ANTHONY HUNTER

PARACONSISTENT LOGICS

1 INTRODUCTION

In practical reasoning, it is common to have "too much" information about some situation. In other words, it is common for there to be classically inconsistent information in a practical reasoning database [Besnard *et al.*, 1995]. The diversity of logics proposed for aspects of practical reasoning indicates the complexity of this form of reasoning. However, central to practical reasoning seems to be the need to reason with inconsistent information without the logic being trivialized [Gabbay and Hunter, 1991; Finkelstein *et al.*, 1994]. This is the need to derive reasonable inferences without deriving the trivial inferences that follow the ex falso quodlibet proof rule that holds in classical logic.

$$\frac{\alpha\ ,\quad \neg\alpha}{\beta}\quad \text{[Ex falso quodlibet]}$$

So for example, from a database $\{\alpha, \neg\alpha, \alpha \rightarrow \beta, \delta\}$, reasonable inferences might include α, $\neg\alpha$, $\alpha \rightarrow \beta$, and δ by reflexivity, β by modus ponens, $\alpha \wedge \beta$ by and introduction, $\neg\beta \rightarrow \neg\alpha$ and so on. In contrast, trivial inferences might include γ, $\gamma \wedge \neg\delta$, etc, by ex falso quodlibet.

Solutions to the problem of inconsistent data include database revision and paraconsistent logics. The first approach effectively removes data from the database to produce a new consistent database. In contrast, the second approach leaves the database inconsistent, but prohibits the logics from deriving trivial inferences. Unfortunately, the first approach means we may loose useful information — we may be forced to make a premature selection of our new database, or we may not even be able to make a selection. We consider here the advantages and disadvantages of the paraconsistent approach.

The primary objective of this chapter is to present a range of paraconsistent logics that give sensible inferences from inconsistent information. We consider (1) Weakly-negative logics which use the full classical language, but a subset of the classical proof theory; (2) Four-valued logics which uses a subset of the classsical language and a subset of the classical proof theory, together with an intuitive four-valued semantics; (3) Quasi-classical logic which uses the full classical language, though data and queries are effectively rewritten by the logic to a conjunctive normal form and reasoning is essentially that of clause finding; and (4) Argumentative logics which reason with consistent subsets of classical formulae.

These options behave in quite different ways with data. None can be regarded as perfect for handling inconsistent information in general. Rather, they provide a spectrum of approaches. However, in all the approaches we cover, we aim to stay close to classical reasoning, since classical logic has many appealing features for knowledge representation and reasoning.

2 CLASSICAL REASONING

In this section, we consider classical reasoning in more detail by presenting some basic definitions that are needed for developing paraconsistent logics.

2.1 Language and proof theory

DEFINITION 1 *Let $\mathcal{L}$ be the set of classical propositional formulae formed from a set of atoms and the $\wedge, \vee, \rightarrow$ and $\neg$ connectives. A database Δ is some subset of $\mathcal{L}$.*

DEFINITION 2 *For each atom $\alpha \in \mathcal{L}$, α is a literal and $\neg\alpha$ is a literal. For $\alpha_1 \vee .. \vee \alpha_n \in \mathcal{L}$, $\alpha_1 \vee .. \vee \alpha_n$ is a clause iff each of $\alpha_1, .., \alpha_n$ is a literal. For $\alpha_1 \wedge .. \wedge \alpha_n \in \mathcal{L}$, $\alpha_1 \wedge .. \wedge \alpha_n$ is in a conjunctive normal form (CNF) iff each of $\alpha_1, .., \alpha_n$ is a clause.*

DEFINITION 3 *For $\alpha_1 \wedge .. \wedge \alpha_n \in \mathcal{L}$, and $\beta \in \mathcal{L}$, $\alpha_1 \wedge .. \wedge \alpha_n$ is in a conjunctive normal form (CNF) of β iff $\alpha_1, .., \alpha_n$ is classically equivalent to β, and $\alpha_1, .., \alpha_n$ is in a CNF.*

For any $\alpha \in \mathcal{L}$, a CNF of α can be produced by the application of distributivity, double negation elimination, and de Morgan laws.

DEFINITION 4 *The relation $\vdash$ is classical consequence, defined in the standard way over $\mathcal{L}$. For a database Δ, $\mathsf{Cn}(\Delta)$ is the set $\{\phi | \Delta \vdash \phi\}$.*

2.2 Properties of consequence relations

The following standard properties of consequence relations have been adapted from those given by Gabbay [Gabbay, 1985] and Gärdenfors and Makinson [Gärdenfors and Makinson, 1993].

DEFINITION 5 *Let $\vdash_x$ be some consequence relation, where $\vdash_x \subseteq \wp(\mathcal{L}) \times \mathcal{L}$. We introduce the following properties:*

$\Delta \vdash_x \alpha$ *if* $\Delta \vdash \alpha$	*(Supraclassicality)*
$\Delta \cup \{\alpha\} \vdash_x \alpha$	*(Reflexivity)*
$\Delta \cup \{\beta\} \vdash_x \gamma$ *if* $\Delta \cup \{\alpha\} \vdash_x \gamma$ *and* $\vdash \alpha \leftrightarrow \beta$	*(Left logical equivalence)*
$\Delta \vdash_x \alpha$ *if* $\Delta \vdash_x \beta$ *and* $\vdash \beta \rightarrow \alpha$	*(Right weakening)*
$\Delta \vdash_x \alpha \wedge \beta$ *if* $\Delta \vdash_x \alpha$ *and* $\Delta \vdash_x \beta$	*(And)*
$\Delta \cup \{\alpha\} \vdash_x \beta$ *if* $\Delta \not\vdash_x \neg\alpha$ *and* $\Delta \vdash_x \beta$	*(Rational monotonicity)*
$\Delta \cup \{\alpha\} \vdash_x \beta$ *if* $\Delta \vdash_x \alpha$ *and* $\Delta \vdash_x \beta$	*(Cautious monotonicity)*
$\Delta \cup \{\alpha\} \vdash_x \beta$ *if* $\Delta \vdash_x \beta$	*(Monotonicity)*
$\Delta \vdash_x \beta$ *if* $\Delta \vdash_x \alpha$ *and* $\Delta \cup \{\alpha\} \vdash_x \beta$	*(Cut)*
$\Delta \vdash \bot$ *if* $\Delta \vdash_x \bot$	*(Consistency preservation)*
$\Delta \vdash_x \alpha \rightarrow \beta$ *if* $\Delta \cup \{\alpha\} \vdash_x \beta$	*(Conditionalization)*
$\Delta \cup \{\alpha\} \vdash_x \beta$ *if* $\Delta \vdash_x \alpha \rightarrow \beta$	*(Deduction)*
$\Delta \cup \{\alpha \vee \beta\} \vdash_x \gamma$ *if* $\Delta \cup \{\alpha\} \vdash_x \gamma$ *and* $\Delta \cup \{\beta\} \vdash_x \gamma$	*(Or)*

These properties have been proposed as desirable conditions of a consequence relation. In particular, identifying the properties that fail indicates the deviation from classical logic.

2.3 Notions of trivialization

In the following, we define the notion of a clause being trivial with respect to a set of formulae, and the notion of a clause being pure with respect to a set of formulae.

DEFINITION 6 *Let $\vdash_x$ be a consequence relation, where $\vdash_x$ is defined by some proof rules. Hence, $\vdash_x \subseteq \wp(\mathcal{L}) \times \mathcal{L}$. The relation $\vdash_x$ is trivializable iff for all α, β, $\{\alpha, \neg\alpha\} \vdash_x \beta$.*

We now require the following function $Atoms(\Delta)$ which gives the set of atoms used in the set of formulae in Δ.

DEFINITION 7 *Let* $\Delta \in \wp(\mathcal{L})$*, and* α*,* β*,* $\gamma_1 \wedge .. \wedge \gamma_n$*,* $\delta_1 \vee .. \vee \delta_n \in \mathcal{L}$*,*

$$Atoms(\Delta \cup \{\beta\}) = Atoms(\{\beta\}) \cup Atoms(\Delta)$$

$$Atoms(\emptyset) = \emptyset$$

$$Atoms(\{\beta\}) = Atoms(\{\gamma\}) \text{ where } \gamma \text{ is the CNF of } \beta$$

$$Atoms(\{\gamma_1 \wedge .. \wedge \gamma_n\}) = Atoms(\{\gamma_1\}) \cup .. \cup Atoms(\{\gamma_n\})$$

$$Atoms(\{\delta_1 \vee .. \vee \delta_n\}) = Atoms(\{\delta_1\}) \cup .. \cup Atoms(\{\delta_n\})$$

$$Atoms(\{\neg\alpha\}) = Atoms(\{\alpha\})$$

$$Atoms(\{\alpha\}) = \{\alpha\} \text{ if } \alpha \text{ is an atom}$$

PROPOSITION 8 *Let* $\Delta \subseteq \mathcal{L}$*, and let* $\mathsf{Cx}(\Delta)$ *denote the consequence closure of* Δ *by* $\vdash_x$*, ie* $\mathsf{Cx}(\Delta) = \{\alpha \mid \Delta \vdash_x \alpha\}$*. For any* $\Delta \subseteq \mathcal{L}$*, where* $\Delta \vdash \bot$*, the consequence relation* $\vdash_x$ *is trivial iff* $\mathsf{Cx}(\Delta) = \mathcal{L}$*.*

A paraconsistent logic has a non-trivializable consequence relation. However, since the notion of trivial is quite general, we have introduced the definition of pure. This captures a notion of relevancy between premises and consequences.

