A Bilattice-based Approach to Recover Consistent Data from Inconsistent Knowledge-Bases

Ofer Arieli
Department of Computer Science
School of Mathematical Sciences
Tel-Aviv University
Ramat-Aviv 69978, ISRAEL.
Email: ofera@math.tau.ac.il

Arnon Avron
Department of Computer Science
School of Mathematical Sciences
Tel-Aviv University
Ramat-Aviv 69978, ISRAEL.
Email: aa@math.tau.ac.il

Abstract

***Bilattices*, which have been shown to be very useful in logic programming, are used here for recovering consistent data from an inconsistent knowledge-base. Our method is *conservative* in the sense that it considers the contradictory data as useless, and regards all the remaining information unaffected. This kind of approach is nonmonotonic and paraconsistent in nature.**

1 Introduction

One of the most significant drawbacks of the classical calculus is its inability to deduce nontrivial results in the presence of an inconsistency. The warranty of drawing *any* conclusion whatsoever just because of the existence of a contradiction, is certainly unrealistic. Nevertheless, the classical calculus is still a very convenient framework to work with; adding new mechanisms or connectives to it generally cause a considerable growth in the computational complexity needed to maintain the resulting system. The purpose of this work, then, is to propose means that would allow drawing conclusions from systems that are based on classical logic, although the information might become temporarily inconsistent.

The scenario we think of is the following: Suppose that a given first-order classical knowledge-base has become inconsistent. For the reason stated above, any attempt to deduce meaningful inferences from this "polluted" knowledge-base is useless. Our approach solves this "dead-end" by temporarily expanding the semantics to be multi-valued, and this *without changing the knowledge-base syntacticly.*

How do we practically recover consistent data from an inconsistent knowledge-base without changing it? The first step, as we have implied, is to switch into a special multi-valued framework. For this, we use a special algebraic structures called *bilattices.* Bilattices were first proposed by Ginsberg (see [Gi88]) as a basis for a general framework for many applications. This notion was further developed by Fitting ([Fi90a, Fi94]), who showed that bilattices are most suitable for logic programming ([Fi89, Fi90b, Fi91, Fi93]). In bilattices the elements (which are also referred to as "truth values") are simultaneously arranged in two partial orders: one, $\leq_t$, may intuitively be understood as a measure of the degree of *truth* that each element represents; the other, $\leq_k$, describes (again, intuitively) differences in the amount of *knowledge* (or *information*) that each element exhibits on the assertions that it is supposed to represent.

The next step is to develop a mechanism that enables paraconsistent inferences. For this, we use an epistemic entailment proposed in [KL92] as the consequence relation of the logic [1]. This relation can be viewed as a kind of a "closed word assumption", since it considers only the "most consistent" models

[1] See [AA94c] for a detailed discussion on the advantages of the present logic with respect to the logic of [KL92] in particular, and to annotated logics [Su90a, Su90b, KS92] in general.

(mcms) of a given set of assertions. As was shown in [AA94a, AA94b], this relation enjoys some appealing features, such as being non-monotonic, paraconsistent ([dC74]), and a "plausibility logic" ([Le92]).

By using $\models_{con}$ we are able to: a) discover easily the "core" of the inconsistency in KB, and b) construct consistent subsets of the knowledge-base (called "support sets"), which are useful means to override the contradictions when focusing the attention on certain (recoverable) formulae. These support sets are the candidates to be the "recovered" knowledge-base. The common feature shared by each support set is that it considers some contradictory information as useless, and regards all the remaining information not depending on it as unaffected. This kind of approach is called in [Wa94] *conservative* (*skeptical*), and it apparently has not yet been studied in the literature (refer also to [Wa94, p.107]).

2 Preliminary definitions and notations

In this section we briefly review the notions that will be significant in what follows. For a more detailed presentation of the following notions, refer to [AA94b].

2.1 Bilattices

Definition 2.1 [Gi88] A *bilattice* is a structure $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ such that B is a non empty set containing at least two elements; $(B, \leq_t)$, $(B, \leq_k)$ are complete lattices; and $\neg$ is a unary operation on B s.t.: (a) if $a \leq_t b$ then $\neg a \geq_t \neg b$, (b) if $a \leq_k b$ then $\neg a \leq_k \neg b$, (c) $\neg\neg a = a$.

Notation 2.2 Following Fitting, we shall use $\wedge$ and $\vee$ for the meet and join which correspond to $\leq_t$, and $\otimes$, $\oplus$ for the meet and join under $\leq_k$. He suggested to intuitively understand $\otimes$ and $\oplus$ as representing the "consensus" and "accept all" operations, respectively [2]. f and t will denote, respectively, the least and the greatest element w.r.t. $\leq_t$, while $\bot$ and $\top$ – the least and the greatest element w.r.t. $\leq_k$. While t and f may have their usual intuitive meaning, $\bot$ and $\top$ could be thought of as representing no information and inconsistent knowledge, respectively. Obviously, $f, t, \bot$ and $\top$ are all different (see lemma 2.3).

Lemma 2.3 [Gi88] Let $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ be a bilattice, and suppose that $a, b \in B$.
a) $\neg(a \wedge b) = \neg a \vee \neg b$; $\neg(a \vee b) = \neg a \wedge \neg b$; $\neg(a \otimes b) = \neg a \otimes \neg b$; $\neg(a \oplus b) = \neg a \oplus \neg b$.
b) $\neg f = t$; $\neg t = f$; $\neg\bot = \bot$; $\neg\top = \top$.

