Logical comparison of inconsistent perspectives using scoring functions

Anthony Hunter
Department of Computer Science
University College London
Gower Street
London WC1E 6BT
UK

February 21, 2003

Abstract

The language for describing inconsistency is underdeveloped. If a database (a set of formulae) is inconsistent, there is usually no qualification of that inconsistency. Yet, it would seem useful to be able to say how inconsistent a database is, or to say whether one database is "more inconsistent" than another database. In this paper, we provide a more general characterization of inconsistency in terms of a scoring function for each database Δ. A scoring function S is from the power set of Δ into the natural numbers defined so that $S(\Gamma)$ gives the number of minimally inconsistent subsets of Δ that would be eliminated if the subset Γ was removed from Δ. This characterization offers an expressive and succinct means for articulating, in general terms, the nature of inconsistency in a set of formulae. We then compare databases using their scoring functions. This gives an intuitive ordering relation over databases that we can describe as "more inconsistent than". These techniques are potentially useful in a wide range of problems including monitoring progress in negotiations between a number of participants, and in comparing heterogeneous sources of information.

Keywords: inconsistency handling; conflict resolution; logic-based negotiation; heterogeneous knowledge

1 Introduction

Inconsistency handling is a big problem in computer science and IT. Techniques for comparing sets of formulae on the basis of inconsistency should be important instruments in helping to address this problem. Currently techniques for measuring the degree of inconsistency in a set of formulae are underdeveloped. Some approaches touch on the topic. In diagnostic systems, there are proposals that offer preferences for certain kinds of consistent subsets of inconsistent information [KW87, Rei87], in proposals for belief revision, epistemic entrenchment is an ordering over formulae which reflects the preference for which formulae to give up in case of inconsistency [Gar88], in proposals for drawing inferences from inconsistent information there is a preference for inferences from certain consistent subsets (E.g. [MR70, Bre89, BDP93, CRS93, Cho95, EGH95]), in proposals for approximating entailment, two sequences of entailment relation are defined (the first is sound but not complete, and the second is complete but not sound) which converge to classical entailment [SC95], and in proposals for partial consistency checking, checking is terminated after the search space exceeds a threshold which gives a measure of partial consistency of the data (E.g. Maximum generalized satisfiability [Pap94]). However, none of these proposals provide a direct definition for degree of inconsistency.

To address this need for a more direct way of comparing sets of formulae on the basis of the inconsistencies arising, we present a new approach based on scoring functions. A scoring function can be determined for any set of formulae Δ. The scoring function S is from the power set of Δ into the natural numbers defined so that $S(\Gamma)$ gives the number of minimally inconsistent subsets of Δ that would be eliminated if the subset Γ was removed from Δ.

Scoring functions are a straightforward way of summarizing the nature of the inconsistencies arising in a database. They are a more expressive approach than counting the number of minimally inconsistent subsets, or looking at the cardinality of the union of the minimally inconsistent subsets. They are more succinct than presenting all the minimally inconsistent subsets, or presenting the union of them.

Furthermore, we can compare databases by comparing the scoring function for each database. Consider the scoring function S_i for database Δ_i and S_j for Δ_j. If there is a bijection f from the subsets of Δ_i to the subsets of Δ_j such that for all $\Gamma \in \wp(\Delta_i)$, $S_i(\Gamma) \leq S_j(f(\Gamma))$, then Δ_j is more inconsistent than Δ_i. We believe that there are a variety of applications that could benefit from this proposal including:

Monitoring progress in negotiations Negotiations often start from a position of conflict. A group of participants may each have their own perspectives and agenda, and the group as a whole has to move to a position that is consistent. Suppose, each participant represents their perspective by a classical logic formula, then the group view can be represented by a set of formulae (for example in requirements engineering [HN98]). If this set of formulae is inconsistent, then each participant is able to revise their contribution. This revision may be iterated a number of times until eventually, a consistent set of formulae is obtained. Now if we have an intermediate position Δ, and this is revised by the participants to give Δ', then it would be desirable to know whether Δ' is less inconsistent than Δ. If it is not, then there is a danger that the negotiations are going off track. Ideally, we would like a sequence of positions $\Delta_1, \ldots, \Delta_n$ where Δ_1 is the starting position, $\ldots, \Delta_n$ is the final position, and that for each i, Δ_{i+1} is less inconsistent than Δ_i.

Comparing heterogeneous sources Suppose we have a number of sources providing information on some topic. Maybe we are dealing with a group of clinicians advising on some patient, a group of witnesses of some incident, or a set of newspaper reports covering some event. These are all situations where we expect some degree of inconsistency in the information. Also suppose that the information by each source i is represented by the set Φ_i. Each source may provide information that conflicts with the domain knowledge Ψ. Let us represent $\Phi_i \cup \Psi$ by Δ_i for each source i. Now if we were to offer some tool support for analysing the information from each of these sources, we may want to know whether one source is more inconsistent than another — so whether Δ_i is more inconsistent that Δ_j — and in particular determine which is the least inconsistent of the sources and so identify a minimal Δ_i in this inconsistency ordering. We may then view this minimal database as the least problematical or most reliable source of information.

The notions of scoring functions and score orderings are novel approaches to analysing inconsistent information. In this paper, we define, motivate, and analyse, them.

2 Related work

In belief revision theory, and the related field of knowledgebase merging, there are some proposals that do provide some description of the degree of inconsistency of a set of formulae. For example, the Dalal distance, essentially the Hamming distance between two propositional interpretations, can be used to give a profile of an inconsistent knowledgebase. Let $dalal(w, w')$ denote the Dalal distance from w to w', let $[\alpha]$ denote the set of classical models of α, and let $d(w, \alpha)$ be the $w' \in [\alpha]$ such that $dalal(w, w')$ is minimized. Now suppose we have a knowledgebase $\{\alpha_1, .., \alpha_n\}$ where each α_i is inconsistent but the knowledgebase may be inconsistent. We can then obtain a value of $d(w, \alpha_i)$ for each world w and each formula α_i in the knowledgebase. Unfortunately, this does not provide a very succinct way of describing the degree of inconsistency in a given set of formulae, and it is not clear how we could compare sets of formulae using this approach. Furthermore, operators for aggregating these distances such as the majority operator [LM98], egalitarist operator [Rev97], or the leximax operator [KP98], do not seem to be appropriate summaries of the degree of inconsistency in the original knowledgebase since they seek to find the most appropriate model for particular kinds of compromise of the orginal knowledge. Related techniques for knowledge base revision (for a review see [Pap00]) are similarly inappropriate for describing inconsistency in a set of formulae.

Another approach to handling inconsistent information is that of possibility theory [DLP94]. Let (ϕ, α) be a weighted formula where ϕ is a classical formula and $\alpha \in [0, 1]$. A possibilistic knowledgebase is a set of weighted formulae. An α-cut of a possibilistic knowledgebase, denoted $B_{\geq \alpha}$, is $\{(\psi, \beta) \in B \mid \beta \geq \alpha\}$. The inconsistency degree of B, denoted $Inc(B)$, is the maximum value of α such that the α-cut is inconsistent. As presented, the problem with this measure is that it assumes weighted formulae. In other words, we need some form of preference ordering in addition to the set of classical formulae in the knowledgebase. The knowledgebase can be used to induce such an ordering as suggested in [BDKP00], where an ordering over inferentially weaker forms of the orginal formulae are generated. Again this does not offer a direct lucid view on the inconsistency in the original set of formulae.