DEFINITION 9 *A clause* $\alpha \in \mathcal{L}$ *is pure with respect to* $\Delta \in \wp(\mathcal{L})$ *iff* $Atoms(\Delta) \cap Atoms(\{\alpha\}) \neq \emptyset$*. A consequence relation* $\vdash_x$ *is pure iff for all* $\alpha \in \mathcal{L}, \Delta \in \wp(\mathcal{L}), \alpha$ *is pure with repect to* Δ*.*

PROPOSITION 10 *If a consequence relation* $\vdash_x$ *is pure, then* $\vdash_x$ *is non-trivializable. However, the converse does not necessarily hold.*

Clearly, the classical consequence relation, $\vdash$, is trivializable and not pure.

2.4 Sectioning the database

One of the most obvious strategies for handling inconsistency in a database is to reason with consistent subsets of the database. Closely related to this approach is to remove information from the database that is causing an inconsistency. Here, we explore some of the issues relating these approaches in the context of classical proof theory.

DEFINITION 11 *Let Δ be a database. Then:*

$$\mathsf{CON}(\Delta) = \{\Pi \subseteq \Delta | \Pi \nvdash \bot\}$$
$$\mathsf{INC}(\Delta) = \{\Pi \subseteq \Delta | \Pi \vdash \bot\}$$
$$\mathsf{MC}(\Delta) = \{\Pi \in \mathsf{CON}(\Delta) | \forall \Phi \in \mathsf{CON}(\Delta) \Pi \not\subset \Phi\}$$
$$\mathsf{MI}(\Delta) = \{\Pi \in \mathsf{INC}(\Delta) | \forall \Phi \in \mathsf{INC}(\Delta) \Phi \not\subset \Pi\}$$
$$\mathsf{FREE}(\Delta) = \bigcap \mathsf{MC}(\Delta)$$

Hence $\mathsf{MC}(\Delta)$ is the set of maximally consistent subsets of Δ; $\mathsf{MI}(\Delta)$ is the set of minimally inconsistent subsets of Δ; and $\mathsf{FREE}(\Delta)$ is the set of information that all maximally consistent subsets of Δ have in common.

PROPOSITION 12 ([Elvang-Goransson and Hunter, 1995]) *Let Δ be a database.*

$$\bigcap \mathsf{MC}(\Delta) = \Delta - \bigcup \mathsf{MI}(\Delta)$$

We can consider a maximally consistent subset of a database as capturing a "plausible" or "coherent" view on the database. For this reason, the set $\mathsf{MC}(\Delta)$ is important in many of the definitions presented in the next section. Furthermore, we consider $\mathsf{FREE}(\Delta)$, which is equal to $\bigcap \mathsf{MC}(\Delta)$, as capturing all the "uncontroversial" information in Δ. In contrast, we consider the set $\bigcup \mathsf{MI}(\Delta)$ as capturing all the "problematical" data Δ.

EXAMPLE 13 Let $\Delta = \{\alpha, \neg\alpha, \alpha \rightarrow \beta, \neg\alpha \rightarrow \beta, \gamma\}$. This gives two maximally consistent subsets, $\Phi_1 = \{\alpha, \alpha \rightarrow \beta, \neg\alpha \rightarrow \beta, \gamma\}$, and $\Phi_2 = \{\neg\alpha, \alpha \rightarrow \beta, \neg\alpha \rightarrow \beta, \gamma\}$. From this $\bigcap \mathsf{MC}(\Delta)$=$\{\alpha \rightarrow \beta, \neg\alpha \rightarrow \beta, \gamma\}$, and a minimally inconsistent subset $\Psi = \{\alpha, \neg\alpha\}$.

PROPOSITION 14 ([Elvang-Goransson and Hunter, 1995]). *Let Δ be a database and $\alpha \in \mathcal{L}$, and let* max *be an operator picking $\subseteq$-maximal elements from a set of sets.*

$$\mathsf{MC}(\Delta \cup \{\alpha\}) = \{\Phi \in \mathsf{MC}(\Delta) | \Phi \vdash \neg\alpha\} \ \cup$$
$$\{\Phi \cup \{\alpha\} | \Phi \in \mathsf{max}\{\Psi \in \mathsf{CON}(\Delta) | \Psi \nvdash \neg\alpha\}\}$$

We now use this proposition to show that $\mathsf{MC}(\Delta \cup \{\beta\})$ can be constructed directly from $\mathsf{MC}(\Delta)$.

EXAMPLE 15 Let $\Delta = \{\alpha, \gamma \wedge (\alpha \vee \neg\beta), \neg\gamma \wedge (\neg\alpha \vee \neg\beta)\}$. Then $\mathsf{MC}(\Delta) = \{\{\alpha, \gamma \wedge (\alpha \vee \neg\beta)\}, \{\alpha, \neg\gamma \wedge (\neg\alpha \wedge \neg\beta)\}\}$. And $\mathsf{MC}(\Delta \cup \{\beta\}) = \{\{\alpha, \gamma \wedge (\alpha \vee \neg\beta), \beta\}, \{\alpha, \neg\gamma \wedge (\neg\alpha \vee \neg\beta)\}, \{\neg\gamma \wedge (\neg\alpha \vee \neg\beta), \beta\}\}$.

PROPOSITION 16 ([Elvang-Goransson and Hunter, 1995]). *Let Δ be a database and $\alpha \in \mathcal{L}$.*

$$\mathsf{FREE}(\Delta \cup \{\alpha\}) \subseteq \mathsf{FREE}(\Delta) \cup \{\alpha\}$$

This result has ramifications for deriving inferences from $\mathsf{FREE}(\Delta)$, since the choice of updating (in the form of either $\mathsf{FREE}(\Delta \cup \{\alpha\})$ or $\mathsf{FREE}(\Delta) \cup \{\alpha\}$) can affect the reasoning.

Reasoning with consistent subsets of the database contrasts significantly with weakly-negative logics, four-valued logic, and quasi-classical logic. However, it forms the basis of argumentative logics.

3 WEAKLY-NEGATIVE LOGICS

To avoid trivialization, weakly-negative logics compromise on classical proof theory. They allow, for example, normal notions of conjunction, such as $\alpha \wedge \beta$ gives α, but they are substantially weaker in terms of negation.

There are a number of ways in which this can be achieved. One way is to weaken classical logic so that ex falso quodlibet and reductio ad absurdum do not hold. This gives a paraconsistent logic called C_ω logic proposed by da Costa [da Costa, 1974].

3.1 Proof theory for C_ω

Below we give a presentation of C_ω. All the schema in the logic C_ω are schema in classical logic.

DEFINITION 17 *The logic C_ω is defined by the following axiom schema together with the modus ponens proof rule.*

$$\begin{gathered}
\alpha \rightarrow (\beta \rightarrow \alpha) \\
(\alpha \rightarrow \beta) \rightarrow ((\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow (\alpha \rightarrow \gamma)) \\
\alpha \wedge \beta \rightarrow \alpha \\
\alpha \wedge \beta \rightarrow \beta \\
\alpha \rightarrow (\beta \rightarrow \alpha \wedge \beta) \\
\alpha \rightarrow \alpha \vee \beta \\
\beta \rightarrow \alpha \vee \beta \\
(\alpha \rightarrow \gamma) \rightarrow ((\beta \rightarrow \gamma) \rightarrow (\alpha \vee \beta \rightarrow \gamma)) \\
\alpha \vee \neg\alpha \\
\neg\neg\alpha \rightarrow \alpha
\end{gathered}$$

This proof theory gives the C_ω consequence relation.

EXAMPLE 18 To illustrate the use of C_ω, consider the following example. In this example, there is a symmetry about whether or not α is a δ. In other words, there is an argument that α is a δ, and an argument that α is $\neg\delta$.

$$\begin{array}{l} \alpha \rightarrow (\beta \wedge \gamma) \\ \gamma \rightarrow \delta \\ \beta \rightarrow \neg\delta \\ \alpha \end{array}$$

Using the proof theory we can derive inferences including α, β and γ. We can also derive both δ and $\neg\delta$.

In C_ω, rules such as modus tollens and disjunctive syllogism fail.

$$\frac{\alpha \rightarrow \beta, \neg\beta}{\neg\alpha} \qquad \text{[Modus tollens]}$$

$$\frac{\alpha \vee \beta, \neg\beta}{\alpha} \qquad \text{[Disjunctive syllogism]}$$

Many useful equivalences fail also such as the following,

$$\neg\alpha \vee \beta \not\equiv \alpha \rightarrow \beta$$

$$\neg\neg\alpha \not\equiv \alpha$$

In this sense weakly-negative logics are sub-systems of classical logic. In particular compromising on negation means that many classical inference steps involving negation fail in weakly-negative logics. But to illustrate the sensitivity of this compromise, consider the following example of reasoning which is not valid in C_ω.

EXAMPLE 19 From the schema,

$$\alpha \rightarrow (\beta \rightarrow \alpha)$$

we can derive in C_ω an axiom

$$\alpha \rightarrow (\neg\beta \rightarrow \alpha)$$

Now assume contraposition, which does not hold in C_ω,

$$(\neg\beta \rightarrow \alpha) \rightarrow (\neg\alpha \rightarrow \beta)$$

By transitivity, this would give

$$\alpha \rightarrow (\neg\alpha \rightarrow \beta)$$

which is a form of ex falso quodlibet. Hence, contraposition cannot be a part of C_ω.