Definition 2.4 [AA94a] Let $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ be a bilattice.
a) A *bifilter* is a nonempty set $\mathcal{F} \subset B$, s.t: (i) $a \wedge b \in \mathcal{F}$ iff $a \in \mathcal{F}$ and $b \in \mathcal{F}$, (ii) $a \otimes b \in \mathcal{F}$ iff $a \in \mathcal{F}$ and $b \in \mathcal{F}$.
b) A bifilter $\mathcal{F}$ is *prime*, if it also satisfies: (i) $a \vee b \in \mathcal{F}$ iff $a \in \mathcal{F}$ or $b \in \mathcal{F}$, (ii) $a \oplus b \in \mathcal{F}$ iff $a \in \mathcal{F}$ or $b \in \mathcal{F}$.

Given a bilattice $\mathcal{B}$, it may contain many bifilters. The elements of a bifilter are taken to be *designated* truth values of $\mathcal{B}$; i.e, they represent formulae that are considered true.

Definition 2.5 Given a bilattice $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$. Two natural candidates for being the set of the designated values of $\mathcal{B}$ are $\mathcal{D}_k(\mathcal{B}) = \{b \in B \mid b \geq_k t\}$ and $\mathcal{D}_t(\mathcal{B}) = \{b \in B \mid b \geq_t \top\}$.

Lemma 2.6 Let $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ be a bilattice. For every $b \in B$, $\{b, \neg b\} \subseteq \mathcal{D}_t(\mathcal{B})$ iff $b = \top$.
Proof: $\{b, \neg b\} \subseteq \mathcal{D}_t(\mathcal{B})$ iff $b \geq_t \top$ and $\neg b \geq_t \top$ iff $b \geq_t \top$ and $b \leq_t \neg\top = \top$ iff $b = \top$. □

Definition 2.7 [AA94a] A *logical bilattice* is a pair $(\mathcal{B}, \mathcal{F})$, where $\mathcal{B}$ is a bilattice, and $\mathcal{F}$ is a prime bifilter.

Example 2.8 Belnap's *FOUR*, Ginsberg's *DEFAULT*, and *NINE* (figure 1) are all logical bilattices with $\mathcal{F} = \mathcal{D}_k(\cdot) = \mathcal{D}_t(\cdot)$. In case of *FOUR* and *DEFAULT* there is no other bifilter. *NINE* induces also another logical bilattice, in which $\mathcal{F} = \mathcal{D}_k(NINE) \cup \{of, d\top, dt\}$.

[2] These operators would not play a central role in what follows, since we will be most interested in the "classical" operators $\wedge$ and $\vee$. However, our method allows the usage of these operators without any further effort, so we shall refer to then as well.

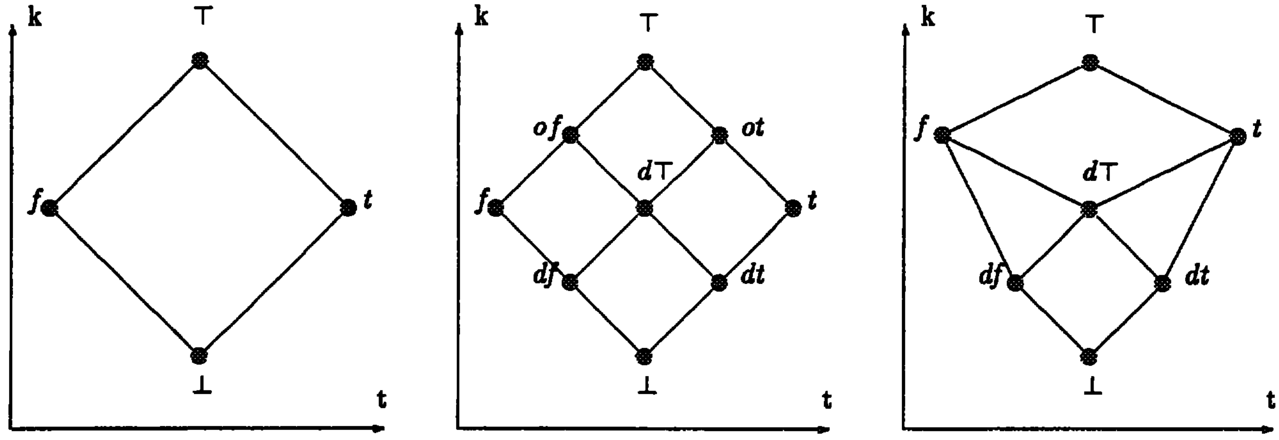

Figure 1: *FOUR*, *NINE*, and *DEFAULT*

2.2 The logic

Definition 2.9 Let $\mathcal{B}$ be an arbitrary bilattice. The language $BL(\mathcal{B})$ (Bilattice-based Language over $\mathcal{B}$) is the standard propositional language over $\{\wedge, \vee, \neg, \otimes, \oplus\}$ enriched with a propositional constant for each element in B. In what follows we shall fix $\mathcal{B}$, so we shall shorten $BL(\mathcal{B})$ to BL.

Definition 2.10 Let KB be a set of formulae above BL.
a) $\mathcal{A}(KB)$ denotes the set of the atomic formulae that appear in some formula of KB.
b) $\mathcal{L}(KB)$ denotes the set of the literals that appear in some formula of KB.

Definition 2.11
a) A *valuation* ν is a function that assigns a truth value from B to each atomic formula, and map every constant to its corresponding value in B. Any valuation is extended to complex formulas in the standard way. We shall sometimes denote $\psi{:}b \in \nu$ (or $\nu = \{\psi{:}b, \ldots\}$) instead of $\nu(\psi) = b$.
b) Given $(\mathcal{B}, \mathcal{F})$, we will say that ν *satisfies* ψ ($\nu \models \psi$), iff $\nu(\psi) \in \mathcal{F}$.
c) A valuation that satisfies every formula in a given set of formulas, KB, is said to be a *model* of KB. The set of the models of KB will be denoted *mod*(KB).