Some notions of measuring the "amount information" is related to the idea of measuring information. Information theory can be used to measure the information content of sets of inconsistent formulae. Developing Shannon's measure of information, Lozinskii proposes that the information in a set of propositional formulae Γ, that has been composed from n different atom symbols, is the the logarithm of the number of models (2^n) divided by the number of models for the maximum consistent subsets of Γ. This information theoretic measure increases with additions of consistent information and decreases with additions of inconsistent information.

However, as highlighted by Wong and Besnard, the measure by Lozinskii is sensitive to the presence of tautologies in Γ. To address, they suggest the use of a normal form for the formulae in Γ that is obtained by rewriting Γ into conjunctive normal form, and then applying disjunction elimination and resolution exhaustively [WB01]. However, neither this approach nor Lozinskii's approach provide a direct measure of inconsistency since for example, the value for $\{\alpha\}$ is the same as for $\{\alpha, \neg\alpha, \beta\}$.

3 Basic definitions

We start with a few definitions for classical logic.

Definition 3.1 *Let $\mathcal{L}$ be the set of* **classical propositional formulae** *formed from a set of atoms, and the logical connectives $\{\vee, \wedge, \neg, \rightarrow\}$. Let $\perp$ denote any inconsistent formula.*

Definition 3.2 *Let $\mathcal{D}$ be the set of* **databases** *formed from $\mathcal{L}$, where $\mathcal{D} = \wp(\mathcal{L})$. Let $\mathbb{N}$ be the set of natural numbers. For $n \in \mathbb{N}$, $\mathcal{D}^n$ is the set of databases of size n where*

$$\mathcal{D}^n = \{\Gamma \in \mathcal{D} \mid |\Gamma| = n\}$$

Definition 3.3 *Let $\Delta \in \mathcal{D}$, $Con(\Delta) = \{\Gamma \subseteq \Delta \mid \Gamma \not\vdash \bot\}$, and $Incon(\Delta) = \{\Gamma \subseteq \Delta \mid \Gamma \vdash \bot\}$.*

$$MC(\Delta) = \{\Phi \in Con(\Delta) \mid \forall \Psi \in Con(\Delta)\ \Phi \not\subset \Psi\}$$

$$MI(\Delta) = \{\Phi \in Incon(\Delta) \mid \forall \Psi \in Incon(\Delta)\ \Psi \not\subset \Phi\}$$

We call $MI(\Delta)$ the set of minimally inconsistent subsets of Δ, and $MC(\Delta)$ the set of maximally consistent subsets of Δ.

Definition 3.4 *Let $Free(\Delta)$ be the intersection of the maximally consistent subsets of Δ and $Core(\Delta)$ be the union of the minimally inconsistent subsets of Δ. We consider $Free(\Delta)$ as the unproblematical formulae in Δ, and we consider $Core(\Delta)$ as the problematical formulae in Δ.*

The value of $|MI(\Delta)|$ does not uniquely determine the value for $|MC(\Delta)|$, and vice versa, as illustrated by Example 3.1. This indicates the difficulty in abstracting sufficiently expressive yet concise parameters for describing inconsistency in a set of formulae.

Example 3.1 *Consider $\Delta_1 = \{\alpha \wedge \neg\alpha\}$, $\Delta_2 = \{\alpha, \neg\alpha\}$, $\Delta_3 = \{\alpha, \neg\alpha \wedge \beta, \neg\alpha \wedge \gamma\}$, and $\Delta_4 = \{\alpha, \neg\alpha\}$. Here, we have the following relationships:*

$$|MI(\Delta_1)| = |MI(\Delta_2)| \quad |MC(\Delta_1)| \neq |MC(\Delta_2)|$$

$$|MC(\Delta_3)| = |MC(\Delta_4)| \quad |MI(\Delta_3)| \neq |MI(\Delta_4)|$$

For further discussion of some of the relationships between MI, MC, $Free$, and $Core$ see [MR70, BDP93, CRS93, EGH95].

4 Scoring functions

For each $\Gamma \in \wp(\Delta)$, the scoring function for Δ gives the number of minimally inconsistent subsets of Δ that would be removed if Γ were removed from Δ.

Definition 4.1 *Let $\Delta \in \mathcal{D}$. Let S be the* **scoring function** *for Δ defined as follows, where $S : \wp(\Delta) \mapsto \mathbb{N}$ and $\Gamma \in \wp(\Delta)$*

$$S(\Gamma) = |MI(\Delta)| - |MI(\Delta - \Gamma)|$$

The scoring function for a database is an abstraction of the information we have about the database, and it says much about the inconsistencies arising in the database.

Example 4.1 *Let $\Delta = \{\alpha, \neg\alpha, \beta\}$, where S is the scoring function for Δ, defined as follows:*

$$S(\{\alpha\}) = 1 \qquad S(\{\neg\alpha\}) = 1 \qquad S(\{\beta\}) = 0$$
$$S(\{\alpha, \neg\alpha\}) = 1 \qquad S(\{\alpha, \beta\}) = 1 \qquad S(\{\neg\alpha, \beta\}) = 1$$
$$S(\{\alpha, \neg\alpha, \beta\}) = 1$$

Example 4.2 *Let* $\Delta = \{\alpha, \neg\alpha, \beta \wedge \neg\beta\}$*, where* S *is the scoring function for* Δ*, defined as follows:*

$$S(\{\alpha\}) = 1 \quad S(\{\neg\alpha\}) = 1 \quad S(\{\beta \wedge \neg\beta\}) = 1$$
$$S(\{\alpha, \neg\alpha\}) = 1 \quad S(\{\alpha, \beta \wedge \neg\beta\}) = 2 \quad S(\{\neg\alpha, \beta \wedge \neg\beta\}) = 2$$
$$S(\{\alpha, \neg\alpha, \beta \wedge \neg\beta\}) = 2$$

Example 4.3 *Let* $\Delta = \{\alpha, \neg\alpha \vee \neg\beta, \beta\}$*, where* S *is the scoring function for* Δ*, defined as follows:*

$$S(\{\alpha\}) = 1 \quad S(\{\neg\alpha \vee \neg\beta\}) = 1 \quad S(\{\beta\}) = 1$$
$$S(\{\alpha, \neg\alpha \vee \neg\beta\}) = 1 \quad S(\{\alpha, \beta\}) = 1 \quad S(\{\neg\alpha \vee \neg\beta, \beta\}) = 1$$
$$S(\{\alpha, \neg\alpha \vee \neg\beta, \beta\}) = 1$$

Example 4.4 *Let* $\Delta = \{\alpha \wedge \neg\alpha, \beta, \gamma\}$*, where* S *is the scoring function for* Δ*, defined as follows:*

$$S(\{\alpha \wedge \neg\alpha\}) = 1 \quad S(\{\beta\}) = 0 \quad S(\{\gamma\}) = 0$$
$$S(\{\alpha \wedge \neg\alpha, \beta\}) = 1 \quad S(\{\alpha \wedge \neg\alpha, \gamma\}) = 1 \quad S(\{\beta, \gamma\}) = 0$$
$$S(\{\alpha \wedge \neg\alpha, \beta, \gamma\}) = 1$$

4.1 Score orderings

We can compare databases using the scoring function for each database. For this we define score orderings.

