des symboles de variations.

Un littéral de $(\Delta\mathcal{LP}, \models)$ est une formule de la forme a^v où $a \in \mathcal{V}$ et $v \in \mathcal{D}$. Un *cube* de cette logique est une conjonction de littéraux dans laquelle une variable n'apparaît au plus qu'une fois. Un cube est *primitif* si tous les symboles de variation qui y apparaissent sont primitifs. Un cube est *complet* s'il est primitif et si toutes les variables apparaissent dans ce cube.

Pour $\varphi \in \Delta\mathcal{LP}$, $x \in \mathcal{V}$ et $\alpha \in \mathcal{LP}$, $\varphi[x\backslash\alpha]$ est le résultat de la substitution de x par α dans φ. L'ensemble des variables ayant des occurrences dans φ est noté $\mathcal{V}(\varphi)$. La taille de φ, notée $|\varphi|$, est le nombre d'occurrences de connecteurs qu'elle contient.

Sémantique de $(\Delta\mathcal{LP}, \models)$**.** Soit $\Delta\Omega = \Omega \times \Omega$. Un élément $(\mathcal{I}, \mathcal{J})$ de $\Delta\Omega$ est appelé interprétation (pour cette logique) et sera noté simplement $\mathcal{I}\mathcal{J}$. Pour $\varphi \in \Delta\mathcal{LP}$, la relation $\mathcal{I}\mathcal{J} \models \varphi$ est définie comme suit :

- $\mathcal{I}\mathcal{J} \models \alpha \triangleright \beta$ si $\mathcal{I} \models \alpha$ et $\mathcal{J} \models \beta$.
- $\mathcal{I}\mathcal{J} \models \neg\psi$ si $\mathcal{I}\mathcal{J} \not\models \psi$.
- $\mathcal{I}\mathcal{J} \models \psi_1 \wedge \psi_2$ si $\mathcal{I}\mathcal{J} \models \psi_1$ et $\mathcal{I}\mathcal{J} \models \psi_2$.
- $\mathcal{I}\mathcal{J} \models \psi_1 \vee \psi_2$ si $\mathcal{I}\mathcal{J} \models \psi_1$ ou $\mathcal{I}\mathcal{J} \models \psi_2$.

On définit alors $\mathcal{M}(\varphi) = \{\mathcal{I}\mathcal{J} \in \Delta\Omega \mid \mathcal{I}\mathcal{J} \models \varphi\}$. On note en particulier que $\mathcal{M}(\alpha \triangleright \beta) = \mathcal{M}(\alpha) \times \mathcal{M}(\beta)$. On définit les notions de satisfiabilité, de conséquence logique ($\models$) et d'équivalence logique ($\equiv$) de la même façon qu'en logique propositionnelle.

Pour $\varphi \in \Delta\mathcal{LP}$, on notera $\mathsf{G}(\varphi)$ et $\mathsf{D}(\varphi)$ des formules propositionnelles (uniques à l'équivalence logique près) telles que

$$\mathcal{M}(\mathsf{G}(\varphi)) = \{\mathcal{I} \mid \mathcal{I}\mathcal{J} \in \mathcal{M}(\varphi)\}$$
$$\mathcal{M}(\mathsf{D}(\varphi)) = \{\mathcal{J} \mid \mathcal{I}\mathcal{J} \in \mathcal{M}(\varphi)\}$$

En particulier, pour $\alpha, \beta \in \mathcal{LP}$, $\mathsf{G}(\alpha \triangleright \beta) \equiv \alpha$ et $\mathsf{D}(\alpha \triangleright \beta) \equiv \beta$ (G comme « gauche », D comme « droite »).

Quelques résultats. Les points suivants résument et complètent certains résultats de [7] :

— Tout sous-ensemble A de $\Delta\Omega$ est *représentable* dans la logique : il existe $\varphi \in \Delta\mathcal{LP}$ telle que $\mathcal{M}(\varphi) = A$.

— On a la plupart des résultats classiques de la logique propositionnelle : théorème de la déduction, lois de De Morgan, formes normales conjonctives et disjonctives, etc.

— Toute formule φ peut se mettre sous la forme d'une formule obtenue à partir d'une formule de logique propositionnelle, en substituant à ses variables des formules de la forme a^v où $a \in \mathcal{V}$ et $v \in \mathcal{D}_0$. Par exemple :

$$a \wedge b \triangleright \neg a \wedge c \quad \equiv \quad a^{-} \wedge (b^{=1} \vee b^{-}) \wedge (c^{=1} \vee c^{+})$$

— En particulier, on peut écrire toute formule de $(\Delta\mathcal{LP}, \models)$ sous forme normale disjonctive (FND), i.e. sous la forme de disjonction de cubes.

— On a la propriété de distributivité de $\triangleright$ sur $\wedge$ (modulo l'équivalence) :

$$(\alpha_1 \wedge \alpha_2) \triangleright \beta \equiv (\alpha_1 \triangleright \beta) \wedge (\alpha_2 \triangleright \beta)$$
$$\alpha \triangleright (\beta_1 \wedge \beta_2) \equiv (\alpha \triangleright \beta_1) \wedge (\alpha \triangleright \beta_2)$$

pour $\alpha, \alpha_1, \alpha_2, \beta, \beta_1, \beta_2 \in \mathcal{LP}$. On a également la distributivité de $\triangleright$ sur $\vee$.

— Pour $a \in \mathcal{V}$, soit $\ell(a)$ et $m(a)$ deux littéraux construits sur a ($\ell(a), m(a) \in \{a, \neg a\}$). On a :

$$\bigwedge_{a\in\mathcal{V}} \ell(a) \triangleright \bigwedge_{a\in\mathcal{V}} m(a) \quad \equiv \quad \bigwedge_{a\in\mathcal{V}} \ell(a) \triangleright m(a) \quad (1)$$

et chaque $\ell(a) \triangleright m(a)$ peut s'écrire a^v où $v \in \mathcal{D}_0$.

— On peut définir un système formel correct et complet pour $(\Delta\mathcal{LP}, \models)$.

— Le problème de décision $\mathcal{B} \models \varphi$ est NP-complet et plusieurs approches pour l'implanter ont été envisagées.

— On a le résultat suivant (avec $\varphi, \psi \in \Delta\mathcal{LP}$) :

$$\mathsf{D}(\varphi \vee \psi) \equiv \mathsf{D}(\varphi) \vee \mathsf{D}(\psi) \qquad (2)$$

— Pour $\varphi \in \Delta\mathcal{LP}$, on peut définir facilement $\varphi^{-1} \in \Delta\mathcal{LP}$ telle que $\mathcal{IJ} \models \varphi^{-1}$ ssi $\mathcal{JI} \models \varphi$ (il suffit d'inverser les paramètres de $\triangleright$).

2.2 Rappels sur le raisonnement à partir de cas

Généralités. Le RàPC est un type de raisonnement s'appuyant sur une base de cas où un cas est une représentation d'un épisode de résolution de problème.

Un domaine d'application pour le RàPC est donné par un triplet $(\mathcal{P}, \mathcal{S}, \rightsquigarrow)$ où $\mathcal{P}$ et $\mathcal{S}$ sont des ensembles et $\rightsquigarrow$ est une relation binaire sur $\mathcal{P} \times \mathcal{S}$. Un élément de $\mathtt{x}$ est appelé *problème*, un élément de $\mathtt{y}$, *solution* et $\mathtt{x} \rightsquigarrow \mathtt{y}$ se lit « $\mathtt{x}$ a pour solution $\mathtt{y}$ ».

En général, la relation $\rightsquigarrow$ n'est pas connue du système de RàPC dont l'objectif est de construire une *hypothèse de solution* $\mathtt{y}^{\mathtt{cible}}$ à un problème donné, le *problème cible*, noté $\mathtt{x}^{\mathtt{cible}}$.

On considère souvent (et cela sera le cas dans cet article) qu'un cas est donné par un couple $(\mathtt{x}, \mathtt{y}) \in \mathcal{P}\times\mathcal{S}$ où $\mathtt{x} \rightsquigarrow \mathtt{y}$: la relation $\rightsquigarrow$ est connue pour un ensemble fini qui constitue la *base de cas*. On note $(\mathtt{x}^s, \mathtt{y}^s)$ un élément de la base de cas et on l'appelle *cas source* ($\mathtt{x}^s$ est un *problème source*).

Le processus de RàPC le plus courant consiste en une étape de remémoration (sélection de k éléments de la base de cas jugés similaires au problème cible) et d'adaptation (réutilisation des cas remémorés pour résoudre $\mathtt{x}^{\mathtt{cible}}$). Dans cet article, ne seront considérés que des processus de remémoration et d'adaptation simples, i.e. $k = 1$. Pour l'adaptation, cela signifie qu'un *problème d'adaptation* sera donné par un cas source et un problème cible : on le notera $((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^{\mathtt{cible}})$.

L'approche de l'adaptation considérée dans cet article, parfois appelée *adaptation transformationnelle* (voir [2]) vise à résoudre le problème d'adaptation en le lisant sous la forme « La solution cherchée, $\mathtt{y}^{\mathtt{cible}}$, est à $\mathtt{y}^s$ ce que $\mathtt{x}^{\mathtt{cible}}$ est à $\mathtt{x}^s$. » On peut l'interpréter ainsi : des variations entre les problèmes $\mathtt{x}^s$ et $\mathtt{x}^{\mathtt{cible}}$, notées $\Delta\mathtt{x}$, on infère des variations entre la solution $\mathtt{y}^s$ et l'inconnue $\mathtt{y}^{\mathtt{cible}}$, notée $\Delta\mathtt{y}$, puis, on infère $\mathtt{y}^{\mathtt{cible}}$ à partir de $\mathtt{y}^s$ et $\Delta\mathtt{y}$.

Modèle de connaissances du RàPC. On considère généralement qu'une base de connaissances d'un système de RàPC est constitué de quatre « conteneurs de connaissances » : la base de cas $\mathtt{BC}$, les connaissances du domaine $\mathtt{CD}$, les connaissances de remémoration $\mathtt{CR}$ et les connaissances d'adaptation $\mathtt{CA}$ [11]. $\mathtt{BC}$ et $\mathtt{CA}$ ont été évoqués ci-dessus. $\mathtt{CR}$ est souvent représenté par une mesure de similarité ou une distance entre problèmes. $\mathtt{CD}$ peut être vu comme une conjonction de contraintes d'intégrité qui donnent des conditions nécessaires pour qu'un cas soit licite (exemple en cuisine : « Les salsifis, c'est pas bon [1]. »). Les inférences dans le formalisme représentant les cas se feront en général sur la base de ces connaissances du domaine (on utilisera la relation $\models_{\mathtt{CD}}$ plutôt que simplement $\models$).

Représentation des cas en logique propositionnelle. Pour certaines applications, on peut représenter les cas dans $(\mathcal{LP}, \models)$: on considère un partitionnement de $\mathcal{V}$ en $\{\mathcal{V}_{\mathcal{P}}, \mathcal{V}_{\mathcal{S}}\}$ et un problème $\mathtt{x}$ (resp. une solution $\mathtt{y}$) sera représenté(e) par une formule dont les variables appartiennent à $\mathcal{V}_{\mathcal{P}}$ (resp. à $\mathcal{V}_{\mathcal{S}}$). Un cas source $(\mathtt{x}^s, \mathtt{y}^s)$ sera représenté par la conjonction $\mathtt{x}^s \wedge \mathtt{y}^s$ ([2]).

