A General Framework for Reasoning about Inconsistency

Leila Amgoud, Venkatramanan Siva Subrahmanian

A General Framework for Reasoning about Inconsistency

V. S. Subrahmanian
Computer Science Dept. and UMIACS
University of Maryland
College Park, Maryland 20742
vs@cs.umd.edu

Leila Amgoud
IRIT - CNRS
118, route de Narbonne,
31062 Toulouse Cedex 9, France
amgoud@irit.fr

Abstract

Numerous logics have been developed for reasoning about inconsistency which differ in (i) the logic to which they apply, and (ii) the criteria used to draw inferences. In this paper, we propose a general framework for reasoning about inconsistency in a wide variety of logics including ones for which inconsistency resolution methods have not yet been studied (e.g. various temporal and epistemic logics). We start with Tarski and Scott's axiomatization of logics, but drop their monotonicity requirements that we believe are too strong for AI. For such a logic L, we define the concept of an option. Options are sets of formulas in L that are closed and consistent according to the notion of consequence and consistency in L. We show that by defining an appropriate preference relation on options, we can capture several existing works such as Brewka's subtheories. We also provide algorithms to compute most preferred options.

1 Introduction

Inconsistency management has been intensely studied in various parts of AI, often in slightly disguised form [Poole, 1985; G. Pinkas, 1992; Rescher and Manor, 1970; Gardenfors, 1988]. For example, default logics [Reiter, 1980] use syntax to distinguish between strict facts and default rules, and identify different extensions of the default logic as potential ways of "making sense" of seemingly conflicting information. Likewise, inheritance networks [Touretzkey, 1984] define extensions based on analyzing paths in the network and using notions of specificity to resolve conflicts. Argumentation frameworks [Dung, 1995] study different ways in which an argument for or against a proposition can be made, and then determine which arguments defeat which other arguments in an effort to decide what can be reasonably concluded. All these excellent works provide an *a priori* conflict resolution mechanism. A user who uses a system based on these papers is more or less committed to the semantics implemented in the system, and has little say in the matter (besides which most users querying KBs are unlikely to be experts in even classical logic, let alone default logics and argumentation methods).

The aims of the paper are:

1. to propose a unified framework for reasoning about inconsistency, which captures existing approaches as a special case and provides an easy basis for comparison.
2. to apply the framework using any logic (*monotonic* or *nonmonotonic*), including ones for which inconsistency management has not been studied before (e.g. temporal, probabilistic logics).
3. to allow end-users to bring their domain knowledge to bear, allowing them to voice an opinion on *what works for them, not what a system manager decided was right for them.*, in other words, to take into account the preferences of the end-user.
4. to propose efficient algorithms for computing the preferred options.

We do this by building upon Dana Scott's celebrated notions of an abstract logic. We start with a simple example to illustrate why conflicts can often end up being resolved in different ways by human beings, and why it is important to allow end-users to bring their knowledge to bear when a system resolves conflicts.

Example 1 (Salary example) *Suppose a university payroll system says that John's salary is 50K, while the university personnel database says it is 60K. In addition, there may be an axiom that says that everyone has exactly one salary. One simple way to model this is via the theory, denoted* sal^{base}*, below.*

$$\mathsf{salary}(\mathsf{John}, 50\mathsf{K}) \leftarrow \quad (1)$$

$$\mathsf{salary}(\mathsf{John}, 60\mathsf{K}) \leftarrow \quad (2)$$

$$\mathsf{S} = \mathsf{S}' \leftarrow \mathsf{salary}(\mathsf{X}, \mathsf{S}) \wedge \mathsf{salary}(\mathsf{X}, \mathsf{S}'). \quad (3)$$

sal^{base} *is obviously inconsistent. Suppose (3) is definitely known to be true. Then a bank manager considering John for a loan may choose the 50K number to determine a maximal loan amount that John qualifies for. But a national tax agency may use the 60K figure to send John a letter asking him why he underpaid his taxes.*

Neither the bank manager nor the tax officer is making any attempt to find out the truth (thus far) - however, both of them are making different decisions based on the same facts.

The rest of this paper proceeds as follows. In Section 2, we recall Scott's notion of what a logic is [Scott, 1982]. Then, in Section 3, we define our general framework for reasoning about inconsistency for any Scott logic. Section 4 shows how to adapt the general framework to the particular case of inconsistent bases. In section 5, we show how existing works can be retrieved in our general framework. Section 6 presents some examples of how the general framework can be applied in other logics such as temporal and probabilistic logics. In Section 7, we develop algorithms to compute optimal options based on various types of monotonicity assumptions about the objective function. Note that due to lack of space, we only include a few proofs - the full version of this paper will contain all proofs.

2 Scott's Abstract Consequence Operation

Dana Scott [Scott, 1982] defines an abstract logic to consist of a set $\mathcal{L}$ (whose members are called well-formed formulas) and a consequence operator CN. CN is any function from $2^{\mathcal{L}}$ (the powerset of $\mathcal{L}$) to $2^{\mathcal{L}}$ (intuitively, $\mathsf{CN}(X)$ returns the set of formulas that are logical consequences of X according to the logic in question) that satisfies the following axioms:

Expansion $X \subseteq \mathsf{CN}(X)$.

Idempotence $\mathsf{CN}(\mathsf{CN}(X)) = \mathsf{CN}(X)$.