Definition 4.2 *A* **score ordering***, denoted* $\leq$*, is defined as follows*[1]*. Assume* $\Delta_i, \Delta_j \in \mathcal{D}^n$*, for some* n*, and* S_i *is the scoring function for* Δ_i*, and* S_j *is the scoring function for* Δ_j*.* $S_i \leq S_j$ *holds iff there is a bijection* $f : \wp(\Delta_i) \mapsto \wp(\Delta_j)$ *such that the following condition is satisfied:*

$$\forall \Gamma \in \wp(\Delta_i), S_i(\Gamma) \leq S_j(f(\Gamma))$$

Note, $S_i < S_j$ *iff* $S_i \leq S_j$ *and* $S_j \not\leq S_i$*. Also,* $S_i \simeq S_j$ *iff* $S_i \leq S_j$ *and* $S_j \leq S_i$*. We say* Δ_j *is* **more inconsistent** *than* Δ_i *iff* $\Delta_i \leq \Delta_j$*.*

Example 4.5 *Let* $\Delta_1 = \{\alpha, \neg\alpha\}$ *and* $\Delta_2 = \{\alpha, \beta \wedge \neg\beta\}$*. Let* S_1 *be the scoring function for* Δ_1 *and* S_2 *be the scoring function for* Δ_2*, and so* $S_2 < S_1$*.*

$$S_1(\{\alpha\}) = 1 \quad S_2(\{\alpha\}) = 0$$
$$S_1(\{\neg\alpha\}) = 1 \quad S_2(\{\beta \wedge \neg\beta\}) = 1$$
$$S_1(\{\alpha, \neg\alpha\}) = 1 \quad S_2(\{\alpha, \beta \wedge \neg\beta\}) = 1$$

Example 4.6 *Consider* $\Delta_1 = \{\alpha \wedge \neg\alpha, \beta, \gamma\}$ *and* $\Delta_2 = \{\alpha \wedge \neg\alpha, \beta \wedge \neg\beta, \delta\}$*. If* S_1 *is the scoring function for* Δ_1*, and* S_2 *is the scoring function for* Δ_2*, then* $S_1 < S_2$*.*

We can consider scoring functions as giving information about the overlaps of the minimally inconsistent subsets. For example, for $\Delta_i, \Delta_j \in \mathcal{D}^n$, if $|MI(\Delta_i)| = |MI(\Delta_j)|$ and $S_i \leq S_j$ then the inconsistencies are more overlapping in Δ_j. In other words, more of the formulae are in more minimally inconsistent subsets.

In case we want to compare sets of different cardinality, we can add dummy propositions to the smaller set to make it the same size as the larger set. These dummy propositions are literals that do not appear elsewhere and so can be assumed to not be in any of the minimally inconsistent subsets of the database.

[1]Note, we are now using the $\leq$ symbol for the usual ordering over the natural numbers and as defined here for an ordering over score functions. Hopefully, this overloading of the symbol will not cause confusion.

4.2 Comparing inconsistent perspectives

Our motivation for using scoring functions is to compare inconsistent perspectives. This includes applications in monitoring progress in negotiations and in comparing heterogeneous sources of information.

For the following example of negotiation, we will keep the domain knowledge separate from the perspectives of the participants. In other words, we will consider the domain knowledge as being correct and not subject to negotiation. This will allow us to focus our attention on the perspectives of the participants.

Example 4.7 *Consider three members of a family who are discussing their wishes for their next family car. Let the domain knowledge* Ψ *be:*

$$\begin{array}{l} \texttt{red} \rightarrow \texttt{fast} \\ \texttt{fast} \rightarrow \neg\texttt{fuelEfficient} \\ \texttt{offRoad} \rightarrow \texttt{expensive} \\ \texttt{sporty} \rightarrow (\texttt{expensive} \wedge (\texttt{black} \vee \texttt{red} \vee \texttt{white})) \\ \neg\texttt{expensive} \rightarrow \texttt{under\$20K} \\ \texttt{cabriolet} \rightarrow \neg\texttt{bigCapacity} \\ \texttt{fuelEfficient} \rightarrow \neg\texttt{offRoad} \end{array}$$

Let the initial preferences (requirements or demands) for each family member (participant 1, participant 2, and participant 3) be represented by Φ_1^1*,* Φ_1^2 *and* Φ_1^3 *respectively.*

$$\begin{array}{l} \Phi_1^1 = \{\texttt{red}, \texttt{offRoad}\} \\ \Phi_1^2 = \{\neg\texttt{expensive}, \texttt{fuelEfficient}\} \\ \Phi_1^3 = \{\texttt{sporty}, \texttt{cabriolet}, \texttt{bigCapacity}\} \end{array}$$

So the starting point of the discussions is captured by Δ_1*.*

$$\Delta_1 = \Psi \cup \Phi_1^1 \cup \Phi_1^2 \cup \Phi_1^3$$

Let S_1 *be the scoring function for* Δ_1*. Now consider* S_1 *for some subsets of* Δ_1*.*

$$\begin{array}{cc} S_1(\{\texttt{red}\}) = 1 & S_1(\{\texttt{bigCapacity}\}) = 1 \\ S_1(\{\texttt{sporty}\}) = 1 & S_1(\{\texttt{offRoad}\}) = 2 \\ S_1(\{\texttt{fuelEfficient}\}) = 2 & S_1(\{\neg\texttt{expensive}\}) = 2 \end{array}$$

$$\begin{array}{c} S_1(\{\texttt{cabriolet}\}) = 1 \\ S_1(\{\texttt{red}, \texttt{bigCapacity}\}) = 2 \\ S_1(\{\neg\texttt{expensive}, \texttt{fuelEfficient}\}) = 4 \\ S_1(\{\texttt{red}, \texttt{offRoad}\}) = 3 \\ S_1(\{\texttt{sporty}, \texttt{cabriolet}, \texttt{bigCapacity}\}) = 2 \\ S_1(\Delta_1) = 5 \end{array}$$

We see from S_1 *that each of the preferences is individually inconsistent with the domain knowledge. We also see that* Φ_1^2 *has the highest score (4) of the initial preferences and it would be a good starting point for discussion.*

Suppose after some discussion, Φ_1^1 *is changed to* Φ_2^1 *by participant 1,* Φ_1^2 *to* Φ_2^2 *by participant 2, and* Φ_1^3 *to* Φ_2^3 *by participant 3, as follows.*

$$\begin{array}{l} \Phi_2^1 = \{\texttt{red} \vee \texttt{black}, \texttt{sporty} \vee \texttt{offRoad}\} \\ \Phi_2^2 = \{\neg\texttt{expensive}\} \\ \Phi_2^3 = \{\texttt{sporty}, \texttt{bigCapacity}\} \end{array}$$

