Plateforme de raisonnement basée sur la valeur d'incohérence de Shapley MI

Sébastien Konieczny Stéphanie Roussel
konieczny@cril.fr sroussel@cril.fr

CRIL - CNRS, UMR 8188
Université d'Artois
F-62307 Lens, France

Résumé :
Dans cet article nous montrons comment construire une plate-forme de raisonnement à partir d'une valeur d'incohérence. L'idée est d'utiliser la valeur d'incohérence pour calculer la responsabilité de chaque formule d'une base de croyances dans l'incohérence de la base. Cette évaluation permet de définir une stratification (pré-ordre total) de la base qui peut alors être utilisée pour différentes tâches de raisonnement comme l'inférence, la révision de croyances ou la conciliation. Nous montrons que les opérateurs alors obtenus sont intéressants et ont de bonnes propriétés logiques. La valeur d'incohérence que nous utilisons est la valeur d'incohérence de Shapley MI. Cette dernière est connue pour ses bonnes propriétés et peut de plus être calculée à partir des sous-ensembles minimaux incohérents. Nous avons développé une plate-forme en Java, PRISM, qui utilise la librairie Sat4j pour calculer les sous-ensembles minimaux incohérents, nous donnant ainsi une manière efficace de calculer les valeurs de Shapley MI. Nous avons implémenté plusieurs méthodes d'inférence, de révision et de conciliation à partir de cette valeur d'incohérence. Nous fournissons ainsi une plate-forme de raisonnement complète, pouvant par exemple être utilisée à des fins acédémiques.

1 Introduction

L'évolution des croyances et le raisonnement en présence d'incohérence sont deux thèmes qui ont reçu une attention particulière dans la littérature. Les résultats théoriques sur les méthodes de raisonnement sont nombreux : caractérisation logique pour l'inférence non-monotone [1, 2], révision de croyances [3, 4, 5], fusion de croyances [6, 7, 8], etc. Plusieurs méthodes particulières ont également été proposées pour la révision de croyances [9], la fusion de croyances [8], l'inférence en présence d'incohérences [10], etc.

Par contre il existe très peu d'approches pour lesquelles des implémentations ont été proposées. Les implémentations sont pourtant particulièrement utiles pour tester les opérateurs proposés, expérimenter les différentes méthodes de raisonnement et pour diffuser ces opérateurs plus largement dans la communauté IA. En fait, il n'existe à notre connaissance que *deux* [1] plates-formes qui implémentent plusieurs méthodes de raisonnement. La première plate-forme est SATEN [12, 13]. Elle a été développée par Williams et Sims et permet d'effectuer de l'extraction de théorie, de la révision itérée de croyances, du raisonnement non-monotone, du raisonnement possibiliste et du raisonnement hypothétique. La plate-forme SATEN est écrite en Java 1.1 et est basée sur un prouveur de théorèmes. Elle utilise la représentation Spohnienne des états épistémiques [14]. La seconde plate-forme est COBA [15, 16]. Elle a été développée par Delgrance, Liu, Schaub et Thiele et permet d'effectuer de la révision et de la contraction de croyances basée sur des approches de projection de langage définies dans [17]. COBA est une applet Java et utilise un solveur SAT.

Dans cet article nous proposons une nouvelle plate-forme de raisonnement. Celle-ci est basée sur le calcul d'une valeur d'incohérence, la valeur d'incohérence de Shapley MI [18], qui se calcule facilement à partir des sous-ensembles minimaux incohérents de la base de croyances. Ainsi, nous obtenons non seulement une mesure de l'incohérence de chacune des formules de la base et de la base elle-même mais également une stratification (pré-ordre total) de la base. Nous utilisons cette dernière pour les différentes tâches de raisonnement que sont l'inférence, la révision de croyances et la conciliation.

L'article est organisé de la manière suivante : nous présentons des définitions préliminaires, les mesures et les valeurs d'incohérence dans la section 2. Nous étudions ensuite trois opérations de raisonnement : l'inférence, la révision et la conciliation respectivement dans les sections 3, 4 et 5. Nous décrivons la plate-forme de raisonnement PRISM dans la section 6 et concluons sur les perspectives de travaux futurs en section 7.

1. Nous pouvons également mentionner le projet QUIP [11], mais la plate-forme correspondante n'est, à notre connaissance, pas encore disponible.

2 Préliminaires - Mesures d'incohérence - Valeurs d'incohérence

On considère ici un langage $\mathcal{L}$ que l'on construit à partir d'un ensemble fini de symboles propositionnels $\mathcal{P}$.

Une base de croyances K est un ensemble fini de formules propositionnelles. On note $\mathcal{K}_{\mathcal{L}}$ l'ensemble des bases de croyances que l'on peut définir à partir des formules du langage $\mathcal{L}$.

Si une base de croyances K n'est pas cohérente, alors on définit les sous-ensembles minimaux incohérents [2] de K de la manière suivante : $\mathsf{MI}(K) = \{K' \subseteq K \mid K' \vdash \bot \text{ and } \forall K'' \subset K', K'' \nvdash \bot\}$. Une formule est dite *libre* si elle n'appartient à aucun sous-ensemble minimal incohérent.

La notion de sous-ensemble maximal cohérent [3] est duale de la notion de sous-ensembles minimaux incohérents. On définit les sous-ensembles maximaux cohérents de la manière suivante : $\mathsf{MC}(K) = \{K' \subseteq K \mid K' \nvdash \bot \text{ et } \forall K'' \text{ s. t. } K' \subset K'', K'' \vdash \bot\}$.

Un profil Ψ est un vecteur de bases de croyances $\langle K_1, \ldots, K_n \rangle$. L'ensemble de tous les profils est noté $\mathcal{E}$. $\bigwedge \Psi$ représente la conjonction des bases de Ψ.

Des travaux récents ont commencé à étudier comment mesurer l'incohérence d'une base de croyances (voir par exemple [19]). Plusieurs approches raisonnables permettent de traiter ce problème. Cette diversité n'est pas surprenante puisque de la même manière il existe plusieurs approches pour définir des relations d'inférence non-triviales à partir de bases incohérentes.