However, the removal of certain classical inference rules means that the propositional connectives in the language do not behave in a classical fashion. In the case of C_ω the classical "sense" of negation – and as a result also the interdefinability of the classical connectives – has been traded in exchange for non-trivialisation. One manifestation of this, according to Besnard [1991], is the following.

EXAMPLE 20 In C_ω, disjunctive syllogism, $((\alpha \vee \beta) \wedge \neg\beta) \rightarrow \alpha$, does not hold, whereas modus ponens, $(\alpha \wedge (\alpha \rightarrow \beta)) \rightarrow \beta$, does hold. So, for example, α does not follow from the database: $\{(\alpha \vee \beta), \neg\beta\}$, whereas α does follow from the database: $\{(\neg\beta \rightarrow \alpha), \neg\beta\}$.

There are many similar examples that could be considered confusing and counter-intuitive from a practical reasoning perspective.

PROPOSITION 21 ([Hunter, 1996b]). *The following properties succeed for the C_ω consequence relation: Reflexivity, And, Monotonicity, Cut, Deduction, Conditionalization, Consistency preservation, and Or.*

PROPOSITION 22 ([Hunter, 1996b]). *The following properties fail for the C_ω consequence relation: Supraclassicality,Left logical equivalence, and Right weakening.*

PROPOSITION 23 ([da Costa, 1974; Hunter, 1996b]). *The C_ω consequence relation is not pure and not trivializable.*

An alternative presentation of C_ω is given by a weaker-than-classical set of classical deduction proof rules [Raggio, 1978].

3.2 A semantic tableau proceedure for C_ω

We now consider a proof proceedure for C_ω by Carnielli *et al.* [1991; 1992]. The method is derived from the semantic tableau proof procedure for classical logic.

DEFINITION 24 *The formula* $\neg(\alpha \wedge \neg\alpha)$ *is not valid in general, but if it does hold for a formula* α*, it is a well-behaved formula, and is denoted* α°*.*

DEFINITION 25 *Each formula* α *is labelled with either a + symbol or a* $-$ *symbol and we call* $+ : \alpha$ *and* $- : \alpha$ *signed formulae.*

Intuitively, $+ : \alpha$, and $- : \alpha$, can be interpreted as α being true, and α being false, respectively. Any set of sets of signed formulae is called a form.

DEFINITION 26 *Let* α *and* β *be two formulae, and let* δ *be other formulae and/or other forms. Below are a set of production rules that can be used to reduce a set of formulae into either a new set of formulae, or set of sets of formulae.*

$$\{\delta, + : (\alpha \wedge \beta)\} => \{\delta, + : \alpha, + : \beta\}$$

$$\{\delta, - : (\alpha \vee \beta)\} => \{\delta, - : \alpha, - : \beta\}$$

$$\{\delta, - : (\alpha \rightarrow \beta)\} => \{\delta, + : \alpha, - : \beta\}$$

$$\{\delta, + : (\neg\neg\alpha)\} => \{\delta, + : \alpha\}$$

$$\{\delta, - : (\neg\alpha)\} => \{\delta, + : \alpha\}$$

$$\{\delta, - : (\neg\neg\alpha)\} => \{\delta, - : \alpha\}$$

$$\{\delta, - : (\alpha \diamond \beta)^\circ\} => \{\delta, - : (\alpha^\circ \diamond \beta^\circ)\}, \textit{where } \diamond \in \{\wedge, \vee, \rightarrow\}$$

$$\{\delta, - : (\alpha \wedge \beta)\} => \{\{\delta, - : \alpha\}, \{\delta, - : \beta\}\}$$

$$\{\delta, + : (\alpha \vee \beta)\} => \{\{\delta, + : \alpha\}, \{\delta, + : \beta\}\}$$

$$\{\delta, + : (\alpha \rightarrow \beta)\} => \{\{\delta, - : \alpha\}, \{\delta, + : \beta\}\}$$

$$\{\delta, + : (\neg\alpha)\} => \{\{\delta, - : \alpha\}, \{\delta, - : \alpha^\circ\}\}$$

Given a form C, we denote by R(C) the result of applying one of the rules to the form. A tableau is a sequence of forms $C_1, .., C_n$*, such that* $C_{i+1} = R(C_i)$*.*

In order to test if a formulae can be inferred from a set of formulae, we label it with the $-$ symbol, add it to the data, and construct a tableau. The formula can be inferred if the tableau is closed. A tableau is closed if every set of formulae of its form is closed, and a set of formulae is closed if there is a formula α for which $+ : \alpha$ and $- : \alpha$ belong to that set.

EXAMPLE 27 Consider the following formulae,

$$\alpha \rightarrow (\beta \wedge \gamma)$$

$$\beta \rightarrow \delta$$

$$\gamma \rightarrow \neg\delta$$

$$\alpha$$

Running the tableau rules for this set, the resulting open tableau is the proposed solution to the problem introduced by the inconsistency. Here we consider only the two main forms of which one is closed and the other is not closed. The rest of the closed forms will be omitted.

$$C_0 = \{+ : (\alpha \rightarrow (\beta \wedge \gamma)), + : (\beta \rightarrow \delta), \\ + : (\gamma \rightarrow \neg\delta), + : \alpha\}$$

$$C_1 = C_0 \cup \{+ : (\beta \wedge \gamma)\}$$

$$C_2 = C_1 \cup \{+ : \beta, + : \gamma\}$$

$$C_3 = C_2 \cup \{+ : \delta, + : (\neg\delta)\}$$

$$C_4 = \{C_3 \cup \{- : \delta\}, C3 \cup \{- : \delta^\circ\}\}$$

The set $C_3 \cup \{- : \delta\}$ is closed and the set $C3 \cup \{- : \delta^\circ\}$ is not closed. This means that we can restrict our considerations to the following set of signed elementary expressions of the open set $\{- : \delta^\circ, + : \delta, + : \beta, + : \gamma, + : \alpha\}$. This set gives us a solution to the problem in the sense that we consider α, β, γ, and δ as holding, but that δ is controversial. This example shows how even though the database is inconsistent, the technique allows us to identify δ and $\neg\delta$ as being central to this inconsistency problem.

A by-product of the proof method is that, as with the classical semantic tableau method, this proof method indicates an interesting semantic characterization of the weakly negative logic.

3.3 Applicability of weakly-negative logic

The logic C_ω is only one of a number of interesting weakly-negative logics. Further proof rules can be added to C_ω to give a stronger, and yet still non-trivializable, logic. For example, PI^s logic by Batens [1980] and VI logic by Arruda [1977]. Other weakly-negative logics can be defined by alternative, but similar weakenings, such as for relevant logics by Anderson and Belnap [1975].

Weakly-negative logics are useful for rule-based reasoning with information since the logic supports modus ponens. They can be used to give guidance on the inconsistency and facilitate actions that should be taken on the database. Furthermore, they can be used without recourse to consistency checks. Finally, paraconsistent logics can be used as a formal basis for truth maintenance [Martins and Shapiro, 1988].

4 FOUR-VALUED LOGIC

The four-valued logic of Belnap [1977] provides an interesting alternative to the weakly-negative logics in that it has an illuminating and intuitive semantic characterization to complement its proof theory.

DEFINITION 28 *The language for four-valued logic is a subset of classical logic. Let $\mathcal{P}$ be the usual set of formulae of classical logic that is formed using the connectives $\neg$, $\wedge$ and $\vee$. Then the set of formulae of the language, denoted Q, is $\mathcal{P} \cup \{\alpha \rightarrow \beta \mid \alpha, \beta \in \mathcal{P}\}$, and hence implication is not nestable.*

DEFINITION 29 *A formula in the language can be one of "true", "false", "both" or "neither", which we denote by the symbols T, F, B, and N, respectively.*

EXAMPLE 30 For the database $\{\alpha, \neg\alpha, \beta\}$, an acceptable assignment of truth values is such that α is B, $\neg\alpha$ is B, β is T, and γ is N.

Intuitively we can view this form of assignment in terms of an "Approximation" lattice (see Figure 1). As more "information" is obtained about a formula, the truth-value "increases". In other words, if we know nothing about a formula, it is N. Then as we gain some information it becomes either T or F. Finally, if we gain too much information it becomes B.

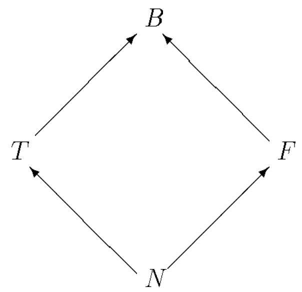

Figure 1. The "Approximation" lattice

4.1 *Semantics for four-valued logic*

DEFINITION 31 *For the semantics, we assume a distributive lattice, the "Logical" lattice (see Figure 2). We also assume an involution operator $*$ satisfying the conditions (1) $\alpha = \alpha^{**}$, and (2) if $\alpha \leq \beta$ then $\beta^* \leq \alpha^*$, where $\leq$ is the ordering relation for the lattice.*

DEFINITION 32 *The semantic assignment function observes monotonicity and complementation, in the logical lattice, so $x \wedge y$ is the meet of $\{x, y\}$ and $x \vee y$ is the join of $\{x, y\}$, giving the following truth tables (Tables 1 – 3) for the $\neg, \wedge, \vee$ connectives. Let α, β be formulae. The inference β from α is valid iff $\beta \leq \alpha$, where $\leq$ is the ordering relation for the logical lattice. Let $\alpha \rightarrow \beta$ signify that the inference from α to β is valid in our four values, ie. that α entails β.*

There is no $\alpha \in \mathcal{Q}$ such that the semantic assignment function always assigns the value T. However, there are formulae that never take the value F, for example $\alpha \vee \neg\alpha$. Though the set of formulae that never take the value F is not closed under conjunction. For example, consider $(\alpha \vee \neg\alpha) \wedge (\beta \vee \neg\beta)$ when α is N and β is B.