This intermediate point is captured by Δ_2.

$$\Delta_2 = \Psi \cup \Phi_2^1 \cup \Phi_2^2 \cup \Phi_2^3$$

Let S_2 *be the scoring function for* Δ_2. *Now consider* S_2 *for some subsets of* Δ_2.

$$S_2(\{\texttt{sporty}\}) = 1$$
$$S_2(\{\neg\texttt{expensive}\}) = 2$$
$$S_2(\{\texttt{sporty} \vee \texttt{offRoad}\}) = 1$$
$$S_2(\Delta_2) = 2$$

We see that $S_2 < S_1$. *Furthermore, we see that the preference for* $\neg$`expensive` *is the most problematical.*

Now suppose after further discussion, Φ_2^1 *is changed to* Φ_3^1 *by participant 1,* Φ_2^2 *to* Φ_3^2*, and* Φ_2^3 *to* Φ_3^3*, as follows.*

$$\Phi_3^1 = \{\texttt{red} \vee \texttt{black}, \texttt{sporty} \vee \texttt{offRoad}\}$$
$$\Phi_3^2 = \{\texttt{interestFreeCredit}, \texttt{diesel}\}$$
$$\Phi_3^3 = \{\texttt{sporty} \vee \texttt{offRoad}, \texttt{bigCapacity}\}$$

This final situation is captured by Δ_3.

$$\Delta_3 = \Psi \cup \Phi_3^1 \cup \Phi_3^2 \cup \Phi_3^3$$

Let S_3 *be the scoring function for* Δ_3. *We see that* $S_3 < S_2$. *Also for all* $\Gamma \in \Delta_3$*, we have* $S_3(\Gamma) = 0$. *So* Δ_3 *could be regarded as an acceptable end-point.*

In the above example, we see that the scoring functions allow us to focus on the more problematical data, and use this to facilitate conflict resolution.

5 Characterising scoring functions

In this section, we show how scoring functions are a concise and yet expressive encodation of the nature of the inconsistencies that arise in a set of formulae.

5.1 Properties of scoring functions

We make a few simple observations regarding scoring functions.

Proposition 5.1 *For* $\Delta \in \mathcal{D}$*, where* S *is the scoring function for* Δ*,*

$$S(Core(\Delta)) = S(\Delta) = |MI(\Delta)|$$
$$S(Free(\Delta)) = 0$$

From the scoring function for a database Δ, it is straightforward to calculate the cardinality of $Free(\Delta)$ and $Core(\Delta)$.

Proposition 5.2 *If* $\Delta \in \mathcal{D}$ *and* S *is a scoring function for* Δ*, then*

$$|Free(\Delta)| = |\{\alpha \in \Delta \mid S(\{\alpha\}) = 0\}|$$
$$|Core(\Delta)| = |\{\alpha \in \Delta \mid S(\{\alpha\}) \neq 0\}|$$

There is no simple counterpart to these propositions for determining the cardinality of the set of maximally consistent subsets of a database directly from the scoring function for the database.

Proposition 5.3 *Let $\leq$ be the usual ordering relation over $\mathbb{N}$. For all $\Delta \in \mathcal{D}$, and $\Gamma_i, \Gamma_j \in \wp(\Delta)$, where S is the scoring function for Δ,*

$$S(\Gamma_i \cap \Gamma_j) \leq min(\{S(\Gamma_i), S(\Gamma_j)\})$$

$$max(\{S(\Gamma_i), S(\Gamma_j)\}) \leq S(\Gamma_i \cup \Gamma_j)$$

Note, $S(\Gamma_i) + S(\Gamma_j) \leq S(\Gamma_i \cup \Gamma_j)$ does not necessarily hold as illustrated below.

Example 5.1 *Let S be the scoring function for Δ, and let $\Gamma_1 = \{\neg\alpha, \alpha \wedge \beta\}$, and let $\Gamma_2 = \{\neg\alpha, \alpha \wedge \neg\beta\}$, and let $\Delta = \Gamma_1 \cup \Gamma_2$. So $S(\Gamma_1) = S(\Gamma_2) = 2$, but $S(\Gamma_1 \cup \Gamma_2) = 3$.*

Also, a pair of scoring functions may agree on all singleton sets modulo some bijection, but this does not necessarily mean the scoring functions are equal as illustrated below.

Example 5.2 *Consider Δ_1 and Δ_2 below. Let S_1 be the scoring function for Δ_1 and S_2 be the scoring function for Δ_2.*

$$\Delta_1 = \{\alpha, (\neg\alpha \vee \neg\beta) \wedge \phi, \beta, (\neg\alpha \vee \neg\beta) \wedge \psi\}$$
$$\Delta_2 = \{\neg\alpha, \alpha, \alpha \wedge \beta, \neg\beta\}$$

For singleton sets Γ, $S_1(\Gamma) = S_2(f(\Gamma))$, where f is a bijection defined as follows $f(\{\alpha\}) = \neg\alpha$, $f(\{(\neg\alpha \vee \neg\beta) \wedge \phi\}) = \alpha$, $f(\{\beta\}) = \alpha \wedge \beta$, and $f(\{(\neg\alpha \vee \neg\beta) \wedge \psi\}) = \neg\beta$. Hence, $S_1(\{\alpha\}) = S_2(\{\neg\alpha\}) = 2$, $S_1(\{(\neg\alpha \vee \neg\beta) \wedge \phi\}) = S_2(\{\alpha\}) = 1$, $S_1(\{\beta\}) = S_2(\{\alpha \wedge \beta\}) = 2$, and $S_1(\{(\neg\alpha \vee \neg\beta) \wedge \psi\}) = S_2(\{\neg\beta\}) = 1$. However, $S_1(\Delta_1) \neq S_2(f(\Delta_1))$. Since $S_1(\Delta_1) = 2$ and $S_2(\Delta_2) = 3$. There are a further three bijections f where $S_1(\Gamma) = S_2(f(\Gamma))$ for each singleton set Γ, but none where $S_1(\Delta_1) = S_2(f(\Delta_2))$.

5.2 Properties of score orderings

The following two results indicate the implicit constraints on a pair of databases for one to be more inconsistent than the other.