Monotonicity $X \subseteq Y \Rightarrow \mathsf{CN}(X) \subseteq \mathsf{CN}(Y)$.

Coherence $\mathsf{CN}(\emptyset) \neq \mathcal{L}$.

It is easy to see that most well known monotonic logics (such as propositional logic [Shoenfield, 1967], first order logic [Shoenfield, 1967], modal logic, temporal logic, fuzzy logic, probabilistic logic [Bacchus, 1990], etc.) can be viewed as a special case of Scott's notion of an abstract logic. The non-monotonic logics introduced in AI do not satisfy the monotonicity axiom, though Marek et. al. [Marek *et al.*, 1990] have defined the notion of non-monotone rule systems that extend Scott's ideas to non-monotonic logics by dropping the Monotonicity Axiom above. *In the remainder of this paper, we drop monotonicity axiom.* Once $(\mathcal{L}, \mathsf{CN})$ are fixed, we can define a notion of consistency as follows:

Definition 1 (Consistency) *Let $X \subseteq \mathcal{L}$. X is* consistent *in logic $(\mathcal{L}, \mathsf{CN})$ iff $\mathsf{CN}(X) \neq \mathcal{L}$.*

This says that X is consistent iff its set of consequences is not the set of all well formed formulas. Note that the coherence requirement characterizing CN forces $\emptyset$ to always be consistent - this is reasonable for pretty much any reasonable logic as saying nothing should intuitively be consistent.

3 A general framework for handling inconsistency

This section proposes a general framework for handling inconsistency under any logic. Reasoning with inconsistent knowledge bases is a process which follows three steps:

1. Constructing consistent subbases,
2. Selecting among all the subbases the preferred ones, called preferred subbases,
3. Applying classical entailment on a choice of the preferred subbases.

Throughout the rest of this paper, we assume that we have an arbitrary, but fixed (*monotonic* or *non-monotonic*) logic $(\mathcal{L}, \mathsf{CN})$.

The basic idea behind our framework is to construct what we call *options*, and then to define a preference relation on these options. The *preferred options* are intended to support the conclusions to be drawn from the inconsistent knowledge base. Intuitively, an option is a set of formulas that is both consistent and closed w.r.t. the consequence in logic $(\mathcal{L}, \mathsf{CN})$.

Definition 2 (Options) *An* option *is any set $\mathcal{O}$ of elements of $\mathcal{L}$ such that:*

- *$\mathcal{O}$ is consistent.*
- *$\mathcal{O}$ is closed, i.e. $\mathcal{O} = \mathsf{CN}(\mathcal{O})$.*

Let $\mathtt{Opt}(\mathcal{L})$ be the set of all options that can be built from $(\mathcal{L}, \mathsf{CN})$.

Les us illustrate the above concept.

Example 2 *Let $\mathcal{L}$ be a propositional language, and let $\mathcal{K} \subseteq \mathcal{L}$ be the knowledge base $\mathcal{K} = \{a, a \rightarrow b, \neg b\}$. The following options, for instance, can be built from $\mathcal{K}$: $\mathcal{O}_1 = \mathsf{CN}(\{a\})$, $\mathcal{O}_2 = \mathsf{CN}(\{\neg b\})$, $\mathcal{O}_3 = \mathsf{CN}(\{a \rightarrow b\})$, $\mathcal{O}_4 = \mathsf{CN}(\{a, a \rightarrow b\})$, $\mathcal{O}_5 = \mathsf{CN}(\{a, \neg b\})$, $\mathcal{O}_6 = \mathsf{CN}(\{a, b\})$.*

The framework for reasoning about inconsistency has three components: a set of all options that can be built from the considered logic, a preference relation between the options, and an entailment mechanism.

Definition 3 (General framework) *Let $(\mathcal{L}, \mathsf{CN})$ be a fixed logic. A* general framework for reasoning about inconsistency *in the logic $(\mathcal{L}, \mathsf{CN})$ is a triple $\langle \mathtt{Opt}(\mathcal{L}), \succeq, \mid\!\sim \rangle$ such that:*

- *$\mathtt{Opt}(\mathcal{L})$ is the set of options built from $(\mathcal{L}, \mathsf{CN})$*
- *$\succeq \subseteq \mathtt{Opt}(\mathcal{L}) \times \mathtt{Opt}(\mathcal{L})$. $\succeq$ is a partial (or total) preorder, that is, it is reflexive and transitive.*
- *$\mid\!\sim : 2^{\mathtt{Opt}(\mathcal{L})} \rightarrow \mathtt{Opt}(\mathcal{L})$ is an entailment mechanism.*

The second important concept of the above general framework is the preference relation $\succeq$ between options. Indeed, $\mathcal{O}_1 \succeq \mathcal{O}_2$ means that the option $\mathcal{O}_1$ is at least as preferred as $\mathcal{O}_2$. This relation captures the idea that some options are better than others because, for instance, the user has decided that, or because those preferred options satisfy the requirements imposed by the developer of a conflict management system. For instance, in Example 1, the user chooses certain options (e.g. the options where the salary is minimal or where the salary is maximal based on his needs). In most approaches for handling inconsistency, maximal consistent subsets of the inconsistent knowledge base have an important role. However, if we consider a KB such as $\mathcal{K} = \{(a \wedge b); a \rightarrow c; b \rightarrow \neg c\}$, there are three maximal consistent subsets. One of these is $MCS_1 = \{a \rightarrow c; b \rightarrow \neg c\}$. If we represent this KB as the default theory $\Delta = (W, D)$ where $W = \emptyset$ and $D = \{\frac{:(a \wedge b)}{(a \wedge b)}; \frac{:a \rightarrow c}{a \rightarrow c}; \frac{:b \rightarrow \neg c}{b \rightarrow \neg c}\}$, we would get three extensions corresponding to the three maximal consistent subsets. However, one could argue that MCS_1 (and the extension corresponding to it) are too weak - we

could have included either a or b by weakening the formula $(a \wedge b)$ instead of dropping it altogether.