α	N	F	T	B
$\neg\alpha$	N	T	F	B

Table 1. Truth table for negation

$\wedge$	N	F	T	B
N	N	F	N	F
F	F	F	F	F
T	N	F	T	B
B	F	F	B	B

Table 2. Truth table for conjunction

$\vee$	N	F	T	B
N	N	N	T	T
F	N	F	T	B
T	T	T	T	T
B	T	B	T	B

Table 3. Truth table for disjunction

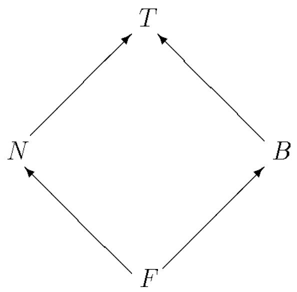

Figure 2. The "Logical" lattice

4.2 *Proof theory for four-valued logic*

To complement the semantics, the following is a definition for the FV consequence relation for the proof theory for four-valued logic.

DEFINITION 33 *Let $\alpha, \beta, \gamma \in \mathcal{L}$. The following are the proof rules for the FV consequence relation.*

$$\alpha_1 \wedge ... \wedge \alpha_m \rightarrow \beta_1 \vee ... \vee \beta_n \textit{ provided some } \alpha_i \textit{ is some } \beta_j$$
$$(\alpha \vee \beta) \rightarrow \gamma \textit{ iff } \alpha \rightarrow \gamma \textit{ and } \beta \rightarrow \gamma$$
$$\alpha \rightarrow (\beta \wedge \gamma) \textit{ iff } \alpha \rightarrow \beta \textit{ and } \alpha \rightarrow \gamma$$
$$\alpha \rightarrow \beta \textit{ iff } \neg\beta \rightarrow \neg\alpha$$
$$\alpha \rightarrow \beta \textit{ and } \beta \rightarrow \gamma \textit{ implies } \alpha \rightarrow \gamma$$
$$\alpha \rightarrow \beta \textit{ iff } \alpha \leftrightarrow (\alpha \wedge \beta) \textit{ iff } \beta \leftrightarrow (\alpha \vee \beta)$$

In addition, the following extends the defintion of the FV consequence relation. Let $\alpha \leftrightarrow \beta$ signify that α and β are semantically equivalent, and can be intersubstituted in any context.

$$\alpha \vee \beta \leftrightarrow \beta \vee \alpha$$
$$\alpha \wedge \beta \leftrightarrow \beta \wedge \alpha$$
$$\alpha \vee (\beta \vee \gamma) \leftrightarrow (\alpha \vee \beta) \vee \gamma$$
$$(\alpha \wedge \beta) \wedge \gamma \leftrightarrow \alpha \wedge (\beta \wedge \gamma)$$
$$\alpha \wedge (\beta \vee \gamma) \leftrightarrow (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$
$$\alpha \vee (\beta \wedge \gamma) \leftrightarrow (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

$$\neg\neg\alpha \leftrightarrow \alpha$$
$$\neg(\alpha \wedge \beta) \leftrightarrow \neg\alpha \vee \neg\beta$$
$$\neg(\alpha \vee \beta) \leftrightarrow \neg\alpha \wedge \neg\beta$$

Also,

$\alpha \leftrightarrow \beta$ *and* $\beta \leftrightarrow \gamma$ *implies* $\alpha \leftrightarrow \gamma$.

EXAMPLE 34 To illustrate the use of the FV consequence relation consider the following example. As with the use of C_ω, there is an argument for δ and an argument for $\neg\delta$.

$$\alpha \rightarrow (\beta \wedge \gamma)$$

$$\gamma \rightarrow \delta$$

$$\beta \rightarrow \neg\delta$$

$$\alpha$$

From $\alpha \rightarrow (\beta \wedge \gamma)$, we get $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$. From $\alpha \rightarrow \beta$ and $\beta \rightarrow \neg\delta$, we get $\alpha \rightarrow \neg\delta$. From $\alpha \rightarrow \gamma$ and $\gamma \rightarrow \delta$, we get $\alpha \rightarrow \delta$. Hence, α is equivalent to $\alpha \wedge \delta$ and $\alpha \wedge \neg\delta$.

However, the FV consequence relation deviates from the C_ω consequence relation in that we cannot detach δ from α nor $\neg\delta$ from α. This is in part due to FV incorporating neither modus ponens nor and elimination.

PROPOSITION 35 ([Hunter, 1996b]). *The following properties succeed for the FV consequence relation: Reflexivity, Consistency Preservation, Monotonicity, and Cut.*

PROPOSITION 36 ([Hunter, 1996b]). *The following properties fail for the FV consequence relation: And, Supraclassicality, Or, Left Logical Equivalence, Deduction, Conditionalization, and Right Weakening.*

PROPOSITION 37 ([Belnap, 1977; Hunter, 1996b]). *The FV consequence relation is not pure and not trivializable.*

PROPOSITION 38 ([Hunter, 1996b]). *For* $\Delta \in \wp(\mathcal{Q})$*, let* $\mathsf{C}\omega(\Delta)$ *denote the set of consequences from* Δ *by the* C_ω *consequence relation, and let* $\mathsf{CFV}(\Delta)$ *denote the set of consequences from* Δ *by the FV consequence relation. For this* $\mathsf{C}\omega(\Delta) \not\subseteq \mathsf{CFV}(\Delta)$*, and* $\mathsf{CFV}(\Delta) \not\subseteq \mathsf{C}\omega(\Delta)$

4.3 Applicability of four-valued logic

Four-valued logic provides a natural and intuitive alternative to weakly-negative logics. The semantic characterization based on the approximation lattice and logical lattice could be applicable for reasoning with facts. In particular, the logic seems useful for aggregating conflicting information. However, there are problems with reasoning with rules, particularly with respect to the lack of modus ponens. As with weakly-negative logics, the FV consequence relation can be used without recourse to consistency checks.

5 QUASI-CLASSICAL LOGIC

As we have seen with weakly-negative logics and with four-valued logics, the weakening of the proof theory means that the connectives do not behave in a classical fashion. To address this, an alternative called quasi-classical logic has been proposed by Besnard and Hunter [Besnard and Hunter, 1995]. In this, queries are rewritten in conjunctive normal form, and the proof theory is restricted to that of finding clauses that follow from the data.

5.1 Proof theory for QC logic

In the following, we present the QC proof rules, which are a subset of the classical proof rules, and we define the notion of a QC proof, which is a restricted version of a classical proof.

DEFINITION 39 *Assume that $\wedge$ is a commutative and associative operator, and $\vee$ is a commutative and associative operator.*

$$\frac{\alpha \wedge \beta}{\alpha} \quad \textit{[Conjunct elimination]}$$

$$\frac{\alpha \vee \alpha \vee \beta}{\alpha \vee \beta} \quad \textit{[Disjunct contraction]}$$

$$\frac{\alpha \vee \beta}{\neg\neg\alpha \vee \beta} \quad \textit{[Negation introduction]}$$

$$\frac{\neg\neg\alpha \vee \beta}{\alpha \vee \beta} \quad \textit{[Negation elimination]}$$

$$\frac{\alpha}{\neg\neg\alpha} \qquad \frac{\neg\neg\alpha}{\alpha\vee}$$

$$\frac{\alpha \vee \beta \quad \neg\alpha \vee \gamma}{\beta \vee \gamma} \; \frac{\alpha \quad \neg\alpha \vee \gamma}{\gamma} \quad \textit{[Resolution]}$$

$$\frac{\alpha \vee (\beta \rightarrow \gamma)}{\alpha \vee \neg\beta \vee \gamma} \quad \frac{\alpha \vee \neg(\beta \rightarrow \gamma)}{\alpha \vee (\beta \wedge \neg\gamma} \quad \textit{[Arrow elimination]}$$

$$\frac{\beta \rightarrow \gamma}{\neg\beta \vee \gamma} \quad \frac{\neg(\beta \rightarrow \gamma)}{\beta \wedge \neg\gamma}$$

$$\frac{\alpha \vee (\beta \wedge \gamma)}{(\alpha \vee \beta) \wedge (\alpha \vee \gamma)} \quad \frac{(\alpha \wedge \beta) \vee) \alpha \wedge \gamma)}{\alpha \wedge (\beta \vee \gamma)} \quad \textit{[Distribution]}$$

$$\frac{\neg(\alpha \wedge \beta) \vee \gamma}{\neg\alpha \vee \neg\beta \vee \gamma} \quad \frac{\neg(\alpha \vee \beta) \vee \gamma}{(\neg\alpha \wedge \neg\beta) \vee \gamma} \quad \textit{[de Morgan laws]}$$

$$\frac{\neg(\alpha \wedge \beta)}{\neg\alpha \vee \neg\beta} \quad \frac{\neg(\alpha \vee \beta)}{\neg\alpha \wedge \neg\beta}$$

$$\frac{\alpha}{\alpha \vee \beta} \quad \textit{[Disjunct introduction—only used as a last step in a proof]}$$

DEFINITION 40 *T is a proof-tree iff T is a tree where (1) each node is an element of $\mathcal{L}$; (2) for the trees with more than one node, the root is derived by application of any QC proof rule, where the premises for the proof rule are the parents of the root; (3) the leaves are the assumptions for the root; and (4) any node, that is not a leaf or root, is derived by the application of any QC proof rule - except the disjunct introduction rule - and the premises for the proof rule are the parents of the node.*

DEFINITION 41 *Let $\Delta \in \wp(\mathcal{L})$. For a clause β, there is a QC proof of β from Δ iff there is a QC proof tree, where each leaf is an element of Δ, and the root is β.*

DEFINITION 42 *Let $\Delta \in \wp(\mathcal{L})$, and $\alpha \in \mathcal{L}$. We define the QC consequence relation, denoted $\vdash_Q$, as follows:*

$$\Delta \vdash_Q \alpha \text{ iff for each } \beta_i \ (1 \leq i \leq n) \text{ there is a QC proof of } \beta_i \text{ from } \Delta$$

where $\beta_1 \wedge .. \wedge \beta_n$ is a CNF of α.