Proposition 5.4 *For each $n \in \mathbb{N}$, the score ordering $\leq$ over $\mathcal{D}^n$ is reflexive and transitive, but not antisymmetric.*

Proposition 5.5 *Let $\Delta_i, \Delta_j \in \mathcal{D}^n$, where S_i is a scoring function for Δ_i, and S_j is a scoring function for Δ_j. If $S_i \leq S_j$ holds, and $S_j \leq S_i$ holds, then there is a bijection f from $\wp(\Delta_i)$ to $\wp(\Delta_j)$ such that for all $\Gamma \in \wp(\Delta_i)$, $S_i(\Gamma) = S_j(f(\Gamma))$.*

Proof: We assume there is a bijection $f : \wp(\Delta_i) \mapsto \wp(\Delta_j)$, such that $\forall\Gamma \in \wp(\Gamma_i)\ S_i(\Gamma) \leq S_j(f(\Gamma))$, and there is a bijection $f' : \wp(\Delta_i) \mapsto \wp(\Delta_j)$, such that $\forall \in \Gamma\wp(\Delta_i)\ S_i(\Gamma) \geq S_j(f'(\Gamma))$. It is no loss of generality to assume that for each $\Gamma \in \wp(\Delta_i)$, we have $S_i(\Gamma) = S_j(f'(\Gamma))$ iff $S_i(\Gamma) = S_j(f(\Gamma))$. Let $E(\Delta_i) = \{\Gamma \in \wp(\Delta_i) \mid S_i(\Gamma) = S_j(f(\Gamma))\}$, and $E'(\Delta_i) = \{\Gamma \in \wp(\Delta_i) \mid S_i(\Gamma) = S_j(f'(\Gamma))\}$. So $E(\Delta_i) = E'(\Delta_i)$. Also let $N(\Delta_i) = \{\Gamma \in \wp(\Delta_i) \mid S_i(\Gamma) < S_j(f(\Gamma))\}$, and $N'(\Delta_i) = \{\Gamma \in \wp(\Delta_i) \mid$

$S_i(\Gamma) > S_j(f'(\Gamma))\}$. So $N(\Delta_i) = \wp(\Delta_i) - E(\Delta_i)$, and $N'(\Delta_i) = \wp(\Delta_i) - E'(\Delta_i)$. Hence, $N(\Delta_i) = N'(\Delta_i)$. But this is only possible, if $N(\Delta_i) = N'(\Delta_i) = \emptyset$. As a result, $E(\Delta_i) = E'(\Delta_i) = \wp(\Delta_i)$. Therefore, $\forall \Gamma \in \wp(\Delta_i)\ S_i(\Gamma) = S_j(f(\Gamma)) = S_j(f'(\Gamma))$. □

The following two results show in part how a score ordering can be viewed as an aggregation of parameters including the relative number of minimally inconsistent formulae and the relative number of free formulae.

Proposition 5.6 *For $n \in \mathbb{N}$, and $\Delta_i, \Delta_j \in \mathcal{D}^n$, if S_i is the scoring function for Δ_i, and S_j is the scoring function for Δ_j, then*

$$S_i \leq S_j \text{ implies } |MI(\Delta_i)| \leq |MI(\Delta_j)|$$

Note, the converse does not hold.

Proof: If $S_i \leq S_j$, then there is a bijection f such that $S_i(\Delta_i) \leq S_j(f(\Delta_j))$, and hence $|MI(\Delta_i)| \leq |MI(\Delta_j)|$. As a counterexample for the converse, consider the sets $\Delta_1 = \{\alpha, \neg\alpha, \alpha \rightarrow \beta, \neg\beta\}$, and $\Delta_2 = \{\alpha, \neg\alpha, \beta \wedge \neg\beta, \gamma \wedge \neg\gamma\}$. If S_1 is the scoring function for Δ_1, and S_2 is the scoring function for Δ_2, then $S_1 \not\leq S_2$ and $S_2 \not\leq S_1$.□

Proposition 5.7 *For $n \in \mathbb{N}$, and $\Delta_i, \Delta_j \in \mathcal{D}^n$, if S_i is the scoring function for Δ_i, and S_j is the scoring function for Δ_j, then*

$$S_i \leq S_j \text{ implies } |Free(\Delta_i)| \geq |Free(\Delta_j)|$$

Note, the converse does not hold.

Proof: If $S_i \leq S_j$, then there is a bijection f such that $\forall \Gamma \in \Delta_i$, $S_i(\Gamma) \leq S_j(f(\Gamma))$. This implies $\forall \Gamma \in \Delta_i$ if $S_j(f(\Gamma)) = 0$, then $S_i(\Gamma) = 0$. Finally this implies $|Free(\Delta_i)| \geq |Free(\Delta_j)|$. As a counterexample for the converse, let $\Delta_1 = \{\alpha \wedge \neg\alpha, \beta \wedge \neg\beta, \gamma\}$ and $\Delta_2 = \{\phi, \neg\phi \vee \psi, \neg\psi\}$. Hence, $|Free(\Delta_1)| > |Free(\Delta_2)|$, but there is no bijection such that $S_1 \leq S_2$ holds. □

With the same assumptions as those for Proposition 5.6, we do not get that $S_i \leq S_j$ implies $|MC(\Delta_i)| \leq |MC(\Delta_j)|$ or that it implies $|MC(\Delta_i)| \geq |MC(\Delta_j)|$. This is captured in the following example.

Example 5.3 *Consider $\Delta_1 = \{\alpha, \beta\}$ and $\Delta_2 = \{\alpha, \neg\alpha\}$. So $S_1 \leq S_2$ and $|MC(\Delta_1)| \leq |MC(\Delta_2)|$. Now consider $\Delta_3 = \{\alpha, \neg\alpha\}$ and $\Delta_4 = \{\beta \wedge \neg\beta, \gamma \wedge \neg\gamma\}$. So $S_3 \leq S_4$ and $|MC(\Delta_3)| \geq |MC(\Delta_4)|$.*

5.3 Comparison with ⊥-isomorphisms

Another approach to characterizing inconsistency in sets of formulae is based on inconsistency isomorphisms. In this, we provide a way to say that a pair of databases are equivalent with regard to inconsistency. This is based on the structure of the minimally inconsistent subsets of each set including the number and overlap of the minimally inconsistent subsets.

Definition 5.1 *Let $n \in \mathbb{N}$, and $\Delta_i, \Delta_j \in \mathcal{D}^n$, and Δ_i and Δ_j are* **isomorphic with respect to inconsistency** *iff there is a bijection $h : \Delta_i \mapsto \Delta_j$ such that for each subset $\{\phi_1, .., \phi_p\}$ of Δ_i the following equivalence holds:*

$$\{\phi_1, .., \phi_p\} \vdash \bot \text{ iff } \{h(\phi_1), .., h(\phi_p)\} \vdash \bot$$

If Δ_i and Δ_j are isomorphic with respect to inconsistency, then we say Δ_j is an **inconsistency isomorphism** *(abbreviated to* **$\perp$-isomorphism***) of Δ_i.*

Example 5.4 *The following two sets are isomorphic with respect to inconsistency:*

$$\Delta_1 = \{\alpha, \neg\alpha, \alpha \to \beta, \neg\beta\}$$
$$\Delta_2 = \{\alpha \wedge \delta, \neg\alpha, \neg\alpha \vee \beta, \neg\beta\}$$

Consider the bijection $h(\alpha) = \alpha \wedge \delta$, $h(\neg\alpha) = \neg\alpha$, $h(\alpha \to \beta) = \neg\alpha \vee \beta$, and $h(\neg\beta) = \neg\beta$.

Example 5.5 *The following two sets are not isomorphic with respect to inconsistency:*

$$\Delta_1 = \{\alpha, \neg\alpha\}$$
$$\Delta_2 = \{\alpha \wedge \neg\alpha, \beta\}$$

Example 5.6 *The following two sets are isomorphic with respect to inconsistency:*

$$\Delta_1 = \{\alpha \wedge \beta, \neg(\alpha \wedge \beta) \vee \gamma, \neg\gamma\}$$
$$\Delta_2 = \{\phi, \phi \to \psi, \neg\psi\}$$

Consider the bijection $h(\alpha \wedge \beta) = \phi$, $h(\neg(\alpha \wedge \beta) \vee \gamma) = \phi \to \psi$, and $h(\neg\gamma) = \neg\psi$.