The third component of the framework is a mechanism for selecting the inferences to be drawn from the knowledge base. In our framework, the set of inferences is itself an option. Thus, it should be consistent. This requirement is of great importance, since it ensures that the framework delivers *safe* conclusions.

The set of inferences is generally computed from the different preferred options. Let $\mathcal{O}^{\succeq}$ be the set of all preferred options, i.e. $\mathcal{O}^{\succeq} = \{\mathcal{O}_i \in \mathtt{Opt}(\mathcal{L}) \mid \nexists \mathcal{O}_j \in \mathtt{Opt}(\mathcal{L})$ with $\mathcal{O}_j \succeq \mathcal{O}_i\}$. Different entailment mechanisms can be defined for selecting the inferences to be drawn. Here are some examples of such mechanisms.

Definition 4 *Let $\langle \mathtt{Opt}(\mathcal{L}), \succeq, \mid\sim \rangle$ be a framework, and $\mathcal{O}^{\succeq}$ the set of its preferred options. Let ψ be an element of $\mathcal{L}$.*

Universal criterion: *$\mathcal{L} \mid\sim \psi$ iff $\psi \in \mathcal{O}_i$, $\forall \mathcal{O}_i \in \mathcal{O}^{\succeq}$. ψ is called a* universal consequence *of $\mathcal{L}$.*

Argumentative criterion: *$\mathcal{L} \mid\sim \psi$ iff $\exists \mathcal{O}_i \in \mathcal{O}^{\succeq}$ such that $\psi \in \mathcal{O}_i$, and $\nexists \mathcal{O}_j \in \mathcal{O}^{\succeq}$ such that $\neg\psi \in \mathcal{O}_j$. ψ is called an* argumentative consequence *of $\mathcal{L}$.*

We can show that the set of inferences made using the universal criterion is itself an option of the language $\mathcal{L}$, thus it is an entailment mechanism. Moreover, it is included in every preferred option.

Proposition 1 *Let $\langle \mathtt{Opt}(\mathcal{L}), \succeq, \mid\sim \rangle$ be a framework built from a monotonic logic ($\mathcal{L}$, CN).*

- *The set $\{\psi \mid \psi$ is a universal consequence of $\mathcal{L}\}$ is an option of $\mathtt{Opt}(\mathcal{L})$.*
- *$\forall \mathcal{O}_i \in \mathcal{O}^{\succeq}$, $\{\psi \mid \psi$ is a universal conseq. of $\mathcal{L}\} \subseteq \mathcal{O}_i$.*

Proof (Sketch) *Let $\mathcal{C} = \{\psi \mid \psi$ is a universal consequence of $\mathcal{L}\}$. By definition, any element in $\mathcal{C}$ belongs to all the options in $\mathcal{O}^{\succeq}$. Consequently, $\mathcal{C} \subseteq \mathcal{O}_i$, $\forall \mathcal{O}_i \in \mathcal{O}^{\succeq}$.*

As each $\mathcal{O}_i \in \mathcal{O}^{\succeq}$ is an option, $\mathcal{O}_i$ is consistent. Thus, $\mathcal{C}$ (which is a subset of $\mathcal{O}_i$) is also consistent. Similarly, since $\mathcal{C} \subseteq \mathcal{O}_i$, thus $\mathsf{CN}(\mathcal{C}) \subseteq \mathcal{O}_i$ as well, $\forall \mathcal{O}_i \in \mathcal{O}^{\succeq}$. Consequently, $\mathsf{CN}(\mathcal{C}) \subseteq \mathcal{C}$ (according to the above definition of universal consequences). Thus, $\mathcal{C}$ is closed, and consequently it is an option. ■

Similarly, we can show that the set of argumentative consequences is an option. Thus, it is a valid entailment mechanism.

Proposition 2 *Let $\langle \mathtt{Opt}(\mathcal{L}), \succeq, \mid\sim \rangle$ be a framework built from a monotonic logic ($\mathcal{L}$, CN). The set $\{\psi \mid \psi$ is an argumentative consequence of $\mathcal{L}\}$ is an option of $\mathtt{Opt}(\mathcal{L})$.*

However, the following criterion is not a valid entailment mechanism since the set of consequences returned by it may be inconsistent, thus it is not an option.

Example 3 *$\mathcal{L} \mid\sim \psi$ iff $\exists \mathcal{O}_i \in \mathcal{O}^{\succeq}$ such that $\psi \in \mathcal{O}_i$.*

4 Optimal Options that Handle Inconsistency

In the preceding section, we have developed the concepts of an option and a preferred option for any logic $(\mathcal{L}, \mathsf{CN})$. However, this has been defined in a way that is independent of a knowledge base $\mathcal{K}$. This section shows how to associate a set of preferred options with an inconsistent knowledge base $\mathcal{K}$.