The next notion describes the truth values of B that represent inconsistent beliefs:

Definition 2.12 [AA94a] Let $(\mathcal{B}, \mathcal{F})$ be a logical bilattice. A subset $\mathcal{I}$ of B is an *inconsistency set*, if it has the following properties: (a) $b \in \mathcal{I}$ iff $\neg b \in \mathcal{I}$, (b) $b \in \mathcal{F} \cap \mathcal{I}$ iff $b \in \mathcal{F}$ and $\neg b \in \mathcal{F}$ [3].

Example 2.13 $\mathcal{I}_1 = \{b \mid b \in \mathcal{F} \wedge \neg b \in \mathcal{F}\}$ is the minimal inconsistency set in every logical bilattice. $\mathcal{I}_2 = \{b \mid b = \neg b\}$ is inconsistency set in *FOUR*, *DEFAULT*, and *NINE* with $\mathcal{F} = \mathcal{D}_k(\cdot)$ [4].

In the following discussion we fix some logical bilattice $(\mathcal{B}, \mathcal{F})$ as well as an inconsistency set $\mathcal{I}$ of it. $l \in \mathcal{L}(KB)$ will denote an arbitrary literal, $\bar{l}$ – its complementary, and $p, q \in \mathcal{A}(KB)$ – atomic formulae.

Notation 2.14 Given a valuation M on KB. Denote: $Inc_M(KB) = \{p \in \mathcal{A}(KB) \mid M(p) \in \mathcal{I}\}$.

[3] Note that by (b) of definition 2.12, always $\top \in \mathcal{I}$ and $t \notin \mathcal{I}$. Hence, by (a), $f \notin \mathcal{I}$.

[4] Note that $\bot \notin \mathcal{I}_1$, while $\bot \in \mathcal{I}_2$. Indeed, one of the major considerations when choosing an inconsistency set, is whether to include $\bot$ in $\mathcal{I}$ or not. Although in every bilattice $\neg\bot = \bot$ (see lemma 2.3), $\bot$ intuitively reflects no knowledge whatsoever about the assertion it represents; in particular one might not take such assertions as inconsistent.

Definition 2.15 Let M, N be two models of a finite set of formulae, KB.
a) M is *more consistent* than N ($M >_{con} N$), iff $Inc_M(KB) \subset Inc_N(KB)$.
b) M is a *most consistent* model of KB (mcm, in short), if there is no other model of KB which is more consistent than M. The set of all the mcms of KB will be denoted by $con(KB)$.
c) M is *smaller* than N (with respect to $<_k$), $M <_k N$, if for any $p \in \mathcal{A}(KB)$, $M(p) \leq_k N(p)$, and there is $q \in \mathcal{A}(KB)$ s.t $M(q) <_k N(q)$.
d) M is a *minimal* model of KB, if there in no other model of KB which is smaller than M. The set of all the minimal models of KB will be denoted by $min(KB)$.

Definition 2.16 Let KB be a finite sets of formulae and ψ – a formula. Let S be any set of valuations. We denote $KB \models_S \psi$ if each model of KB which is in S, is also a model of ψ.

Some particularly interesting instances of definition 2.16 are the following: (1) $KB \models_{mod(KB)} \psi$ if every model of KB is a model of ψ (abbreviation: $KB \models \psi$), (2) $KB \models_{con(KB)} \psi$ if every mcm of KB is a model of ψ (abbreviation: $KB \models_{con} \psi$), and (3) $KB \models_{min(KB)} \psi$ if every minimal model of KB is a model of ψ (abbreviation: $KB \models_{min} \psi$).

Example 2.17 Let $KB = \{s, \neg s, r_1, r_1 \rightarrow \neg r_2, r_2 \rightarrow d\}$ and $\mathcal{B} = FOUR$ with $\mathcal{F} = \{t, \top\}$. The models of KB are listed in figure 2 below. It follows that $con(KB) = \{M_1, M_2, M_3\}$ provided that $\bot \notin \mathcal{I}$, while if $\bot \in \mathcal{I}$, $con(KB) = \{M_2, M_3\}$. Also, $min(KB) = \{M_1, M_9\}$, thus $KB \models_{con} \neg r_2$, while $KB \not\models \neg r_2$ and $KB \not\models_{min} \neg r_2$.

Model No.	s	r_1	r_2	d	Model No.	s	r_1	r_2	d
M_1	$\top$	t	f	$\bot$	M_9	$\top$	$\top$	$\bot$	t
M_2	$\top$	t	f	t	M_{10}	$\top$	$\top$	$\bot$	$\top$
M_3	$\top$	t	f	f	$M_{11} - M_{12}$	$\top$	$\top$	t	$t, \top$
M_4	$\top$	t	f	$\top$	$M_{13} - M_{16}$	$\top$	$\top$	f	$\bot, t, f, \top$
$M_5 - M_8$	$\top$	t	$\top$	$\bot, t, f, \top$	$M_{17} - M_{20}$	$\top$	$\top$	$\top$	$\bot, t, f, \top$

Figure 2: The models of KB

2.3 The knowledge-bases

Definition 2.18 A formula ψ over BL is an *extended clause*, if ψ is a literal, or $\psi = \phi \vee \varphi$, where ϕ and φ are extended clauses, or $\psi = \phi \oplus \varphi$, where ϕ and φ are extended clauses.