1. Cette connaissance du domaine, quelque peu subjective, nous a été fournie par une des sœurs d'un des auteurs.

2. Parfois, comme on le verra pour l'exemple suivi, le partitionnement entre variables problèmes et variables solutions se fait une fois le problème d'adaptation spécifié : connaissant $\mathtt{x}^{\mathtt{cible}}$ et le cas source $\mathtt{c}^s$, on détermine

Dans beaucoup d'applications, les cas sources et le problème cible sont considérés comme *spécifiques*, i.e. issus d'une expérience particulière complètement instanciée. Dans ce cas, pour un cas source $(\mathtt{x}^s, \mathtt{y}^s)$ et un problème cible $\mathtt{x}^{\mathtt{cible}}$, les problèmes $\mathtt{x}^s$ et $\mathtt{x}^{\mathtt{cible}}$ peuvent s'écrire sous la forme $\bigwedge_{a \in \mathcal{V}_\mathcal{P}} \ell(a)$ et la solution $\mathtt{y}^s$ sous la forme $\bigwedge_{a \in \mathcal{V}_S} \ell(a)$, où $\ell(a) \in \{a, \neg a\}$. La formule $\mathtt{x}^s \wedge \mathtt{y}^s$ est alors un cube complet et donc $|\mathcal{M}(\mathtt{x}^s \wedge \mathtt{y}^s)| = 1$.

L'apprentissage de connaissances d'adaptation. L'adaptation peut s'appuyer sur des connaissances d'adaptation. Ces connaissances peuvent être acquises auprès d'un expert (voir p. ex. [5]). Elles peuvent être aussi apprises à partir de la base de cas, selon le principe appelé *difference heuristics* dans [10] et défini initialement dans [9]. Ce principe s'inscrit dans le cadre de l'adaptation transformationnelle : étant donné deux cas sources différents $(\mathtt{x}^i, \mathtt{y}^i)$ et $(\mathtt{x}^j, \mathtt{y}^j)$, on construit les variations $\Delta\mathtt{x}^{ij}$ de $\mathtt{x}^i$ à $\mathtt{x}^j$ et $\Delta\mathtt{y}^{ij}$ de $\mathtt{y}^i$ à $\mathtt{y}^j$. L'apprentissage de connaissances d'adaptation utilise un ensemble de couples $(\Delta\mathtt{x}^{ij}, \Delta\mathtt{y}^{ij})$ et permet de construire un modèle d'une fonction $\Delta\mathtt{x} \mapsto \Delta\mathtt{y}$, qui permet, à partir d'une variation entre problèmes d'obtenir une variation entre solution, ce qui est utile pour un processus d'adaptation transformationnelle.

Des travaux dans ce cadre se sont appuyés sur une notation a^v, où a est un attribut d'un problème ou d'une solution et v représente une relation entre valeurs du domaine de cet attribut. Cette notation permet de coder ces variations, dans une optique d'apprentissage de règles d'adaptation, utilisant des techniques telles que l'extraction de motifs fermés fréquents [4]. Par exemple, $\mathtt{âge}^{\mathtt{ajouter(5)}}$ indique une variation de 5 pour l'attribut $\mathtt{âge}$. Le résultat d'une extraction de motifs fermés fréquents avec des $\Delta\mathtt{x}^{ij}$ et des $\Delta\mathtt{y}^{ij}$ est un ensemble de motifs, chaque motif étant un ensemble d'expressions a^v. En se limitant à des attributs booléens, les a^v correspondaient dans ce travail à ceux de la section 2.1 : c'est ce travail qui a d'ailleurs motivé au départ l'étude de la logique $(\Delta\mathcal{LP}, \models)$.

Plus précisément, si les cas sources sont supposés spécifiques, d'après le résultat (1), on peut écrire $\Delta\mathtt{x}^{ij}$ et $\Delta\mathtt{y}^{ij}$ sous les formes $\bigwedge_{a \in \mathcal{V}_\mathcal{P}} a^{v(a)}$ et $\bigwedge_{a \in \mathcal{V}_S} a^{v(a)}$, où $v(a) \in \mathcal{D}_0$. Un algorithme de recherche de motifs fréquents permettra alors de calculer des motifs M, ensemble d'éléments de la forme a^v, motifs qui peuvent être interprétés en règles d'adaptation.

3 Nouveaux résultats sur $(\Delta\mathcal{LP}, \models)$

Cette section introduit et étudie des notions relatives à la logique $(\Delta\mathcal{LP}, \models)$ utiles pour la suite de l'article et qui étaient peu ou pas introduits dans [7].

$\mathcal{V}_\mathcal{P}$ et $\mathcal{V}_S$ pour décomposer $\mathtt{c}^s$ en $\mathtt{x}^s \wedge \mathtt{y}^s$.

3.1 Résultats généraux

Le premier résultat énonce le fait que seules les variables apparaissant dans une formule de $(\Delta\mathcal{LP}, \models)$ ont une influence dans les inférences déductives (ce qui peut sembler une évidence mais mérite peut-être d'être énoncé) :

$$\begin{aligned} &\text{si } \mathcal{I}_1\mathcal{J}_1, \mathcal{I}_2\mathcal{J}_2 \in \Delta\Omega \text{ vérifient} \\ &\quad \mathcal{I}_1(x) = \mathcal{I}_2(x) \text{ et } \mathcal{J}_1(x) = \mathcal{J}_2(x) \quad \text{pour } x \in \mathcal{V}(\varphi) \\ &\text{alors } \mathcal{I}_1\mathcal{J}_1 \models \varphi \text{ ssi } \mathcal{I}_2\mathcal{J}_2 \models \varphi \end{aligned} \tag{3}$$

pour $\varphi \in \Delta\mathcal{LP}$. La preuve de ce résultat se fait par induction structurelle.

Le résultat (2) ne se généralise pas en remplaçant $\vee$ par $\wedge$. Cependant, on a le résultat suivant :

$$\begin{aligned} &\text{si } \varphi, \psi \in \Delta\mathcal{LP} \text{ vérifient } \mathcal{V}(\varphi) \cap \mathcal{V}(\psi) = \emptyset \\ &\text{alors } \mathsf{G}(\varphi \wedge \psi) \equiv \mathsf{G}(\varphi) \wedge \mathsf{G}(\psi) \\ &\text{et } \mathsf{D}(\varphi \wedge \psi) \equiv \mathsf{D}(\varphi) \wedge \mathsf{D}(\psi) \end{aligned} \tag{4}$$

Preuve. La preuve ne sera faite que pour l'opérateur D (elle se transpose immédiatement pour l'opérateur G). Soit $\mathcal{J} \in \mathcal{M}(\mathsf{D}(\varphi \wedge \psi))$. Il existe donc $\mathcal{I} \in \Omega$ telle que $\mathcal{I}\mathcal{J} \models \varphi \wedge \psi$. Donc, il existe $\mathcal{I} \in \Omega$ telle que $\mathcal{I}\mathcal{J} \models \varphi$. Donc $\mathcal{J} \models \mathsf{D}(\varphi)$. Le même raisonnement conduit à $\mathcal{J} \models \mathsf{D}(\psi)$. Donc $\mathcal{J} \models \mathsf{D}(\varphi) \wedge \mathsf{D}(\psi)$. Par conséquent, $\mathsf{D}(\varphi \wedge \psi) \models \mathsf{D}(\varphi) \wedge \mathsf{D}(\psi)$.

Pour ce sens direct, l'hypothèse $\mathcal{V}(\varphi) \cap \mathcal{V}(\psi) = \emptyset$ n'était pas utile. Elle l'est pour la réciproque.

Supposons que $\mathcal{J} \models \mathsf{D}(\varphi) \wedge \mathsf{D}(\psi)$. Par conséquent, il existe $\mathcal{I}_1 \in \Omega$ telle que $\mathcal{I}_1\mathcal{J} \models \varphi$. De même, il existe $\mathcal{I}_2 \in \Omega$ telle que $\mathcal{I}_2\mathcal{J} \models \psi$. Comme $\mathcal{V}(\varphi) \cap \mathcal{V}(\psi) = \emptyset$ on peut définir $\mathcal{I} \in \Omega$ par

$$\mathcal{I}(x) = \begin{cases} \mathcal{I}_1(x) & \text{si } x \in \mathcal{V}(\varphi) \\ \mathcal{I}_2(x) & \text{si } x \in \mathcal{V}(\psi) \\ 1 & \text{sinon} \end{cases} \qquad (\text{pour tout } x \in \mathcal{V})$$

D'après le résultat (3), $\mathcal{I}_1\mathcal{J} \models \varphi$ entraîne $\mathcal{I}\mathcal{J} \models \varphi$ (puisque pour $x \in \mathcal{V}(\varphi)$, $\mathcal{I}_1(x) = \mathcal{I}(x)$ et, évidemment, $\mathcal{J}(x) = \mathcal{J}(x)$). De la même façon, on prouve que $\mathcal{I}\mathcal{J} \models \psi$. Donc, $\mathcal{I}\mathcal{J} \models \varphi \wedge \psi$ et, par conséquent, $\mathcal{J} \models \mathsf{D}(\varphi \wedge \psi)$. On en déduit que $\mathsf{D}(\varphi) \wedge \mathsf{D}(\psi) \models \mathsf{D}(\varphi \wedge \psi)$ ce qui conclut la preuve. ■

3.2 Caractérisation alternative de $\mathcal{B} \models \varphi$

La relation $\models$ entre une base de connaissances $\mathcal{B}$ et une formule φ de la logique des variations propositionnelles peut être redéfinie en s'appuyant sur quatre « valeurs de vérité » sur les variations, données par les éléments de $\mathcal{D}_0 = \{\mathtt{=1}, \mathtt{=0}, \mathtt{+}, \mathtt{-}\}$.

Soit $\Delta\Omega_{\text{alt}}$ l'ensemble des fonctions $\omega : \mathcal{V} \to \mathcal{D}_0$. On définit la relation $\models_{\text{alt}}$ entre $\omega \in \Delta\Omega_{\text{alt}}$ et $\varphi \in \Delta\mathcal{LP}$ par :

$$\omega \models_{\text{alt}} \varphi \text{ si } \bigwedge_{a \in \mathcal{V}} a^{\omega(a)} \models \varphi$$

On note alors $\mathcal{M}_{\text{alt}}(\varphi) = \{\omega \in \Delta\Omega_{\text{alt}} \mid \omega \models_{\text{alt}} \varphi\}$. Par exemple, si $\mathcal{V} = \{a, b\}$, $\mathcal{M}_{\text{alt}}(a^{=} \wedge b^{+}) = \{\omega_1, \omega_2\}$ où $\omega_1(a) = \mathtt{=1}$, $\omega_2(a) = \mathtt{=0}$ et $\omega_1(b) = \omega_2(b) = \mathtt{+}$. Pour une base de connaissances $\mathcal{B}$, on notera $\mathcal{M}_{\text{alt}}(\mathcal{B}) = \bigcap_{\varphi\in\mathcal{B}} \mathcal{M}_{\text{alt}}(\varphi)$. On définit alors la relation $\models_{\text{alt}}$ entre une base de connaissances $\mathcal{B}$ et une formule φ par $\mathcal{B} \models_{\text{alt}} \varphi$ si $\mathcal{M}_{\text{alt}}(\mathcal{B}) \subseteq \mathcal{M}_{\text{alt}}(\varphi)$.

Cette relation $\models_{\text{alt}}$ ne définit pas une nouvelle sémantique de la logique des variations comme le montre la proposition ci-dessous ; elle permet juste une caractérisation alternative de cette sémantique.