Dans [18], une distinction est faite entre les *mesures d'incohérence*, qui mesurent l'incohérence d'une base de croyances, et les *valeurs d'incohérence*, qui mesurent (la responsabilité de) l'incohérence de chaque formule d'une base de croyances.

Evidemment, les valeurs d'incohérence, définies formule par formule, peuvent être utilisées pour définir les mesures d'incohérence correspondantes. Il suffit d'agréger les valeurs d'incohérence calculées.

Commençons par rappeler la définition de la Valeur d'Incohérence de Shapley (SIV) [18] :

2. Notés parfois MUS (pour Minimally Unsatisfiable Subsets).
3. Notés parfois MSS (pour Maximally Satisfiable Subsets).

Définition 1 ([18]) *Une mesure d'incohérence I est appelée* mesure d'incohérence basique *si elle satisfait les propriétés suivantes* [4]*, $\forall K, K' \in \mathcal{K}_{\mathcal{L}}$, $\forall \alpha, \beta \in \mathcal{L}$:*

- *$I(K) = 0$ ssi K est cohérent* **(Cohérence)**
- *$I(K \cup K') \geq I(K)$* **(Monotonie)**
- *Si α est une formule libre de K, alors $I(K) = I(K \setminus \{\alpha\})$* **(Indépendance de Formule Libre)**

Nous pouvons maintenant définir la valeur d'incohérence de Shapley.

Définition 2 ([18]) *Soit I une mesure d'incohérence basique. On définit la* Valeur de Shapley Incohérente *(SIV) correspondante, notée S^I, comme la valeur de Shapley du jeu de coalition défini par la fonction I, i.e. soit $\alpha \in K$:*

$$S^I_\alpha(K) = \sum_{C \subseteq K} A_c(I(C) - I(C \setminus \{\alpha\}))$$

où $A_c = \frac{(c-1)!(n-c)!}{n!}$, n étant le cardinal de K et c le cardinal de C.

A partir de cette valeur, nous définissons la mesure d'incohérence (globale) d'une base de croyances comme étant la valeur de son plus mauvais élément :

Définition 3 ([18]) *Soit K une base de croyances, $\hat{S}^I(K) = \max_{\alpha \in K} S^I_\alpha(K)$*

Un exemple simple de mesure d'incohérence basique est la valeur d'incohérence drastique : $I_d(K) = 0$ si K est cohérent, et $I_d(K) = 1$ sinon. Cette mesure d'incohérence est la plus simple que l'on puisse définir et n'est donc pas très intéressante en soi. Cependant, la SIV correspondante est déjà une mesure intéressante.

Un autre exemple de mesure d'incohérence basique est celle qui compte le nombre de conflits d'une base à partir du cardinal des sous-ensembles minimaux cohérents :

$$I_{MI}(K) = |\mathsf{MI}(K)|$$

4. Dans [18], une propriété supplémentaire, la **Dominance** est également requise.

La SIV correspondante est intéressante et a été caractérisée logiquement dans [18]. Commençons par définir ces propriétés sur les valeurs d'incohérence : soient I une mesure d'incohérence basique et S^I la valeur d'incohérence de Shapley correspondante :

• $\sum_{\alpha \in K} S^I_\alpha(K) = I(K)$ **(Distribution)**

• Si $\alpha, \beta \in K$ sont telles que pour tout $K' \subseteq K$ avec $\alpha, \beta \notin K'$, $I(K' \cup \{\alpha\}) = I(K' \cup \{\beta\})$, alors $S^I_\alpha(K) = S^I_\beta(K)$ **(Symétrie)**

• Si α est une formule libre de K, alors $S^I_\alpha(K) = 0$ **(Minimalité)**

• Si $|\mathsf{MI}(K_1 \cup \ldots \cup K_n)| = |\mathsf{MI}(K_1)| + \ldots + |\mathsf{MI}(K_n)|$, alors $S^I_\alpha(K_1 \cup \ldots \cup K_n) = S^I_\alpha(K_1) + \ldots + S^I_\alpha(K_n)$ **(Décomposabilité)**

• Si $M \in \mathsf{MI}(K)$, alors $I(M) = 1$ **(MinInc)**

Proposition 1 ([18]) *Une valeur d'incohérence satisfait Distribution, Symétrie, Minimalité, Décomposabilité et MinInc si et seulement si elle est la valeur d'incohérence de Shapley MI $S^{I_{MI}}_\alpha$.*

De plus, cette valeur est équivalente à celle définie ci-dessous :

Définition 4 ([18]) *MIV_C est définie de la manière suivante :*

$$MIV_C(K, \alpha) = \sum_{\{M \in \mathsf{MI}(K) \ t.q.\ \alpha \in M\}} \frac{1}{|M|}$$

Proposition 2 ([18]) $S^{I_{MI}}_\alpha(K) = MIV_C(K, \alpha)$

Cette deuxième définition montre que cette valeur peut être calculée directement si les sous-ensembles minimaux incohérents de la base de croyances sont connus.

Exemple 1 *Considérons la base $K = \{\varphi_1, \ldots, \varphi_7\}$ avec les formules suivantes :*
– $\varphi_1 = a \wedge b$,
– $\varphi_2 = a \wedge (c \vee d)$,
– $\varphi_3 = a \wedge \neg d$,
– $\varphi_4 = a \wedge \neg c \wedge e$,
– $\varphi_5 = \neg a \wedge \neg b$,
– $\varphi_6 = a \wedge (\neg c \rightarrow \neg e)$,
– $\varphi_7 = a \wedge \neg c \wedge f$.
On a :
– $S^{I_{MI}}_{\varphi_1} = \frac{1}{2}$,
– $S^{I_{MI}}_{\varphi_2} = \frac{7}{6}$,
– $S^{I_{MI}}_{\varphi_3} = \frac{7}{6}$,
– $S^{I_{MI}}_{\varphi_4} = \frac{4}{3}$,
– $S^{I_{MI}}_{\varphi_5} = 3$,
– $S^{I_{MI}}_{\varphi_6} = 1$,
– $S^{I_{MI}}_{\varphi_7} = \frac{5}{6}$.

Notre plate-forme de raisonnement est basée sur ce calcul de la valeur de Shapley $S^{I_{MI}}_\alpha$. Nous utilisons cette valeur pour définir de nouvelles relations d'inférence, opérateurs de révision et de conciliation.

