A reasoning platform based on the MI Shapley inconsistency value

Sébastien Konieczny and Stéphanie Roussel

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

Abstract. In this paper we show how to build a reasoning platform using an inconsistency value. The idea is to use an inconsistency value for evaluating how much each formula of the belief base is responsible of the inconsistency of the base. Then this evaluation allows us to obtain a stratification (total pre-order) of the base, that can be used as the preferential input for different reasoning tasks, such as inference, belief revision, or conciliation. We show that the obtained operators are interesting and have good logical properties. We use as inconsistency value, the MI Shapley inconsistency value, that is known to have good properties, and that can be computed from minimal inconsistent subsets. We developed a java-based platform, that use the Sat4j library for computing the minimal inconsistent subsets, and that allows to have an effective way to compute the MI Shapley inconsistent subsets. We implemented also several inference, revision and conciliation methods, that use this inconsistency value. So this provides a complete reasoning platform, that can be used for instance for academic purposes.

1   Introduction

Belief change and reasoning under inconsistency are two topics that have received considerable attention. There are a lot of theoretical results on these reasoning methods, such as logical characterizations for non-monotonic inference [1,2], belief revision [3,4,5], belief merging [6,7,8], etc. There are also numerous particular methods that have been proposed for belief revision [9], belief merging [8], inference under inconsistency [10], etc.

   In contrast, there are very few proposed implemented approaches. Although these implementations can be useful to test the proposed operators, to experiment the different reasoning method, and to disseminate these operators more widely in the AI community. In fact we are aware of only *two*[1] reasoning platforms, that implement several reasoning methods. The first one is the SATEN platform [12,13], developed by Williams and Sims, that allows to perform theory extraction, iterated belief revision, non-monotonic reasoning, possibilistic reasoning and hypothetical reasoning. This platform is written in Java 1.1 and is based on a theorem prover. It basically uses the Spohnian representation of epistemic states [14]. The second

---

[1] We can mention also the QUIP project [11], but there is not yet, as far as we know, a corresponding available platform.

one is the COBA platform [15,16], developed by Delgrande, Liu, Schaub and Thiele, that performs belief revisions and contractions, based on the langage projection approaches developed in [17]. COBA is a Java applet that uses a SAT-solver.

In this work we propose such a reasoning platform. The whole platform is based on the effective computation of a given inconsistency value, namely the MI Shapley inconsistency value $S^{I_{MI}}$ [18], that can be computed easily from the minimal inconsistent subsets of a belief base. In addition to the measure of inconsistency of the formulae of the base and of the whole base given directly by this computation, we use the obtained stratification of the base (a total pre-order) as the preferential input for different reasoning tasks, such as inference, belief revision, or conciliation.

The paper is organized as follows. Section 2 presents preliminary definitions and inconsistency measures and values. In sections 3, 4 and 5, we formally study three reasoning operations that are respectively inference, revision and conciliation. Section 6 is dedicated to the platform description. We finally conclude on future works in section 7.

2 Preliminaries - Inconsistency Measures & Values

We consider a propositional language $\mathcal{L}$ built from a finite set of propositional symbols $\mathcal{P}$. A *belief base* K is a finite set of propositional formulae. Let us note $\mathcal{K}_{\mathcal{L}}$ the set of belief bases definable from formulae of the language $\mathcal{L}$. If a belief base K is not consistent, then one can define the minimal inconsistent subsets [2] of K as: $\mathsf{MI}(K) = \{K' \subseteq K \mid K' \vdash \bot \text{ and } \forall K'' \subset K', K'' \nvdash \bot\}$.

The notion of maximal consistent subset[3] is the dual of that of minimal inconsistent subset. Each maximal consistent subset represents a maximal (regarding set inclusion) subset of the base that is consistent:

$\mathsf{MC}(K) = \{K' \subseteq K \mid K' \nvdash \bot \text{ and } \forall K'' \text{ s. t. } K' \subset K'', K'' \vdash \bot\}$

A profile Ψ is a vector of belief bases $\langle K_1, \ldots, K_n \rangle$. The set of all profiles is denoted $\mathcal{E}$. $\bigwedge \Psi$ denotes the conjunction of the elements of Ψ.