EXAMPLE 43 For $\Delta = \{\alpha \vee \beta, \alpha \vee \neg\beta, \neg\alpha \wedge \delta\}$, consequences of Δ include $\alpha \vee \beta$, $\alpha \vee \neg\beta$, α, $\neg\alpha$, and δ, but do not include $\neg\delta$, γ, $\gamma \vee \phi$, or $\neg\psi \wedge \neg\phi$. For $\Delta = \{\alpha \vee (\beta \wedge \gamma), \neg\beta\}$, consequences of Δ include $\alpha \vee \beta$, $\alpha \vee \gamma$, α, and $\neg\beta$.

PROPOSITION 44 ([Besnard and Hunter, 1995; Hunter, 1996a]). *The following properties succeed for the QC consequence relation: Reflexivity, Consistency preservation, and Monotonicity.*

PROPOSITION 45 ([Besnard and Hunter, 1995; Hunter, 1996a]). *The following properties fail for the QC consequence relation: Cut, Right weakening, Left logical equivalence, and Supraclassicality.*

PROPOSITION 46 ([Besnard and Hunter, 1995; Hunter, 1996a]). *The QC consequence relation is pure, and hence not trivializable.*

PROPOSITION 47 ([Besnard and Hunter, 1995; Hunter, 1996a]). *For $\Delta = \emptyset$, there are no $\alpha \in \mathcal{L}$ such that $\Delta \vdash_Q \alpha$.*

The QC consequence relation offers many more non-tautological inferences from data than either the weakly-negative or four-valued logics. For example, via disjunctive syllogism, QC logic gives β from $\{\neg\alpha, \alpha \vee \beta\}$, whereas neither the weakly-negative logic C_ω nor the four-valued logic gives β.

PROPOSITION 48 ([Hunter, 1996b]). *For $\Delta \in \wp(\mathcal{L})$, let $\mathsf{C}\omega(\Delta)$ denote the set of C_ω consequences from Δ, and let $\mathsf{CQ}(\Delta)$ denote the set of QC consequences from Δ. For this $\mathsf{CQ}(\Delta) \not\subseteq \mathsf{C}\omega(\Delta)$ and $\mathsf{C}\omega(\Delta) \not\subseteq \mathsf{CQ}(\Delta)$ hold.*

This proposition follows from the classical tautologies from the empty set not being derivable in QC logic. However, if we exclude consideration of these tautologies, then we see that QC logic is stronger than C_ω.

PROPOSITION 49 ([Hunter, 1996b]). *For $\Delta \in \wp(\mathcal{Q})$, let $\mathsf{CFV}(\Delta)$ denote the set of FV consequences from Δ, and let $\mathsf{CQ}(\Delta)$ denote the set of QC consequences from Δ. For this $\mathsf{CQ}(\Delta) \not\subseteq \mathsf{CFV}(\Delta)$ and $\mathsf{CFV}(\Delta) \not\subseteq \mathsf{CQ}(\Delta)$ hold.*

5.2 Semantics for QC logic

There is a semantic counterpart to the QC proof theory [Hunter, 1996a]. To simplify the discussion for this chapter, we restrict the coverage to just clauses — i.e. disjunctions $\alpha_1 \vee .. \vee \alpha_n$, where $\{\alpha_1, .., \alpha_n\}$ is a set of literals.

DEFINITION 50 *Let S be some set. Let O be a set of objects defined as follows, where $+\alpha$ is a positive object, and $-\alpha$ is a negative object.*

$$O = \{+\alpha \mid \alpha \in S\} \cup \{-\alpha \mid \alpha \in S\}$$

We call any $X \in \wp(O)$ a model.

We can consider the following meaning for positive and negative objects being in or out of some model X,

- $+\alpha \in X$ means α is "satisfiable" in the model
- $-\alpha \in X$ means $\neg\alpha$ is "satisfiable" in the model
- $+\alpha \notin X$ means α is not "satisfiable" in the model
- $-\alpha \notin X$ means $\neg\alpha$ is not "satisfiable" in the model

This semantics can also be regarded as giving four truth values, called "Both", "True", "False", "Neither". For a literal α, and its complement α^*,

- α is "Both" if α is satisfiable and α^* is satisfiable
- α is "True" if α is satisfiable and α^* is not satisfiable

- α is "False" if α is not satisfiable and α^* is satisfiable
- α is "Neither" if α is not satisfiable and α^* is not satisfiable

This intuition coincides with that of four-valued logic. However, we will not follow the four-valued lattice-theoretic interpretation of the connectives, and instead provide a significantly different semantics.

DEFINITION 51 *Let* $\models_s$ *be a satisfiability relation, where* $X \in \wp(O)$*, and* α*,* α_1*,..,*α_n *are literals. Let* α_i *be a literal in the set of literals* $\{\alpha_1, .., \alpha_n\}$*, and let* β *be the disjunction formed from the set of literals* $\{\alpha_1, .., \alpha_n\} - \{\alpha_i\}$*.*

$$X \models_s \alpha \textit{ if } +\alpha \in X$$

$$X \models_s \neg\alpha \textit{ if } -\alpha \in X$$

$$X \models_s \alpha_1 \vee .. \vee \alpha_n \textit{ iff } ((X \models_s \alpha_1 \textit{ or .. or } X \models_s \alpha_n) \textit{ and (if } X \models_s \neg\alpha_i \textit{ then } X \models_s \beta)$$

Satisfaction for disjunction therefore incorporates a link between the satisfaction of the complement of a disjunct and the satisfaction of the corresponding resolvant.

DEFINITION 52 *We extend the notion of satisfaction to that of weak satisfaction, denoted as* $\models_w$*, as follows, where* $X \in \wp(O)$ *and* α *and* β *are clauses.*

$$X \models_w \alpha \textit{ if } X \models_s \alpha$$

$$X \models_w \alpha \vee \beta \textit{ if } X \models_s \alpha$$

DEFINITION 53 *Let* $\models_Q$ *be an entailment relation defined as follows, where* $\{\alpha_1, ..\alpha_n\}$ *is a set of clauses and* β *is a clause.*

$$\{\alpha_1, .., \alpha_n\} \models_Q \beta$$
$$\textit{iff for all models } X \textit{ if } X \models_s \alpha_1 \textit{ and ... and } X \models_s \alpha_n \textit{ then } X \models_w \beta$$

EXAMPLE 54 Let $\Delta = \{\alpha\}$, and let $X1 = \{+\alpha\}$ and $X2 = \{+\alpha, -\alpha\}$. Now $X1 \models_s \alpha$, and $X2 \models_s \alpha$, whereas $X1 \models_s \alpha \vee \beta$, and $X2 \not\models_s \alpha \vee \beta$. However, $X1 \models_w \alpha \vee \beta$, and $X2 \models_w \alpha \vee \beta$, and indeed $\Delta \models_Q \alpha \vee \beta$. As another example, let $\Delta = \{\alpha \vee \beta, \neg\alpha\}$. For all models X, if $X \models_s \alpha \vee \beta$, and $X \models_s \neg\alpha$, then $X \models_s \beta$. Hence, $\Delta \models_Q \alpha \vee \beta$, $\Delta \models_Q \neg\alpha$, and $\Delta \models_Q \beta$.

PROPOSITION 55 *For all* $\alpha \in \mathcal{L}$, $\{\} \not\models_Q \alpha$.

EXAMPLE 56 Consider $\alpha \vee \neg\alpha$. Here models that satisfy $\{\}$ include those, for example X, where $+\alpha \notin X$ and $-\alpha \notin X$, and so $X \not\models_Q \alpha$ and $X \not\models_Q \neg\alpha$ hold. Hence, it is not the case that for all models X that satisfies the empty set, X also satisfies $\alpha \vee \neg\alpha$.

PROPOSITION 57 *For a language restricted to clauses, the* $\vdash_Q$ *relation is sound and complete with respect to the* $\models_Q$ *relation.*

The semantics and correctness result for the full QC language is in [Hunter, 1996a].

5.3 Applicability of quasi-classical logic

Developing a non-trivializable, or paraconsistent logic, necessitates some compromise, or weakening, of classical logic. The compromises imposed to give QC logic seem to be more appropriate than other paraconsistent logics for applications in computing. QC logic provides a means to obtain all the non-trivial resolvants from a set of formulae, without the problem of trivial clauses also following. Though the constraints on QC logic result in tautologies from an empty set of assumptions being non-derivable, this is not usually a problem for applications.

