Journal of Automated Reasoning **22:** 263–309, 1999.

A Model-Theoretic Approach for Recovering Consistent Data from Inconsistent Knowledge Bases ⋆

OFER ARIELI and ARNON AVRON
Department of Computer Science, School of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel. e-mail: ofera@math.tau.ac.il; aa@math.tau.ac.il

(Received: 21 November 1997)

Abstract. One of the most significant drawbacks of classical logic is its being useless in the presence of an inconsistency. Nevertheless, the classical calculus is a very convenient framework to work with. In this work we propose means for drawing conclusions from systems that are based on classical logic, although the information might be inconsistent. The idea is to detect those parts of the knowledge base that 'cause' the inconsistency, and isolate the parts that are 'recoverable'. We do this by temporarily switching into Ginsberg/Fitting multivalued framework of bilattices (which is a common framework for logic programming and nonmonotonic reasoning). Our method is conservative in the sense that it considers the contradictory data as useless and regards all the remaining information unaffected. The resulting logic is nonmonotonic, paraconsistent, and a plausibility logic in the sense of Lehmann.

Key words: inconsistent knowledge bases, nonclassical logics, bilattices.

1. Introduction

Most classical theorem provers are based on refutation procedures. In order to find out whether a given formula ψ follows from a given knowledge base KB, the negation of ψ is temporarily added to KB. The question is then: Is the resulting knowledge base consistent or not? If it is, then ψ follows from KB. Otherwise it does not.

A question now arises: What if the original KB is already inconsistent? The above approach necessarily leads then to the conclusion that ψ follows from KB. Classically this is fine. In classical logic an inconsistent theory entails everything. From a practical point of view, however, this leaves something to be desired. Drawing *any* conclusion whatsoever just because of the existence of a contradiction is certainly unrealistic, for example, in knowledge bases in which the information comes from several sources.

One possible solution to this problem is to use some kind of a *paraconsistent* logic [13], that is, a logic in which trivial reasoning from a contradiction

⋆ A preliminary version of this paper appears in [3].

is not allowed. Several candidates has been suggested in the literature (see, e.g., [7, 8, 12, 23, 28, 32, 33, 38–40]). The use of such logic has, however, its own drawbacks. The major one is the fact that it forces us to give up classical reasoning altogether. This is certainly not justified if the given knowledge base *is* consistent. Moreover, the classical calculus is a very convenient framework to work with; adding new mechanisms or connectives to it generally causes a considerable growth in the computational complexity needed to maintain the resulting system. The fact that many relatively efficient theorem provers that are based on classical logic already exist is also significant from a pragmatic point of view.

The purpose of this work is to propose means for drawing conclusions from systems that are based on classical logic, even though the information might be inconsistent. The idea is to detect first those parts of the knowledge base that 'cause' the inconsistency, and to isolate the parts that are 'recoverable'. The outcome of this approach is a construction of a subset of the original knowledge base, which *is* consistent and preserves most of the original data without changing its meaning. Since we are not changing any assertion (and in particular we are not damaging the syntax), we can continue handling the 'recovered' knowledge base by the usual methods of current theorem provers.

Consider, for example, the following set of assertions: $S = \{\neg s,\ r_1,\ r_1 \to \neg r_2,\ r_2 \to i\}$. This set is consistent; therefore, classical logic might be used for drawing conclusions from it. Assume now that new information arrives, and we are told that s is known to be true. The new knowledge base, $S' = S \cup \{s\}$, is inconsistent. Since everything classically follows from S', another mechanism for drawing plausible conclusions is needed. A common approach for doing so is to consider some maximal consistent subset of the knowledge base (see, e.g., [7, 28, 31, 36]). The drawback of this method is that it might lead to conclusions that are in a direct conflict with the original information. In the case of S', for instance, every maximal consistent subset must contain either s or $\neg s$ (but not both), and therefore such a set classically entails formulae that *contradict an explicit data of the original knowledge base*.