Proposition 1. *Pour toute base de connaissances $\mathcal{B}$ de $(\Delta\mathcal{LP}, \models)$ et toute $\varphi \in \Delta\mathcal{LP}$, $\mathcal{B} \models_{alt} \varphi$ ssi $\mathcal{B} \models \varphi$.*

Preuve. À $\mathcal{I}\mathcal{J} \in \Delta\Omega$, on associe $\omega_{\mathcal{I}\mathcal{J}} \in \Delta\Omega_{\text{alt}}$ de la façon suivante, pour $a \in \mathcal{V}$

$$\omega_{\mathcal{I}\mathcal{J}}(a) = \begin{cases} \mathtt{=1} & \text{si } \mathcal{I}(a) = \mathcal{J}(a) = 1 \\ \mathtt{=0} & \text{si } \mathcal{I}(a) = \mathcal{J}(a) = \mathtt{0} \\ \mathtt{+} & \text{si } \mathcal{I}(a) = \mathtt{0} \text{ et } \mathcal{J}(a) = 1 \\ \mathtt{-} & \text{si } \mathcal{I}(a) = 1 \text{ et } \mathcal{J}(a) = \mathtt{0} \end{cases}$$

La fonction $\mathcal{I}\mathcal{J} \in \Delta\Omega \mapsto \omega_{\mathcal{I}\mathcal{J}} \in \Delta\Omega_{\text{alt}}$ est une bijection dont la fonction inverse est définie ainsi : à $\omega \in \Delta\Omega_{\text{alt}}$ elle associe $\mathcal{I}\mathcal{J} \in \Delta\Omega$ définie ainsi, pour $a \in \mathcal{V}$:

$$\mathcal{I}(a) = \begin{cases} \mathtt{0} & \text{si } \omega(a) \in \{\mathtt{=0}, \mathtt{+}\} \\ 1 & \text{sinon} \end{cases}$$

$$\mathcal{J}(a) = \begin{cases} \mathtt{0} & \text{si } \omega(a) \in \{\mathtt{=0}, \mathtt{-}\} \\ 1 & \text{sinon} \end{cases}$$

(on montre que c'est une bijection en utilisant le fait que $|\Delta\Omega_{\text{alt}}| = |\Delta\Omega| = 4^{|\mathcal{V}|}$ et en vérifiant que, pour $\omega \in \Delta\Omega_{\text{alt}}$, en y associant $\mathcal{I}\mathcal{J}$ comme ci-dessus, on a $\omega_{\mathcal{I}\mathcal{J}} = \omega$).

On montre ensuite que pour $\mathcal{I}\mathcal{J} \in \Delta\Omega$ et $\varphi \in \Delta\mathcal{LP}$ on a $\mathcal{I}\mathcal{J} \models \varphi$ ssi $\omega_{\mathcal{I}\mathcal{J}} \models_{\text{alt}} \varphi$.

Enfin, pour une base de connaissances $\mathcal{B}$ et $\varphi \in \Delta\mathcal{LP}$, on montre que $\mathcal{B} \models \varphi$ ssi $\mathcal{B} \models_{\text{alt}} \varphi$ en s'appuyant sur la bijection introduite ci-dessus et l'équivalence entre satisfaction dans $\Delta\Omega$ et dans $\Delta\Omega_{\text{alt}}$. ■

3.3 Variables invariantes d'une formule de variations

Soit $\varphi \in \Delta\mathcal{LP}$, on s'intéresse (et ce sera justifié par la suite) aux variables a dont le changement laisse φ inchangée. On va définir $\mathsf{Inv}(\varphi)$ en s'appuyant sur la définition alternative de la sémantique présentée à la section précédente. L'idée est que $a \in \mathsf{Inv}(\varphi)$ signifie que la valeur d'$\omega(a)$ n'influe pas sur la satisfaction de φ par ω. Formellement, cela mène à la définition suivante :

Définition 1. *Pour $\omega \in \Delta\Omega_{alt}$, $a \in \mathcal{V}$ et $v \in \mathcal{D}_0$, on définit $\omega[a^v] \in \Delta\Omega_{alt}$ par*

$$\omega[a^v] : x \in \mathcal{V} \mapsto \begin{cases} v & \text{si } x = a \\ \omega(x) & \text{sinon} \end{cases}$$

Pour $\varphi \in \Delta\mathcal{LP}$, l'ensemble des variables invariantes de φ est

$$\mathsf{Inv}(\varphi) = \left\{ a \in \mathcal{V} \;\middle|\; \begin{array}{l} \text{pour toute } \omega \in \mathcal{M}_{alt}(\varphi) \\ \text{et tout } v \in \mathcal{D}_0,\ \omega[a^v] \models_{alt} \varphi \end{array} \right\}$$

Par exemple,

$$\begin{aligned} &\text{si } \mathcal{V} = \{a, b, c\} \text{ et } \varphi = (a^{+} \wedge b^{=}) \vee (a^{+} \wedge b^{+}) \vee (a^{+} \wedge b^{-}) \\ &\text{alors } \mathsf{Inv}(\varphi) = \{b, c\} \end{aligned} \tag{5}$$

Cette notion est indépendante de la syntaxe : si $\varphi \equiv \psi$ alors $\mathsf{Inv}(\varphi) = \mathsf{Inv}(\psi)$. On montre, en s'appuyant sur (3) que si une variable x n'apparaît pas dans φ alors elle est invariante. Donc :

$$\mathcal{V} \setminus \mathcal{V}(\varphi) \subseteq \mathsf{Inv}(\varphi) \subseteq \mathcal{V} \tag{6}$$

Comme le montre l'exemple (5), une variable apparaissant dans φ peut très bien être invariante pour φ.

Le cas particulier de la proposition suivante est utile dans la suite de l'article.

Proposition 2. *Si φ est un cube de $(\Delta\mathcal{LP}, \models)$ alors $\mathsf{Inv}(\varphi) = \mathcal{V} \setminus \mathcal{V}(\varphi)$.*

Preuve. Comme $\mathsf{Inv}(\varphi) \supseteq \mathcal{V} \setminus \mathcal{V}(\varphi)$ (cf. (6)), il suffit de montrer que si $a \in \mathsf{Inv}(\varphi)$ alors $a \notin \mathcal{V}(\varphi)$, ce qu'on va montrer par contraposée. Soit $a \in \mathcal{V}(\varphi)$. Soit $v \in \mathcal{D}$ tel que a^v est l'atome de φ ayant a comme variable. Soit $v_0 \in \mathcal{D}_0$ tel que $a^{v_0} \models a^v$ (si v est un symbole de variation primitif, $v_0 = v$, sinon, il y a plusieurs choix possibles pour v_0, p. ex., si $v = \mathtt{\neq}$ alors on peut choisir $v_0 = \mathtt{+}$ ou $v_0 = \mathtt{-}$). Soit $w \in \mathcal{D}_0 \setminus \{v_0\}$. On a donc $\varphi[a^{v_0}] \models a^{v_0}$ et $\varphi[a^w] \not\models a^{v_0}$ (puisque, par hypothèse sur les cubes, une variable n'apparaît qu'une seule fois dans un cube et donc, les cubes sont satisfiables). Par conséquent, $\varphi[a^{v_0}] \not\equiv \varphi[a^w]$ et donc $a \notin \mathsf{Inv}(\varphi)$. ■

3.4 Opérateur *ceteris paribus*

Considérons la formule a^{+} : elle indique que a « passe » de $\mathtt{0}$ à 1, les autres variables étant libres de suivre toutes les variations. Ainsi, si $\mathcal{V} = \{a, b\}$, $\mathcal{M}(a^{+}) = \{(\overline{a}\overline{b}, a\overline{b}), (\overline{a}\overline{b}, ab), (\overline{a}b, a\overline{b}), (\overline{a}b, ab)\}$. Il apparaît utile dans certains cas d'exprimer par exemple que a « passe » de $\mathtt{0}$ à 1, le reste des variables restant *inchangées*. On va exprimer cela par une formule $\mathsf{cp}(a^{+})$ qui, sur l'exemple avec deux variables, donnera $\mathcal{M}(\mathsf{cp}(a^{+})) = \{(\overline{a}\overline{b}, a\overline{b}), (\overline{a}b, ab)\}$: dans les modèles de ces formules, b reste à $\mathtt{0}$ ou reste à 1 (cp pour *ceteris paribus*), en d'autres termes, $\mathsf{cp}(a^{+}) \models b^{=}$. Dans cet exemple, b est une variable dont l'interprétation ne change pas l'interprétation de φ : $b \in \mathsf{Inv}(\varphi)$.

Plus généralement, on cherche un opérateur cp tel que, pour $\varphi, \psi \in \Delta\mathcal{LP}$, on ait

$$\mathsf{cp}(\varphi) \models \varphi \tag{7}$$
$$\text{pour toute } x \in \mathsf{Inv}(\varphi),\ \mathsf{cp}(\varphi) \models x^{=} \tag{8}$$
$$\text{si } \varphi \equiv \psi \text{ alors } \mathsf{cp}(\varphi) \equiv \mathsf{cp}(\psi) \tag{9}$$
$$\text{si } \varphi \text{ est satisfiable alors } \mathsf{cp}(\varphi) \text{ l'est aussi} \tag{10}$$

On propose la définition suivante de cp et on prouve ensuite qu'elle vérifie ces propriétés.

Définition 2. cp *est la fonction de* $\Delta\mathcal{LP}$ *dans lui-même définie, pour* $\varphi \in \Delta\mathcal{LP}$ *par*

$$\mathsf{cp}(\varphi) = \varphi \wedge \bigwedge \{x^{=} \mid x \in \mathsf{Inv}(\varphi)\}$$

Proposition 3. cp *vérifie (7), (8), (9) et (10).*

Preuve. (7), (8) et (9) découlent immédiatement de la définition.

Pour (10), on le prouve de la façon suivante. Supposons que φ soit satisfiable. Elle l'est donc aussi au sens de la définition alternative de la sémantique et donc, il existe $\omega \in \Delta\Omega_{\text{alt}}$ telle que $\omega \models_{\text{alt}} \varphi$. Soit a_1, a_2, ..., a_p telles que $\mathsf{Inv}(\varphi) = \{a_1, a_2, \ldots, a_p\}$. Soit $\omega' = \omega[a_1^{=1}][a_2^{=1}]\ldots[a_p^{=1}]$. On a, par définition de $\mathsf{Inv}(\varphi)$, $\omega' \models_{\text{alt}} \varphi$. Par ailleurs, $\omega' \models_{\text{alt}} a_k^{=1}$ pour tout $k \in \{1, 2, \ldots, p\}$ et donc $\omega' \models_{\text{alt}} \bigwedge\{x^{=1} \mid x \in \mathsf{Inv}(\varphi)\}$. Or, pour toute variable a, $a^{=1} \models a^{=}$. Par conséquent, $\omega' \models_{\text{alt}} \bigwedge\{x^{=} \mid x \in \mathsf{Inv}(\varphi)\}$. Donc, $\omega' \models_{\text{alt}} \mathsf{cp}(\varphi)$ et donc, $\mathsf{cp}(\varphi)$ est satisfiable. ■

Une conséquence immédiate de la proposition 2 est la suivante :

$$\text{si } \varphi \text{ est un cube alors } \mathsf{cp}(\varphi) = \varphi \wedge \bigwedge_{x \in \mathcal{V} \setminus \mathcal{V}(\varphi)} x^{=} \tag{11}$$

Par ailleurs, les modèles de $\mathsf{cp}(\top)$ sont les $\mathcal{I}\mathcal{I}$ pour $\mathcal{I} \in \Omega$.