QC logic exhibits the nice feature that no attention need to be paid to a special form that premises should have. This is in contrast with other paraconsistent logics where two formulae identical by definition of a connective in classical logic may not yield the same set of conclusions. An example given earlier in this paper is $\{(\neg\alpha \rightarrow \beta), \neg\alpha\}$ yielding the conclusion β, whereas $\{\alpha \vee \beta, \neg\alpha\}$ does not. QC logic is much better behaved in this respect, as illustrated by the the fact that more non-trivial classical conclusions are captured by QC-logic.

QC logic is also more appropriate than various approaches to reasoning from consistent subsets of inconsistent sets of formulae (for a review, see [Benferhat, Dubois and Prade, 1993]). In particular, QC logic does not suffer from the limitation due to spliting sets of formulae into compatible subsets: QC logic can make use of the contents of the formulas without being constrained by a consistency check. Moreover, it is obviously an advantage of QC logic to dispense with the costly consistency checks that are needed in all approaches to reasoning from consistent subsets.

6 ARGUMENT SYSTEMS

In Section 2.4, we reviewed the strategy of handling inconsistency by reasoning with consistent subsets of the database. The problem for such an approach is that inferences that follow from consistent subsets of an inconsistent database are only weakly justified in general. To handle this problem the notion of an argument from a database, and a notion of acceptability of an argument have been developed by Elvang and Hunter [1995]. An argument is a subset of the database, together with an inference from that subset. Using the notion of acceptability, the set of all arguments can be partitioned into sets of (arguments of) different degrees of acceptability. This can then be used to define a class of consequence relations.

DEFINITION 58 *Let Δ be a database. An argument from Δ is a pair, (Π, ϕ), such that $\Pi \subseteq \Delta$ and $\Pi \vdash \phi$. An argument is consistent, if Π is consistent. We denote the set of arguments from Δ as $\mathsf{An}(\Delta)$, where $\mathsf{An}(\Delta) = \{(\Pi, \phi) | \Pi \subseteq \Delta \wedge \Pi \vdash \phi\}$. Γ is an argument set of Δ iff $\Gamma \subseteq \mathsf{An}(\Delta)$.*

DEFINITION 59 *Let Δ be a database. Let (Π, ϕ) and (Θ, ψ) be any arguments constructed from Δ. If $\vdash \phi \leftrightarrow \neg\psi$, then (Π, ϕ) is a rebutting defeater of (Θ, ψ). If $\gamma \in \Theta$ and $\vdash \phi \leftrightarrow \neg\gamma$, then (Π, ϕ) is an undercutting defeater of (Θ, ψ).*

Rebutting defeat, as defined here, is a symmetrical relation. One way of changing this is by use of priorities, such as in epistemic entrenchment [Gärdenfors, 1988] or as in specificity [Poole, 1985].

For a database Δ, an argumentative structure is any set of subsets of $\mathsf{An}(\Delta)$. The intention behind the definition for an argumentative structure is that different subsets of $\mathsf{An}(\Delta)$ have different degrees of acceptability. Below, we present one particular argumentative structure A^*, and then explain how the definition captures notions of acceptability.

DEFINITION 60 *The following sets constitute the argumentative structure A^*, where Δ is a database.*

$$\begin{array}{lcl}
\mathsf{AT}(\Delta) & = & \{(\emptyset, \phi) | \emptyset \vdash \phi\} \\
\mathsf{AF}(\Delta) & = & \{(\Pi, \phi) | \Pi \subseteq \mathsf{FREE}(\Delta) \wedge \Pi \vdash \phi\} \\
\mathsf{AB}(\Delta) & = & \{(\Pi, \phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge \\
& & (\forall \Phi \in \mathsf{MC}(\Delta), \psi \in \Pi \Phi \vdash \psi)\} \\
\mathsf{ARU}(\Delta) & = & \{(\Pi, \phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge \\
& & \quad (\forall \Phi \in \mathsf{MC}(\Delta) \Phi \nvdash \neg\phi) \wedge \\
& & \quad (\forall \Phi \in \mathsf{MC}(\Delta), \psi \in \Pi \Phi \nvdash \neg\psi)\}
\end{array}$$

$$
\begin{aligned}
\mathsf{AU}(\Delta) &= \{(\Pi,\phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge \\
&\quad (\forall \Phi \in \mathsf{MC}(\Delta), \psi \in \Pi \Phi \nvdash \neg\psi)\} \\
\mathsf{A}\forall(\Delta) &= \{(\Pi,\phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge \\
&\quad (\forall \Phi \in \mathsf{MC}(\Delta) \Phi \vdash \phi)\} \\
\mathsf{AR}(\Delta) &= \{(\Pi,\phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge \\
&\quad (\forall \Phi \in \mathsf{MC}(\Delta) \Phi \nvdash \neg\phi)\} \\
\mathsf{A}\exists(\Delta) &= \{(\Pi,\phi) | \Pi \in \mathsf{CON}(\Delta) \wedge \Pi \vdash \phi\}
\end{aligned}
$$

The naming conventions for the argument sets are motivated as follows. T is for the tautological arguments—i.e. those that follow from the empty set of premises. F is for the free arguments—(due to Benferhat *et al.* [1993])—which are the arguments that follow from the data that is free of inconsistencies. B is for the backed arguments - i.e. those for which all the premises follow from all the maximally consistent subsets of the data. RU is for the arguments that are not subject to either rebutting or undercutting. U is for the arguments that are not subject to undercutting. $\forall$ is for the universal arguments—(essentially due to Manor and Rescher [1970]), where it was called inevitable arguments)—which are the arguments that follow from all maximally consistent subsets of the data. R is for the arguments that are not subject to rebutting. $\exists$ is for existential arguments—(essentially due to Manor and Rescher [1970])—which are the arguments with consistent premises.

The definitions for $\mathsf{A}\exists$, AF, AT should be clear. We therefore focus on the remainder. AR allows an argument (Π,ϕ) only if there is no maximally consistent subset that gives $\neg\phi$. AU allows an argument (Π,ϕ) only if for all items ψ in Π, there is no maximally consistent subset that gives $\neg\psi$. ARU combines the conditions of the AR and AU. Notice that AR and $\mathsf{A}\forall$ have very similar definitions, with the only difference being "$\Phi \nvdash \neg\phi$" in AR versus "$\Phi \vdash \phi$" in $\mathsf{A}\forall$. A similar remark applies to AU and AB. Therefore $\mathsf{A}\forall$ and AB are strengthenings of AR and AU, respectively (i.e. "$\nvdash \neg\phi$" replaced with "$\vdash \phi$").

EXAMPLE 61 We give an example of a database, and some of the items in each argument set. Take $\Delta = \{\alpha, \neg\alpha\}$. Then $(\{\alpha,\neg\alpha\}, \alpha \wedge \neg\alpha) \in \mathsf{An}(\Delta)$, $(\{\alpha\},\alpha) \in \mathsf{A}\exists(\Delta)$, $(\{\alpha\}, \alpha \vee \beta) \in \mathsf{AR}(\Delta)$, if $\beta \nvdash \alpha$, $(\{\}, \alpha \vee \neg\alpha) \in \mathsf{A}\forall(\Delta)$. Furthermore, $\mathsf{A}\forall(\Delta) = \mathsf{AF}(\Delta) = \mathsf{AB}(\Delta) = \mathsf{ARU}(\Delta) = \mathsf{AU}(\Delta) = \mathsf{AT}(\Delta)$.

EXAMPLE 62 As another example, consider $\Delta = \{\neg\alpha \wedge \beta, \alpha \wedge \beta\}$. Then for $\Pi = \{\alpha \wedge \beta\}$, $(\Pi,\beta) \in \mathsf{A}\exists(\Delta)$, $(\Pi,\beta) \in \mathsf{AR}(\Delta)$, and $(\Pi,\beta) \in \mathsf{A}\forall(\Delta)$. But there is no $\Pi \subseteq \Delta$ such that $(\Pi,\beta) \in \mathsf{AU}(\Delta)$, $(\Pi,\beta) \in \mathsf{ARU}(\Delta)$, $(\Pi,\beta) \in \mathsf{AB}(\Delta)$, or $(\Pi,\beta) \in \mathsf{AF}(\Delta)$.

$$
\begin{array}{c}
\mathsf{AT}(\Delta) \\
| \\
\mathsf{AF}(\Delta) = \mathsf{AB}(\Delta) = \mathsf{ARU}(\Delta) = \mathsf{AU}(\Delta) \\
| \\
\mathsf{A}\forall(\Delta) \\
| \\
\mathsf{AR}(\Delta) \\
| \\
\mathsf{A}\exists(\Delta) \\
| \\
\mathsf{An}(\Delta)
\end{array}
$$

Figure 3. Partial order on A^* induced by $\subseteq$

PROPOSITION 63 ([Elvang-Goransson and Hunter, 1995]).

$$
\begin{array}{c}
\mathsf{AT}(\Delta) \subseteq \mathsf{AF}(\Delta) = \mathsf{AB}(\Delta) = \mathsf{ARU}(\Delta) = \\
\mathsf{AU}(\Delta) \subseteq \mathsf{A}\forall(\Delta) \subseteq \mathsf{AR}(\Delta) \subseteq \mathsf{A}\exists(\Delta) \subseteq \mathsf{An}(\Delta)
\end{array}
$$

We summarize this result by the diagram in Figure 3. The main features to notice are that A^* is a linear structure, and that there is an equivalence of AF, AB, ARU, and AU.

6.1 Argumentative logics induced by A^*

Each argument set in A^* induces a consequence relation. In the following we let "x" syntactically denote an arbitrary member of the suffixes: "T, F, B, RU, U, $\forall$, R, $\exists$, n".