Definition 2.19 A formula ψ is said to be *normalized*, if it has no subformula of the form $\phi \vee \phi$, $\phi \wedge \phi$, $\phi \oplus \phi$, $\phi \otimes \phi$, or $\neg\neg\phi$.

Lemma 2.20 For every formula ψ there is an equivalent normalized formula ψ' such that for every valuation ν, $\nu(\psi) \in \mathcal{F}$ iff $\nu(\psi') \in \mathcal{F}$.

From now on, unless otherwise stated, the knowledge-bases that we shall consider are finite sets of normalized extended-clauses. As the next proposition shows, presenting the formulae in an (normalized) extended clause form does not reduce the generality:

Proposition 2.21 For every formula ψ over BL there is a set S of normalized extended clauses such that for every valuation M, $M \models \psi$ iff $M \models S$.
Proof: By an induction on the structure of the negation normal form of ψ. □

Lemma 2.22 Let ψ be an extended clause over BL, l_i ($i = 1 \ldots n$) – its literals, and ν – a valuation on $\mathcal{A}(\psi)$. Then $\nu \models \psi$ iff there is $1 \leq i \leq n$ s.t. $\nu \models l_i$.
Proof: By an induction on the structure of ψ. □

3 Classification of the atomic formulae

The first step to recover inconsistent situations is to identify the atomic formulae that are involved in the conflicts. In order to do so, we divide the atomic formulae that appear in the clauses of the knowledge-base into four subsets as follows:

Definition 3.1 Let KB be a set of formulae, and $l \in \mathcal{L}(KB)$.
a) If $KB \models_{con} l$ and $KB \models_{con} \bar{l}$, then l is said to be *spoiled.*
b) If $KB \models_{con} l$ and $KB \not\models_{con} \bar{l}$, then l is said to be *recoverable.*
c) If $KB \not\models_{con} l$ and $KB \not\models_{con} \bar{l}$, then l is said to be *damaged.* [5]

Example 3.2 In example 2.17, s is spoiled, r_1 and $\neg r_2$ are recoverable, and d is damaged.

3.1 The spoiled literals

We treat first those literals that form, as their name suggests, the "core" of the inconsistency in KB. As the following theorem suggests, those literals are very easy to detect:

Theorem 3.3 Let KB be a knowledge-base, and $l \in \mathcal{L}(KB)$. The following conditions are equivalent: **(a)** l is a spoiled literal of KB. **(b)** $M(l) \in \mathcal{I} \cap \mathcal{F}$ for every model M of KB. **(c)** $M'(l) \in \mathcal{I} \cap \mathcal{F}$ for every mcm M' of KB. **(d)** $\{l, \bar{l}\} \subseteq KB$.
Proof: The only nontrivial transition in the sequence (a) → (c) → (d) → (b) → (a) is (c) → (d): Suppose that for every mcm M' of KB, $M'(l) \in \mathcal{I} \cap \mathcal{F}$. Since KB is finite, then for every model of KB there is an mcm of KB which is more consistent or equal to it. Hence l is assigned some inconsistent truth value in *every* model of KB. Assume that $l \in \{p, \neg p\}$ for some $p \in \mathcal{A}(KB)$, and consider the following valuations: $\nu_t = \{q : \top \mid q \in \mathcal{A}(KB),\ q \neq p\} \cup \{p : t\}$, $\nu_f = \{q : \top \mid q \in \mathcal{A}(KB),\ q \neq p\} \cup \{p : f\}$. Since ν_t is *not* a model of KB (because p has a consistent value under ν_t), $\neg p \in KB$ (otherwise, every formula $\psi \in KB$ contains a literal l' s.t. $\nu_t(l') \in \mathcal{F}$, and so $\nu_t \models \psi$ by lemma 2.22). Similarly, since ν_f is not a model of KB, $p \in KB$. □

Corollary 3.4 If $\mathcal{F} = \mathcal{D}_t(\mathcal{B})$ then every model of KB assigns $\top$ to every spoiled literal.
Proof: Immediate from (d) of theorem 3.3, since $\{b, \neg b\} \in \mathcal{D}_t(\mathcal{B})$ iff $b = \top$ (see lemma 2.6). □

3.2 The recoverable literals and their support sets

The recoverable literals are those that may be viewed as the "robust" part of a given inconsistent knowledge-base, since all the mcms "agree" on their validity. As we shall see, each recoverable literal l can be associated with a consistent subset, which preserves the information about l.

Definition 3.5 Let KB be a finite set of normalized extended clause in BL.
a) A model M of KB is *consistent* if it assigns a consistent value to every atom that appears in $\mathcal{A}(KB)$.
b) KB is *consistent* if it has a consistent model.
c) A subset $KB' \subseteq KB$ is *consistent in KB* if KB' is a consistent set, and (at least) one of its consistent models is expandable to a (not necessarily consistent) model of KB.

Example 3.6 $KB' = \{q\}$ is a consistent set, which is *not* consistent in $KB = \{q, \neg q\}$, since there is no consistent model of KB' that is expandable to a model of KB.

Definition 3.7 A set of normalized extended clauses $SS(l)$ is a *support set* of l (or: $SS(l)$ *supports* l), if $SS(l)$ satisfies the following conditions: **(a)** $SS(l) \subseteq KB$; $SS(l)$ is not empty, **(b)** $SS(l)$ is consistent in KB, **(c)** $SS(l) \models_{con} l$ and $SS(l) \not\models_{con} \bar{l}$.

[5] As noted before, these notions (but not the context) are taken from [KL92].

Definition 3.8 If $SS(l)$ supports l, and there is no support set $SS'(l)$ s.t. $SS(l)\subset SS'(l)$, then $SS(l)$ is said to be a *maximal support set* of l, or *recovered subset* of KB. A knowledge-base that has a recovered subset is called *recoverable*.