Recently some works have started to study how to measure the inconsistency in a propositional belief base (see e.g. [19]). There are several sensible ways to do that. This is not surprising since it parallels the fact that there are several sensible ways to define non-trivial inference relations from inconsistent bases.

In [18] a distinction has been made between *inconsistency measures*, that measure the inconsistency of a belief base, and *inconsistency values*, that measure the (responsibility for) inconsistency of each formula of a belief base.

Of course the inconsistency values, which work formula-by-formula, can be used to define corresponding inconsistency measures, just by aggregating the obtained inconsistency values.

Let us recall the definition of the Shapley Inconsistency Values (SIV) [18]:

Definition 1 ([18]). *An inconsistency measure I is called a* basic inconsistency measure *if it satisfies the following properties*[4]*, $\forall K, K' \in \mathcal{K}_{\mathcal{L}}$, $\forall \alpha, \beta \in \mathcal{L}$:*

---

[2] also called Minimally Unsatisfiable Subsets - MUS
[3] also called Maximally Satisfiable Subsets - MSS
[4] In [18] an additional **Dominance** property is also asked.

- $I(K) = 0$ *iff K is consistent*   **(Consistency)**
- $I(K \cup K') \geq I(K)$ **(Monotony)**
- *If α is a free formula of K, then $I(K) = I(K \setminus \{\alpha\})$*
                                                      (Free Formula Independence)

Now we are able to define the Shapley inconsistency value.

Definition 2 ([18]). *Let I be a basic inconsistency measure. We define the corresponding* Shapley Inconsistency Value (SIV), *noted S^I, as the Shapley value of the coalitional game defined by the function I, i.e. let $\alpha \in K$:*

$$S_\alpha^I(K) = \sum_{C \subseteq K} \frac{(c-1)!(n-c)!}{n!}(I(C) - I(C \setminus \{\alpha\}))$$

where n is the cardinality of K and c is the cardinality of C.

From this value, one can define an inconsistency value for the whole belief base as in the next definition which essentially says that a base is as bad as its worst element.

Definition 3 ([18]). *Let K be a belief base, $\hat{S}^I(K) = \max_{\alpha \in K} S_\alpha^I(K)$*

As examples of simple basic inconsistency measures, one can consider the drastic inconsistency value[5], that is the simplest inconsistency measure one can define, and that is not really interesting by itself. But the corresponding SIV is interesting.

Another example of basic inconsistency measures is the one that counts the conflicts of a base using the number of minimal inconsistent subsets: $I_{MI}(K) = |\mathsf{MI}(K)|$.

The corresponding SIV is interesting, and has been logically characterized [18]. Let us first define these properties on inconsistency values: assume a given basic inconsistency measure I, and the corresponding Shapley inconsistency value S^I :

- $\sum_{\alpha \in K} S_\alpha^I(K) = I(K)$ **(Distribution)**
- If $\alpha, \beta \in K$ are such that for all $K' \subseteq K$ s.t. $\alpha, \beta \notin K'$,
  $I(K' \cup \{\alpha\}) = I(K' \cup \{\beta\})$, then $S_\alpha^I(K) = S_\beta^I(K)$ **(Symmetry)**
- If α is a free formula of K, then $S_\alpha^I(K) = 0$ **(Minimality)**
- If $|\mathsf{MI}(K_1 \cup \ldots \cup K_n)| = |\mathsf{MI}(K_1)| + \ldots + |\mathsf{MI}(K_n)|$,
  then $S_\alpha^I(K_1 \cup \ldots \cup K_n) = S_\alpha^I(K_1) + \ldots + S_\alpha^I(K_n)$ **(Decomposability)**
- If $M \in \mathsf{MI}(K)$, then $I(M) = 1$ **(MinInc)**

Proposition 1 ([18]). *An inconsistency value satisfies Distribution, Symmetry, Minimality, Decomposability and MinInc if and only if it is the MI Shapley Inconsistency Value $S_\alpha^{I_{MI}}$.*

---

[5] $I_d(K) = 0$ if K is consistent, and $I_d(K) = 1$ otherwise.

Second, this value is equivalent to the following one:

Definition 4 ([18]). *MIV_C is defined as follows:*

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

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

This alternative definition shows that this value can be computed directly if one knows the minimal inconsistent subsets of a belief base.