Notons que cp est non monotone, dans le sens où on peut avoir $\varphi \models \psi$ et $\mathsf{cp}(\varphi) \not\models \mathsf{cp}(\psi)$. Par exemple, $a^{+} \models \top$ et $\mathsf{cp}(a^{+}) \not\models \mathsf{cp}(\top)$.

Un problème pratique se pose avec cet opérateur : pour $\varphi \in \Delta\mathcal{LP}$, $|\mathsf{cp}(\varphi)|$ est en $O(|\varphi| + n)$ et la manipulation de formules longues prend du temps de calcul. Ce problème devient théorique si on étend le cadre logique à un ensemble de variables infini : la définition de cp ci-dessus ne s'applique pas, les formules étant nécessairement de tailles finies. Une perspective pour pallier ce problème serait l'étude d'une possible extension de la logique par un *connecteur* cp (plutôt que par un opérateur).

Il apparaît parfois utile de restreindre le *ceteris paribus* à un ensemble prédéfini de variables, d'où la définition suivante.

Définition 3. *Soit* $\mathcal{W} \subseteq \mathcal{V}$ *et* $\varphi \in \Delta\mathcal{LP}$. *Le* ceteris paribus *de* φ *restreint à l'ensemble de variables* $\mathcal{W}$ *est :*

$$\mathsf{cp}_{\mathcal{W}}(\varphi) = \varphi \wedge \bigwedge \{x^{=} \mid x \in \mathsf{Inv}(\varphi) \cap \mathcal{W}\}$$

Et on a, évidemment, $\mathsf{cp}_{\mathcal{V}}(\varphi) = \mathsf{cp}(\varphi)$.

3.5 Composition des variations

La composition sur $(\Delta\mathcal{LP}, \models)$ a été introduite dans [7] mais peu étudiée. Comme une formule $\varphi \in \Delta\mathcal{LP}$ représente un sous-ensemble $\mathcal{M}(\varphi)$ de $\Delta\Omega = \Omega \times \Omega$, on peut la considérer comme une représentation d'une relation binaire sur Ω. Les relations binaires sur un ensemble peuvent se composer et la composition des formules de variations correspond à la composition entre relations binaires.

Plus formellement, soit $\varphi, \psi \in \Delta\mathcal{LP}$. La composition de φ et ψ est une formule $\varphi \ ;\ \psi$ définie à la syntaxe près par

$$\mathcal{M}(\varphi \ ;\ \psi) = \left\{ \mathcal{I}\mathcal{K} \in \Delta\Omega \,\middle|\, \begin{array}{l} \text{il existe } \mathcal{J} \in \Omega \text{ telle que} \\ \mathcal{I}\mathcal{J} \models \varphi \text{ et } \mathcal{J}\mathcal{K} \models \psi \end{array} \right\}$$

Le résultat suivant est une conséquence directe de l'associativité de la composition de relations binaires :

$$\text{pour } \varphi, \psi, \chi \in \Delta\mathcal{LP},\ (\varphi \ ;\ \psi) \ ;\ \chi \equiv \varphi \ ;\ (\psi \ ;\ \chi)$$

Cela permet d'omettre des parenthèses dans les compositions.

La recherche d'une formule équivalente à $\varphi \ ;\ \psi$ va être étudiée via les propositions ci-dessous.

On commence par le cas particulier où les formules à composer sont de la forme $\alpha \triangleright \beta$.

Proposition 4. *Soit* $\alpha, \beta, \gamma, \delta \in \mathcal{LP}$. *On a :*

$$(\alpha \triangleright \beta) \ ;\ (\gamma \triangleright \delta) \equiv \begin{cases} \bot & \textit{si } \beta \wedge \gamma \models \bot \\ \alpha \triangleright \delta & \textit{sinon} \end{cases}$$

On s'intéresse ensuite aux liens entre composition et disjonction :

Proposition 5. *Soit* $\varphi, \varphi_1, \varphi_2, \psi, \psi_1, \psi_2 \in \Delta\mathcal{LP}$. *On a les équivalences suivantes :*

$$\varphi \ ;\ (\psi_1 \vee \psi_2) \equiv (\varphi \ ;\ \psi_1) \vee (\varphi \ ;\ \psi_2) \tag{12}$$
$$(\varphi_1 \vee \varphi_2) \ ;\ \psi \equiv (\varphi_1 \ ;\ \psi) \vee (\varphi_2 \ ;\ \psi) \tag{13}$$

Définition 4. *On introduit deux symboles de variations supplémentaires* $\bot_{\mathcal{D}}$ *et* $\top_{\mathcal{D}}$ *tels que, pour tout* $\alpha \in \mathcal{LP}$, $\alpha^{\bot_{\mathcal{D}}} = \bot \triangleright \bot = \bot_{\Delta}$ *et* $\alpha^{\top_{\mathcal{D}}} = \top \triangleright \top = \top_{\Delta}$.

À un cube φ *de* $(\Delta\mathcal{LP}, \models)$ *on associe* $\mathtt{var}(x, \varphi) \in \mathcal{D} \cup \{\top_{\mathcal{D}}\}$ *de la façon suivante :*

$$\mathtt{var}(x, \varphi) = \begin{cases} \top_{\mathcal{D}} & \textit{si } x \notin \mathcal{V}(\varphi) \\ v & \textit{si } x^{v} \textit{ est un des termes de } \varphi \end{cases}$$

La proposition suivante permet de calculer la composition de deux atomes de $(\Delta\mathcal{LP}, \models)$ construits sur la même variable.

Proposition 6. *Il existe une opération binaire* ; *sur* $\mathcal{D} \cup \{\bot_{\mathcal{D}}, \top_{\mathcal{D}}\}$ *telle que pour une variable* a, *on a* $a^{v} \ ;\ a^{w} \equiv a^{v;w}$.

Principe de la preuve. À chaque symbole de variation v on associe A^v, une matrice 2×2 de booléens telle que les lignes et les colonnes sont indexées par $(i, j) \in \{\mathbb{0}, 1\}^2$ et tel que $A^v = [A^v_{ij}]_{ij}$ est défini comme suit (où a est une variable arbitraire et $\mathcal{I}\mathcal{J} \in \Delta\Omega$) :

$$A^v_{\mathcal{I}(a)\mathcal{J}(a)} = \begin{cases} 1 & \text{si } \mathcal{I}\mathcal{J} \models a^v \\ \mathbb{0} & \text{sinon} \end{cases}$$

Ainsi, $A^{\perp_\mathcal{D}} = \begin{pmatrix} \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} \end{pmatrix}$, $A^{+} = \begin{pmatrix} \mathbb{0} & 1 \\ \mathbb{0} & \mathbb{0} \end{pmatrix}$ et $A^{\neq} = \begin{pmatrix} \mathbb{0} & 1 \\ 1 & \mathbb{0} \end{pmatrix}$.

On montre ensuite que, pour $v, w \in \mathcal{D} \cup \{\perp_\mathcal{D}, \top_\mathcal{D}\}$, le produit des matrices $A^v \times A^w$ (où la somme et le produit correspondent aux `ou` et au `et` sur les booléens) donne une matrice A^u où $u \in \mathcal{D} \cup \{\perp_\mathcal{D}, \top_\mathcal{D}\}$ (et on notera $u = v \ ; \ w$).

Il reste à montrer que le produit des matrices correspond bien à une composition au sens où, pour $a \in \mathcal{V}$, $a^v \ ; \ a^w = a^{v;w}$. Pour ce faire, un programme Python a été développé pour automatiser le calcul d'une table de composition. ■

Cela permet alors de calculer la composition de cubes :

Proposition 7. *Soit φ et ψ, deux cubes de $(\Delta\mathcal{LP}, \models)$. On a :*

$$\varphi \ ; \ \psi \equiv \bigwedge_{a \in \mathcal{V}} a^{\mathtt{var}(a,\varphi);\mathtt{var}(a,\psi)} \tag{14}$$

Pour composer deux cubes, on applique la proposition 7 et on simplifie en se rappelant que $x^{\top_\mathcal{D}} \equiv \top$ et $x^{\perp_\mathcal{D}} \equiv \perp$. Par exemple, si l'ensemble des variables est $\{a, b, c, d, e\}$ alors

$$\begin{aligned} & a^{+} \wedge b^{=} \wedge d^{=\mathbb{0}} \ ; \ a^{-} \wedge c^{-} \wedge d^{+} \\ & \equiv a^{=\mathbb{0}} \wedge b^{\top_\mathcal{D}} \wedge c^{\bullet\mathbb{0}} \wedge d^{+} \wedge e^{\top_\mathcal{D}} \\ & \equiv a^{=\mathbb{0}} \wedge c^{\bullet\mathbb{0}} \wedge d^{+} \end{aligned}$$

(en s'appuyant sur une table de composition sur $\mathcal{D}$ disponible dans [8]).

De façon générale, une façon d'avoir une expression syntaxique pour $\varphi \ ; \ \psi$ consiste à mettre ces deux formules de $(\Delta\mathcal{LP}, \models)$ sous FND puis à appliquer les équivalences de la proposition 5 tant que c'est possible pour enfin appliquer la proposition 7 et se débarrasser complètement du symbole $;$.

4 Représenter des règles d'adaptation

Cette section vise de façon générale à formaliser dans $(\Delta\mathcal{LP}, \models)$ ce qu'est une règle d'adaptation et comment raisonner avec.

4.1 Introduction d'un exemple

On considère un exemple dans le domaine culinaire : un problème représente une requête de recette et une solution, une recette. On suppose que les problèmes et les solutions s'expriment en logique propositionnelle. Dans cet exemple, on suppose que l'ensemble des connaissances du domaine est vide : $\mathtt{CD} = \top$. Le problème d'adaptation est $(\mathtt{c}^s, \mathtt{x}^{\mathtt{cible}})$ où

- $\mathtt{c}^s$ est le cas source correspondant à une recette de salade, avec de la batavia, du magret de canard, des tomates, de l'huile d'olive et du vinaigre :

  $$\begin{aligned} \mathtt{c}^s = {} & \mathtt{rSalade} \wedge \mathtt{iBatavia} \wedge \mathtt{iMagret} \wedge \mathtt{iTomate} \\ & \wedge \mathtt{iHuileDOlive} \wedge \mathtt{iVinaigre} \wedge \textit{rien d'autre} \end{aligned}$$

  où $\mathtt{rSalade}$ se lit « recette de salade », $\mathtt{iMachin}$, « recette avec l'ingrédient machin » et *rien d'autre* est une notation pour la conjonction des littéraux négatifs $\neg a$ tels que a ne peut être déduit du début de la formule : $\alpha \wedge \textit{rien d'autre} \equiv \alpha \wedge \bigwedge\{\neg a \mid a \in \mathcal{V}$ et $\alpha \not\models_{\mathtt{CD}} a\}$. En particulier, $\mathtt{c}^s \models_{\mathtt{CD}} \neg\mathtt{iAil}$. Notons que $\mathtt{c}^s$ est un cube complet.
- $\mathtt{x}^{\mathtt{cible}}$ est le problème cible « Je veux une recette de salade avec des tomates, du jus de citron, mais pas d'ail. » :

  $$\mathtt{x}^{\mathtt{cible}} = \mathtt{rSalade} \wedge \mathtt{iTomate} \wedge \mathtt{iJusCitron} \wedge \neg\mathtt{iAil}$$

La connaissance du problème d'adaptation $(\mathtt{c}^s, \mathtt{x}^{\mathtt{cible}})$ permet de partitionner $\mathcal{V}$ en $\{\mathcal{V}_\mathcal{P}, \mathcal{V}_\mathcal{S}\}$: $\mathcal{V}_\mathcal{P} = \{\mathtt{rSalade}, \mathtt{iTomate}, \mathtt{iJusCitron}, \mathtt{iAil}\}$ (variables de $\mathtt{x}^{\mathtt{cible}}$) et $\mathcal{V}_\mathcal{S} = \mathcal{V} \setminus \mathcal{V}_\mathcal{P}$. Par conséquent, $\mathtt{c}^s$ s'écrit $\mathtt{c}^s = \mathtt{x}^s \wedge \mathtt{y}^s$ avec

$$\begin{aligned} \mathtt{x}^s &= \mathtt{rSalade} \wedge \mathtt{iTomate} \wedge \neg\mathtt{iJusCitron} \wedge \neg\mathtt{iAil} \\ \mathtt{y}^s &= \mathtt{iBatavia} \wedge \mathtt{iMagret} \wedge \mathtt{iHuileDOlive} \\ & \quad \wedge \mathtt{iVinaigre} \wedge \ldots \end{aligned}$$

(les points de suspension correspondant à la conjonction des littéraux $\neg a$ où $a \in \mathcal{V}_\mathcal{S}$ et a n'est pas un littéral positif de $\mathtt{y}^s$).