Instead of looking for maximal consistent subsets, it seems to us more reasonable to try to do the following things:

1. Detect and isolate the cause of the inconsistency together with what is related to it. Any data that is not related to the conflicting information should not be affected or changed.
2. Make sure that the remaining information yields conclusions that are semantically coherent with the original data (i.e., only inferences that do not contradict any previously drawn conclusions are allowed).

In the specific example considered above, for instance, it is clear that the ambiguity in S' is connected only to $\{s, \neg s\}$, while $\{r_1,\ r_1 \to \neg r_2,\ r_2 \to i\}$ seems to be the part of S' from which one would like to draw (classical) conclusions.

How do we *practically* recover consistent data from an inconsistent knowledge base without changing it? The method that we suggest in this paper is to switch into a multivalued framework. For this we use special algebraic structures called *bilattices.* Bilattices were first proposed by Ginsberg (see [22]) as a basis for a general framework for many applications such as first-order theorem provers, truth maintenance systems, and implementations of default inferences. This notion was further developed by Fitting, who used bilattices for extending some well-known logics (like Kleene 3-valued logics; see [16, 20]), and introduced bilattice-based logic programming methods [15, 17 – 19]. In bilattices the elements (which are also referred to as 'truth values') are arranged in two partial orders simultaneously: 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 *information* that each element exhibits on the assertions that it is supposed to represent.⋆

Just adding truth values is not enough, of course. To recover the information truthfully, we have to develop a mechanism that enables paraconsistent inferences. For this we use an epistemic entailment proposed in [23] (denoted here by $\models_{\text{con}}$) as the consequence relation of the logic. This relation can be viewed as a kind of 'closed world assumption', since it considers only the 'most consistent' models (mcms) of a given set of assertions. As was shown in [2, 4], this consequence relation enjoys some appealing properties, such as being nonmonotonic and paraconsistent, and a 'plausibility logic' in the sense of Lehmann [25].

By using $\models_{\text{con}}$ we would be able to construct subsets of the knowledge base (called 'support sets'), which are useful means to override the contradictions when attention is focused on certain (recoverable) formulae. These sets are the candidates to be the recovered knowledge base. The common property shared by every recovered knowledge base 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 *conservative* (*skeptical*) [41] or *coherent* [7, 8].

It should be emphasized at this point that our method, like many other well-known approaches in the literature of AI, does not act as a complete reasoner. That is, it does not always propose a unique solution that is the best interpretation of the conflicts in the knowledge base. Instead, it suggests several support sets that can be extracted from the original inconsistent set of assertions. The reasoners are then expected to choose the one that is most suitable according to their actual epistemic beliefs. In the sequel we provide some heuristics that might guide them as to which support set to choose.

Before turning to the technical details, a few words on implementation issues. A major challenge encountered by every reasoning method is to turn the proposed *formalism* into a computationally feasible *process.* There are many ways of dealing with this problem: The method proposed in [26, 41], for example, is to restrict the representation language, taking into account the trade-off between expressiveness

⋆ The idea of using *two* partial orders may be traced back to Belnap and his well-known four-valued logic [9, 10], which is exactly the simplest bilattice $FOUR$ (see below). Belnap was also the one who first proposed $FOUR$ as being useful for computer-based reasoning.

and efficiency. Here we use a different approach: for practical implementations we restrict ourselves to a particular family of common knowledge bases (called *stratified* – see Section 3.3). We provide an efficient algorithm for recovering consistent data from this type of knowledge base.

The paper is organized as follows. In the next section we describe the framework of the discussion. We survey some basic notions related to bilattices, define bilattice-based logics, and present the general kinds of knowledge bases to be considered in the sequel. Section 3 contains the basic ideas of recovering consistent data from conflicting information. In this section we also examine some of the properties of the recovered data and propose a method for producing it in several important cases. In Section 4 we consider a special family of models that is sufficient for the task of recovering consistent data, and in Section 5 we show how to extend the results to first-order logic. In Section 6 we summarize the ideas developed along the paper and provide several examples of their use (the most important of which seems to be in the area of model-based diagnostic systems). In Section 7 we compare our approach with other formalisms that deal with inconsistency. Finally, in Section 8 we give some concluding remarks and discuss directions for further study.