DEFINITION 64 *A consequence closure for each argumentative structure is denoted* Cx*, where* $\mathsf{x} \in \{\mathsf{T}, \mathsf{F}, \mathsf{B}, \mathsf{RU}, \mathsf{U}, \forall, \mathsf{R}, \exists, \mathsf{n}\}$*, and defined as follows,*

$$\mathsf{Cx}(\Delta) = \{\phi | \exists \Pi \subseteq \Delta (\Pi, \phi) \in \mathsf{Ax}(\Delta)\}.$$

A consequence relation, denoted $\vdash_{\mathsf{x}}$, can then be generated from the consequence closure as expected.

$$\Delta \vdash_{\mathsf{x}} \phi \text{ iff } \phi \in \mathsf{Cx}(\Delta).$$

Clearly, $\vdash_n$ is classical entailment, but we continue to omit the subscript.

PROPOSITION 65 ([Elvang-Goransson and Hunter, 1995]).

$$\begin{gathered}\mathsf{CT}(\Delta) \subseteq \mathsf{CF}(\Delta) = \mathsf{CB}(\Delta) = \mathsf{CRU}(\Delta) = \\ \mathsf{CU}(\Delta) \subseteq \mathsf{C}\forall(\Delta) \subseteq \mathsf{CR}(\Delta) \subseteq \mathsf{C}\exists(\Delta) \subseteq \mathsf{Cn}(\Delta).\end{gathered}$$

In Figure 4, we summarize properties of the argumentative consequence relations [Elvang-Goransson and Hunter, 1995]. The weaker conditions on $\mathsf{C}\exists$ and CR allow some of the classical axioms on the consequence relations to hold such as left logical equivalence and right weakening, but that others such as cut, the and property, and the or property fail. In contrast for the more restricted definitions for $\mathsf{C}\forall$, CU, CRU, CB and CF, monotonicity fails, but cut and the and property are preserved. A number of properties succeed for all these consequence relations such as left logical equivalence, right weakening, and conditionalization, whereas some properties fail for all (except the Cn-consequence relation) such as supraclassicality and deduction.

Note how even though a property may fail for a consequence relation, further restrictions on the acceptability of arguments may cause the property to hold. For example, the properties of and, cautious monotony, and cut fail for CR, but succeed for $\mathsf{C}\forall$. For the or property, increasing the restrictions on acceptability causes failure for $\mathsf{C}\exists$ and CR, success for $\mathsf{C}\forall$, and then failure for CU.

The argumentative logics are, in a key sense, far more restricted than the other paraconsistent logics considered in this chapter: If a pair of formulae are mutually inconsistent, then none of these argumentative logics will derive any consequences from the conjunction of the two formulae. This is not the case with any of the weakly-negative, four-valued or quasi-classical logics. However, the following results show that argumentative logics offer useful alternatives to the other paraconsistent logics that we have considered.

PROPOSITION 66 ([Hunter, 1996b]). *For all* $\mathsf{x} \in \mathsf{T}, \mathsf{F}, \mathsf{B}, \mathsf{RU}, \mathsf{U}, \forall, \mathsf{R}, \exists$, *the consequence relation* $\vdash_{\mathsf{x}}$ *is not pure and not trivializable.*

PROPOSITION 67 ([Hunter, 1996b]). *For* $\Delta \in \wp(\mathcal{L})$*, let* $\mathsf{C}\omega(\Delta)$ *denote the set of* C_ω *consequences from* Δ*, and let* $\mathsf{Cx}(\Delta)$ *denote the set of consequences from* Δ *by a* $\vdash_{\mathsf{x}}$ *consequence relation. For this* $\mathsf{Cx}(\Delta) \not\subseteq \mathsf{C}\omega(\Delta)$ *and* $\mathsf{C}\omega(\Delta) \not\subseteq \mathsf{Cx}(\Delta)$ *hold.*

PROPOSITION 68 ([Hunter, 1996b]). *For* $\Delta \in \wp(\mathcal{Q})$*, let* $\mathsf{CFV}(\Delta)$ *denote the set of FV consequences from* Δ*, and let* $\mathsf{Cx}(\Delta)$ *denote the set of consequences from* Δ *by a* x *consequence relation. For this* $\mathsf{Cx}(\Delta) \not\subseteq \mathsf{CFV}(\Delta)$ *and* $\mathsf{CFV}(\Delta) \not\subseteq \mathsf{Cx}(\Delta)$ *hold.*

Properties	Cn	C∃	CR	C∀	CU	CT
Supraclassicality	•	∘	∘	∘	∘	∘
Reflexivity	•	∘	∘	∘	∘	∘
Left logical equivalence	•	•	•	•	•	•
Right weakening	•	•	•	•	•	•
And	•	∘	∘	•	•	•
Rational monotony	•	•	∘	∘	∘	•
Cautious monotony	•	•	∘	•	•	•
Monotonicity	•	•	∘	∘	∘	•
Cut	•	∘	∘	•	•	•
Consistency preservation	•	•	•	•	•	•
Conditionalization	•	•	•	•	•	•
Deduction	•	∘	∘	∘	∘	∘
Or	•	∘	∘	•	∘	•

Figure 4. Summary of properties. Symbols: $\bullet$, for success and $\circ$, for failure

PROPOSITION 69 ([Hunter, 1996b]). *For* $\Delta \in \wp(\mathcal{L})$*, let* $\mathsf{CQ}(\Delta)$ *denote the set of QC consequences from* Δ*, and let* $\mathsf{Cx}(\Delta)$ *denote the set of consequences from* Δ *by a* x *consequence relation. For this* $\mathsf{Cx}(\Delta) \not\subseteq \mathsf{CQ}(\Delta)$ *and* $\mathsf{CQ}(\Delta) \not\subseteq \mathsf{Cx}(\Delta)$ *hold.*

The proposition above follows from the classical tautologies from the emptyset not being derivable in QC logic. However, if we exclude consideration of these tautologies, then we see that QC logic is stronger than $\mathsf{C}\exists$ consequence relation.

6.2 Applicability of argumentative logics

The concept of an argumentative structure, with the two notions of argument and acceptability, are a convenient framework for developing practical reasoning tools. Although, they are based on simple definitions of arguments and acceptability, the concepts carry many possibilities for further refinement. It remains to be seen whether there is a general taxonomy of argumentative structures, such as suggested by Pinkas and Loui [1992], and universal properties of the logics that they induce.

There are also a number of other argument-based systems that have been proposed, including by Vreeswijk [1991], Prakken [1993], and Simari and Loui [1992]. These differ from argumentative logics in that they focus on defeasible reasoning: They incorporate defeasible, or default, connectives into their languages, together with associated machinery.

Another approach to acceptability of arguments is by Dung [1993]. This approach assumes a set of arguments, and a binary "attacks" relation between pairs of subsets of arguments. A hierarchy of arguments is then defined in terms of the relative attacks "for" and "against" each argument in each subset of arguments. In this way, for example, the plausibility of an argument could be defended by another argument in its subset.

7 DISCUSSION

Paraconsistent logics allow useful conclusions from data. They are robust, in the sense that the conclusions are "sensible" with respect to the data. There is no obligation to resolve the inconsistency, the logics functions satisfactorily irrespective of the inconsistency remaining. The logics can give guidance on the source of the inconsistency, up to the deductive power of the paraconsistent logic.

Unfortunately paraconsistent logics only localize inconsistency. They don't offer strategies for acting on inconsistency. In contrast, many approaches force consistency without consideration of the environment in which the data is used. Truth maintenance systems (de Kleer [1986], Doyle [1979]), and belief revision theory (Gardenfors [1988]) ensure consistency by rejecting formulae upon finding inconsistency. Similarly, Fagin et al [1983] propose amending the database when finding inconsistency during updating. Even more restrictive is the use of integrity constraints in databases, which prohibit inconsistent data even entering the database.

Yet intellectual activities usually involve reasoning with different perspectives. For example, consider negotiation, learning, or merging multiple opinions. For these, maintaining absolute consistency is not always possible. Often it is not even desirable since this can unnecessarily constrain the intellectual activity, and can lead to the loss of important information. Indeed since the real-world forces us to work with inconsistencies, we should formalize some of the usually informal or extra-logical ways of responding to them. This is not necessarily done by eradicating inconsistencies, but rather by supplying logical rules specifying how we should act on them [Gabbay and Hunter, 1991;

Gabbay and Hunter, 1993a].

In this chapter, we have restricted consideration to a classical language with no extra connectives or other notation. However, there have been a variety of interesting proposals for reasoning with inconsistent information that do extend the language. These include extending the language with labels and markers, priorities, default and defeasible connectives, and modal operators. We leave a more detailed discussion of usage of defeasible, default, and modal operators to chapters X, Y and Z respectively in this volume, and consider the remainder in the following.

In practical reasoning, object-level formulae can represent many kinds of useful information. However, it is possible to augment the syntactic information by semantic or meta-level information as proposed by Gabbay in Labelled Deductive Systems [Gabbay, 1993]. For a formula α, this augmentation can allow the clear expression of extra information such as:

- fuzzy reliability of α
- origin of α
- priority of α
- time when α holds
- possible world where α holds
- a proof of α - in for example truth maintenance systems

The extra information contained in this kind of augmentation is then used by the logic to affect the outcome of the reasoning. We can represent the extra information for a formula α by some label i, and then represent this as $i : \alpha$, where the label is always juxtaposed to the formula.