On considère également un motif M obtenu par un système d'extraction de motifs (p. ex., de motifs fermés fréquents) pour l'acquisition de connaissances d'adaptation. Un tel motif contient un ensemble fini d'atomes a^v indiquant des variations « simultanées » entre cas et s'interprète par la conjonction φ de ces éléments. φ peut s'écrire comme un cube de $(\Delta\mathcal{LP}, \models)$: si plusieurs atomes a^{v_1}, a^{v_2}, ... construits sur la même variable a apparaissent dans φ, on peut montrer qu'ils peuvent être remplacés par un seul a^v. Dans cet exemple, on va considérer la formule suivante :

$$\varphi = \mathtt{rSalade}^{=1} \wedge \mathtt{iVinaigre}^{-} \wedge \mathtt{iJusCitron}^{+} \wedge \mathtt{iSel}^{+}$$

Le motif M dont est issu φ traduit le fait qu'on a observé un nombre suffisant de couples de recettes de salades avec du vinaigre dans la première mais pas dans la seconde, et du jus de citron et du sel dans la seconde mais pas dans la première. Par conséquent, si on considère que φ

est une règle d'adaptation, alors, on va considérer l'applicabilité et l'application de φ sur un problème d'adaptation $((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^{\mathtt{cible}})$.

4.2 Représenter une règle d'adaptation

Une règle d'adaptation est définie comme étant un couple $R = (\varphi, \mathsf{C})$ où $\varphi \in \Delta\mathcal{LP}$ et C est un réel positif appelé *coût* de R.

Que signifie que R est *applicable* sur un problème d'adaptation $((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^{\mathtt{cible}})$ et comment définir le résultat de cette application ? Pour répondre à cette question, on va s'appuyer sur une fonction qui à $(\alpha, \varphi, \gamma) \in \mathcal{LP} \times \Delta\mathcal{LP} \times \mathcal{LP}$ associe $\mathsf{appl}(\alpha, \varphi, \gamma) \in \mathcal{LP}$, l'application sur α de φ sous contrainte γ, fonction qui va être définie et étudiée dans la section 4.3. Comme les cas sources et le problème cible doivent être considérés avec les connaissances du domaine $\mathtt{CD}$, on calculera la formule suivante :

$$\beta = \mathsf{appl}(\mathtt{CD} \wedge \mathtt{x}^s \wedge \mathtt{y}^s, \varphi, \mathtt{CD} \wedge \mathtt{x}^{\mathtt{cible}}) \tag{15}$$

Si $\beta \models \bot$ alors on dira que R n'est pas applicable sur le problème d'adaptation. Sinon, ce résultat s'écrira $\beta \equiv \mathtt{CD} \wedge \mathtt{x}^{\mathtt{cible}} \wedge \mathtt{y}^{\mathtt{cible}}$ où $\mathtt{y}^{\mathtt{cible}}$ sera la solution proposée pour $\mathtt{x}^{\mathtt{cible}}$ par application de R sur le problème d'adaptation.

Le RàPC est généralement hypothétique et l'adaptation ne constitue pas nécessairement une solution correcte au problème cible. Ainsi, une règle d'adaptation $R = (\varphi, \mathsf{C})$ peut produire une telle solution incorrecte. Pour choisir entre plusieurs règles d'adaptation, on utilise C : le coût associé à cette règle est indicatif de la qualité de R. Plus C est élevé, moins la confiance en l'application de la règle sera grande ; on pourrait, par exemple, associer C a une mesure de la probabilité P que la règle donnera un résultat correct par $\mathsf{C} = -\log P$ (sous des hypothèses de distribution données).

4.3 Appliquer une formule des variations sur une formule propositionnelle

Cette section vise à définir une notion d'application d'une formule $\varphi \in \Delta\mathcal{LP}$ sur une formule $\alpha \in \mathcal{LP}$, en présence d'une contrainte γ sur le résultat.

On va noter $\mathsf{appl}(\alpha, \varphi, \gamma)$ l'application sur α de φ sous contrainte γ. Pour arriver à sa définition, on s'intéressera d'abord à la transformation de α par φ sous contrainte γ, une formule $\mathsf{transf}(\alpha, \varphi, \gamma) \in \Delta\mathcal{LP}$ dont le résultat (la partie droite) sera $\mathsf{appl}(\alpha, \varphi, \gamma) = \mathsf{D}(\mathsf{transf}(\alpha, \varphi, \gamma))$.

On propose les trois conditions suivantes sur $\mathcal{I}\mathcal{J} \in \Delta\Omega$ pour que $\mathcal{I}\mathcal{J} \models \mathsf{transf}(\alpha, \varphi, \gamma)$: $\mathcal{I} \models \alpha$, $\mathcal{I}\mathcal{J} \models \varphi$ et $\mathcal{J} \models \gamma$. Ces trois conditions peuvent s'écrire en une seule :

$$\mathcal{I}\mathcal{J} \models (\alpha \triangleright \gamma) \wedge \varphi \tag{16}$$

Cette condition est-elle suffisante ? Considérons l'exemple suivant : $\alpha = a \wedge \neg b \wedge c$, $\varphi = a^- \wedge b^+$ et $\gamma = \top$. Si (16) était une condition nécessaire et suffisante pour que $\mathcal{I}\mathcal{J} \models \mathsf{transf}(\alpha, \varphi, \gamma)$ alors $\mathsf{appl}(\alpha, \varphi, \gamma)$ serait équivalente à $\mathsf{D}(\varphi)$ donc à $\neg a \wedge b$. Donc, que $\alpha \models c$ ou que $\alpha \models \neg c$ ne changerait pas l'application de φ sur α. Nous supposerons au contraire que si une variable (dans cet exemple c) est invariante dans φ, sa valeur doit être inchangée par la transformation, d'où l'utilisation du *ceteris paribus* (et la justification *a posteriori* de son étude) : on supposera donc que $\mathcal{I}\mathcal{J} \models \mathsf{cp}\,(\varphi)$. L'idée est que la formule φ représente de façon compacte une transformation et que quand elle ne dit rien sur une variable x, celle-ci est inchangée par la transformation : si $x \in \mathsf{Inv}(\varphi)$ et $\alpha \models x$ (resp. $\alpha \models \neg x$) alors, avec $\beta = \mathsf{appl}(\alpha, \varphi, \gamma)$, $\beta \models x$ (resp. $\beta \models \neg x$). On aboutit ainsi à la définition suivante :

Définition 5. *Pour $\varphi \in \Delta\mathcal{LP}$ et $\alpha, \gamma \in \mathcal{LP}$, on définit la transformation d'α par φ sous contrainte γ ainsi :*

$$\mathsf{transf}(\alpha, \varphi, \gamma) = (\alpha \triangleright \gamma) \wedge \mathsf{cp}\,(\varphi)$$

L'application de φ sur α sous contrainte γ est le résultat de cette transformation :

$$\mathsf{appl}(\alpha, \varphi, \gamma) = \mathsf{D}(\mathsf{transf}(\alpha, \varphi, \gamma))$$

On dira que φ est applicable sur α sous contrainte γ si $\mathsf{appl}(\alpha, \varphi, \gamma) \not\models \bot$. Deux conditions nécessaires (mais non suffisantes) d'applicabilité sont $\alpha \wedge \mathsf{G}(\varphi) \not\models \bot$ et $\mathsf{D}(\varphi) \wedge \gamma \not\models \bot$.

Malgré l'usage du *ceteris paribus*, si α, φ et γ contiennent peu de variables (relativement à $|\mathcal{V}|$) on peut calculer $\mathsf{appl}(\alpha, \varphi, \gamma)$ sans avoir à considérer toutes les variables, comme la proposition suivante le montre :

Proposition 8. *Soit $\alpha, \gamma \in \mathcal{LP}$ et $\varphi \in \Delta\mathcal{LP}$ telles que φ peut s'appliquer sur α sous contrainte γ. Soit $\mathcal{W} = \mathcal{V}(\alpha) \cup \mathcal{V}(\varphi) \cup \mathcal{V}(\gamma)$. On a :*

$$\mathsf{appl}(\alpha, \varphi, \gamma) \equiv \mathsf{D}((\alpha \triangleright \gamma) \wedge \mathsf{cp}_{\mathcal{W}}\,(\varphi))$$

Preuve. Soit $\beta_1 = \mathsf{appl}(\alpha, \varphi, \gamma) = \mathsf{D}((\alpha \triangleright \gamma) \wedge \mathsf{cp}\,(\varphi))$ et $\beta_2 = \mathsf{D}((\alpha \triangleright \gamma) \wedge \mathsf{cp}_{\mathcal{W}}\,(\varphi))$. On cherche à montrer que $\beta_1 \equiv \beta_2$.

Comme $\mathsf{cp}\,(\varphi) \models \mathsf{cp}_{\mathcal{W}}\,(\varphi)$, on en déduit que $\beta_1 \models \beta_2$ (en particulier parce que si $\psi \models \psi'$ alors $\mathsf{D}(\psi) \models \mathsf{D}(\psi')$ pour toute $\psi \in \Delta\mathcal{LP}$).

Montrons que $\beta_2 \models \beta_1$. Soit $\mathcal{J} \in \mathcal{M}\,(\beta_2)$: il reste donc à montrer que $\mathcal{J} \models \beta_1$. Par définition de β_2, il existe $\mathcal{I} \in \Omega$ telle que $\mathcal{I}\mathcal{J} \models (\alpha \triangleright \gamma) \wedge \mathsf{cp}_{\mathcal{W}}\,(\varphi)$. Soit alors $\mathcal{I}' \in \Omega$ définie (pour $x \in \mathcal{V}$) par $\mathcal{I}'(x) = \begin{cases} \mathcal{I}(x) & \text{si } x \in \mathcal{W} \\ \mathcal{J}(x) & \text{sinon} \end{cases}$. On va montrer que $\mathcal{I}'\mathcal{J} \models (\alpha \triangleright \gamma) \wedge \mathsf{cp}\,(\varphi)$ ce qui impliquera que $\mathcal{J} \models \beta_2$. Pour cela, il suffit de montrer les assertions suivantes :

(A1) $\mathcal{I}'\mathcal{J} \models \alpha \triangleright \gamma$;

(A2) $\mathcal{I}'\mathcal{J} \models \varphi$;

(A3) Pour toute $x \in \mathsf{Inv}(\varphi)$, $\mathcal{I}'\mathcal{J} \models x^{=}$.