Definition 5 *We say an option O* handles *$\mathcal{K}$ iff there is a subset $\mathcal{K}' \subseteq \mathcal{K}$ such that $O = \mathsf{CN}(\mathcal{K}')$. Let* `Base` *be a function that returns for any option $\mathcal{O}$, the subbase $\mathcal{K}'$. Thus,* $\mathtt{Base}(\mathcal{O}) = \mathcal{K}'$.

Intuitively, an option handles $\mathcal{K}$ iff it is the closure of some subset of $\mathcal{K}$. Clearly, such a subset $\mathcal{K}'$ must be consistent because $\mathcal{O}$ is consistent (by virtue of being an option) and as $\mathcal{O} = \mathsf{CN}(\mathcal{K}')$.

Example 4 (Handling inconsistent default logic theories) *Let us consider default logic as our base logic, and suppose we consider the default theory $\Delta = (W, D)$ where $W = \{p\}$ and $D = \{\frac{:p}{\neg p}\}$. Most researchers in default logic would consider this theory to be inconsistent as it has no extension. However, it has two valid options: $\Delta_1 = (\emptyset, \emptyset)$ and $\Delta_2 = (\{p\}, \emptyset)$. A user may specify a preference relation that prefers Δ_2. This shows how our framework can be applied to default logic.*

Definition 6 *Suppose $(\mathtt{Opt}(\mathcal{L}), \succeq, \mid\sim)$ is a general framework for reasoning about inconsistency in logic $(\mathcal{L}, \mathsf{CN})$ and suppose $\mathcal{K}$ is an inconsistent knowledge base. An* optimal option for *$\mathcal{K}$ is any member $\mathcal{O} \in \mathtt{Opt}(\mathcal{L})$ such that:*

- *$\mathcal{O}$ handles $\mathcal{K}$ and*
- *there is no other option $\mathcal{O}' \in \mathtt{Opt}(\mathcal{L})$ that handles $\mathcal{K}$ such that $\mathcal{O}' \succeq \mathcal{O}$.*

What this definition says is that when reasoning about an inconsistent knowledge base $\mathcal{K}$, we always look at the set of optimal options that handle $\mathcal{K}$.

Example 5 *Let us return to Example 1. Suppose we use $\mathcal{O}_1 = \{(1), (2)\}, \mathcal{O}_2 = \{(1), (3)\}, \mathcal{O}_3 = \{(2), (3)\}\}$. Of course, we assume all of these options are also closed under usual first order consequence. First, let us say that these three options are preferred to all other options in the language.*

- *Suppose the* score *$sc(\mathcal{O}_i)$ of option $\mathcal{O}_i$ is the sum of the multiset $\{S \mid sal(John, S) \in \mathcal{O}_i\}$. In this case, the score of $\mathcal{O}_1$ is $50K$, that of $\mathcal{O}_2$ is 110K, and that of $\mathcal{O}_3 = 60K$. We could now say that $\mathcal{O}_i \succeq \mathcal{O}_j$ if $sc(\mathcal{O}_i) \leq sc(\mathcal{O}_j)$. In this case, the only option that handles $\mathcal{K}$ is $\mathcal{O}_1$ which corresponds to the bank manager's viewpoint.*
- *On the other hand, suppose we say that $\mathcal{O}_i \succeq \mathcal{O}_j$ iff $sc(\mathcal{O}_i) \geq sc(\mathcal{O}_j)$. In this case, the only optimal option that handles $\mathcal{K}$ is $\mathcal{O}_2$ — this corresponds to the view that the rule saying everyone has only one salary is wrong (perhaps the database has John being paid out of two projects simultaneously and 50K of his salary is charged to one project and 60K to another).*
- *Now consider the case where we change our scoring method and say that $sc(\mathcal{O}_i) = \min\{S \mid sal(John, S) \in \mathcal{O}_i\}$. In this case, $sc(\mathcal{O}_1) = 50K, sc(\mathcal{O}_2) = 60K, sc(\mathcal{O}_3) = 50K$. Let us suppose that the preference relation says that $\mathcal{O}_i \succeq \mathcal{O}_j$ iff $sc(\mathcal{O}_i) \geq sc(\mathcal{O}_j)$. Then the only optimal option is $\mathcal{O}_2$ which corresponds exactly to the tax agency's viewpoint.*

Thus, we see that our general framework for optimally handling inconsistency is very powerful - it can be used to handle inconsistencies in different ways based upon how the preference relation between options is defined.

5 Link with existing approaches

Two kinds of approaches have been proposed in the literature for solving the problem of inconsistency. The first one consists of revising the knowledge base and restoring its consistency. The second approach accepts inconsistency and copes with it. The first approach initiated in [Rescher and Manor, 1970] proposes to give up some formulas of the knowledge base in order to get one or several consistent subbases of the original base. Then plausible conclusions may be obtained by applying classical entailment on these subbases. In what follows, we consider the case of propositional bases. When the knowledge base is *flat*, i.e. its formulas are *not weighted*, maximal consistent subbases are built.

Let $\mathcal{K}$ be a knowledge base that may be inconsistent, and $\mathcal{S}$ = $\{S_1, \ldots, S_n\}$ its set of maximal (for set inclusion) consistent subbases. We can show that these subbases correspond to the preferred options of the above framework when the preference relation $\succeq$ between options is *monotonic*. Let us first define that notion of monotonicity.