Example 1. Consider the base $K = \{\varphi_1, \ldots, \varphi_7\}$ with the following formulae $\varphi_1 = a \wedge b$, $\varphi_2 = a \wedge (c \vee d)$, $\varphi_3 = a \wedge \neg d$, $\varphi_4 = a \wedge \neg c \wedge e$, $\varphi_5 = \neg a \wedge \neg b$, $\varphi_6 = a \wedge (\neg c \to \neg e)$, $\varphi_7 = a \wedge \neg c \wedge f$. We have $S_{\varphi_1}^{I_{MI}} = \frac{1}{2}$, $S_{\varphi_2}^{I_{MI}} = \frac{7}{6}$, $S_{\varphi_3}^{I_{MI}} = \frac{7}{6}$, $S_{\varphi_4}^{I_{MI}} = \frac{4}{3}$, $S_{\varphi_5}^{I_{MI}} = 3$, $S_{\varphi_6}^{I_{MI}} = 1$, $S_{\varphi_2}^{I_{MI}} = \frac{5}{6}$.

Our reasoning platform is based on this computation of the $S_\alpha^{I_{MI}}$ Shapley value. We use this value for defining new inference relations, revision operators, and conciliation operators.

3   Inference relations

When one wants to draw non-trivial inferences from an inconsistent propositional belief base then it has either to leave classical logic for choosing a paraconsistent logic, or to reason from the maximal consistent subsets of the base. We will focus on this last class of methods.

Unfortunately, there are not a lot of possibilities when the input is a simple propositional belief base. Let $K = \{\varphi_1, \ldots, \varphi_n\}$ be a belief base, and let $\mathsf{MC}(K) = \{M_1, \ldots M_k\}$ be the set of maximal consistent subsets of K. Then the three main possibilities are [10]:

- Skeptical: $K \vdash_s \varphi$ if $\forall M \in \mathsf{MC}(K) \ M \vdash \varphi$
- Credulous: $K \vdash_s \varphi$ if $\exists M \in \mathsf{MC}(K) \ M \vdash \varphi$
- Argumentative: $K \vdash_a \varphi$ if $\exists M \in \mathsf{MC}(K) \ M \vdash \varphi$ and $\nexists M \in \mathsf{MC}(K) \ M \vdash \neg \varphi$

The credulous inference is not that interesting, in particular it does not guarantee to obtain a consistent inference relation, in the sense that it is possible to obtain both φ and $\neg \varphi$ as result. So this leaves only two different possible inference relations: skeptical and argumentative.

Let us now show how to obtain a whole family of inference relation for each given inconsistency measure. The idea is to use the inconsistency measure to order the base, from the least inconsistent formulae to the most inconsistent one. This means that we use the inconsistency measure to transform this flat propositional belief base into a stratified one. Then we can use any of the defined inference relations on stratified bases. We recall just the definition of the possibilisitic, linear and preferred inference relations here, see [10] for other ones and explanations. Consider a stratified belief base $\hat{K} = \langle K_1, \ldots, K_m \rangle$, where formulae in the stratum K_i are considered as more important/reliable/prioritary than the formulae in strata K_j with $j > i$.

- **possibilistic.** Define $\pi(\hat{K})$ as $\pi(\hat{K}) = K_1 \cup \ldots \cup K_i$ with $K_1 \cup \ldots \cup K_i$ consistent and $K_1 \cup \ldots \cup K_i \cup K_{i+1}$ inconsistent. $\hat{K} \vdash_\pi \varphi$ if $\pi(\hat{K}) \vdash \varphi$
- **linear.** Define $\lambda(\hat{K})$ inductively as: $\lambda(K_1) = K_1$ if K_1 is consistent, otherwise $\lambda(K_1) = \emptyset$. For i from 2 to m: if $\lambda(\{K_1, \ldots, K_{i-1}\}) \cup K_i$ is consistent then $\lambda(\{K_1, \ldots, K_i\}) = \lambda(\{K_1, \ldots, K_{i-1}\}) \cup K_i$, otherwise $\lambda(\{K_1, \ldots, K_i\}) = \lambda(\{K_1, \ldots, K_{i-1}\})$. $\hat{K} \vdash_l \varphi$ if $\lambda(\hat{K}) \vdash \varphi$
- **preferred.** Define $SMC(\hat{K})$ as the set of sets $A = A_1 \cup \ldots \cup A_m$ where $\forall i \in 1...m$ $A_1 \cup \ldots \cup A_i \in \mathsf{MC}(K_1 \cup \ldots \cup K_i)$. $\hat{K} \vdash_p \varphi$ if $\forall X \in SMC(\hat{K})$ $X \vdash \varphi$