Example 3.9 Consider again the example given in 2.17 and 3.2: $KB=\{s,\ \neg s,\ r_1,\ r_1\rightarrow\neg r_2,\ r_2\rightarrow d\}$. Here KB is a recoverable knowledge-base, since $S=\{r_1,\ r_1\rightarrow\neg r_2,\ r_2\rightarrow d\}$ is a maximal support set of both r_1 and $\neg r_2$. Note that S does *not* support d, since $S\not\models_{con} d$.

Theorem 3.10 Every recoverable literal has a support set.
Outline of proof: Without loss of generality, suppose that $l=p$, where $p\in\mathcal{A}(KB)$ is recoverable. Let M be an mcm of KB, s.t. $M(p)\in\mathcal{F}\setminus\mathcal{I}$. Suppose that $\{r_1,r_2,\ldots,r_n\}$ are the members of $\mathcal{A}(KB)\setminus Inc_M(KB)$. Define: $SS'(p)=\{\psi\in KB \mid \mathcal{A}(\psi)\subseteq\{r_i\}_{i=1}^{m}\}$. This $SS'(p)$ is a support set for p. □

Theorem 3.11 If l is a recoverable literal in KB, then no subset of KB supports $\bar{l}$.
Proof: Without a loss of generality, suppose that $l=p$, and assume that $SS'(p)$ is a support set for $\neg p$. $SS'(p)$ has a consistent model, M', which is expandable to a model M of KB. M preserves the valuations of M' on $\mathcal{A}(SS'(p))$, so in particular $M(q)=M'(q)\notin\mathcal{I}$ for every $q\in\mathcal{A}(SS'(p))$. Let N be an mcm of KB s.t. $N\geq_{con}M$ (such an N exists since KB is finite). Since $N\geq_{con}M$ then still $N(q)\notin\mathcal{I}$ for every $q\in\mathcal{A}(SS'(p))$. Also, N is an mcm of KB and p is a recoverable atom of KB, hence $N(p)\in\mathcal{F}$. Let N' be the reduction of N to $SS'(p)$. Since N' is identical to N on $\mathcal{A}(SS'(p))$, and since N is a model of KB, then: (a) N' is a model of $SS'(p)$, (b) $N'(q)\notin\mathcal{I}$ for every $q\in\mathcal{A}(SS'(p))$, and (c) $N'(p)\in\mathcal{F}$. Form (a) and (b), then, N' is a consistent model of $SS'(p)$, and so from (c), $N'(\neg p)\notin\mathcal{F}$. Thus $SS'(p)\not\models_{con}\neg p$; a contradiction. □

Proposition 3.12 Let l be a literal s.t. $KB\models_{con}l$. l is recoverable iff it has a support set.
Proof: The "only if" part was proved in theorem 3.10. For the "if" direction note that since l has a support set, it cannot be spoiled. l cannot be damaged either, since $KB\models_{con}l$. This is also the reason why $\bar{l}$ cannot be recoverable. The only possibility left, then, is that l is recoverable. □

Corollary 3.13 Every literal l such that $\{l\}\subseteq KB$ and $\{\bar{l}\}\not\subseteq KB$ is recoverable [6].
Proof: It is easy to see that if $\{\bar{l}\}\not\subseteq KB$ then $SS(l)=\{l\}$ is a support set of l (not necessarily maximal). Since $SS(l)\models l$ and $\models$ is monotonic, then $KB\models l$ as well, and so $KB\models_{con}l$. By 3.12, l is recoverable. □

Another refinment of proposition 3.12 is the following: For $KB'\subset KB$ denote by $con(KB)\downarrow KB'$ the *reductions* of the mcms of KB to the language of KB'. Then:

Proposition 3.14 l is a recoverable literal iff it has a support set $SS(l)$ s.t. $SS(l)\models_{con(KB)\downarrow SS(l)}l$.
Proof: If l is recoverable, then by theorem 3.10 it must have a support set $SS(l)$. Also, since l is recoverable, it is assigned a designated truth value by every mcm. These values are kept when reducing the mcms to the language of $SS(l)$, hence $SS(l)\models_{con(KB)\downarrow SS(l)}l$. For the converse, it is easy to verify that l cannot be either spoiled or damaged, and that $\bar{l}$ cannot be recoverable. □

When a recoverable literal has several support sets it seems reasonable to prefer those that are maximal (w.r.t. containment relation). We next consider such sets:

Definition 3.15 Let l be a recoverable literal in KB, and M – an mcm of KB such that $M(l)\in\mathcal{F}\setminus\mathcal{I}$. The support set of l that is *associated with* M is: $SS_M(l)=\{\psi\in KB \mid \mathcal{A}(\psi)\cap Inc_M(KB)=\emptyset\}$ [7].

[6] The converse of corollary 3.13 is, of course, not true. Consider, e.g. $KB=\{p\vee\neg p,\ p\rightarrow q,\ \neg p\rightarrow q\}$: q is recoverable although $\{q\}\not\subseteq KB$. Moreover, this knowledge-base contains a recoverable literal although there is no $l\in\mathcal{L}(KB)$ s.t. $\{l\}\subseteq KB$.

[7] This is indeed a support set of l. See the proof of theorem 3.10.

Proposition 3.16 Every maximal support sets of a recoverable literal l is associated with some mcm M of KB s.t. $M(l)\notin\mathcal{I}$.