3 Relations d'inférence

Lorsque l'on veut calculer des inférences non-triviales à partir d'une base de croyances propositionnelle, alors il faut soit abandonner la logique classique pour une logique paraconsistante, soit raisonner à partir des sous-ensembles maximaux incohérents. Nous nous intéressons ici au deuxième type de méthodes.

Malheureusement, les possibilités sont peu nombreuses lorsque l'on travaille avec une simple base propositionnelle. Soit $K = \{\varphi_1, \ldots, \varphi_n\}$ une base de croyances, et soit $\mathsf{MC}(K) = \{M_1, \ldots M_k\}$ l'ensembles de sous-ensembles minimaux incohérents de K. Alors les trois possibilités principales sont [10] :

– Sceptique :
  $K \vdash_s \varphi$ si $\forall M \in \mathsf{MC}(K)\ M \vdash \varphi$
– Crédule :
  $K \vdash_c \varphi$ si $\exists M \in \mathsf{MC}(K)\ M \vdash \varphi$
– Argumentative :
  $K \vdash_a \varphi$ si $\exists M \in \mathsf{MC}(K)\ M \vdash \varphi$ et
  $\nexists M \in \mathsf{MC}(K)\ M \vdash \neg\varphi$

L'inférence crédule n'est pas très intéressante parce qu'elle ne garantit pas l'obtention d'une relation d'inférence cohérente : il est possible d'inférer à la fois φ et $\neg\varphi$. Ceci laisse donc deux relations d'inférences possibles : la sceptique et l'argumentative.

Montrons maintenant comment obtenir une famille complète de relations d'inférence pour chaque mesure d'incohérence donnée.

L'idée est d'utiliser une mesure d'incohérence pour ordonner la base, depuis la formule la moins incohérente à la formule la plus incohérente. Autrement dit, nous utilisons la mesure d'incohérence pour transformer la base de croyances propositionnelle (« plate ») en une base stratifiée.

Nous rappelons simplement la définition des relations d'inférence possibiliste, linéaire et préférée (voir [10] pour d'autres relations et pour des explications plus détaillées). Soit $\hat{K} = \langle K_1, \ldots, K_m \rangle$ une base de croyances stratifiée dans laquelle les formules dans la strate K_i sont considérées comme étant plus importantes/fiables/prioritaires que les formules dans les strates K_j avec $j > i$.

- **possibiliste** ($\vdash_\pi$) **:** Définissons $\pi(\hat{K})$ comme $\pi(\hat{K}) = K_1 \cup \ldots \cup K_i$ avec $K_1 \cup \ldots \cup K_i$ cohérent et $K_1 \cup \ldots \cup K_i \cup K_{i+1}$ incohérent.
  $$\hat{K} \vdash_\pi \varphi \text{ si } \pi(\hat{K}) \vdash \varphi$$
- **linéaire** ($\vdash_l$) **:** Définissons $\lambda(\hat{K})$ de manière inductive comme suit : $\lambda(K_1) = K_1$ si K_1 est cohérent, sinon $\lambda(K_1) = \emptyset$. Pour i allant de 2 à m : si $\lambda(\{K_1, \ldots, K_{i-1}\}) \cup K_i$ est cohérent alors $\lambda(\{K_1, \ldots, K_i\}) = \lambda(\{K_1, \ldots, K_{i-1}\}) \cup K_i$, sinon $\lambda(\{K_1, \ldots, K_i\}) = \lambda(\{K_1, \ldots, K_{i-1}\})$.
  $$\hat{K} \vdash_l \varphi \text{ si } \lambda(\hat{K}) \vdash \varphi$$
- **préférée** ($\vdash_p$) **:** Définissons $SMC(\hat{K})$ comme étant l'ensemble des ensembles $A = A_1 \cup \ldots \cup A_m$ pour lesquels $\forall i \in 1...m$ $A_1 \cup \ldots \cup A_i \in \mathsf{MC}(K_1 \cup \ldots \cup K_i)$.
  $$\hat{K} \vdash_p \varphi \text{ si } \forall X \in SMC(\hat{K})\ X \vdash \varphi$$

Nous pouvons maintenant définir formellement nos relations d'inférérence. Utilisons d'abord une valeur d'incohérence pour stratifier la base :

Définition 5 *Soit $K = \{\varphi_1, \ldots, \varphi_n\}$ une base de croyances, et V une valeur d'incohérence, alors la stratification de K sous V est l'ensemble des bases $K^V = \langle K_1, \ldots, K_m \rangle$ avec :*
– $\bigcup K_i = K$
– $K_i \cap K_j = \emptyset\ \forall i, j$
– $\forall \varphi \in K_i, \varphi' \in K_j$, $V(\varphi) \leq V(\varphi')$ iff $i \leq j$

Les différentes strates sont donc les ensembles quotients du pré-ordre total induit par la valeur d'incohérence.

Définition 6 *Soient V une valeur d'incohérence et $\vdash_A$ une relation d'inférence sur les bases stratifiées.*
La relation de (V, A)-inférence $\vdash_A^V$ est définie de la manière suivante : $K \vdash_A^V \varphi$ si $K^V \vdash_A \varphi$.

Ainsi, si la relation d'inférence stratifiée utilisée a de bonnes propriétés logiques, celles-ci se répercutent directement sur la relation de (V, A)-inférence. A partir des résultats montrés dans [10], nous pouvons déduire directement que :

Proposition 3 *Soit V une valeur d'incohérence, alors la relation de (V, π)-inférence, la relation de (V, l)-inférence et la relation de (V, p)-inférence sont des relations d'inférence préférentielles [2].*

Exemple 2 *Reprenons la base de l'exemple 1. La stratification induite de la base est $K^{S^{I_{MI}}} = \langle \{\varphi_1\}, \{\varphi_7\}, \{\varphi_6\}, \{\varphi_2, \varphi_3\}, \{\varphi_4\}, \{\varphi_5\}, \rangle$. Et nous avons par exemple $K \vdash_A^V a$ et $K \vdash_A^V \neg c \wedge \neg e$, alors qu'aucune de ces deux formules ne peut être inférée à partir de l'inférence sceptique ou argumentative.*

4 Opérateurs de révision

La révision de croyances traite le problème de l'intégration d'une nouvelle information dans une base de croyances d'un agent. Cette nouvelle information est souvent en conflit avec des formules de la base de croyances et certaines de ces formules doivent donc être retirées de la base. Une information préférentielle est généralement utilisée pour identifer des priorités entre les formules de la base, exprimant ainsi celles qui devraient de préférence être préservées. Cela peut être encodé par un pré-ordre sur les formules (par exemple les enracinements épistémiques [3]), par un pré-ordre sur les sous-ensembles maximaux cohérents (par exemple les fonctions de contraction partial-meet [4]), par un pré-ordre sur les interprétations (par exemple les assignations fidèles [9]), etc.