Establishing that a pair of databases are isomorphic with respect to inconsistency allows for a number of inferences regarding their relationship.

Proposition 5.8 *For $n \in \mathbb{N}$, and $\Delta_i, \Delta_j \in \mathcal{D}^n$, if Δ_i and Δ_j are isomorphic with respect to inconsistency then the following equivalences hold, with a bijection $h : \Delta_i \mapsto \Delta_j$.*

$$\{\phi_1, .., \phi_p\} \in MC(\Delta_i) \textit{ iff } \{h(\phi_1), .., h(\phi_p)\} \in MC(\Delta_j)$$
$$\{\phi_1, .., \phi_p\} \in MI(\Delta_i) \textit{ iff } \{h(\phi_1), .., h(\phi_p)\} \in MI(\Delta_j)$$

Proposition 5.9 *For $n \in \mathbb{N}$, and $\Delta_i, \Delta_j \in \mathcal{D}^n$, if S_i is the scoring function for Δ_i, and S_j is the scoring function for Δ_j, then the following equivalence holds.*

$$\Delta_i \textit{ and } \Delta_j \textit{ are } \perp\textit{-isomorphisms iff } S_i \simeq S_j$$

Proof: ($\Rightarrow$) If Δ_i and Δ_j are isomorphic with respect to inconsistency, then Δ_i and Δ_j are identical with respect to their inconsistent subsets modulo the differences in names as captured by the bijection between the two sets of formulae. Hence, for each $\{\phi_1, .., \phi_p\} \subseteq \Delta_i$, $S_i(\{\phi_1, .., \phi_p\}) = S_j(\{h(\phi_1), .., h(\phi_p)\})$. So if we let $f(\{\phi_1, .., \phi_p\}) = \{h(\phi_1), .., h(\phi_p)\}$, then for each $\{\phi_1, .., \phi_p\} \subseteq \Delta_i$, $S_i(\{\phi_1, .., \phi_p\}) = S_j(f(\{\phi_1, .., \phi_p\}))$. ($\Leftarrow$) Assume $S_i \simeq S_j$. So there is a bijection $f : \wp(\Delta_i) \mapsto \wp(\Delta_j)$ such that $\forall\Gamma \in \wp(\Delta_i)$ $[S_i(\Gamma) \leq S_j(f(\Gamma))]$ and there is a bijection $f : \wp(\Delta_j) \mapsto \wp(\Delta_i)$ such that $\forall\Gamma \in \wp(\Delta_j)$ $[S_j(\Gamma) \leq S_i(f(\Gamma))]$. This implies, by Proposition 5.5, there is a bijection $f : \wp(\Delta_i) \mapsto \wp(\Delta_j)$ such that $\forall\Gamma \in \wp(\Delta_i)$ $[S_i(\Gamma) = S_j(f(\Gamma))]$. So $\forall\{\phi_1, .., \phi_p\} \in \wp(\Delta_i)$, let $f(\{\phi_1, .., \phi_p\}) = \{h(\phi_1), .., h(\phi_p)\}$, and hence there is a bijection $h : \Delta_i \mapsto \Delta_j$ such that $\{\phi_1, .., \phi_p\} \vdash \perp$ iff $\{h(\phi_1), .., h(\phi_p)\} \vdash \perp$. Therefore, Δ_i and Δ_j are $\perp$-isomorphisms. □

The equivalence identified in Proposition 5.9 shows that the scoring functions subsume the $\perp$-isomorphisms.

5.4 Comparison with the MI function

The scoring function for a database Δ encodes the membership of the minimally inconsistent subsets of Δ. In this section, we consider generating $MI(\Delta)$ directly from the scoring function for Δ. To do this, we need to use the scoring functions to eliminate sets from $\wp(\Delta)$ that are not in $MI(\Delta)$. Each $\Gamma \in \wp(\Delta)$ can be classified as exactly one of five types of set.

1. Γ contains free items. Free items are formulae that are in $Free(\Delta)$ and so are not in any minimally inconsistent subset of Δ. These can be identified and eliminated easily by the scoring function since these items, and only these items, have score 0 as a singleton set.
2. Γ contains a mixed pair. For a set of formulae Δ, $\{\gamma_1, \gamma_2\} \in \wp(\Delta)$ is a mixed pair iff γ_1 and γ_2 are not free items and they do not appear in the same minimally inconsistent subset of Δ. So any set containing a mixed pair is not a minimally inconsistent subset of Δ.
3. Γ is a combi set. A combi set of Δ is a non-singleton set $\Gamma \in \wp(\Delta)$ such that (1) each immediate subset[2] of Γ is in $MI(\Delta)$ and (2) Γ does not contain a mixed pair.
4. Γ is strict subset of a minimally inconsistent subset
5. Γ is a minimally inconsistent subset

These five types of set are disjoint, and they cover all possibilities for $\wp(\Delta)$. We illustrate some of these types in the following examples.

Example 5.7 *Let $\Delta = \{\alpha, \neg\alpha, \beta\}$. Here, $S(\{\beta\}) = 0$, and so $\{\beta\}$, $\{\alpha, \beta\}$ and $\{\alpha, \neg\alpha, \beta\}$ contain free items.*

Example 5.8 *Let $\Delta = \{\alpha, \neg\alpha, \beta, \neg\beta\}$. Here there are four mixed pairs in $\wp(\Delta)$ which are $\{\alpha, \beta\}$, $\{\alpha, \neg\beta\}$, $\{\neg\alpha, \beta\}$, and $\{\neg\alpha, \neg\beta\}$.*

Example 5.9 *Let $\Delta = \{\neg\alpha, \alpha \vee \beta, \neg\beta, \beta \wedge \gamma, \neg\delta, \delta\}$. Mixed pairs in $\wp(\Delta)$ include $\{\neg\delta, \neg\alpha\}$, $\{\neg\alpha, \beta \wedge \gamma\}$ and $\{\beta \wedge \gamma, \delta\}$.*

Example 5.10 *Let $\Delta = \{\alpha \wedge \neg\alpha, \beta \wedge \neg\beta\}$. There is one mixed pair in $\wp(\Delta)$ which is Δ.*

Example 5.11 *Let $\Delta = \{\neg\alpha, \alpha \wedge \beta, \alpha \wedge \neg\beta\}$. So Δ is a combi set of Δ.*

Example 5.12 *Let $\Delta = \{\neg\alpha_1 \wedge (\alpha_2 \vee \alpha_4), \neg\alpha_2 \wedge (\alpha_1 \vee \alpha_3), \neg\alpha_3 \wedge (\alpha_2 \vee \alpha_4), \neg\alpha_4 \wedge (\alpha_1 \vee \alpha_3)\}$. So Δ is a combi set of Δ.*

The process for generating $MI(\Delta)$ is involved, but the aim is really to demonstrate the expressibility of scoring functions. We sketch the process as follows, where we start with any database Δ, and go through the following four steps eliminating elements of $\wp(\Delta)$:

1. To remove sets containing free items, remove sets $\{\alpha\} \in \wp(\Delta)$, where $S(\{\alpha\}) = 0$, and then remove any supersets of $\{\alpha\}$.