Using labelling we generalize our notion of a database. If a formula $i : \alpha$ is in a database Δ, then we do not necessarily assume that α is "true". The meaning assigned to α is in part dependent on the label. So for example, i could mean that α was "true" yesterday, but not necessarily today, or i could mean that α is "true" if there is no formula $k : \beta$ such that is k more preferred to i. In this way, we actually only reason with the subset of our data that is actually "true".

Such languages, for example Gabbay and Hunter [1993b], involve uniquely labelling the data, and amending the proof rules to propagate the labels: The consequent of each proof rule has a label that is a function of the labels of the premises. In this way, any inferences from the logic are labelled with information about the data and proof rules used to derive them. This means we can track information used in reasoning and hence analyse inconsistencies as they arise. We can identify likely sources of the problem, and use this to suggest appropriate actions [Hunter and Nuseibeh, 1995].

There are a number of attempts to accommodate inconsistent data in a database by labelling. We can view ideas of truth maintenance in this way [Fehrer, 1993]. Also, Balzer [Balzer, 1991] suggests "guards" on inconsistent data to minimize the negative ramifications, and then to warn the user of the inconsistency, and in Naqvi and Rossi [1990] inconsistent data is allowed to enter the database, but the time that the data is entered is recorded, and newer the data takes precedence over the older data when resolving inconsistencies.

Finally, priorities have been used in a number of ways in the management of inconsistency (for example see [Benferhat, Dubois and Prade, 1995]), and in the closely related problem of non-monotonic reasoning. This includes the use of specificity [Poole, 1985], ordered theory presentations [Ryan, 1992], and prioritized syntax-based entailment [Benferhat *et al.*, 1993].

Acknowledgements

This work has been partly supported by the ESPRIT BRA DRUMS2 project and by the EPSRC VOILA project. Thanks are due to Salem Benferhat and Torsten Schaub for reading an earlier draft of this chapter.

University College, London.

REFERENCES

[Anderson and Belnap, 1975] A. Anderson and N. Belnap. *Entailment: The Logic of Relevance and Necessity*. Princeton University Press, 1975.

[Arruda, 1977] A. Arruda. On the imaginary logic of NA Vasilev. In A Arruda, N Da Costa, and R Chuaqui, editors, *Non-classical logics, model theory and computability*. North Holland, 1977.

[Balzer, 1991] R. Balzer. Tolerating inconsistency. In *proceedings of the 13th International Conference on Software Engineering*, pages 158–165. IEEE Press, 1991.

[Batens, 1980] D. Batens. Paraconsistent extensional propositional logics. *Logique et Analyse*, 90–91:195–234, 1980.

[Benferhat *et al.*, 1993] S. Benferhat, C. Cayrol, D. Dubois, J. Lang and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 1993.

[Besnard *et al.*, 1995] Ph. Besnard, L. Farinas del Cerro, D. Gabbay and A. Hunter. Logical handling of default and inconsistent information. In Ph Smets and A Motro, editors, *Uncertainty management in information systems*. Kluwer, 1995.

[Benferhat, Dubois and Prade, 1993] S. Benferhat, D. Dubois and H. Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proceedings of Uncertainty in Artificial Intelligence*, pages 1449–1445. Morgan Kaufmann, 1993.

[Benferhat, Dubois and Prade, 1995] S. Benferhat, D. Dubois and H. Prade. A logical approach to reasoning under inconsistency in stratified knowledge bases. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, volume 956 of *Lecture Notes in Computer Science*, pages 36–43. Springer, 1995.

[Belnap, 1977] N. Belnap. A useful four-valued logic. In G. Epstein, editor, *Modern Uses of Multiple-valued Logic*, pages 8–37. Reidel, 1977.

[Besnard, 1991] Ph. Besnard. Paraconsistent logic approach to knowledge representation. In M. de Glas and D. Gabbay, editors, *Proceedings of the First World Conference on Fundamentals of Artificial Intelligence*. Angkor, 1991.

[Besnard and Hunter, 1995] Ph. Besnard and A. Hunter. Quasi-classical logic: Non-trivializable classical reasoning from inconsistent information. In C. Froidevaux and J. Kohlas, editors, *Symbolic and Quantitative Approaches to Uncertainty*, Lecture Notes in Computer Science. Springer, 1995.

[Carnielli *et al.*, 1991] W. Carnielli, L. Farinas del Cerro and M. Lima-Marques. Contextual negation and reasoning with contradiction. In *Proceedings of the Eleventh International Conference on Artificial Intelligence*. Morgan Kaufmann, 1991.

[Carnielli and Lima-Marques, 1992] W. Carnielli and M. Lima-Marques. Reasoning under inconsistent knowledge. *Journal of Applied Non-Classical Logics*, 2:49–79, 1992.

[da Costa, 1974] N. C. da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15:497–510, 1974.

[Doyle, 1979] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[Dung, 1993] P. Dung. The acceptability of arguments and its fundamental role in non-monotonic reasoning and logic programming. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 852–857, 1993.

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

[Fehrer, 1993] D. Fehrer. A unifying framework for reason maintenance. In M. Clarke, R. Kruse, and S. Moral, editors, *Symbolic and Qualitative Approaches to Reasoning and Uncertainty (ECSQARU'93)*, volume 747 of *Lecture Notes in Computer Science*, pages 113–120. Springer, 1993.

[Finkelstein *et al.*, 1994] A. Finkelstein, D. Gabbay, A. Hunter, J. Kramer and B. Nuseibeh. Inconsistency handling in multi-perspective specifications. *Transactions on Software Engineering*, 20(8):569–578, 1994.

[Fagin, Ullman and Vardi, 1983] R. Fagin, J. Ullman and M. Vardi. On the semantics of updates in databases. In *Proceedings of the Second Annual Association of Computing Machinery Symposium on Principles of Database Systems*. ACM Press, 1983.

[Gabbay, 1985] D. Gabbay. Theoretical foundations of non-monotonic reasoning in expert systems. In K. Apt, editor, *Logics and Models of Concurrent Systems*. Springer, 1985.

[Gabbay, 1993] D. Gabbay. Labelled deductive systems: A position paper. In J. Oikkonnen and J. Vaananen, editors, *Proceedings of the Logic Colloquium'90*, volume 2 of *Lecture Notes on Logic*. Springer, 1993.

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

[Gabbay and Hunter, 1991] D. Gabbay and A. Hunter. Making inconsistency respectable 1: A position paper. In Ph. Jorrand and J. Keleman, editors, *Fundamentals of Artificial Intelligence Research*, volume 535 of *Lecture Notes in Artificial Intelligence*. Springer, 1991.

[Gabbay and Hunter, 1993a] D. Gabbay and A. Hunter. Making inconsistency respectable 2: Meta-level handling of inconsistent data. In M. Clarke, R. Kruse and S. Moral, editors, *Symbolic and Qualitative Approaches to Reasoning and Uncertainty (ECSQARU'93)*, volume 747 of *Lecture Notes in Computer Science*, pages 129–136. Springer, 1993.

[Gabbay and Hunter, 1993b] D. Gabbay and A. Hunter. Restricted access logics for inconsistent information. In M. Clarke, R. Kruse and S. Moral, editors, *Symbolic and Qualitative Approaches to Reasoning and Uncertainty (ECSQARU'93)*, volume 747 of *Lecture Notes in Computer Science*, pages 137–144. Springer, 1993.

[Gärdenfors and Makinson, 1993] P. Gärdenfors and D. Makinson. Non-monotonic inference based on expectations. *Artificial Intelligence*, 65:197–246, 1993.

[Hunter and Nuseibeh, 1995] A. Hunter and B. Nuseibeh. Managing inconsistent specifications: Reasoning, analysis, and action. Technical report, Department of Computing, Imperial College, London, 1995.

[Hunter, 1996a] A. Hunter. Reasoning with inconsistent information using quasi-classical logic. Technical report, Department of Computing, Imperial College, London, 1996.

[Hunter, 1996b] A. Hunter. Some results on paraconsistent logics. Technical report, Department of Computing, Imperial College, London, 1996.

[De Kleer, 1986] J. De Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

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

[Martins and Shapiro, 1988] J. Martins and S. Shapiro. A model of belief revision. *Artificial Intelligence*, 35:25–79, 1988.

[Naqvi and Rossi, 1990] S. Naqvi and F. Rossi. Reasoning in inconsistent databases. In *Logic Programming: Proceedings of the North American Conference*. MIT Press, 1990.

[Pinkas and Loui, 1992] G. Pinkas and R. Loui. Reasoning from inconsistency: A taxonomy of principles for resolving conflict. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*. Morgan Kaufmann, 1992.

[Poole, 1985] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1985.

[Prakken, 1993] H. Prakken. An argumentation framework in default logic. *Annals of Mathematics and Artificial Intelligence*, 9, 1993.

[Raggio, 1978] A. Raggio. A proof-theoretic analysis of da Costa's C_ω. In A. Arrunda, N. C. da Costa and R. Chuaqui, editors, *Mathematical Logic: Proceedings of the First Brazilian Conference*. Marcel Defalnes, 1978.

[Ryan, 1992] M. Ryan. Representing defaults as sentences with reduced priority. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*. Morgan Kaufmann, 1992.

[Simari and Loui, 1992] G. Simari and R. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53:125–157, 1992.

[Vreeswijk, 1991] G. Vreeswijk. Abstract argumentation systems. In M. de Glas and D. Gabbay, editors, *Proceedings of the First World Conference on Fundamentals of Artificial Intelligence*. Angkor, 1991.