Nous proposons ici de définir cette information préférentielle à partir d'une mesure d'incohérence. Cette mesure permet de classer les sous-ensembles maximaux cohérents et de sélectionner les meilleurs d'entre eux.

Définition 7 *Soient $K = \{\varphi_1, \ldots, \varphi_n\}$ une base de croyances, et φ une formule. L'ensemble $K \bot \varphi$ est l'ensemble des ensembles X tels que :*

– $X \subseteq K$
– $X \nvdash \varphi$
– *Il n'existe pas de X' tel que $X \subset X' \subseteq K \cup \{\varphi\}$ et $X' \nvdash \varphi$*

Définissons maintenant le score d'un sous-ensemble maximal cohérent à partir des valeurs d'incohérence des formules le composant.

Définition 8 *Soient $K = \{\varphi_1, \ldots, \varphi_n\}$ une base de croyances et φ une formule. Soit I une valeur d'incohérence. On définit alors le score d'une formule φ_i comme étant sa valeur d'incohérence pour la base $K \cup \{\varphi\}$:*

$$s_I(\varphi_i) = I_{\varphi_i}(K \cup \{\varphi\})$$

Et le score d'un sous-ensemble maximal cohérent $X \in K \perp \varphi$ est le score agrégé de ses formules : soit g une fonction d'agrégation,

$$s_{I,g}(X) = g_{\alpha \in X}(s_I(\alpha))$$

Définition 9 *Soit $S = K \perp \varphi = \{X_1, \ldots, X_k\}$. Une* fonction de sélection *sur S est une fonction γ telle que :*

– *Si $X \perp \varphi$ est un ensemble non-vide, alors $\gamma(S)$ est une sous-ensemble non-vide de S.*
– *Si $X \perp \varphi$ est vide, alors $\gamma(X \perp \varphi)$ est vide*

On définit une fonction de sélection basée sur le score $\gamma_{I,g,f}$ à partir d'une mesure d'incohérence I, d'une fonction d'agrégation g et d'une fonction de sélection f de la manière suivante :

$$\gamma_{I,g,f}(S) = argmin_{X_i \in S} f(s_{I,g}(X_i))$$

On choisira en général min pour la fonction de sélection f, de manière à ne sélectionner que les meilleurs résultats. Cependant, on pourrait vouloir récupérer non seulement les meilleurs MC (c'est-à-dire ceux ayant le score minimum), mais également ceux dont le score est proche du meilleur, par exemple les 50% les meilleurs. C'est pourquoi nous avons choisi de laisser f comme paramètre supplémentaire.

Soit $A = \{A_1, \ldots A_m\}$ un ensemble, alors $A \oplus \alpha$ représente l'ensemble $\{A_1 \cup \{\alpha\}, \ldots A_m \cup \{\alpha\}\}$.

Définition 10 *L'opérateur* MC $\star_{\mathsf{MC}}$ *est défini par :*

$$K \star_{\mathsf{MC}} \varphi = \gamma(K \perp \neg\varphi) \oplus \varphi$$

L'opérateur MC *basé sur le score $\star_{I,g,f}$ est défini par :*

$$K \star_{I,g,f} \varphi = \gamma_{I,g,f}(K \perp \neg\varphi) \oplus \varphi$$

Ceci nous permet de définir le résultat de la révision comme un ensemble de bases de croyances. Il reste alors à choisir une politique d'inférence pour cet ensemble. Dans la suite nous nous intéressons à l'inférence sceptique, mais d'autres politiques peuvent également être utilisées.

Définition 11
$K \star_{\mathsf{MC}} \varphi \vdash \alpha$ si $\forall B \in \gamma(K \perp \neg\varphi) \oplus \varphi$, $B \vdash \alpha$.

Pour la révision de base de croyances (i.e. lorsque la base n'est pas fermée déductivement, au contraire des ensembles de croyances), Hansson [5] définit le résultat de la révision comme la conjonction de l'intersection de tous les ensembles restants avec la nouvelle information ($\cap\gamma(K \perp \neg\varphi) \cup \varphi$). Cependant, cette conjontion retire trop d'informations, comme illustré dans l'exemple suivant. C'est pourquoi nous préférons garder l'ensemble complet des résultats possibles, comme défini précédemment.

Exemple 3 *Considérons la base $K = \{a \wedge c, b \wedge c\}$ et la formule $\varphi = \neg a \vee \neg b$. Alors $K \perp \varphi = \{\{a \wedge c\}, \{b \wedge c\}\}$. Supposons que $\gamma = id$, alors $\cap\gamma(K \perp \neg\varphi) = \emptyset$, donc il n'est pas possible d'inférer c à partir de $K \star \varphi$, alors qu'il est possible d'inférer c de l'ensemble $K \star_{\mathsf{MC}} \varphi = \{\{a \wedge c, \neg a \vee \neg b\}, \{b \wedge c, \neg a \vee \neg b\}\}$.*

Ainsi, bien que cette définition soit plus compliquée, elle nous permet d'obtenir plus d'inférences.

Nous nous intéressons maintenant à l'opérateur $\star_{S^{I_{MI}},max,min}$ qui utilise la valeur de Shapley MI, la fonction max comme fonction d'agrégation (g) et la fonction min comme fonction de sélection (f).