Proof: Suppose that $SS'(l)$ is an arbitrary support set of l. Let N' be a consistent model of $SS'(l)$, and N – its expansion to the whole KB. Consider any mcm M that satisfies $N\leq_{con} M$. Since $\mathcal{A}(SS'(l))\subseteq \mathcal{A}(KB)\setminus Inc_N(KB)\subseteq \mathcal{A}(KB)\setminus Inc_M(KB)$, then each formula $\psi\in SS'(l)$ consists only of literals that are assigned consistent truth values under M. Hence $SS'(l)\subseteq SS_M(l)$. □

Corollary 3.17 A knowledge-base is recoverable iff it has a recoverable literal.

Proof: By definition 3.8, a recoverable knowledge-base KB must have a maximal support set, and by proposition 3.16, such a set is of the form $SS_M(l)$, where l is a recoverable literal of KB. The converse direction: let l be a recoverable literal of KB. We have shown that there is an mcm M of KB such that $SS_M(l)$ is a support set of l. By the proof of proposition 3.16 this support set is contained in some maximal support set of l, and so KB is a recoverable. □

3.3 The damaged literals

The last class of literals according to the $\models_{con}$-categorization consists of those literals that a consistent truth value cannot be reliably attached to them (at least, not according to the most consistent models of the knowledge-base). The following theorem strengthens this intuition:

Theorem 3.18 l is damaged iff there exist mcms M_1 and M_2 s.t. $M_1(l)=f$ and $M_2(l)=t$.

Proof: "If": follows directly from the definition of damaged literals. For the converse, suppose that p is the atomic part of l. Since l is damaged iff p is damaged, it suffices to prove the claim for p. Now, p is damaged, thus there are mcms N_1 and N_2 s.t. $N_1(p)\notin\mathcal{F}$ and $N_2(\neg p)\notin\mathcal{F}$. Suppose that M_1 is a valuation that assigns f to p and is equal to N_1 for all the other members of $\mathcal{A}(KB)$. Similarly, suppose that M_2 assigns t to p and is equal to N_2 otherwise. Since N_1 and N_2 are mcms of KB, so are M_1 and M_2. □

We conclude this subsection with some observations related to damaged literals:

- The existence of a support set for a damaged literal is not assured. For instance, the damaged literal d of example 3.9 is a member of a support set (S). Still, no support set of KB supports d.
- Even if there are support sets for a damaged literal, there can be other subsets that support its negation: For example, in $KB=\{p,\ \neg p\vee q,\ r,\ \neg r\vee\neg q\}$ where $\mathcal{B}=FOUR$, q is damaged. It has a support set: $SS(q)=\{p,\ \neg p\vee q\}$, but there is a support set for $\neg q$ as well: $SS(\neg q)=\{r,\ \neg r\vee\neg q\}$.
- Consider $KB=\{p\vee q,\ \neg p\vee\neg q\}$. Here both p and q are damaged although KB is a consistent set. Intuitively, this is so because there is not enough data in KB about either p or q. In particular, a literal can be damaged not just because of "over" information, but because of a lack of data as well. In both cases, however, its truth value cannot be recovered safely.

4 The minimal mcms of *KB*

In this section we show that if one is interested only in recovering an inconsistent knowledge-base (that is, discovering the spoiled, damaged, and recoverable literals of KB, as well as the support sets of the latters), then it is sufficient to consider only the $\leq_k$-minimal models of the most consistent models (minimal mcms, in short). The set of the minimal mcms of KB will be denoted by $\Omega(KB)$, or just Ω.

Abstractly, we can view the construction of Ω as a composition of the two consequence relations "$\models_{con}$" and "$\models_{min}$". First, we confine ourselves to the mcms of KB by using $\models_{con}$, then we minimize the valuations that we have got by using $\models_{min}$. This process is a special case of what is called "stratification" in [BS88].

Lemma 4.1 Let KB be a finite set of normalized extended clauses. For every mcm M of KB there is an $N \in \Omega(KB)$ s.t. $N \leq_k M$ and $Inc_N(KB) = Inc_M(KB)$.

Proof: Let $M \in mcm(KB)$ and $N \in \Omega(KB)$ s.t. $N \leq_k M$. Suppose also that $Inc_N(KB) \neq Inc_M(KB)$. Since $M, N \in mcms(KB)$, there are $q_1, q_2 \in \mathcal{A}(KB)$ s.t. $q_1 \in Inc_N(KB) \backslash Inc_M(KB)$ and $q_2 \in Inc_M(KB) \backslash Inc_N(KB)$. Assume that $N(q_1) \in \mathcal{F}$. Since $N(q_1) \in \mathcal{I}$ then $N(\neg q_1) \in \mathcal{F}$ as well. Thus $M(q_1) \geq_k N(q_1) \in \mathcal{F}$ and $M(\neg q_1) \geq_k N(\neg q_1) \in \mathcal{F}$, so $M(q_1) \in \mathcal{I}$ – a contradiction. Hence $N(q_1) \notin \mathcal{F}$. Similarly, $N(\neg q_1) \notin \mathcal{F}$. Now, consider the valuation N' that assigns t to q_1, and is equal to N on every other $p \in \mathcal{A}(KB)$. It is easy to verify that for $\psi \in KB$, $N'(\psi) \in \mathcal{F}$ whenever $N(\psi) \in \mathcal{F}$, thus $N' \in mod(KB)$. But $Inc_N(KB) = Inc_{N'}(KB) \cup \{q_1\}$, therefore $N' >_{con} N$, and so $N \notin mcm(KB)$. In particular $N \notin \Omega(KB)$ – a contradiction. □

Proposition 4.2 Let KB be a finite set of extended clauses in BL. Then:
a) l is spoiled literal in KB iff for every model $M \in \Omega(KB)$, $M(l) \in \mathcal{F}$ and $M(\bar{l}) \in \mathcal{F}$.
b) l is recoverable iff for every $M \in \Omega(KB)$, $M(l) \in \mathcal{F}$, and there is $N \in \Omega(KB)$ s.t. $N(l) \in \mathcal{F} \setminus \mathcal{I}$.
c) l is damaged literal in KB iff there are $M_1, M_2 \in \Omega(KB)$ s.t. $M_1(l) \notin \mathcal{F}$ and $M_2(\bar{l}) \notin \mathcal{F}$.