[2]For a set Γ, an immediate subset of a set $\Gamma' \subset \Gamma$ is such that $\Gamma \backslash \Gamma'$ is a singleton set.

2. To remove sets containing mixed pairs, we use the following means of identification: $\{\gamma_1, \gamma_2\}$ is a mixed pair of Δ iff $S(\{\gamma_1, \gamma_2\}) = S(\{\gamma_1\}) + S(\{\gamma_2\})$. Any sets containing mixed pairs should be removed. The motivation for this is that γ_1 and γ_2 do not appear together in any minimally inconsistent subset iff the set of minimally inconsistent subsets removed from $\wp(\Delta)$ by removing γ_1 is disjoint from the set of minimally inconsistent subsets removed from $\wp(\Delta)$ by removing γ_2.
3. To remove combi sets, we use the following means of identification: For a non-singleton set Γ, $S(\Gamma) = |\Gamma|$ iff Γ is a combi set. The motivation for this is that Γ has $|\Gamma|$ immediate subsets, and so if each is in $MI(\Delta)$, then $S(\Gamma) = |\Gamma|$. Furthermore, if $S(\Gamma) = |\Gamma|$, and Γ is not a singleton set, and Γ contains no mixed pairs, then each member of Γ appears with every other member of Γ in a minimally inconsistent subset of Δ, and no formula in $\Delta \backslash \Gamma$ is in any of these minimally inconsistent subsets, and so every immediate subset of Γ is in $MI(\Delta)$.
4. The remaining items from $\wp(\Delta)$ are sets that are subsets of minimally inconsistent subsets of Δ. So the maximally inconsistent subsets are the maximal items according to the $\subseteq$ relation.

To illustrate this process, consider the following examples.

Example 5.13 *Let $\Delta = \{\alpha, \neg\alpha \wedge \beta, \neg\alpha \wedge \gamma\}$. Here the scoring function is:*

$$S(\Delta) = 2 \quad S(\{\alpha, \neg\alpha \wedge \beta\}) = 2 \quad S(\{\alpha, \neg\alpha \wedge \gamma\}) = 2$$
$$S(\{\neg\alpha \wedge \beta, \neg\alpha \wedge \gamma\}) = 2 \quad S(\{\alpha\}) = 2 \quad S(\{\neg\alpha \wedge \beta\}) = 1$$
$$S(\{\neg\alpha \wedge \gamma\}) = 1$$

So the sets containing mixed pairs are $\{\neg\alpha \wedge \beta, \neg\alpha \wedge \gamma\}$ and Δ. Also there are no combi sets in $\wp(\Delta)$. So after removing these sets, the remainder of $\wp(\Delta)$ is $\{\{\alpha, \neg\alpha \wedge \beta\}, \{\alpha, \neg\alpha \wedge \gamma\}, \{\alpha\}, \{\neg\alpha \wedge \beta\}, \{\neg\alpha \wedge \gamma\}\}$. There are two maximal elements here, and so $MI(\Delta) = \{\{\alpha, \neg\alpha \wedge \beta\}, \{\alpha, \neg\alpha \wedge \gamma\}\}$.

Example 5.14 *Let $\Delta = \{\alpha, \neg\alpha \wedge \beta, \neg\alpha \wedge \neg\beta\}$. Here the scoring function is:*

$$S(\Delta) = 3 \quad S(\{\alpha, \neg\alpha \wedge \beta\}) = 3 \quad S(\{\alpha, \neg\alpha \wedge \neg\beta\}) = 3$$
$$S(\{\neg\alpha \wedge \beta, \neg\alpha \wedge \neg\beta\}) = 3 \quad S(\{\alpha\}) = 2 \quad S(\{\neg\alpha \wedge \beta\}) = 2$$
$$S(\{\neg\alpha \wedge \neg\beta\}) = 2$$

So there are no mixed pairs in Δ. But there is a combi set which is Δ. After removing the combi set, the remainder of $\wp(\Delta)$ is $\{\{\alpha, \neg\alpha \wedge \beta\}, \{\alpha, \neg\alpha \wedge \neg\beta\}, \{\neg\alpha \wedge \beta, \neg\alpha \wedge \neg\beta\}, \{\alpha\}, \{\neg\alpha \wedge \beta\}, \{\neg\alpha \wedge \neg\beta\}\}$. There are three maximal elements here, and so $MI(\Delta) = \{\{\alpha, \neg\alpha \wedge \beta\}, \{\alpha, \neg\alpha \wedge \neg\beta\}, \{\neg\alpha \wedge \beta, \neg\alpha \wedge \neg\beta\}\}$.

We stress the aim of showing that we can generate $MI(\Delta)$ from the scoring function of Δ is to illustrate the expressivity of scoring functions.

5.5 Syntax sensitivity

Clearly, scoring functions are syntax sensitive. As a result of this sensitivity, scoring functions may also be regarded as being prone to semantic insensitivity. To illustrate semantic insensitivity, consider the following two examples.

Example 5.15 *Consider Δ_1 and Δ_2 below. Let S_1 be the scoring function for Δ_1 and S_2 be the scoring function for Δ_2.*

$$\Delta_1 = \{\alpha, \neg\alpha\}$$
$$\Delta_2 = \{\alpha \wedge \beta, \neg\alpha \wedge \beta\}$$

Here, $S_1 \simeq S_2$ and so the scoring functions do not differentiate Δ_1 and Δ_2. Yet it could be argued that semantically Δ_2 implies more (such as if paraconsistent logic inference were used) than Δ_1.

Example 5.16 *Consider Δ_1 and Δ_2 below. Let S_1 be the scoring function for Δ_1 and S_2 be the scoring function for Δ_2.*

$$\Delta_1 = \{\alpha \wedge \beta \wedge \gamma, \alpha \wedge \neg\beta \wedge \gamma, \neg\alpha \wedge \beta \wedge \neg\gamma\}$$
$$\Delta_2 = \{\alpha \wedge \beta \wedge \gamma, \alpha \wedge \neg\beta \wedge \gamma, \neg\alpha \wedge \beta \wedge \gamma\}$$

Here, the formulae in Δ_1 and Δ_2 are pairwise inconsistent, and the resulting scoring functions are such that $S_1 \simeq S_2$. It may be argued that Δ_2 is less inconsistent than Δ_1 since all formulae in Δ_2 agree on γ.