Nous illustrons le comportement de cet opérateur dans l'exemple suivant :

Exemple 4 *Considérons la base $K = \{a \wedge c, b \wedge c, b \wedge d\}$ et la formule $\varphi = \neg a \vee \neg b$. Alors $K \perp \varphi = \{\{a \wedge c\}, \{b \wedge c, b \wedge d\}\}$. Comme $s_{S^{I_{MI}},max}(\{a \wedge c\}) = 1$, et $s_{S^{I_{MI}},max}(\{b \wedge c, b \wedge$*

$d\}) = 0.5$, avec $f = min$ alors seulement $\{b\wedge c, b\wedge d\}$ est sélectionné. Le résultat est donc l'ensemble singleton :

$$K \star_{S^{I_{MI}},max,min} \varphi = \{b \wedge c, b \wedge d, \neg a \vee \neg b\}$$

Traduisons maintenant les propriétés basiques de révision de croyances AGM [4, 3] dans ce cadre.

(K*1) $K * \alpha$ est une théorie

(K*2) $K * \alpha \vdash \alpha$

(K*3) $K * \alpha \subseteq K \cup \{\alpha\}$

(K*4) Si $\neg\alpha \notin K$, alors $K \cup \{\alpha\} \subseteq K * \alpha$

(K*5) $K * \alpha = K_\perp$ si $\vdash \neg\alpha$

(K*6) Si $\vdash \alpha \leftrightarrow \beta$, alors $K * \alpha = K * \beta$

Comme nous travaillons sur une approche syntaxique (non fermée déductivement), alors (K*1) n'est pas satisfaite. Mais toutes les autres propriétés basiques de révision de croyance sont satisfaites.

Proposition 4 *Les opérateurs* MC $\star_{\mathsf{MC}}$ *satisfont (K*2), (K*3), (K*4), (K*5), (K*6).*

5 Opérateurs de conciliation

Les opérateurs de conciliation permettent de résoudre les conflits au sein d'un ensemble de bases de croyances. L'idée est de sélectionner les bases les plus problématiques, de les affaiblir, et d'itérer le processus jusqu'à ce qu'il n'y ait plus de conflit. On commence par définir des modèles de jeu de croyances [20], qui permettent d'obtenir un profil sans conflit. L'opérateur de conciliation correspondant se définit alors comme la conjonction du profil obtenu.

Définition 12 ([20]) *Une* fonction de choix *est une fonction $g : \mathcal{E} \to \mathcal{E}$ telle que :*
- *$g(\Psi) \sqsubseteq \Psi$*
- *Si $\bigwedge \Psi \not\equiv \top$, alors $\exists \varphi \in g(\Psi)$ t.q. $\varphi \not\equiv \top$*
- *Si $\Psi \equiv \Psi'$, alors $g(\Psi) \equiv g(\Psi')$*

Définition 13 ([20]) *Une* fonction d'affaiblissement *est une fonction $\blacktriangledown : \mathcal{L} \to \mathcal{L}$ telle que :*
- *$\varphi \vdash \blacktriangledown(\varphi)$*
- *Si $\varphi \equiv \blacktriangledown(\varphi)$, alors $\varphi \equiv \top$*
- *Si $\varphi \equiv \varphi'$, alors $\blacktriangledown(\varphi) \equiv \blacktriangledown(\varphi')$*

Définition 14 ([20]) *La solution à un profil de croyances Ψ pour un Modèle de Jeu de Croyances $\mathcal{N} = \langle g, \blacktriangledown \rangle$ sous les contraintes d'intégrité μ, est le profil de croyances $\Psi^\mu_\mathcal{N}$ défini par :*
- *$\Psi_0 = \Psi$*
- *$\Psi_{i+1} = \blacktriangledown_{g(\Psi_i)}(\Psi_i)$*
- *$\Psi^\mu_\mathcal{N}$ est le premier Ψ_i cohérent avec μ*

L'opérateur de conciliation $\blacktriangle_\mathcal{N}$ est défini par $\Psi \blacktriangle_\mathcal{N} \mu = \bigwedge \Psi^\mu_\mathcal{N}$

Définition 15 ([20]) *Soit φ une base de croyances.*
- *La* fonction d'affaiblissement drastique *oublie toutes les informations de l'agent, i.e : $\forall \varphi\ \blacktriangledown_\top(\varphi) = \top$.*
- *La* fonction d'affaiblissement de dilatation *est définie par : $mod(\blacktriangledown_\delta(\varphi)) = \{\omega \in \mathcal{W} \mid \exists \omega' \models \varphi\ d_H(\omega, \omega') \leq 1\}$ où d_H est la distance de Hamming entre les interprétations* [5].

Dans ce travail, nous utilisons la valeur d'incohérence pour définir les agents les plus conflictueux (i.e. la fonction de sélection).

Définition 16 *Un* Modèle de Jeu de Croyances de Shapley *est un Modèle de Jeu de Croyances $\mathcal{N} = \langle S_I, \blacktriangledown \rangle$ dans lequel S_I est une Valeur d'Incohérence de Shapley (SIV).*

La solution d'un profil de croyances Ψ pour un Modèle de Jeu de Croyances de Shapley $\mathcal{N} = \langle S_I, \blacktriangledown \rangle$ sous les contraintes d'intégrité μ est le profil de croyances $\Psi^\mu_\mathcal{N}$ défini par :
- *$\Psi_0 = \Psi$*
- *$\Psi_{i+1} = \blacktriangledown_{argmax_{\varphi_j \in \Psi_i}(S_{I_{\varphi_j}}(\Psi_i))}(\Psi_i)$*
- *$\Psi^\mu_\mathcal{N}$ est le premier Ψ_i cohérent avec μ*

Exemple 5 *Considérons le Modèle de Jeu de Croyances de Shapley suivant $\mathcal{N} = \langle S_{I_{LP_m}}, \blacktriangledown_\delta \rangle$. Supposons qu'il y ait sept agents définis par les bases de croyances suivantes :*
- *$\varphi_1 = a \wedge b$,*
- *$\varphi_2 = a \wedge (c \vee d)$,*
- *$\varphi_3 = a \wedge \neg d$,*
- *$\varphi_4 = a \wedge \neg c \wedge e$,*
- *$\varphi_5 = \neg a \wedge \neg b$,*
- *$\varphi_6 = a \wedge (\neg c \to \neg e)$,*
- *$\varphi_7 = a \wedge \neg c \wedge f$.*

5. Soient ω et ω' deux interprétations, alors $d_H(\omega, \omega') = |\{a \in \mathcal{P} \mid \omega(a) \neq \omega'(a)\}|$.