So, let us now define formally our inference relations. First let us use an inconsistency value to stratify the base:

Definition 5. *Let* $K = \{\varphi_1, \ldots, \varphi_n\}$ *be a belief base, and* V *be an inconsistency value, then the stratification of* K *under* V *is the set of bases* $K^V = \langle K_1, \ldots, K_m \rangle$ *where* $- \bigcup K_i = K$
  $- K_i \cap K_j = \emptyset \; \forall i, j$
  $- \forall \varphi \in K_i, \varphi' \in K_j, \; V(\varphi) \leq V(\varphi') \; \text{iff} \; i \leq j$

Definition 6. *Let* V *be an inconsistency value, and* $\vdash_A$ *be an inference relation on stratified bases. The* (V, A)*-inference relation* $\vdash_A^V$ *is defined as* $K \vdash_A^V \varphi$ *if* $K^V \vdash_A \varphi$.

So, if the stratified inference relation that is used has good logical properties, it straightforwardly gives good properties to our (V, A)-inference relation. So as a consequence of results shown in [10] we know that:

Proposition 3. *Let* V *be any inconsistency value, then the* (V, π)*-inference relation, the* (V, l)*-inference relation, and the* (V, p)*-inference relation are preferential inference relations [2].*

Example 2. Consider the base of Example 1. The induced stratification of the base is $K^{S^{IMI}} = \langle \{\varphi_1\}, \{\varphi_7\}, \{\varphi_6\}, \{\varphi_2, \varphi_3\}, \{\varphi_4\}, \{\varphi_5\}, \rangle$. And for instance we have $K \vdash_A^V a$ and $K \vdash_A^V \neg c \wedge \neg e$, whereas none of them can be inferred from the skeptical or the argumentative inference.

4 Revision operators

Belief revision [3,4,9] aims at incorporating a new piece of information into the belief base of an agent. Often, this new piece of information conflicts with some formulae of the belief base, so some of these formulae have to be removed from the base. One usually uses some preferential information for identifying priorities between formulae that can be preferably preserved. This can be encoded by a pre-order on formulae (such as in epistemic entrenchments [3]), by a pre-order between maximal consistent subsets (such as partial meet contraction functions [4]), by a pre-order between interpretations (such as in faithful assignments [9]), etc.

We propose to define this preferential information from an inconsistency measure. This measure is used to rank the maximal consistent subsets, and to select the best of them.

Definition 7. *Let* $K = \{\varphi_1, \ldots, \varphi_n\}$ *be a belief base, and* φ *be a formula. The set* $K \bot \varphi$ *is the set of sets* X *such that:*

- $X \subseteq K$
- $X \nvdash \varphi$
- *There is no X' such that $X \subset X' \subseteq K \cup \{\varphi\}$ and $X' \nvdash \varphi$*

Let us now define the score of a maximal consistent subset, given by the inconsistency values of its formulae.

Definition 8. *Let $K = \{\varphi_1, \ldots, \varphi_n\}$ be a belief base and φ be a formula. Let I be an inconsistency value. Then define the score of a formula φ_i as its inconsistency value for the base $K \cup \{\varphi\}$: $s_I(\varphi_i) = I_{\varphi_i}(K \cup \{\varphi\})$.*

And the score of a maximal consistent subset $X \in K \perp \varphi$ is the aggregated score of its formula: let g be an aggregation function, $s_{I,g}(X) = g_{\alpha \in X}(s_I(\alpha))$.