Proof: We show only part (b); the proofs of the other parts are similar. Suppose that for every $M \in \Omega$, $M(l) \in \mathcal{F}$. Since $\mathcal{F}$ is upwards closed w.r.t. $\leq_k$, this is true for every mcm M', hence $KB \models_{con} l$. Since there is an $N \in \Omega$ s.t. $N(l) \in \mathcal{F} \setminus \mathcal{I}$, then $N(\bar{l}) \notin \mathcal{F}$, and so $KB \not\models_{con} \bar{l}$, thus l is recoverable. The other direction: since l is recoverable, it is assigned a designated truth value in every mcm of KB, in particular it is designated in every minimal mcm. Also, there must be an mcm N s.t. $N(l) \in \mathcal{F} \setminus \mathcal{I}$ (otherwise l is spoiled). By lemma 4.1 there is an $N' \in \Omega$ s.t. $N'(l) \notin \mathcal{I}$ as well. Therefore $N'(l) \in \mathcal{F} \setminus \mathcal{I}$. □

Another result is a characterization of the maximal support sets in terms of minimal mcms (cf. 3.16):

Proposition 4.3 Every maximal support set of a recoverable literal l is associated with some minimal mcm $M \in \Omega$ s.t. $M(l) \notin \mathcal{I}$.

Proof: Follows easily from proposition 3.16 and lemma 4.1. □

5 Extensions to first-order logic

It is possible to directly expand the present discussion to any first-order knowledge-bases provided that: (a) there are no quantifiers within the clauses; each extended clause that contains variables is considered as universally quantified, and (b) no function symbols are allowed. Consequently, a knowledge-base containing non-grounded formula, ψ, will be viewed as representing the corresponding set of ground formulae formed by substituting each variable that appears in ψ with every possible member of the Herbrand universe, U. Since we have not allowed the appearance of function symbols, and since we deal with finite knowledge-bases, U must be finite as well. Formally: $KB^U = \{\rho(\psi) \mid \psi \in KB,\ \rho : var(\psi) \rightarrow U\}$ where ρ is a *ground substitution* from the variables of every $\psi \in KB$ to the individuals of Herbrand universe U.

6 Examples and discussion

In what follows we consider two benchmark problems which are given in [Li88]. The illustrations are considered in $\mathcal{B} = FOUR$ with $\mathcal{I} = \{\top\}$. As it is shown below, our system manages to keep the results very close to those suggested in [Li88].

Consider the following block world knowledge-base:
$KB1 = \{heavy(A),\ heavy(B),\ \neg on\_table(A),\ \neg red(B),\ heavy(x) \rightarrow on\_table(x),\ heavy(x) \rightarrow red(x)\}$
The minimal mcms of $KB1$ are given in figure 3 [8]. Their associated support sets are listed below:

[8] $KB1$ has 16 mcms. We omit the other 12, which are not $\leq_k$-minimal.

mcm	$heavy(A)$	$heavy(B)$	$red(A)$	$red(B)$	$on\_table(A)$	$on\_table(B)$
$M1a$	t	t	t	$\top$	$\top$	t
$M1b$	t	$\top$	t	f	$\top$	$\bot$
$M1c$	$\top$	t	$\bot$	$\top$	f	t
$M1d$	$\top$	$\top$	$\bot$	f	f	$\bot$

Figure 3: The minimal mcms of $KB1$

$KB1a = \{heavy(A),\ heavy(B),\ heavy(A) \rightarrow red(A),\ heavy(B) \rightarrow on\_the\_table(B)\}$
$KB1b = \{heavy(A),\ \neg red(B),\ heavy(A) \rightarrow red(A)\}$
$KB1c = \{\neg on\_the\_table(A),\ heavy(B),\ heavy(B) \rightarrow on\_the\_table(B)\}$
$KB1d = \{\neg on\_the\_table(A),\ \neg red(B)\}$ [9]

$KB1a$ seems to be the preferable support sets according to many criteria: It is the largest set, it supports more literals than any other support set, and it contains maximal information in the sense of [Lo94] [10]. $KB1a$ implies that $on\_the\_table(B)$ and $red(A)$. These are also the conclusions in [Li88, problem A3].

For another example of the block world, consider the following knowledge-base:
$KB2 = \{heavy(A),\ heavy(B),\ heavy(C),\ heavy(x) \rightarrow on\_table(x),\ \neg on\_table(A) \vee \neg on\_table(B)\}$
Note that the last assertion in $KB2$ states that there is an unknown exception in the information. The mcms of $KB2$ are given in figure 4.

mcm	$heavy(A)$	$heavy(B)$	$heavy(C)$	$on\_table(A)$	$on\_table(B)$	$on\_table(C)$
$M2a$	$\top$	t	t	f	t	t
$M2b$	t	$\top$	t	t	f	t
$M2c$	t	t	t	$\top$	t	t
$M2d$	t	t	t	t	$\top$	t

Figure 4: The (minimal) mcms of $KB2$

Hence, $heavy(X)$ for $X = A, B, C$ and $on\_table(C)$ are all recoverable, while $on\_table(A)$ and $on\_table(B)$ are damaged. The support sets of $KB2$ are listed bellow:

$\{heavy(B),\ heavy(C),\ heavy(B) \rightarrow on\_table(B),\ heavy(C) \rightarrow on\_table(C),\ \neg on\_table(A) \vee \neg on\_table(B)\}$
$\{heavy(A),\ heavy(C),\ heavy(A) \rightarrow on\_table(A),\ heavy(C) \rightarrow on\_table(C),\ \neg on\_table(A) \vee \neg on\_table(B)\}$
$\{heavy(A),\ heavy(B),\ heavy(C),\ heavy(B) \rightarrow on\_table(B),\ heavy(C) \rightarrow on\_table(C)\}$
$\{heavy(A),\ heavy(B),\ heavy(C),\ heavy(A) \rightarrow on\_table(A),\ heavy(C) \rightarrow on\_table(C)\}$

Note that every recovered knowledge-base preserves the intuitive conclusions of $KB2$, i.e.: (a) block C is on the table, and (b) either block A or block B is on the table, but there is no evidence that both are on the table. Again, these conclusions are similar to those of [Li88].

Due to the lack of space we have not considered here all the benchmarks of [Li88]. We confined ourselves to two representative examples of category A (default reasoning). However, the reader might want to check that many other test criteria mentioned there are met in our system. Most notable are the inheritance features, and the autoepistemic characterizations.

[9] The "conservative" nature of the system is emphasized here: each solution gets rid of the information it considers as contradictory, and leaves all the other data unchanged.

[10] See [AA94c] for a detailed discussion of methods for choosing the preferred support set.

References

[AA94a] O.Arieli, A.Avron. *Logical bilattices and inconsistent data.* Proc. 9th IEEE Annual Symp. on Logic in Computer Science, IEEE Press (pp.468-476); July 1994.

[AA94b] O.Arieli, A.Avron. *Reasoning with logical bilattices.* Technical Report 291/94, Tel-Aviv University; July 1994.

[AA94c] O.Arieli, A.Avron. *A Model-theoretic approach to recover consistent data from inconsistent knowledge-bases.* Technical Report 297/94, Tel-Aviv University; Nov. 1994.

[Be77a] N.D.Belnap. *A useful four-valued logic.* Modern Uses of Multiple-Valued Logic (G.Epstein, J.M.Dunn - Eds.), Oriel Press (pp. 7-37); 1977.

[Be77b] N.D.Belnap. *How computer should think.* Contemporary Aspects of Philosophy (G.Ryle - Ed.), Oriel Press (pp. 30-56); 1977.

[BS88] A.L.Brown Jr., Y.Shoham. *New results on semantical nonmonotonic reasoning* Proc. 2nd Int. Workshop on Non-Monotonic Reasoning (pp. 19-26); 1988.

[dC74] N.C.A.da-Costa. *On the theory of inconsistent formal systems.* Notre Damm Journal of Formal Logic, Vol.15 (pp. 497-510); 1974.

[Fi89] M.Fitting. *Negation as refutation.* Proc. 4th Annual Symp. on Logic in Computer Science, IEEE Press (pp. 63-70); 1989.

[Fi90a] M.Fitting. *Kleene's logic, generalized.* J. Logic and Computation, Vol.1 (pp.797-810); Dec. 1990.

[Fi90b] M.Fitting. *Bilattices in logic programming.* The 20th Int. Symp. on Multiple-Valued Logic (G.Epstein - Ed.), IEEE Press (pp.238-246); 1990.

[Fi91] M.Fitting. *Bilattices and the semantics of logic programming.* Journal of Logic Programming, Vol.11, No.2 (pp. 91-116); Aug. 1991.

[Fi93] M.Fitting. *The family of stable models.* J. Logic Programming, Vol.17 (pp. 197-225); 1993.

[Fi94] M.Fitting. *Kleene's three-valued logics and their children.* Fundamenta Informaticae Vol.20 (pp. 113-131); 1994.

[Gi88] M.L.Ginsberg. *Multivalued logics: a uniform approach to reasoning in AI.* Computer Intelligence, Vol.4 (pp.256-316); 1988.

[KL92] M.Kifer, E.L.Lozinskii. *A logic for reasoning with inconsistency.* Journal of Automated reasoning. Vol.9, No.2 (pp. 179-215); Oct. 1992.

[KS92] M.Kifer, V.S.Subrahmanian. *Theory of generalized annotated programming and it's applications.* Journal of Logic Programming, Vol.12 (pp. 335-367); Apr. 1992.

[Le92] D.Lehmann. *Plausibility logic.* Proc. of the 5th Workshop on Computer Science Logic, Springer-Verlag (pp. 227-241); 1992.

[Li88] V.Lifschits. *Benchmark problems for formal nonmonotonic reasoning.* Proc. 2nd Int. Workshop on Non-Monotonic Reasoning (pp. 202-219), Springler-Verlag; 1988.

[Lo94] E.L.Lozinskii. *Recovering contradictions: a pluasible semantics for inconsistent systems.* Journal of Automated Reasoning, Vol.12 (pp.1-31); 1994.

[Su90a] V.S.Subrahmanian. *Mechanical proof procedures for many valued lattice-based logic programming.* Journal of Non-Classical Logic, Vol.7 (pp. 7-41); 1990.

[Su90b] V.S.Subrahmanian. *Paraconsistent disjunctive deductive databases.* IEEE 20th Int. Symp. on Multiple-Valued Logic (pp.339-345); 1990.

[Wa94] G.Wagner. *Vivid logic: knowledge-based reasoning with two kinds of negation.* Lecture Notes in AI No.764, Springer-Verlag, Germany; 1994.