Definition 7 (Monotonic relation) *The relation $\succeq$ is said* monotonic *iff for any X, $Y \subseteq \mathcal{L}$, if $X \subseteq Y$, then $Y \succeq X$. $\succeq$ is said* anti-monotonic *iff, if $X \subseteq Y$, then $X \succeq Y$.*

Proposition 3 *Let $\langle \mathtt{Opt}(\mathcal{L}), \succeq, \vdash\!\sim \rangle$ be a framework such that $\succeq$ is monotonic. Let $\mathcal{K}$ be an inconsistent knowledge base, and $\mathcal{O}^{\succeq}$ be the set of preferred/optimal options for $\mathcal{K}$.*

- *$\forall S_i$, $\exists\, \mathcal{O} \in \mathcal{O}^{\succeq}$, such that $\mathcal{O}$ = $\mathsf{CN}(S_i)$.*
- *$\forall \mathcal{O}_i \in \mathcal{O}^{\succeq}$, $\exists\, S$ such that $\mathcal{O}_i$ = $\mathsf{CN}(S)$.*

In the case of prioritized knowledge bases, Brewka has proposed in [Brewka, 1989] a definition of the preferred subbases. The basic idea behind those bases is to take as many important information into account as possible. In this context, a knowledge base $\mathcal{K}$ is supposed to be stratified into $\mathcal{K}_1$, $\ldots$, $\mathcal{K}_n$ ($\mathcal{K}$ = $\mathcal{K}_1 \cup \ldots \cup \mathcal{K}_n$) such that the formulas in the same strata are equally preferred, whereas formulas in a strata $\mathcal{K}_i$ are preferred to formulas in $\mathcal{K}_j$ with $i < j$.

Definition 8 *Let $\mathcal{K}$ = $\mathcal{K}_1 \cup \ldots \cup \mathcal{K}_n$ be a knowledge base. S = $S_1 \cup \ldots \cup S_n$ is a* preferred subbase *of $\mathcal{K}$ if and only if $\forall j = 1, \ldots, n$, $S_1 \cup \ldots \cup S_j$ is a maximal (for set-inclusion) consistent subbase of $\mathcal{K}_1 \cup \ldots \cup \mathcal{K}_j$.*
$INCL(\mathcal{K})$ denotes the set of preferred subbases of $\mathcal{K}$.

We show that in order to capture the results of the above approach, one needs to define the appropriate preference relation between options.

Definition 9 *Let $\mathcal{K}$ be an inconsistent knowledge base, and $\mathcal{O}_{\mathcal{K}}$ be the set of all options that handle $\mathcal{K}$. Let $\mathcal{O}_1$, $\mathcal{O}_2 \in \mathcal{O}_{\mathcal{K}}$. $\mathcal{O}_1 \succeq \mathcal{O}_2$ iff $\mathtt{Base}(\mathcal{O}_1) \in INCL(\mathcal{K})$, and $\mathtt{Base}(\mathcal{O}_2) \notin INCL(\mathcal{K})$.*

Proposition 4 *Let $\langle \mathtt{Opt}(\mathcal{L}), \succeq, \vdash\!\sim \rangle$ be a framework such that $\succeq$ is defined as in Definition 9. Let $\mathcal{K}$ be an inconsistent knowledge base, and $\mathcal{O}^{\succeq}$ be the set of preferred/optimal options for $\mathcal{K}$.*

- *$\forall S_i \in INCL(\mathcal{K})$, $\exists\, \mathcal{O} \in \mathcal{O}^{\succeq}$, such that $\mathcal{O}$ = $\mathsf{CN}(S_i)$.*
- *$\forall \mathcal{O}_i \in \mathcal{O}^{\succeq}$, $\exists\, S \in INCL(\mathcal{K})$ such that $\mathcal{O}_i$ = $\mathsf{CN}(S)$.*

6 New applications of the framework

In this section we show through the two following examples that the above abstract framework can be used for reasoning about inconsistency using temporal logic and even probabilistic logic.

Example 6 *Consider the temporal logic theory T. The $\bigcirc$ operator denotes the "next time instant" operator. Thus, the first rule says that if a is true at time t, b is true at time $(t+1)$. Under standard temporal logic model theory, T is inconsistent.*

$$a \rightarrow \bigcirc b. \quad (4)$$

$$a \quad (5)$$

$$\bigcirc \neg b \quad (6)$$

We may choose to consider just three options: $\mathcal{O}_1$ = $\mathsf{CN}(\{4, 5\}), \mathcal{O}_2 = \mathsf{CN}(\{4, 6\}), \mathcal{O}_3 = \mathsf{CN}(\{5, 6\})$. Suppose now that we can associate a numeric score with each formula, describing the weight of the source that provided the formula. Let us say these scores are 3,1,2 respectively, and the weight of an option $\mathcal{O}_i$ is the sum of the scores of the formulas in $T \cap \mathcal{O}_i$. $\mathcal{O}_i \succeq \mathcal{O}_j$ iff the score of $\mathcal{O}_i$ is greater than or equal to the score of $\mathcal{O}_j$. In this case, the best option is $\mathcal{O}_2$.