Nous considérons ici qu'il n'y a pas de contraintes d'intégrité ($\mu = \top$). On a $S^{I_{MI}}_{\varphi_1} = \frac{1}{2}$, $S^{I_{MI}}_{\varphi_2} = \frac{7}{6}$, $S^{I_{MI}}_{\varphi_3} = \frac{7}{6}$, $S^{I_{MI}}_{\varphi_4} = \frac{4}{3}$, $S^{I_{MI}}_{\varphi_5} = 3$, $S^{I_{MI}}_{\varphi_6} = 1$, $S^{I_{MI}}_{\varphi_2} = \frac{5}{6}$.

La valeur maximale est 3, ce qui signifie que φ_5 est l'agent qui apporte le plus de conflits. Il est donc sélectionné par la fonction de choix pour être affaibli. Comme nous utilisons la fonction d'affaiblissement drastique, φ_5 est alors remplacé par $\top$.

Le profil obtenu n'étant pas cohérent, nous itérons le processus. Le calcul des valeurs d'incohérence montre que l'agent φ_4 est le plus conflictuel, et φ_4 est donc affaibli à $\top$.

Le profil obtenu n'est toujours pas cohérent, un troisième tour est donc nécessaire. Cette fois, ce sont φ_2, φ_3 et φ_7 qui sont sélectionnés et affaiblis. Le profil (cohérent) résultant pour le processus complet est donc

$$\Psi^{\top}_{\mathcal{N}} = \{\{a \wedge b\}, \top, \top, \top, \top, a \wedge (\neg c \rightarrow \neg e), \top\}$$

On a alors $\Psi \blacktriangle_{\mathcal{N}} \top \equiv a \wedge b \wedge (\neg c \rightarrow \neg e)$.

6 Description de PRISM

Dans cette section, nous décrivons la plateforme PRISM (Platform for Reasoning with Inconsistency Shapley Measure) que nous avons développée afin de tester les différents opérateurs présentés dans les parties précédentes. Cette plate-forme peut-être utilisée pour construire une base de croyances et effectuer différentes tâches de raisonnement comme l'inférence, la révision et la conciliation. PRISM peut être téléchargée depuis la page web `http://www.cril.univ-artois.fr/prism`. Cette page contient également la documentation de la plate-forme ainsi qu'un guide d'utilisation de celle-ci.

Dans la suite, nous présentons les caractéristiques de PRISM ainsi que différents détails d'implémentation.

6.1 Caractéristiques

La plate-forme est une application Java. L'interface utilisateur est divisée en 5 onglets principaux :

1. *Base* - permet à l'utilisateur de créer une base de formules
2. *Shapley* - permet de calculer les valeurs d'incohérence de Shapley de chacune des formules de la base
3. *Inference* - permet de raisonner étant donnée une relation d'inférence
4. *Revision* - permet de raisonner sur une base qui a été révisée par une formule
5. *Conciliation* - permet de calculer un profil cohérent à partir d'un ensemble conflictuel de bases.

Tous les onglets sont structurés de la même manière. La base de croyances courante est affichée dans la partie supérieure gauche de la fenêtre. En fonction de la tâche qui doit être effectuée, les options pour les différents opérateurs apparaissent dans la partie supérieure droite. La partie inférieure de la fenêtre est réservée pour l'affichage des résultats des différents calculs.

Onglet "Base". La base de croyances est composée de plusieurs formules. Ces formules peuvent être chargées à partir d'un fichier ou écrites directement par l'utilisateur. Ne sont acceptées que les formules ayant la syntaxe suivante :

φ := (φ) ; φ&φ ; φ|φ ; φ->φ ; φ<->φ ; lit

lit := v ; $\sim v$ *(où v est un nom de variable)*

Chaque formule doit se terminer par un point virgule " ;".

Presque toutes les chaînes alphanumériques conviennent pour désigner les variables (voir la documentation en ligne pour les exceptions).

Les bases créées ou modifiées par l'utilisateur peuvent être sauvegardées. Les formules peuvent être affichées dans leur forme CNF et peuvent être ajoutées, modifiées ou retirées de la base.

Il est également possible de regrouper les formules. Dans ce cas, la notion de groupe peut par exemple représenter un agent : les formules du groupe sont alors les formules de la base de cet agent. En pratique un groupe est représenté par un entier. On distingue un groupe spécifique identifié par l'entier 0 : il s'agit du groupe de contraintes d'intégrité. Les formules de ce groupe sont des contraintes, c'est-à-dire qu'elles ne peuvent pas être falsifiées. Ces formules spécifiques peuvent représenter des connaissances de l'environnement pour les tâches d'inférence ou de révision ou également des contraintes d'intégrité pour la tâche de conciliation. L'utilisateur peut choisir de prendre en compte ou non les groupes de formules.

Décrivons maintenant les autres onglets de la plate-forme. Il est à noter que chacun de ces onglets contient des opérateurs par défaut mais qu'il est possible de définir et ajouter son propre opérateur (voir la table 1).

Onglet "Shapley". Une fois qu'une base de formules a été construite, il est possible de calculer les valeur d'incohérence de Shapley des formules le composant. Cet onglet présente les résultats de ce calcul. Plus précisément, les sous-ensembles minimaux incohérents MI sont affichés dans la partie inférieure gauche de la fenêtre. Les valeurs d'incohérence de Shapey sont quant à elles affichées sur la partie inférieure droite. Si l'utilisateur a choisi de prendre en compte la classification par groupe, alors les mesures d'incohérence de Shapley pour chacun de ces groupes sont également affichées. Notons que l'opérateur d'agrégation pour la mesure d'incohérence peut être choisi dans la partie supérieure droite de la fenêtre. Par défaut, deux opérateurs sont disponibles : *Mean* (Moyenne) et *Max*.

Onglet "Inference". Les valeurs d'incohérence de Shapley peuvent être utilisées pour stratifier les bases de croyances initialement plates : plus la valeur d'une formule est petite, plus la formule est prioritaire dans la base stratifiée. La stratification de la base est affichée sur l'onglet *Inference*. L'utilisateur peut choisir une politique d'inférence dans la liste (partie supérieure droite) et demander si une formule peut être inférée de la base avec l'opérateur choisi. La formule en question doit avoir la même syntaxe que celle présentée précédemment pour la base de croyances.