Comme la seule différence entre $\mathcal{I}$ et $\mathcal{I}'$ concerne des variables qui ne sont pas dans $\mathcal{V}(\alpha)$ (puisque hors de $\mathcal{W}$) et que $\mathcal{I} \models \alpha$, on a donc $\mathcal{I}' \models \alpha$. Comme $\mathcal{J} \models \gamma$, on a donc (A1).

De la même façon, la seule différence entre $\mathcal{I}\mathcal{J}$ et $\mathcal{I}'\mathcal{J}$ concerne des variables qui ne sont pas dans $\mathcal{V}(\varphi)$ (puisque hors de $\mathcal{W}$) et que $\mathcal{I}\mathcal{J} \models \varphi$, on a donc $\mathcal{I}'\mathcal{J} \models \varphi$, d'où (A2).

Soit $x \in \mathsf{Inv}(\varphi)$. Pour montrer (A3), il suffit de montrer que $\mathcal{I}'(x) = \mathcal{J}(x)$. Si $x \notin \mathcal{W}$, par définition de $\mathcal{I}'$, c'est vérifié. Si $x \in \mathcal{W}$, on a donc $x \in \mathsf{Inv}(\varphi) \cap \mathcal{W}$, donc $\mathsf{cp}_{\mathcal{W}}(\varphi) \models x^{=}$ (d'après la définition 3). Comme $\mathcal{I}\mathcal{J} \models \mathsf{cp}_{\mathcal{W}}(\varphi)$ on a donc $\mathcal{I}\mathcal{J} \models x^{=}$, i.e. $\mathcal{I}(x) = \mathcal{J}(x)$. Or, $\mathcal{I}'(x) = \mathcal{I}(x)$ puisque $x \in \mathcal{W}$. On a donc $\mathcal{I}'(x) = \mathcal{J}(x)$, ce qui permet de prouver (A3) et qui conclut la preuve. ■

La proposition suivante permet de calculer en $O(|\mathcal{V}(\alpha)| + |\mathcal{V}(\varphi)| + |\mathcal{V}(\gamma)|)$ l'applicabilité et l'application de φ sur α dans le cas où α et β sont des cubes et φ est un cube primitif.

Proposition 9. *Soit α et γ deux cubes de $(\mathcal{LP}, \models)$ et φ, un cube primitif de $(\Delta\mathcal{LP}, \models)$. Soit $\mathcal{W} = \mathcal{V}(\alpha) \cup \mathcal{V}(\varphi) \cup \mathcal{V}(\gamma)$. On peut donc écrire $\alpha \equiv \bigwedge_{a\in\mathcal{W}} \ell(a)$, $\gamma \equiv \bigwedge_{a\in\mathcal{W}} p(a)$, et $\varphi \equiv \bigwedge_{a\in\mathcal{W}} m(a) \triangleright n(a)$ où les $\ell(a)$, $m(a)$, $n(a)$ et $p(a)$ appartiennent à $\{a, \neg a, \top\}$. On a donc $a \in \mathcal{V}(\alpha)$ ssi $\ell(a) \neq \top$, $a \in \mathcal{V}(\gamma)$ ssi $p(a) \neq \top$ et $a \in \mathcal{V}(\varphi)$ ssi $m(a) \neq \top$ ou (de façon équivalente) $n(a) \neq \top$ (puisque φ est un cube primitif).*

Pour $a \in \mathcal{W}$, on définit $f(a) \in \{a, \neg a, \bot\}$ de la façon suivante. Si $\ell(a) \wedge m(a) \models \bot$, $n(a) \wedge p(a) \models \bot$ ou ($\ell(a) \wedge p(a) \models \bot$ et $m(a) = n(a) = \top$) alors $f(a) = \bot$. Sinon,

$$f(a) = \begin{cases} \ell(a) & \text{si } \ell(a) \neq \top \text{ et } m(a) = n(a) = \top \\ p(a) & \text{si } \ell(a) = \top \text{ et } m(a) = n(a) = \top \,. \\ n(a) & \text{sinon} \end{cases}$$

On a alors :
$$\mathsf{appl}(\alpha, \varphi, \gamma) \equiv \bigwedge_{a\in\mathcal{W}} f(a)$$

Par conséquent, φ est applicable sur α sous contrainte γ ssi pour toute $a \in \mathcal{W}$, $f(a) \neq \bot$.

La preuve de cette proposition (détaillée dans [8]) consiste essentiellement à détailler tous les cas possibles selon l'appartenance ou non de a à $\mathcal{V}(\alpha)$, à $\mathcal{V}(\varphi)$ et à $\mathcal{V}(\gamma)$ ($2^3 = 8$ cas à considérer, donc).

Enfin, la proposition suivante permet de calculer $\mathsf{appl}(\alpha, \varphi, \gamma)$ où φ est un cube primitif mais où α et γ sont quelconques, en s'appuyant sur des mises sous forme normale disjonctive de α et γ et sur la proposition 9.

Proposition 10. *Soit $\alpha, \alpha_1, \alpha_2, \beta, \beta_1, \beta_2 \in \mathcal{LP}$ et $\varphi \in \Delta\mathcal{LP}$. On a les équivalences suivantes :*

$$\mathsf{appl}(\alpha_1 \vee \alpha_2, \varphi, \gamma) \equiv \mathsf{appl}(\alpha_1, \varphi, \gamma) \vee \mathsf{appl}(\alpha_2, \varphi, \gamma)$$
$$\mathsf{appl}(\alpha, \varphi, \gamma_1 \vee \gamma_2) \equiv \mathsf{appl}(\alpha, \varphi, \gamma_1) \vee \mathsf{appl}(\alpha, \varphi, \gamma_2)$$

Preuve. On a :

$$\begin{aligned} \mathsf{transf}(\alpha_1 \vee \alpha_2, \varphi, \gamma) &= ((\alpha_1 \vee \alpha_2) \triangleright \gamma)) \wedge \mathsf{cp}(\varphi) \\ &\equiv ((\alpha_1 \triangleright \gamma) \wedge \mathsf{cp}(\varphi)) \vee ((\alpha_2 \triangleright \gamma) \wedge \mathsf{cp}(\varphi)) \\ &\equiv \mathsf{transf}(\alpha_1, \varphi, \gamma) \vee \mathsf{transf}(\alpha_2, \varphi, \gamma) \end{aligned}$$

D'après (2), on en déduit la première équivalence de la proposition. La deuxième équivalence se prouve de façon similaire. ■

4.4 Application d'une règle d'adaptation sur l'exemple

On considère le problème d'adaptation $((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^{\mathtt{cible}})$ et la formule φ de la section 4.1, ainsi que la règle d'adaptation $R = (\varphi, \mathsf{C})$ où C est choisi arbitrairement. En appliquant (15) sur cet exemple, on obtient un $\beta \in \mathcal{LP}$ qui est satisfiable et tel que $\beta \equiv \mathsf{CD} \wedge \mathtt{x}^{\mathtt{cible}} \wedge \mathtt{y}^{\mathtt{cible}}$ avec

$$\begin{aligned} \mathtt{y}^{\mathtt{cible}} = \mathtt{iBatavia} \wedge \mathtt{iMagret} \wedge \mathtt{iHuileDOlive} \\ \wedge\, \mathtt{iJusCitron} \wedge \mathtt{iSel} \wedge \ldots \end{aligned}$$

(qui est le résultat attendu sur cet exemple).

4.5 Un algorithme pour le RàPC

Une approche du RàPC utilisée dans le système Taaable [3], notamment, consiste à effectuer une remémoration s'appuyant sur une transformation minimale du problème cible afin qu'il s'apparie exactement avec au moins un cas source puis à utiliser une « transformation inverse » sur un cas remémoré pour obtenir une solution.

Dans le cadre de représentation de cet article, pour implanter cette approche, les transformations considérées seront celles associées aux règles d'adaptation $R = (\varphi, \mathsf{C})$: on considérera la transformation par la règle $R^{-1} = (\varphi^{-1}, \mathsf{C})$ (pour garantir un résultat, on pourra ajouter les règles $(a^{\neq}, \mathsf{C})$, pour $a \in \mathcal{V}$). En supposant que les coûts sont additifs, la remémoration retournera non seulement un (ou plusieurs) $(\mathtt{x}^s, \mathtt{y}^s)$ mais également une séquence de règles ${R_1}^{-1}$, ${R_2}^{-1}$, ..., ${R_q}^{-1}$ telle que, avec $\mathtt{x}^0 = \mathsf{CD} \wedge \mathtt{x}^{\mathtt{cible}}$ et, pour $i \in \{1, 2, \ldots, q\}$, $\mathtt{x}^i = \mathsf{appl}(\mathtt{x}^{i-1}, {\varphi_i}^{-1}, \top)$ (où $R_i = (\varphi_i, \mathsf{C}_i)$), avec $\mathtt{x}^s \wedge \mathtt{y}^s \models_{\mathsf{CD}} \mathtt{x}^q$. La transformation sera de coût $\sum_{i=1}^{q} \mathsf{C}_i$.

Une fois un cas $(\mathtt{x}^s, \mathtt{y}^s)$ remémoré avec une séquence de règles d'adaptation ${R_1}^{-1}$, ${R_2}^{-1}$, ..., ${R_q}^{-1}$ associée, l'idée est d'appliquer la séquence inverse de ces règles sur $\mathtt{x}^s \wedge \mathtt{y}^s$ pour résoudre $\mathtt{x}^{\mathtt{cible}}$: avec $\mathtt{y}^q$ tel que $\mathtt{x}^s \wedge \mathtt{y}^s \equiv \mathtt{x}^q \wedge \mathtt{y}^q$, on calcule successivement $\mathtt{y}^{q-1}$, ..., $\mathtt{y}^0$ avec, pour $i \in \{q-1, q-2, \ldots, 0\}$, $\mathtt{x}^i \wedge \mathtt{y}^i \equiv \mathsf{appl}(\mathtt{x}^{i+1} \wedge \mathtt{y}^{i+1}, \varphi_{i+1}, \mathtt{x}^i)$. Avec $\mathtt{y}^{\mathtt{cible}}$ telle que $\mathsf{CD} \wedge \mathtt{x}^{\mathtt{cible}} \wedge \mathtt{y}^{\mathtt{cible}} \equiv \mathtt{x}^0 \wedge \mathtt{y}^0$, on a alors une proposition de solution de $\mathtt{x}^{\mathtt{cible}}$, adaptée en q étapes à partir du cas remémoré.

Pour qu'une telle adaptation soit possible, il faut que les règles d'adaptation R_i soient applicables. Cela est garanti par la proposition suivante, dans le cas où $\mathtt{x}^s \wedge \mathtt{y}^s$, $\mathtt{x}^{\mathtt{cible}}$ et φ_i sont des cubes :

Proposition 11. *Soit* α *et* γ, *deux cubes de* $(\mathcal{LP}, \models)$ *et* φ, *un cube de* $(\Delta\mathcal{LP}, \models)$. *On a :*

$$\mathsf{appl}(\alpha, \varphi, \gamma) \not\models \top \quad \textit{ssi} \quad \mathsf{appl}(\gamma, \varphi^{-1}, \top) \wedge \alpha \not\models \bot$$

La preuve de cette proposition (détaillée dans [8]) consiste à reprendre le découpage en 8 cas de la preuve de la proposition 9.