Example 7 *Consider the probabilistic logic [Bacchus, 1990] theory T consisting of three formulas $p : [0.3, 0.4]$, $p : [0.44, 0.6]$, $p : [0.41, 0.43]$. Suppose we only consider options that assign a single non-empty probability interval to p. For two atoms $A_1 = p : [L, U]$ and $A_2 = p : [L', U']$, let $diff(A_1, A_2) = abs(L_1 - L_2) + abs(U_1 - U_2)$. Let us say that the score of an option $\mathcal{O} = \{A\}$ is given by $\mathcal{K}_{A' \in T} diff(A, A')$. Suppose we say that option $\mathcal{O}_i \succeq \mathcal{O}_j$ iff $score(\mathcal{O}_i) \leq score(\mathcal{O}_j)$. Intuitively, this means that we are preferring options that change the lower and upper bounds in T as little as possible. In this case, $[0.3, 0.6]$ is a preferred option.*

The reader may be tempted to think that given a $\mathcal{K}$ that is inconsistent, an optimal option may always exist because in the worst case, $\mathsf{CN}(\emptyset)$ seems to be an option. However, this is not correct because we do not require that $\mathsf{CN}(\emptyset)$ be in $\mathtt{Opt}(\mathcal{L})$.

7 Algorithms to Compute Optimal Options that Handle Inconsistency

In this section, we develop procedures to find optimal options for any logic $(\mathcal{L}, \mathsf{CN})$ under a varying set of assumptions. We first define what it means for a formula to be compatible with a given set of formulas.

Definition 10 *Given a consistent set $X \subseteq \mathcal{L}$ and a formula $F \in \mathcal{L}$, we say that F is* compatible *with X iff $X \cup \{F\}$ is consistent. We use the notation $Comp(X)$ to denote the set of all formulas that are compatible with X.*

We now develop an iterative fixpoint computational procedure to find an optimal option.

Definition 11 *Suppose $(\mathcal{L}, \mathsf{CN})$ is a logic, $\mathcal{K}$ is a subset of $\mathcal{L}$, and $\succeq$ is a preference relation on options. We define an operator $\Gamma_{\mathcal{K}}$ associated with $\mathcal{K}$ that maps* sets of options to sets of options *as follows.*

$$\Gamma_{\mathcal{K}}(X) = X \cup \{\mathsf{CN}(Y \cup \{F\}) \mid F \in Comp(Y) \cap \mathcal{K} \wedge Y \in X\}.$$

In other words, $\Gamma_{\mathcal{K}}(X)$ works as follows:

- First, it picks a set Y in X to expand.
- It then finds an $F \in Comp(Y) \cap \mathcal{K}$.
- It then closes $Y \cup \{F\}$ and adds this to the answer, i.e. into $\Gamma_{\mathcal{K}}(X)$.

Clearly, the operator $\Gamma_{\mathcal{K}}$ is monotone under inclusion — moreover, the repeated application of $\Gamma_{\mathcal{K}}$ yields a fixpoint. This is defined as follows.

$$\begin{aligned} \Gamma^0_{\mathcal{K}} &= \{\emptyset\}. \\ \Gamma^{i+1}_{\mathcal{K}} &= \Gamma^i_{\mathcal{K}} \cup \Gamma_{\mathcal{K}}(\Gamma^i_{\mathcal{K}}). \\ \Gamma^{\omega}_{\mathcal{K}} &= \bigcup_i \Gamma^i_{\mathcal{K}}. \end{aligned}$$

Proposition 5 *Suppose $(\mathcal{L}, \mathsf{CN})$ is a logic and $\mathcal{K}$ is a subset of $\mathcal{L}$. Then:*

1. *$\Gamma_{\mathcal{K}}$ is monotonic.*
2. *$\Gamma^{\omega}_{\mathcal{K}}$ is a fixpoint of $\gamma_{\mathcal{K}}$.*
3. *Every set in $\Gamma^{\omega}_{\mathcal{K}}$ is consistent.*

Note that the least fixpoint of Γ_k is the empty set and not $\Gamma^{\omega}_{\mathcal{K}}$ because the latter starts its iteration not with the empty set, but with the set containing the empty set! Fortunately, we can make some concrete statements about $\Gamma^{\omega}_{\mathcal{K}}$.

Proposition 6 *Suppose $(\mathcal{L}, \mathsf{CN})$ is a logic, $(\mathtt{Opt}(\mathcal{L}), \succeq, \mid\!\sim)$ is a general framework for handling inconsistency, and $\mathcal{K}$ is a set of wffs. $\mathcal{O}$ is an optimal option that handles $\mathcal{K}$ iff:*

1. *$\mathcal{O} \in \mathtt{Opt}(\mathcal{L}) \cap \Gamma^{\omega}_{\mathcal{K}}$ and*
2. *there is no $\mathcal{O}' \in \mathtt{Opt}(\mathcal{L}) \cap \Gamma^{\omega}_{\mathcal{K}}$ such that $\mathcal{O}' \succeq \mathcal{O}$.*

The above result suggests an immediate algorithm to find an optimal option for $\mathcal{S}$ regardless of whether the preference relation is monotonic or not.

procedure COO-Naive$(\mathcal{L}, \mathsf{CN}, \mathcal{F}, \mathcal{K}, \mathtt{Opt}(\mathcal{L}), \succeq)$
1. $X = \Gamma^{\omega}_{\mathcal{K}}$;
2. $X = X \cap \mathtt{Opt}(\mathcal{L})$;
3. Return any $\mathcal{O} \in X$ such that there is no $\mathcal{O}' \in X$ such that $\mathcal{O}' \succeq \mathcal{O}$.