In response to the arguments raised in Example 5.15 and 5.16, we believe that this kind of semantic insensitivity is useful in some applications. We believe that when a connective is used, it is used with some intent. So for example, whilst $\alpha \wedge \beta$ and α, β are semantically equivalent, we need to differentiate them also. This intent depends on the applications area, but to illustrate in negotiation, consider a strategy for weakening the preferences (represented by a set of classical formulae) of an agent is take a subset of the preferences. So if an agent starts with $\{\alpha \wedge \beta\}$ as its preferences then the only possible weakening (using the $\subseteq$ relation) is $\{\}$. Whereas if the agent starts with $\{\alpha, \beta\}$ then weakenings also include $\{\alpha\}$ and $\{\beta\}$. In this application, the preference $\alpha \wedge \beta$ is intended to mean that $\alpha \wedge \beta$ must occur together, and so if the preference α is dropped then so is the preference β.

The general conclusion we draw from this discussion is that the syntax sensitivity, and the resulting semantic insensitivity, found in scoring functions is useful in some applications.

6 Discussion

In this paper, we have proposed a framework for characterizing inconsistency that can be used in processes for conflict resolution. For example, when a series of resolution steps are taken to remove all inconsistencies in some multi-participant negotiation, then score ordering can be used to ensure each resolution step is decreasing the degree of inconsistency. As another example, in merging heterogeneous sources we may choose to ignore a source because it exceeds some threshold of inconsistency in the score ordering.

Currently, we are investigating the incorporation of weighting of subsets of a database so that we can view some inconsistencies as more significant than others. For example, in a report on a football match, an inconsistency with regard to the final score is more significant than an inconsistency with regard to the name of the player who scored the most goals.

We are also investigating the use of techniques for consistency checking to enable scoring functions to be more efficiently obtained for any given database. Consistency checking is inherently intractable in the propositional case. To address this problem, we can consider using (A) tractable subsets of classical logic (for example binary disjunctions of literals [GJ79]), (B) heuristics to direct the search for a model[3] (for example in semantic tableau [OS88], GSAT [SLM92], and constraint satisfaction [DP87]), and (C) formalization of approximate consistency checking based on notions described below, such as approximate entailment and partial consistency.

[3]Heuristic approaches can be either complete such as semantic tableau or incomplete such as in the GSAT system. Whilst in general, using heuristics to direct search has the same worst-case computational properties as undirected search, it can offer better performance in practice for some classes of theories. Note, heuristic approaches do not tend to be oriented to offering any analysis of theories beyond a decision on consistency.

Approximate entailment Proposed in [Lev84], and developed in [SC95, Kor01], classical entailment is approximated by two sequences of entailment relations. The first is sound but not complete, and the second is complete but not sound. Both sequences converge to classical entailment. For a set of propositional formulae Δ, a formula α, and an approximate entailment relation $\models_i$, the decision of whether $\Delta \models_i \alpha$ holds or $\Delta \not\models_i \alpha$ holds can be computed in polynomial time.

Partial consistency Consistency checking does not necessarily involve an exponential search space. Furthermore, consistency checking for a set of formulae Δ can be prematurely terminated when the search space exceeds some threshold. When the checking of Δ is prematurely terminated, partial consistency is the degree to which Δ is consistent. This can be measured in a number of ways including the proportion of formulae from Δ that can be shown to form a consistent subset of Δ. Maximum generalized satisfiability [Pap94] may be viewed as an example of this.

Probable consistency Determining the probability that a set of formulae is consistent on the basis of polynomial time classifications of those formulae. Classifications for the propositional case can be based on tests including counting the number of different propositional letters, counting the multiple occurrences of each propositional letter, and determining the degree of nesting for each logical symbol. The more a set of formulae is tested, the greater the confidence in the probability value for consistency/inconsistency, but this is at the cost of undertaking the tests.

Identifying approximate consistency for a set of formulae Δ is obviously not a guarantee that Δ is consistent. However, approximate consistency checking is useful because it helps to focus where problems possibly lie in Δ, and to prioritize resolution tasks.

Acknowledgements

The author is very grateful to Sanjay Modgil for helpful feedback on this work.

References

[BDKP00] S Benferhat, D Dubois, S Kaci, and H Prade. Encoding information fusion in possibilistic logic: A general framework for rational syntactic merging. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI'2000)*, pages 3–7. IOS Press, 2000.

[BDP93] S Benferhat, D Dubois, and H Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 411–419. Morgan Kaufmann, 1993.

[Bre89] G Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *Proceedings of the Eleventh International Conference on Artificial Intelligence*, pages 1043–1048, 1989.

[Cho95] L. Cholvy. Automated reasoning with merged contradictory information whose reliability depends on topics. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'95)*, volume 946 of *Lecture Notes in Computer Science*. Springer, 1995.

[CRS93] C Cayrol, V Royer, and C Saurel. Management of preferences in assumption based reasoning. In *Information Processing and the Management of Uncertainty in Knowledge based Systems (IPMU'92)*, volume 682 of *Lecture Notes in Computer Science*. Springer, 1993.

[DLP94] D Dubois, J Lang, and H Prade. Possibilistic logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. Oxford University Press, 1994.

[DP87] R Dechter and J Pearl. Network-based heuristics for constraint-satisfaction problems. *Artificial Intelligence*, 34:1–38, 1987.

[EGH95] M Elvang-Goransson and A Hunter. Argumentative logics: Reasoning from classically inconsistent information. *Data and Knowledge Engineering*, 16:125–145, 1995.

[Gar88] P Gardenfors. *Knowledge in Flux*. MIT Press, 1988.

[GJ79] M Garey and D Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[HN98] A Hunter and B Nuseibeh. Managing inconsistent specifications: Reasoning, analysis and action. *ACM Transactions on Software Engineering Methodology*, 7:335–367, 1998.

[Kor01] F Koriche. On anytime coherence-based reasoning. In *Quantitative and Qualitative Approaches to Reasoning with Uncertainty (Ecsqaru'01)*, volume 2143 of *Lecture Notes in Computer Science*, pages 544–556. Springer, 2001.

[KP98] S Konieczny and R Pino Perez. On the logic of merging. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR98)*, pages 488–498. Morgan Kaufmann, 1998.

[KW87] J De Kleer and B Williams. Diagnosing mulitple faults. *Artificial Intelligence*, 32:97–130, 1987.

[Lev84] H Levesque. A logic of implicit and explicit belief. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'84)*, pages 198–202, 1984.

[LM98] J. Lin and A.O. Mendelzon. Merging databases under constraints. *International Journal of Cooperative Information Systems*, 7(1), 1998.

[MR70] R Manor and N Rescher. On inferences from inconsistent information. *Theory and Decision*, 1:179–219, 1970.

[OS88] F Oppacher and E Suen. HARP: A tableau-based theorem prover. *Journal of Automated Reasoning*, 4:69–100, 1988.

[Pap94] C Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Pap00] O Papini. Knowledge-base revision. *Knowledge Engineering Review*, pages 339–370, 2000.

[Rei87] R Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[Rev97] P Revesz. On the semantics of arbitration. *International Journal of Algebra and Computation*, 7:133–160, 1997.

[SC95] M Schaerf and M Cadoli. Tractable reasoning via approximation. *Artificial Intelligence*, 74:249–310, 1995.

[SLM92] B Selman, H Levesque, and D Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI'92)*, pages 440–446, 1992.

[WB01] P Wong and Ph Besnard. Paraconsistent reasoning as an analytic tool. *Journal of the Interest Group in Propositional Logic*, 9:233–246, 2001.