On montre que $\mathtt{x}^i \wedge \mathtt{y}^i$ est satisfiable pour tout $i \in \{0, 1, \dots, q\}$ en partant de la fin :

$(i = q)$ Comme, par définition, $\mathtt{x}^q \wedge \mathtt{y}^q$ est équivalente au cube complet $\mathtt{x}^s \wedge \mathtt{y}^s$ et que les cubes sont satisfiables, $\mathtt{x}^q \wedge \mathtt{y}^q \not\models \bot$.

$(i = q-1)$ $\mathtt{x}^{q-1} \wedge \mathtt{y}^{q-1} \equiv \mathsf{appl}(\mathtt{x}^q \wedge \mathtt{y}^q, \varphi_q, \mathtt{x}^{q-1})$ qui, d'après la proposition 11, est satisfiable ssi β est satisfiable avec $\beta = \mathsf{appl}(\mathtt{x}^{q-1}, {\varphi_q}^{-1}, \top) \wedge \mathtt{x}^q \wedge \mathtt{y}^q$. Or, par définition de $\mathtt{x}^q$, $\beta \equiv \mathtt{x}^q \wedge \mathtt{x}^q \wedge \mathtt{y}^q \equiv \mathtt{x}^q \wedge \mathtt{y}^q$ qui est satisfiable (cf. cas $(i = q)$), donc β est satisfiable et par conséquent $\mathtt{x}^{q-1} \wedge \mathtt{y}^{q-1} \not\models \bot$.

$(i = q-2)$ $\mathtt{x}^{q-2} \wedge \mathtt{y}^{q-2} \equiv \mathsf{appl}(\mathtt{x}^{q-1} \wedge \mathtt{y}^{q-1}, \varphi_{q-1}, \mathtt{x}^{q-2})$ qui, d'après la proposition 11, est satisfiable ssi β est satisfiable avec $\beta = \mathsf{appl}(\mathtt{x}^{q-2}, {\varphi_{q-1}}^{-1}, \top) \wedge \mathtt{x}^{q-1} \wedge \mathtt{y}^{q-1}$. Or, par définition de $\mathtt{x}^{q-1}$, $\beta \equiv \mathtt{x}^{q-1} \wedge \mathtt{x}^{q-1} \wedge \mathtt{y}^{q-1} \equiv \mathtt{x}^{q-1} \wedge \mathtt{y}^{q-1}$ qui est satisfiable (cf. cas $(i = q-1)$), donc β est satisfiable et par conséquent $\mathtt{x}^{q-2} \wedge \mathtt{y}^{q-2} \not\models \bot$.

$(i = 0)$ $\mathtt{x}^0 \wedge \mathtt{y}^0 \equiv \mathsf{appl}(\mathtt{x}^1 \wedge \mathtt{y}^1, \varphi_1, \mathtt{x}^0)$ qui, d'après la proposition 11, est satisfiable ssi β est satisfiable avec $\beta = \mathsf{appl}(\mathtt{x}^0, {\varphi_1}^{-1}, \top) \wedge \mathtt{x}^1 \wedge \mathtt{y}^1$. Or, par définition de $\mathtt{x}^1$, $\beta \equiv \mathtt{x}^1 \wedge \mathtt{x}^1 \wedge \mathtt{y}^1 \equiv \mathtt{x}^1 \wedge \mathtt{y}^1$ qui est satisfiable (cf. cas $(i = 1)$), donc β est satisfiable et par conséquent $\mathtt{x}^0 \wedge \mathtt{y}^0 \not\models \bot$.

Comme $\mathtt{CD} \wedge \mathtt{x}^{\mathtt{cible}} \wedge \mathtt{y}^{\mathtt{cible}} \equiv \mathtt{x}^0 \wedge \mathtt{y}^0$, $\mathtt{CD} \wedge \mathtt{x}^{\mathtt{cible}} \wedge \mathtt{y}^{\mathtt{cible}}$ est satisfiable : l'adaptation donne bien un résultat cohérent.

Notons enfin que cet algorithme peut être utilisé pour résoudre un problème d'adaptation $((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^{\mathtt{cible}})$ (quand un cas $(\mathtt{x}^s, \mathtt{y}^s)$ a déjà été choisi). Il suffit pour cela de l'appliquer à une base de cas singleton $\mathtt{BC} = \{(\mathtt{x}^s, \mathtt{y}^s)\}$.

4.6 Vers une composition des règles d'adaptation

L'algorithme de la section 4.5 s'appuie sur une séquence d'applications de règles d'adaptation. La question ouverte qui est posée ici est le calcul d'une composition de règles d'adaptation : étant donnés deux telles règles R_1 et R_2, peut-on définir une règle R telle que l'application de R équivaut à l'application successive de R_1 et de R_2 ? *A priori*, la composition des formules présentée à la section 3.5 devrait être utile ici, mais nous n'avons pas de réponse définitive à cette question pour l'heure.

Un intérêt de définir une telle composition de règles serait dans le post-traitement de l'apprentissage de ces règles. Si on utilise, par exemple, une extraction de motifs fermés fréquents, elle produira souvent un grand nombre de motifs M interprétables en règles R. S'il était possible de trouver une famille génératrice minimale pour la composition de ces règles, cela diminuerait leur nombre (et le temps de calcul) sans altérer l'inférence.

5 Discussion et liens avec d'autres travaux

Dans [7], $(\Delta\mathcal{LP}, \models)$ était comparée avec d'autres formalismes. Cette comparaison est complétée en section 5.1, sous l'angle des deux types d'inférences associées aux règles d'adaptation. La section 5.2 traite plus en détail du lien entre $(\Delta\mathcal{LP}, \models)$ et une logique dynamique.

5.1 Observations et actions

Les mécanismes liés à l'adaptation par transformation présentés dans la section 4 peuvent être partagés en mécanismes d'*observations* et mécanismes d'*actions*.

Les mécanismes d'observations apparaissent lors de l'apprentissage de connaissances d'adaptation (évoquée dans la section 2.2) dans laquelle sont observées des variations entre cas sources.

Étant donné une règle d'adaptation $R = (\varphi, \mathtt{C})$ et un problème d'adaptation $((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^{\mathtt{cible}})$, l'application de R sur $((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^{\mathtt{cible}})$ peut être apparentée à un mécanisme d'action : on crée une hypothèse de solution $\mathtt{y}^{\mathtt{cible}}$ de $\mathtt{x}^{\mathtt{cible}}$, partant de $(\mathtt{x}^s, \mathtt{y}^s)$. Donc, en assimilant un cas à un état et une règle d'adaptation à une action, on peut considérer l'application d'une règle d'adaptation comme l'application d'une action sur un état, pour obtenir un nouvel état (sachant que ce nouvel état est contraint par $\mathtt{x}^{\mathtt{cible}}$).

L'application de règles d'adaptation invite donc à se pencher sur les formalismes et inférences relatives aux actions (voir la synthèse [6]). En l'occurrence, les actions considérées sont déterministes : si une action est applicable sur un état, elle génère un état et un seul. Dans la section suivante, nous nous intéresserons à une logique dynamique et à ses liens avec $(\Delta\mathcal{LP}, \models)$.

Une dernière remarque peut être faite. On pourrait imaginer faire le lien entre les observations et les actions épistémiques (également évoquées dans [6]). Cependant, le lien avec les observations dont il est question ici (observer des variations entre deux cas) et les actions épistémiques (agir pour observer et acquérir/modifier des connaissances sur l'état actuel) ne s'est pas révélé fructueux (ou pas encore).

5.2 $(\Delta\mathcal{LP}, \models)$ et DL-PA

La logique DL-PA (*dynamic logic of propositional assignments*) est une logique dynamique dont les programmes atomiques sont des affectations de variables propositionnelles, i.e. des expressions $+a$ et $-a$ pour $a \in \mathcal{V}$ correspondant à une affectation de a par respectivement $\mathtt{1}$ et $\mathbb{0}$ [1].

Les formules de DL-PA sont des variables propositionnelles, des formules construites à partir d'autres formules en utilisant les connecteurs de la logique propositionnelle et les formules de la forme $\langle\pi\rangle\alpha$ où π est un programme et α est une formule. Les programmes sont soit des programmes atomiques, soit d'une des formes π_1 ; π_2, $\pi_1 \cup \pi_2$, π^* et $\alpha?$ (où π, π_1 et π_2 sont des programmes et où α est une formule).

La sémantique d'une formule α est donnée par un sous-ensemble $\mathcal{M}(\alpha)$ de Ω et celle d'un programme π par un sous-ensemble $\mathcal{M}(\pi)$ de $\Omega \times \Omega = \Delta\Omega$, ce qui suggère d'étudier les liens entre les programmes de DL-PA et les formules de $(\Delta\mathcal{LP}, \models)$. Cette sémantique est définie de façon habituelle pour les variables propositionnelles et les formules construites sur les connecteurs, et pour le reste on a (pour $a \in \mathcal{V}$, α, une formule, π, π_1 et π_2, des programmes) :

$$\mathcal{M}(\langle\pi\rangle\alpha) = \left\{ \mathcal{I} \in \Omega \,\middle|\, \begin{array}{l} \text{il existe } \mathcal{J} \in \Omega \text{ telle que} \\ \mathcal{I}\mathcal{J} \models \pi \text{ et } \mathcal{J} \models \alpha \end{array} \right\}$$

$$\mathcal{M}(+a) = \left\{ \mathcal{I}\mathcal{J} \in \Delta\Omega \,\middle|\, \begin{array}{l} \mathcal{J} \models a \\ \text{et, pour toute } b \in \mathcal{V} \setminus \{a\}, \\ \mathcal{I}(b) = \mathcal{J}(b) \end{array} \right\}$$

$$\mathcal{M}(-a) = \left\{ \mathcal{I}\mathcal{J} \in \Delta\Omega \,\middle|\, \begin{array}{l} \mathcal{J} \models \neg a \\ \text{et, pour toute } b \in \mathcal{V} \setminus \{a\}, \\ \mathcal{I}(b) = \mathcal{J}(b) \end{array} \right\}$$

$$\mathcal{M}(\pi_1 ; \pi_2) = \left\{ \mathcal{I}\mathcal{K} \in \Delta\Omega \,\middle|\, \begin{array}{l} \text{il existe } \mathcal{J} \in \Omega \text{ telle que} \\ \mathcal{I}\mathcal{J} \models \pi_1 \text{ et } \mathcal{J}\mathcal{K} \models \pi_2 \end{array} \right\}$$

$$\mathcal{M}(\pi_1 \cup \pi_2) = \mathcal{M}(\pi_1) \cup \mathcal{M}(\pi_2)$$

$$\mathcal{M}(\alpha?) = \{\mathcal{I}\mathcal{I} \mid \mathcal{I} \in \mathcal{M}(\alpha)\}$$

Enfin, $\mathcal{I}\mathcal{J} \models \pi^*$ s'il existe une séquence $\mathcal{I}_0, \mathcal{I}_1, \dots, \mathcal{I}_p$ ($p \geq 0$) telle que $\mathcal{I}_0 = \mathcal{I}$, $\mathcal{I}_p = \mathcal{J}$ et $\mathcal{I}_k\mathcal{I}_{k+1} \models \pi$ pour tout $k \geq 0$ et $k < p$.