Les opérateurs d'inférence par défaut sont *Possibilistic* (possibiliste) et *Linear* (linéaire) et correspondent à ceux présentés en section 3.

Onglet "Revision". Sur cet onglet, l'utilisateur peut réviser une base de croyances par une formule. La formule est rentrée dans le champ de texte juste sous la table dans laquelle la base courante est affichée. La syntaxe est toujours la même. Comme présenté dans la section 4, la révision implique différents opérateurs/fonctions : une fonction d'agrégation, une fonction de sélection et une politique d'inférence (cf la table 1 pour les détails sur les classes). Les fonctions d'agrégation disponibles par défaut sont : *Max*, *Min* et *Sum*. La plate-forme propose trois fonctions de sélection : *No Selection*, *All Min Selection* et *One of the min selection*. La dernière choisit arbitrairement un MC parmi les meilleurs. Finalement, le dernier élément à choisir est la politique d'inférence. L'utilisateur a le choix entre les inférences *Skeptical* (sceptique) et *Credulous* (crédule). Tous les sous-ensembles maximaux cohérents MC sont affichés avec leur scores respectifs dans la partie inférieure gauche de la fenêtre. La table de la partie inférieure droite montre les sous-ensembles sélectionnés par la fonction de sélection. Comme dans l'onglet de révision, il est possible de demander si une formule peut être inférée, étant donnée une politique d'inférence.

Onglet "Conciliation". Cet onglet permet d'effectuer la concilation (modèle de jeu de croyance de Shapley) tel qu'il a été décrit dans la section 5. Ce jeu est basé sur deux fonctions : une fonction de choix et une fonction d'affaiblissement. La première fonction de choix disponible dans la plate-forme s'appelle *Shapley Choice* et correspond à $S_{I_{MI}}$. La seconde, *Weak Shapley Choice*, choisit arbitrairement une formule parmi celles sélectionnées par $S_{I_{MI}}$. Ceci permet de réduire le nombre de formules affaiblies. Une seule fonction d'affaiblissement est disponible par défaut, *Drastic*, et correspond à l'opérateur $\blacktriangledown_{\top}$ défini en section 5.

6.2 Détails sur l'implémentation

PRISM est une plate-forme évolutive, c'est-à-dire que pour toutes les tâches de raisonnement, il est possible d'ajouter sa propre implémentation d'un opérateur. Pour faire cet ajout, il suffit de créer une classe étendant une classe spécifique et d'ajouter cette classe au classpath [6]. La table 1 indique pour chaque opérateur la classe spécifique qu'il faut étendre ainsi que la méthode principale à implémenter.

Pour calculer la valeur d'incohérence de Shapley, nous avons choisi de passer par le calcul de tous les sous-ensembles incohérents minimaux MI de la base de croyances. Dans le cas général, calculer MI est un problème difficile. En effet, le nombre de MI peut être exponentiel : une

6. Les détails sur l'ajout d'un nouvel opérateur peuvent être trouvés sur la documentation en ligne de la plate-forme à l'adresse `http://www.cril.univ-artois.fr/prism`

Onglet	Classe Abstraite	Méthode
Shapley	ShapleyValueSet	computeShapleyValue(List<Formula> l) : **double**
Inference	InferenceOperator	isAFormulaEntailed(List<Formula> b, Formula f) : **boolean**
Revision	MssScoreAggregation	computeMSSScore(MSS m) : **double**
	MssSelectionOperator	selectMss(List<MSS> l) : List<MSS>
	InferenceFromMSSOperator	isAFormulaEntailed(List<MSS> l, Formula f) : **boolean**
Conciliation	ChoiceOperator	chooseFormulae(List<Formula> l) : List<Formula>
	WeakOperator	weakFormula(Formula f) : Formula

TABLE 1 – Classes à étendre et méthodes à implémenter pour ajouter de nouveaux opérateurs

instance SAT de n clauses peut correspondre à $C_n^{n/2}$ MI dans le pire des cas et vérifier si une formule donnée appartient à l'ensemble des MI est dans Σ_2^p [21]. Notre problème ici est encore plus difficile car nous avons besoin de calculer *tous* les MI et ensuite vérifier si les formules appartiennent à ces sous-ensembles.

Il existe plusieurs approches pour extraire un MI d'un ensemble de clauses ([22, 23, 24], etc.) ou pour calculer des couvertures des MI ([25]). Il nous faut ici une approche complète, i.e. une qui extrait tous les MI. Les outils candidats sont peu nombreux mais l'on peut citer CAMUS [26] et HYCAM [27] par exemple.

Dans ce travail, nous avons choisi d'effectuer l'extraction de MI avec un solveur SAT de la librairie Sat4j [28]. La principale justification de ce choix est notre volonté de développer une plate-forme portable (en terme de système d'exploitation). Utiliser des technologies basées sur Java nous permet de respecter cette propriété de portabilité, même si nous ne profitons pas des dernières avancées en terme d'extraction de tous les MI (les outils dédiés présentés plus hauts sont développés en C ou C++). Dans les versions futures de PRISM, une détection du système d'exploitation sera effectuée pour permettre l'utilisation de CAMUS ou de HYCAM si possible.

Le solveur Sat4j (version 2.3.3) extrait tous les MI dans une méthode en deux étapes [29] : on commence par calculer tous les MC, puis les MI sont extraits dans un deuxième temps.

Pour tous les extracteurs de MI cités précédemment, l'entrée est une formule CNF. Notre plate-forme permet à l'utilisateur de peupler la base avec des formules générales. Cela signifie qu'il faut transformer les formules de la base en formules CNF équisatisfiables. Pour effectuer cette transformation, nous utilisons l'encodage de Tseitin ([30]). Cet encodage permet d'obtenir une formule dont la taille augmente linéairement par rapport à la taille de la formule originelle. Il faut par contre ajouter de nouvelles variables [7].