Definition 9. *Let $S = K \perp \varphi = \{X_1, \ldots, X_k\}$, a selection function for S is a function γ such that:*

- *If $X \perp \varphi$ is a non-empty set, then $\gamma(S)$ is a non-empty subset of S.*
- *If $X \perp \varphi$ is empty, then $\gamma(X \perp \varphi)$ is empty*

A score-based selection function $\gamma_{I,g,f}$, generated by the inconsistency measure I, the aggregation function g and the selection function f is such that $\gamma_{I,g,f}(S) = argmin_{X_i \in S} f(s_{I,g}(X_i))$.

min should be usually chosen for f, in order to select only the best results, but one could for instance want to obtain not only the MC with best (minimal) scores, but also close-to-the best ones, as for instance the 50% best ones. This is why we define f as an additional parameter.

Let $A = \{A_1, \ldots A_m\}$ be a set, then $A \oplus \alpha$ denotes the set $\{A_1 \cup \{\alpha\}, \ldots A_m \cup \{\alpha\}\}$.

Definition 10. *The MC operator $\star_{\mathsf{MC}}$ is defined as $K \star_{\mathsf{MC}} \varphi = \gamma(K \perp \neg \varphi) \oplus \varphi$.*

The score-based MC operator $\star_{I,g,f}$ is defined as $K \star_{I,g,f} \varphi = \gamma_{I,g,f}(K \perp \neg \varphi) \oplus \varphi$.

This defines the result of a revision as a set of belief bases. Then one has to choose an inference policy from this set. In the following we will focus on skeptical inference, but other policies can be used:

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

For belief base revision (i.e. when the base is not closed deductively, as opposed to belief sets), Hansson [5] defines the result of the revision as the conjunction of the intersection of all the selected remainder sets with the new piece of information ($\cap \gamma(K \perp \neg \varphi) \cup \varphi$), but this conjunction removes too much information, as illustrated in the next example, so we prefer to keep the full set of possible results as defined above.

Example 3. Consider the base $K = \{a \wedge c, b \wedge c\}$ and the formula $\varphi = \neg a \vee \neg b$. So $K \perp \varphi = \{\{a \wedge c\}, \{b \wedge c\}\}$. Suppose that $\gamma = id$, so $\cap \gamma(K \perp \neg \varphi) = \emptyset$, so it is not possible to infer c from $K \star \varphi$, whereas from the set $K \star_{\mathsf{MC}} \varphi = \{\{a \wedge c, \neg a \vee \neg b\}, \{b \wedge c, \neg a \vee \neg b\}\}$ it is possible to infer c.

So this gives a little more complicated definition, but it allows to obtain more inferences.

We will focus on the $\star_{S^I MI,max,min}$ operator using the MI Shapley inconsistency value, the max as aggregation function g, and the min as selection function f.

Let us illustrate the behavior of this operator on the following:

Example 4. Consider the base $K = \{a \wedge c, b \wedge c, b \wedge d\}$ and the formula $\varphi = \neg a \vee \neg b$.

So $K \perp \varphi = \{\{a \wedge c\}, \{b \wedge c, b \wedge d\}\}$. As $s_{S^I MI,max}(\{a \wedge c\}) = 1$, and $s_{S^I MI,max}(\{b \wedge c, b \wedge d\}) = 0.5$, with $f = min$ only $\{b \wedge c, b \wedge d\}$ is selected, so the result is a singleton set: $K \star_{S^I MI,max,min} \varphi = \{b \wedge c, b \wedge d, \neg a \vee \neg b\}$.

Let us now translate usual AGM belief revision basic properties [4,3] in this framework:

(K*1) $K * \alpha$ is a theory
(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) $K * \alpha = K_\perp$ iff $\vdash \neg \alpha$
(K*6) Si $\vdash \alpha \leftrightarrow \beta$, alors $K * \alpha = K * \beta$

Of course we work in a syntactic (not deductively closed) approach, so (K*1) should not be satisfied. But all other basic revision properties are satisfied:

Proposition 4. *The* MC *operators* $\star_{MC}$ *satisfy (K*2), (K*3), (K*4), (K*5), (K*6).*

5 Conciliation operators

Conciliation operators allow to solve the conflicts between a set of belief bases. The idea is to select the most problematic bases, to weaken them, and to iterate this process until there is no conflict left. We first define belief game models [20], that allow to obtain a conflict-free profile. Then the corresponding conciliation operator is just the conjunction of the obtained profile.