One reason for the inefficiency of **COO-Naive** is that it makes no assumptions about the preference relation. However, if the preference relation is known, for example, to be monotone or anti-monotone, then we can do better. **COO-Anti** below shows that when $\succeq$ is assumed to be anti-monotone, we can do better than **COO-Naive**.

procedure COO-Anti$(\mathcal{L}, \mathsf{CN}, \mathtt{Opt}(call), \succeq, \mathcal{K})$
1. $TODO = \{\mathsf{CN}(\emptyset)\}$
2. $SOL = \{\}$;
3. **while** $TODO \neq \emptyset$ **do**
4. Pick $X \in TODO$ s.t. $\nexists Y \in TODO$ s.t. $\mathsf{CN}(Y) \succeq \mathsf{CN}(X)$.
5. $TODO = TODO - \{X\}$
6. **if** $X \in \mathtt{Opt}(\mathcal{L})$ **then**
7. $SOL = (SOL - \{Y \in SOL \mid X \succeq Y\}) \cup \{X\}$;
8. **else** $TODO = TODO \cup \{$
9. $\mathsf{CN}(X \cup \{F\}) \mid F \in Comp(X) \cap \mathcal{K})\}$;
10. **end-while**
11. **return** any $\succeq$-preferred member of SOL.

Intuitively **procedure COO-Anti** generates $\subseteq$-inclusion minimal closed and consistent subsets that are in $\mathtt{Opt}(\mathcal{L})$ using a bottom up procedure and then chooses the best ones. It starts by checking if $\mathsf{CN}(\emptyset)$ is in $\mathtt{Opt}(\mathcal{L})$ - if so, it can be returned because of the anti-monotonicity of $\succeq$. Otherwise, it finds all formulas compatible with it and for each such formula, it finds the logical consequences. Whenever it finds an option that handles $\mathcal{K}$, it adds it to SOL and deletes any option in SOL that is worse than it (according to $\succeq$). This procedure is continued.

Pruning occurs in step (8) of the algorithm where the anti-monotonicity of $\succeq$ is exploited. Moreover, the algorithm proceeds in a greedy fashion, always choosing the best (or one of the best) sets from $TODO$ to expand in step (4).

The reader may think that the first solution found by this bottom up procedure is optimal. Unfortunately, this may not be the case because anti-monotonicity merely states that $\mathcal{O}_1 \subseteq \mathcal{O}_2 \rightarrow \mathcal{O}_1 \succeq \mathcal{O}_2$. It is possible for $\mathcal{O}_1 \succeq \mathcal{O}_2$ to hold even if $\mathcal{O}_1 \not\subseteq \mathcal{O}_2$ and hence, in the anti-monotonic case, one has to generate all $\subseteq$-minimal options before deciding which one is the best.

Proposition 7 *Suppose $(\mathcal{L}, \mathsf{CN})$ is a logic, $(\mathtt{Opt}(\mathcal{L}), \succeq, \mid\!\sim)$ is a general framework for reasoning about inconsistency, and $\mathcal{K}$ is a set of wffs. Further, suppose $\succeq$ is anti-monotonic.*

1. *If there is an optimal option to handle $\mathcal{K}$, then **COO-Anti**$(\mathcal{L}, \mathsf{CN}, \mathtt{Opt}(\mathcal{L}), \succeq, \mathcal{K})$ will return one.*
2. *If **COO-Anti**$(\mathcal{L}, \mathsf{CN}, \mathtt{Opt}(\mathcal{L}), \succeq, \mathcal{K})$ returns $\mathcal{O}$, then $\mathcal{O}$ is an optimal option that handles $\mathcal{K}$.*

Just as in the case of anti-monotonic $\succeq$ preference relations, we can also improve on the **COO-Naive** algorithm when $\succeq$ is monotonic. The **COO-Mon** algorithm assumes monotonicity of $\succeq$ and works in a similar manner as for **COO-Anti** but starts top down and eliminates members of $\mathsf{CN}(\mathcal{K})$ instead of starting the iteration from the empty set. The **Coo-Mon** algorithm uses the notion of a *deletion candidate*.

Definition 12 *Suppose $(\mathcal{L}, \mathsf{CN})$ is a logic and $\mathcal{K}$ is a set of wffs. A set $Y \subseteq \mathcal{K}$ is a* deletion candidate for $\mathcal{K}$ *iff*

1. *$\mathsf{CN}(\mathcal{K} - Y) \subset \mathcal{K}$ and*
2. *$\mathsf{CN}(\mathcal{K} - Y)$ is consistent*

3. *and there is no set $Y' \subset Y$ that satisfies the previous two conditions.*

In other words, a deletion candidate for $\mathcal{S}$ is a wff F whose removal from $\mathcal{S}$ causes at least one formula to no longer be inferrable from $\mathcal{S}$. We use the notation $\mathsf{DelCand}(X)$ to denote the set of all deletion candidates of a set X.