Pour une formule donnée, les clauses de sa forme CNF sont groupées lorsqu'elles sont transmises au solveur. Cela nous permet de garantir l'équivalence entre le calcul sur les formules de base et sur leurs formes CNF. Notons que le solveur doit pouvoir prendre en compte la notion de groupe.

Afin de modéliser les formules générales, nous utilisons un langage de domaine spécifique (dsl) écrit en Scala. Cela nous permet d'avoir un parsing et une transformation en CNF très efficace. De plus, comme Scala est basé sur la JVM (Java Virtual Machine), nous préservons la portabilité de la plate-forme.

7 Conclusion et perspectives

Nous avons présenté PRISM, une plate-forme évolutive basée sur la valeur d'incohérence de Shapley MI pour effectuer différentes tâches de raisonnement telles que l'inférence, la révision et la conciliation. Des opérateurs sont proposés par défaut pour chacune des opérations mais il est possible d'écrire et d'ajouter sa propre implémentation d'un ou plusieurs opérateurs pour les tester. La plate-forme est écrite en Java et utilise des outils en Java, ce qui permet de rester indépendant au système d'exploitation utilisé.

Par la suite, en plus de développer de nouveaux opérateurs, nous souhaitons proposer une partie de cette plate-forme sous forme de librairie Java. Une telle librairie contiendrait les méthodes pour utiliser la valeur de Shapley MI et les opérateurs associés, ce qui permettrait d'utiliser facilement ces opérateurs pour des applications diverses.

7. Les nouvelles variables ont les noms réservés "_nv#*i*".

Références

[1] Makinson, D. : General Pattern in nonmonotonic reasoning. In : Handbook of Logic in Artificial Intelligence and Logic Programming. Volume III. Clarendon Press, Oxford (1994) 35–110

[2] Kraus, S., Lehmann, D., Magidor, M. : Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence **44** (1990) 167–207

[3] Gärdenfors, P. : Knowledge in flux. MIT Press (1988)

[4] Alchourrón, C.E., Gärdenfors, P., Makinson, D. : On the logic of theory change : Partial meet contraction and revision functions. Journal of Symbolic Logic **50** (1985) 510–530

[5] Hansson, S.O. : A Textbook of Belief Dynamics : Theory Change and Database Updating. Kluwer (1999)

[6] Revesz, P.Z. : On the semantics of arbitration. International Journal of Algebra and Computation **7** (1997) 133–160

[7] Konieczny, S., Pino Pérez, R. : Merging information under constraints : a qualitative framework. Journal of Logic and Computation **12** (2002) 773–808

[8] Konieczny, S., Pino Pérez, R. : Logic based merging. Journal of Philosophical Logic **40** (2011) 239–270

[9] Katsuno, H., Mendelzon, A.O. : Propositional knowledge base revision and minimal change. Artificial Intelligence **52** (1991) 263–294

[10] Benferhat, S., Dubois, D., Prade, H. : Some syntactic approaches to the handling of inconsistent knowledge bases : A comparative study. part ii : The prioritized case. In Orlowska, E., ed. : Logic at work. Volume 24 of Physica-Verlag., Heidelberg (1998) 473–511

[11] Egly, U., Eiter, T., Tompits, H., Woltran, S. : Solving advanced reasoning tasks using quantified boolean formulas. In : Proceedings of AAAI'00. (2000) 417–422

[12] Williams, M.A., Sims, A. : Saten : An object-oriented web-based revision and extraction engine. CoRR **cs.AI/0003059** (2000)

[13] (http ://magic.it.uts.edu.au/systems/saten.html)

[14] Spohn, W. : Ordinal conditional functions : a dynamic theory of epistemic states. In Harper, W.L., Skyrms, B., eds. : Causation in Decision, Belief Change, and Statistics. Volume 2. (1987) 105–134

[15] Delgrande, J.P., Liu, D.H., Schaub, T., Thiele, S. : COBA 2.0 : A consistency-based belief change system. In : Proceedings of ECSQARU'07. (2007) 78–90

[16] (http ://www.cs.sfu.ca/ cl/software/COBA/-coba2.html)

[17] Delgrande, J.P., Schaub, T. : A consistency-based approach for belief change. Artificial Intelligence **151** (2003) 1–41

[18] Hunter, A., Konieczny, S. : On the measure of conflicts : Shapley inconsistency values. Artificial Intelligence **174** (2010) 1007–1026

[19] Hunter, A., Konieczny, S. : Approaches to measuring inconsistent information. In : Inconsistency tolerance, Springer LNCS 3300 (2005) 189–234

[20] Konieczny, S. : Belief base merging as a game. Journal of Applied Non-Classical Logics **14** (2004) 275–294

[21] Eiter, T., Gottlob, G. : On the complexity of propositional knowledge base revision, updates, and counterfactuals. Artificial Intelligence **57** (1992) 227–270

[22] Belov, A., Lynce, I., Marques-Silva, J. : Towards efficient mus extraction. AI Commun. **25** (2012) 97–116

[23] Bruni, R. : On exact selection of minimally unsatisfiable subformulae. Annals of Mathematics and Artificial Intelligence **43** (2005) 35–50

[24] Dershowitz, N., Hanna, Z., Nadel, E. : A scalable algorithm for minimal unsatisfiable core extraction. In : Proceedings of SAT'06, Springer (2006)

[25] Grégoire, E., Mazure, B., Piette, C. : Tracking muses and strict inconsistent covers. In : Proceedings of FMCAD'06, San Jose (USA) (2006) 39–46

[26] Liffiton, M., Sakallah, K. : Algorithms for computing minimal unsatisfiable subsets of constraints. Journal of Automated Reasoning **40** (2008) 1–33

[27] Grégoire, E., Mazure, B., Piette, C. : Using local search to find msses and muses. European Journal of Operational Research **199** (2009) 640–646

[28] Le Berre, D., Parrain, A. : The sat4j library, release 2.2. JSAT **7** (2010) 59–6

[29] Castell, T., Cayrol, C., Cayrol, M., Berre, D.L. : Using the davis and putnam procedure for an efficient computation of preferred models. In : Proceedings of ECAI'96. (1996) 350–354

[30] Tseitin, G.S. : On the complexity of derivations in the propositional calculus. Studies in Mathematics and Mathematical Logic **Part II** (1968) 115–125