Definition 12 ([20]). *A* choice function *is a function* $g : \mathcal{E} \to \mathcal{E}$ *such that:*

- $g(\Psi) \sqsubseteq \Psi$
- *If* $\bigwedge \Psi \not\equiv \top$, *then* $\exists \varphi \in g(\Psi)$ *s.t.* $\varphi \not\equiv \top$
- *If* $\Psi \equiv \Psi'$, *then* $g(\Psi) \equiv g(\Psi')$

Definition 13 ([20]). *A* weakening function *is a function* $\blacktriangledown : \mathcal{L} \to \mathcal{L}$ *such that:*

- $\varphi \vdash \blacktriangledown(\varphi)$
- *If* $\varphi \equiv \blacktriangledown(\varphi)$, *then* $\varphi \equiv \top$
- *If* $\varphi \equiv \varphi'$, *then* $\blacktriangledown(\varphi) \equiv \blacktriangledown(\varphi')$

Definition 14 ([20]). *The solution to a belief profile* Ψ *for a Belief Game Model* $\mathcal{N} = \langle g, \blacktriangledown \rangle$ *under the integrity constraints* μ, *is the belief profile* $\Psi_\mathcal{N}^\mu$ *defined as:*

- $\Psi_0 = \Psi$
- $\Psi_{i+1} = \blacktriangledown_{g(\Psi_i)}(\Psi_i)$

- $\Psi_{\mathcal{N}}^{\mu}$ is the first Ψ_i that is consistent with μ

The conciliation operator $\blacktriangle_{\mathcal{N}}$ is defined as $\Psi \blacktriangle_{\mathcal{N}} \mu = \bigwedge \Psi_{\mathcal{N}}^{\mu}$

Definition 15 ([20]). *Let φ be a belief base.*

- *The* drastic weakening function *forgets all the information about that agent, i.e. :* $\forall \varphi \; \blacktriangledown_{\top}(\varphi) = \top$.
- *The* dilation weakening function *is defined as :*

$$mod(\blacktriangledown_{\delta}(\varphi)) = \{\omega \in \mathcal{W} \mid \exists \omega' \models \varphi \; d_H(\omega, \omega') \leq 1\}$$

where d_H is the Hamming distance between interpretations[6].

In this work we use the inconsistency value for defining the most conflicting agents (i.e. the selection function).

Definition 16. *A* Shapley Belief Game Model *is a Belief Game Model* $\mathcal{N} = \langle S_I, \blacktriangledown \rangle$, *where S_I is a Shapley Inconsistency Value.*

The solution to a belief profile Ψ for a Shapley Belief Game Model $\mathcal{N} = \langle S_I, \blacktriangledown \rangle$ under the integrity constraints μ, is the belief profile $\Psi_{\mathcal{N}}^{\mu}$ defined as:

- $\Psi_0 = \Psi$
- $\Psi_{i+1} = \blacktriangledown_{argmax_{\varphi_j \in \Psi_i}(S_{I_{\varphi_j}}(\Psi_i))}(\Psi_i)$
- $\Psi_{\mathcal{N}}^{\mu}$ is the first Ψ_i that is consistent with μ

Example 5. Consider the Shapley Belief Game Model $\mathcal{N} = \langle S_{I_{LP_m}}, \blacktriangledown_{\delta} \rangle$. There are seven agents $\Psi = \{\varphi_1, \ldots, \varphi_7\}$ with the following belief bases $\varphi_1 = a \wedge b$, $\varphi_2 = a \wedge (c \vee d)$, $\varphi_3 = a \wedge \neg d$, $\varphi_4 = a \wedge \neg c \wedge e$, $\varphi_5 = \neg a \wedge \neg b$, $\varphi_6 = a \wedge (\neg c \rightarrow \neg e)$, $\varphi_7 = a \wedge \neg c \wedge f$. There are no integrity constraints ($\mu = \top$). We have $S_{\varphi_1}^{I_{MI}} = \frac{1}{2}$, $S_{\varphi_2}^{I_{MI}} = \frac{7}{6}$, $S_{\varphi_3}^{I_{MI}} = \frac{7}{6}$, $S_{\varphi_4}^{I_{MI}} = \frac{4}{3}$, $S_{\varphi_5}^{I_{MI}} = 3$, $S_{\varphi_6}^{I_{MI}} = 1$, $S_{\varphi_2}^{I_{MI}} = \frac{5}{6}$. The maximal value is 3, meaning that φ_5 is the agent that brings the most conflicts, and so it is selected by the choice function for weakening. So φ_5 is replaced by $\top$. We have not yet reached a consistent profile, so we must do a further round. Then the new computations of inconsistency values give φ_4 as the most conflictual agent, and it is weakened to $\top$. The profile is still not consistent, so a third round is needed. In this third round φ_2, φ_3 and φ_7 are weakened. The resulting (consistent) profile for the whole process is then: $\Psi_{\mathcal{N}}^{\top} = \{\{a \wedge b\}, \top, \top, \top, \top, a \wedge (\neg c \rightarrow \neg e), \top\}$. So $\Psi \blacktriangle_{\mathcal{N}} \top \equiv a \wedge b \wedge (\neg c \rightarrow \neg e)$.