procedure COO-Mon$(\mathcal{L}, \mathsf{CN}, \mathtt{Opt}(\mathcal{L}), \succeq, \mathcal{S}, \chi)$
1. $TODO = \{\mathsf{CN}(\mathcal{K})\}$;
2. $SOL = \{\emptyset\}$;
3. **while** $TODO \neq \emptyset$ **do**
4. Pick $X \in TODO$ s.t. $\nexists Y \in TODO\ Y \succeq X$;
5. $TODO = TODO - \{X\}$;
6. **If** $\mathsf{CN}(X) \in \mathtt{Opt}(\mathcal{L}) \wedge \mathsf{CN}(X) \neq \mathcal{L}$ **then**
7. $SOL = (SOL - (\{ Y \in SOL \mid \mathsf{CN}(X) \succeq Y\})) \cup \{\mathsf{CN}(X)\}$;
8. **else**
9. $TODO = TODO \cup \mathsf{DelCand}(X)$;
10. **end while**
11. **return** $\succeq$-preferred members of SOL.

The **COO-Mon** algorithm works as follows. It starts with $\mathsf{CN}(\mathcal{K})$ and checks if it is consistent. If so, it adds it to SOL and halts. Otherwise, it deletes every possible deletion candidate (minimal set of elements from $\mathcal{K}$ that restore consistency). If any of the resulting options are valid options, then the best one according to $\succeq$ is returned. Otherwise, a set from $TODO$ is selected and further deletion candidates are found. This is repeated until we either find an optimal option that handles $\mathcal{K}$ or there is none.

Proposition 8 *Suppose $(\mathcal{L}, \mathsf{CN})$ is a logic, $(\mathtt{Opt}(\mathcal{L}), \succeq, \mid\!\sim)$ is a general framework for reasoning about inconsistency, and $\mathcal{K}$ is a set of wffs. Further, suppose $\succeq$ is monotonic.*

1. *If there is an optimal option to handle $\mathcal{K}$, then **COO-Anti**$(\mathcal{L}, \mathsf{CN}, \mathtt{Opt}(call), \succeq, \mathcal{K})$ will return one.*
2. *If **COO-Anti**$(\mathcal{L}, \mathsf{CN}, \mathtt{Opt}(call), \succeq, \mathcal{K})$ returns $\mathcal{O}$, then $\mathcal{O}$ is an optimal option that handles $\mathcal{K}$.*

8 Conclusion

In the literature, there are several proposals for handling inconsistency in knowledge bases. These proposals differ in i) the logic to which they apply, and (ii) the criteria used to draw inferences.

In this paper, we have proposed a general and unified framework for reasoning about inconsistency in a wide variety of logics. Indeed, the proposed framework is based on an abstract logic as suggested by Scott in [Scott, 1982]. The framework has three basic components: a set of options that are consistent and closed subsets of well-founded formulas of the logic, a preference relation between the options and en entailment mechanism. The preference relation between options is a general notion. It may capture the requirements used in the literature, such as the maximality, but also other criteria maybe provided by the user for choosing among the options. We have shown that by defining an appropriate preference relation on options, we can capture several existing works such as the subbases defined in [Rescher and Manor, 1970], default logic and Brewkas subtheories. The third component of the framework consists of an entailment mechanism that allows the selection of the inferences to be drawn from the knowledge base. Such mechanism should return an option. This enforces the system to make safe inferences. We have also shown through examples how this abstract framework can be used in different logics such as temporal and probabilistic logics. Another important contribution of the paper is the set of algorithms provided for computing most preferred options.

Acknowledgments. This work was partly supported by AFOSR grants FA95500610405 and FA95500510298, ARO grant DAAD190310202 and by the Joint Institute for Knowledge Discovery.

References

[Bacchus, 1990] F. Bacchus. Representing and reasoning with probabilistic knowledge. In *MIT Press, Cambridge*, pages 268–271, 1990.

[Brewka, 1989] G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. pages 1043–1048. Morgan-Kaufmann, 1989.

[Dung, 1995] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[G. Pinkas, 1992] R. P. Loui G. Pinkas. Reasoning from inconsistency: a taxonomy of principles for resolving conflicts. In *3rd International Conference on Principles of Knowledge Representation and Reasoning, KR'92*, pages 709 –719, 1992.

[Gardenfors, 1988] P. Gardenfors. The dynamics of belief systems: Foundations vs. coherence. *International journal of Philosophy*, 1988.

[Marek *et al.*, 1990] V. Wiktor Marek, A. Nerode, and J. B. Remmel. A theory of nonmonotonic rule systems. *LICS 1990*, pages 79–94, 1990.

[Poole, 1985] D. Poole. On the comparison of theories: preferring the most specific explanation. In *9th International Joint Conference on Artificial Intelligence, IJCAI'85*, pages 144–147, 1985.

[Reiter, 1980] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[Rescher and Manor, 1970] N. Rescher and R. Manor. On inference from inconsistent premises. *Theory and decision*, 1:179–219, 1970.

[Scott, 1982] D. S. Scott. Lectures on a mathematical theory of computation. *Theoretical Foundations of Programming Methodology, . Reidel Publ.*, pages 145–292, 1982.

[Shoenfield, 1967] J. Shoenfield. Mathematical logic. *AK Peters Ltd*, 1967.

[Touretzkey, 1984] D. S. Touretzkey. Implicit ordering of defaults in inheritance systems. In *National Conference on Artificial Intelligence, AAAI'84*, pages 322 – 325, 1984.