Le *ceteris paribus* sur des formules de $(\Delta\mathcal{LP}, \models)$ va être utile pour faire le lien de formules de cette logique vers des programmes de DL-PA. On peut aussi définir une opération sur les programmes de DL-PA permettant de faire un lien dans l'autre sens. Cela suppose que l'ensemble des variables propositionnelles est fini et fixé, contrairement à l'hypothèse d'un ensemble dénombrable de variables qui est faite dans [1]. On fera donc cette restriction dans la suite de cette section.

On part de l'observation que le programme $+a \cup -a$ (pour $a \in \mathcal{V}$) fait l'inverse d'une affectation : même si a est affecté à une valeur ($\mathbb{0}$ ou 1) dans l'état de départ, l'exécution de ce programme fait que a n'a pas d'affection après l'exécution de ce programme. On généralise cela à un ensemble $\mathcal{W} = \{a_1, \dots, a_p\}$ de variables :

$$\mathsf{d\acute{e}saff}(\{a_1, \dots, a_p\}) = (+a_1 \cup -a_1) ; \dots ; (+a_p \cup -a_p)$$

La table 1 donne des égalités entre ensembles de modèles de certaines formules de $(\Delta\mathcal{LP}, \models)$ et ensembles de modèles de certains programmes de DL-PA. On notera que dans

Pour $a \in \mathcal{V}$, $\varphi_1, \varphi_2 \in \Delta\mathcal{LP}$ et π_1 et π_2 des programmes de DL-PA tels que $\mathcal{M}(\varphi_1) = \mathcal{M}(\pi_1)$ et $\mathcal{M}(\varphi_2) = \mathcal{M}(\pi_2)$, on a le tableau suivant, dans lequel, pour chaque ligne, $\mathcal{M}(\varphi) = \mathcal{M}(\pi)$:

$\varphi \in \Delta\mathcal{LP}$	π : programme de DL-PA
$\mathsf{cp}(a^{\bullet 1})$	$+a$
$\mathsf{cp}(a^{\bullet \mathbb{0}})$	$-a$
$a^{=1}$	$a?$; $+a$; $\mathsf{d\acute{e}saff}(\mathcal{V} \setminus \{a\})$
$a^{=\mathbb{0}}$	$\neg a?$; $-a$; $\mathsf{d\acute{e}saff}(\mathcal{V} \setminus \{a\})$
a^{+}	$\neg a?$; $+a$; $\mathsf{d\acute{e}saff}(\mathcal{V} \setminus \{a\})$
a^{-}	$a?$; $-a$; $\mathsf{d\acute{e}saff}(\mathcal{V} \setminus \{a\})$
$\varphi_1 \vee \varphi_2$	$\pi_1 \cup \pi_2$
φ_1 ; φ_2	π_1 ; π_2

TABLE 1 – Des correspondances entre formules de $(\Delta\mathcal{LP}, \models)$ et programmes de DL-PA.

cette table ne sont pas distingués les opérations sur les logiques des connecteurs de cette logique : par exemple, dans π_1 ; π_2, le ; appartient à la syntaxe de la logique alors que dans φ_1 ; φ_2, le ; est un opérateur. Par ailleurs, pour $\alpha \in \mathcal{LP}$ (qui est donc un cas particulier de formule de DL-PA), on a $\mathcal{M}(\alpha?) = \mathcal{M}(\widehat{\alpha})$ où $\widehat{\alpha}$ est défini dans [7] comme étant le plongement de α dans $(\Delta\mathcal{LP}, \models)$ défini syntaxiquement en remplaçant toutes les occurrences de $a \in \mathcal{V}(\alpha)$ par $a^{=1}$. Enfin, pour $\alpha \in \mathcal{LP}$, $\varphi \in \Delta\mathcal{LP}$ et un programme π de DL-PA tel que $\mathcal{M}(\varphi) = \mathcal{M}(\pi)$, on a $\mathcal{M}(\langle\pi\rangle\alpha) = \mathsf{G}(\varphi \wedge \alpha^{\bullet 1})$.

De façon générale, on peut traduire toute $\varphi \in \Delta\mathcal{LP}$ en un programme π de DL-PA de façon à ce que $\mathcal{M}(\varphi) = \mathcal{M}(\pi)$, comme nous le verrons ci-dessous, du moins sous l'hypothèse d'un ensemble fini de variables. Notons néanmoins que la traduction proposée n'est pas efficace et qu'il y a peu d'espoir d'en trouver une qui le soit puisque la satisfiabilité dans $(\Delta\mathcal{LP}, \models)$ est NP-complète alors que la complexité de la satisfiabilité dans DL-PA est EXPTIME-difficile (même si cette complexité concerne les formules et pas les programmes, c'est une indication de la complexité relative aux inférences sur les programmes).

Une preuve simple du fait que $\varphi \in \Delta\mathcal{LP}$ puisse se traduire en π tient au fait que tout sous-ensemble de $\Delta\Omega$ est représentable par un programme π (sous l'hypothèse d'un ensemble fini de variables). Une preuve plus constructive consiste à associer d'abord à chaque cube complet un programme de DL-PA. Soit ψ un cube complet de $(\Delta\mathcal{LP}, \models)$: $\psi = \bigvee_{a \in \mathcal{V}} a^{\omega(a)}$ pour une $\omega \in \Delta\Omega_{\text{alt}}$. Pour $a \in \mathcal{V}$, soit le programme

$$\pi_a = \begin{cases} a? ; +a & \text{si } \omega(a) = {=}1 \\ \neg a? ; -a & \text{si } \omega(a) = {=}\mathbb{0} \\ \neg a? ; +a & \text{si } \omega(a) = + \\ a? ; -a & \text{si } \omega(a) = - \end{cases}$$

Avec $\mathcal{V} = \{a_1, a_2, \ldots, a_n\}$ soit $\pi = \pi_{a_1}$; π_{a_2} ; ... ; π_{a_n}. On peut montrer alors que $\mathcal{M}(\pi) = \mathcal{M}(\psi)$. De façon générale, toute φ peut s'écrire comme une disjonction de cubes complets : $\varphi = \psi_1 \vee \psi_2 \vee \ldots \vee \psi_p$. En associant à chaque cube ψ_k un programme π_k comme ci-dessus, on peut construire le programme $\pi = \pi_1 \cup \pi_2 \cup \ldots \cup \pi_p$. On a alors $\mathcal{M}(\pi) = \bigcup_{k=1}^{p} \mathcal{M}(\pi_k) = \bigcup_{k=1}^{p} \mathcal{M}(\psi_k) = \mathcal{M}(\varphi)$.

Il y a donc des liens forts entre $(\Delta\mathcal{LP}, \models)$ et DL-PA qui doivent être étudiés davantage. Il semblerait que $(\Delta\mathcal{LP}, \models)$ soit davantage utile pour *observer* des variations (et c'est pour cela que cette logique a été définie initialement) alors que DL-PA qui, en quelque sorte, intègre naturellement le *ceteris paribus* soit appropriée pour *agir* avec ces variations (en considérant l'application d'une règle d'adaptation comme l'exécution d'un programme). Ainsi, raisonner avec des règles d'adaptation pourrait se faire avec $(\Delta\mathcal{LP}, \models)$ pour leur apprentissage et avec DL-PA (ou un de ses fragments) pour leur application.

6 Conclusion

Cet article a abordé l'usage de la logique des variations propositionnelles au raisonnement à partir de cas, en se penchant sur une première problématique : celle de la représentation des règles d'adaptation.

De nombreuses perspectives suivent cette étude, en particulier celles mentionnées dans [7] et pas traitées dans cet article et celles qui apparaissent dans ce présent article. Nous considérerons en particulier l'étude de l'extension de la logique des variations propositionnelles à d'autres logiques, en particulier, des logiques avec des variables de types entier ou réel. Cela suppose de définir un langage de symboles de variations qui sera nécessairement incomplet : une logique étant supposée s'appuyer sur un langage dénombrable (du moins, c'est une hypothèse que nous considérons) et l'ensemble des relations binaires sur $\mathbb{N}$ étant indénombrable... Une façon de conduire cette étude passe par les logiques dynamiques au-delà de DL-PA, en tentant de résoudre l'équation analogique « Une telle logique des variations serait à $(\Delta\mathcal{LP}, \models)$ ce qu'une logique dynamique serait à DL-PA. »

Références

[1] Balbiani, Ph., A. Herzig et N. Troquard: *Dynamic logic of propositional assignments : a well-behaved variant of PDL*. Dans *28th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 143–152. IEEE, 2013.

[2] Carbonell, J. G.: *Learning by analogy : Formulating and generalizing plans from past experience*. Dans Michalski, R. S., J. G. Carbonell et T. M. Mitchell (rédacteurs) : *Machine Learning, An Artificial Intelligence Approach*, chapitre 5, pages 137–161. Morgan Kaufmann, Inc., 1983.

[3] Cordier, A., V. Dufour-Lussier, J. Lieber, E. Nauer, F. Badra, J. Cojan, E. Gaillard, L. Infante-Blanco, P. Molli, A. Napoli et H. Skaf-Molli: *Taaable : a Case-Based System for personalized Cooking*. Dans Montani, S. et L. C. Jain (rédacteurs) : *Successful Case-based Reasoning Applications-2*, tome 494 de *Studies in Computational Intelligence*, pages 121–162. Springer, 2014.

[4] d'Aquin, M., F. Badra, S. Lafrogne, J. Lieber, A. Napoli et L. Szathmary: *Case base mining for adaptation knowledge acquisition*. Dans Veloso, M. M. (rédacteur) : *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI'07)*, pages 750–755. Morgan Kaufmann, Inc., 2007.

[5] d'Aquin, M., J. Lieber et A. Napoli: *Adaptation Knowledge Acquisition : a Case Study for Case-Based Decision Support in Oncology*. Computational Intelligence (an International Journal), 22(3/4) :161–176, 2006.

[6] Dupin de Saint-Cyr, F., A. Herzig, J. Lang et P. Marquis: *Raisonnement sur l'action et le changement*. Dans *Panorama de l'intelligence artificielle — ses bases méthodologiques, ses développements*, tome 1, chapitre 12, pages 363–392. Cépaduès, 2014.

[7] François, N., Th. Laure et J. Lieber: *Une logique pour représenter des variations propositionnelles*. Dans Bouraoui, Z., F. Schwarzentruber et A. Wilczynski (rédacteurs) : *Journées d'intelligence artificielle fondamentale – plateforme AFIA 2023*, page 11, Strasbourg, France, juillet 2023. Z. Bouraoui and F. Schwarzentruber and A. Wilczynski.

[8] François, N. et J. Lieber: *Appliquer la logique des variations propositionnelles à la représentation de règles d'adaptation pour le raisonnement à partir de cas (version étendue)*. rapport technique, Loria, 2024. (Ce rapport se trouve à l'adresse `https://hal.science/hal-04497632`).

[9] Hanney, K. et M. T. Keane: *Learning adaptation rules from a case-base*. Dans Smith, I. et B. Faltings (rédacteurs) : *Advances in Case-Based Reasoning – Proc. of the Third Eur. Workshop, EWCBR'96*, LNAI 1168, pages 179–192. Springer Verlag, Berlin, 1996.

[10] Jalali, V., D. Leake et N. Forouzandehmehr: *Learning and applying adaptation rules for categorical features : An ensemble approach*. AI Communications, 30(3-4) :193–205, 2017.

[11] Richter, M. M. et R. O. Weber: *Case-based reasoning, a textbook*. Springer, 2013.

[12] Riesbeck, C. K. et R. C. Schank: *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1989. Available on line.