6 Platform description

In this section, we describe the PRISM (Platform for Reasoning with Inconsistency Shapley Measure) platform, that we have built in order to test the different operators presented in the previous section. One can use this platform to build a base and perform the different reasoning tasks such as inference, revision and conciliation. PRISM can be downloaded from the following page : http://www.cril.univ-artois.fr/prism. This page also contains pieces of information and detailed documentation about the platform. In the following, we present features and details about the implementation.

---

[6] Let ω and ω' be two interpretations, then $d_H(\omega, \omega') = \#(\{a \in \mathcal{P} \mid \omega(a) \neq \omega'(a)\})$.

6.1 Features

The platform is available as a Java application. The user interface is divided into 5 main tabs:

1. *Base* - allows the user to create a base of formulae
2. *Shapley* - computes the Shapley value of each formulae of the base
3. *Inference* - allows to reason given some inference relation
4. *Revision* - allows to reason on the base revised by a new formula
5. *Conciliation* - computes a consistent base from a set of (conflicting) bases

All tabs are structured in the same way. The top-left part of the panel represents the belief base currently used. Depending on the task that has to be performed, the top-right part displays the operators options. The bottom part displays the result of the computations.

Base tab The belief base is composed of several formulae. These formulae can either be loaded from a file or be directly written by the user. Accepted formulae have the following syntax :

$$\varphi := (\ \varphi\ )\ ;\ \varphi\ \&\ \varphi\ ;\ \varphi\ |\ \varphi\ ;\ \varphi\ \text{->}\ \varphi\ ;\ \varphi\ \text{<->}\ \varphi\ ;\ lit$$
$$lit := v\ ;\ \sim v\ \textit{where } v \textit{ is a variable name}$$

Each formula must end with a semicolon ";". Almost all alphanumeric strings are accepted for variable names[7].

Bases can be saved and loaded into the platform. Formulae can be viewed in CNF and can be added, modified or removed from the base.

It is also possible to group formulae. Groups can for instance represent agents to which the formulae belong. In practice, a group is represented by an integer. We distinguish a specific group that is identified by 0 : the constraints group. Formulae belonging to this group are constraints, i.e. formulae that cannot be falsified. These specific formulae can encode background knowledge for inference or revision, and integrity constraints for conciliation. The user can choose to take the groups into account or not.

In the following, we detail tabs of the platform. Each tab includes default operators but it is possible to define its own operator (see table 1).

Shapley tab Once a base is available, Shapley values of the formulae composing it can be computed. Details of this computation are available on the Shapley tab. More precisely, the MI are displayed at the bottom left panel. Shapley values are displayed at the bottom right panel. If group classification is taken into account, then the Shapley measures for each group is also displayed. Note that the operator for aggregating an inconsistency measure for a group of formulae can be chosen at the top right panel. By default, two operators are available : *Mean* and *Max*.

Inference tab Shapley values can be used to stratify belief bases: the lower the value, the higher in the layers of the base. The belief base stratification is displayed on the inference tab. The user can choose an inference operator from the list and ask whether a formula can be entailed from the base and the chosen operator. The formula must follow the same syntax as presented previously for the base. The default inference operators are named *Possibilistic* and *Linear* and correspond to the ones presented in section 3.

---

[7] The exceptions are the symbols in the following list : $'$, $-$, $\&$, $|$, $\sim$, $($, $)$, $<$, $>$ and ;. Strings "_nv#i" where i is an integer are also reserved.

Revision tab This tab allows to revise the belief base by some formula. The formula must be given in the field just under the table representing the base. The syntax must be the same as the one described previously for new formulae of the base. As presented in section 4, revision operator is compounded of different sub-operators : an aggregation function, a selection function and an inference policy. The default aggregation functions are *Max, Min* and *Sum*. See table 1 for classes details. The platform proposes three selection functions: *No Selection, All Min Selection* and *One of the min selection*. The last one arbitrary takes one MC into account. Finally, the last option is the inference policy. The user can initially choose between *Skeptical* and *Credulous* inferences but, once again, it is possible to implement its own inference policy. All MC are displayed in the bottom left panel, along with their respective score. The table in the bottom right panel shows selected MC. And, as in the revision tab, it is possible to ask whether a formula can be entailed according to the chosen type of inference.

Conciliation tab This tab allows to test the conciliation (belief game model) as it has been presented in section 5. This game is based on two operators: a choice operator and a weakening operator. The first choice operator is named *Shapley Choice* and corresponds to $S_{I_{MI}}$. The second operator, *Weak Shapley Choice*, selects randomly one formula amongst the ones selected by the Shapley operator. This allows to reduce the number of weakened formulae. The default weakening operator is named *Drastic* and corresponds to operator $\blacktriangledown_{\top}$ defined in section 5.

6.2 Implementation Details

PRISM is an evolutive platform, i.e. for all the reasoning tasks, one can add its own implementation. Main conditions are to create a class that extends a specific one and to add this class to the classpath[8]. The following table indicates for each operator the abstract class to extend and the main method to implement.

Tab	Abstract Class	Method
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 and methods to implement in order to add new operators

Assigning a Shapley value to a formula of a base necessitates the computation of all the MI that can be derived from the belief base. In the general case, computing MI is intractable. First, the number of MI can be exponential: a *n*-clauses SAT instance can exhibit $C_n^{n/2}$ MI in the worst case. Then, checking whether a formula belongs to the set of MI is in Σ_2^p [21]. Our problem is even more difficult since we want to compute all MI and check whether formulae belong to them.

Many approaches have been proposed to extract one MI from a set of clauses ([22,23,24] and many others) or to compute MI covers ([25]). We need here a com-

---

plete approach, i.e. one that extracts all MI. Candidate tools are thus less numerous: CAMUS [26], HYCAM [27].

We choose here to perform the MI extraction with the Sat4j SAT solver [28]. The main reason for this choice is our willingness to develop a platform independent tool. Using Java based technology is a way to preserve this property, even if we don't get the benefit of the last advances in the extraction af all MI (all dedicated tools cited previously are developed in C or C++). In further versions of PRISM, a detection of the running platform will be made in order to allow the use of CAMUS or HYCAM.

This Sat4j solver (2.3.3 release) extracts all MI in a two steps method [29]: all MC of the base are first computed, then MI are obtained through a second pass.

For all the MI extractors cited previously, the input is a CNF formula. Our platform allows the user to populate the belief base with general formulae. This means that we have to transform given formulae into equisatisfiable CNF formulae. To perform this transformation, we use the Tseitin encoding ([30]). This encoding results in a formula with a linear size increase in expense of the addition of new variables[9].

For a given formula, clauses composing its CNF form are grouped when given to the solver. This allows us to maintain equivalence between CNF and formulae.

In order to model general formulae, we have used a domain specific language (dsl) written in Scala. This allows us to have a very efficient parsing and CNF transformation. Moreover, since scala is built on top of the JVM, we preserve the platform independency.

7 Conclusion and future works

In this paper, we propose an evolutive platform that uses the MI Shapley inconsistency value to perform different reasoning tasks such as inference, revision and conciliation. Although operators are already implemented for each of these operations, one can write its own implementation and add it dynamically to the platform to test it. The platform is Java-based, which brings us full operating system independency.

In future works, on top of developing more operators, we plan to propose part of this platform as a Java library. Such a library would provide methods to use the MI Shapley inconsistency value and its associated operators for various applications.

References

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)

---

[9] New variables resulting from the encoding have the reserved names "_nv#i".

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 SAT06, 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