2. Preliminaries

2.1. BILATTICES

DEFINITION 2.1 [22]. A *bilattice* is a structure $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ such that B is a nonempty set containing at least two elements; $(B, \leq_t)$, $(B, \leq_k)$ are complete lattices; and $\neg$ is a unary operation on B with the following properties:

(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) $\neg\neg a = a$.

Bilattices are therefore algebraic structures that contain two partial orders simultaneously. Each one reflects a different concept: $\leq_t$ intuitively reflects differences in the 'measure of truth' that the bilattice elements are supposed to represent, while $\leq_k$ might intuitively be understood as reflecting differences in the amount of *knowledge* (or in the amount of *information*) that each one of these elements exhibits. The basic relation between these two partial orders is via negation. Note that negation is order preserving w.r.t $\leq_k$. This reflects the intuition that while one expects negation to invert the notion of truth, it should keep the amount of information: we know no more and no less about $\neg p$ than we know about p (see [22, p. 269] and [17, p. 239] for further discussion).

NOTATION 2.2. Following Fitting, we shall use $\wedge$ and $\vee$ for the meet and join that 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.* f and t will denote the least and the greatest element w.r.t. $\leq_t$, while $\bot$ and $\top$ will denote 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 also Lemma 2.5 below). Finally, unlike in [2, 4], $\psi \to \phi$ is here just an abbreviation for $\neg\psi \vee \phi$.

DEFINITION 2.3.

(a) [22] A *distributive bilattice* is a bilattice in which all the (twelve) possible distributive laws concerning $\wedge, \vee, \otimes$, and $\oplus$ hold. (In other words, $a\triangle(b\nabla c) = (a\triangle b)\nabla(a\triangle c)$ for every $\triangle, \nabla \in \{\wedge, \vee, \otimes, \oplus\}$, $\triangle \neq \nabla$.)

(b) [17] An *interlaced bilattice* is a bilattice in which each one of $\wedge, \vee, \otimes$, and $\oplus$ is monotonic with respect to both $\leq_t$ and $\leq_k$. In other words,

if $a \leq_t b$, then $a \otimes c \leq_t b \otimes c$, and $a \oplus c \leq_t b \oplus c$;
if $a \leq_k b$, then $a \wedge c \leq_k b \wedge c$, and $a \vee c \leq_k b \vee c$.

LEMMA 2.4 [17]. *Every distributive bilattice is also interlaced.*

LEMMA 2.5. *Let $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ be a bilattice, and let a, b be arbitrary elements of B.*

(a) [22] $\neg(a \wedge b) = \neg a \vee \neg b$; $\neg(a \vee b) = \neg a \wedge \neg b$; $\neg(a \otimes b) = \neg a \otimes \neg b$; $\neg(a \oplus b) = \neg a \oplus \neg b$.

(b) [22] $\neg f = t$; $\neg t = f$; $\neg\bot = \bot$; $\neg\top = \top$.

(c) [17] *If $\mathcal{B}$ is interlaced, then* $\bot \wedge \top = f$; $\bot \vee \top = t$; $f \otimes t = \bot$; $f \oplus t = \top$.

DEFINITION 2.6 [2]. Let $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ be a bilattice.

(a) A *bifilter* is a nonempty proper subset $\mathcal{F} \subset B$ such that

$a \wedge b \in \mathcal{F}$ iff $a \in \mathcal{F}$ and $b \in \mathcal{F}$,
$a \otimes b \in \mathcal{F}$ iff $a \in \mathcal{F}$ and $b \in \mathcal{F}$.

(b) A bifilter $\mathcal{F}$ is called *prime* if it satisfies also

$a \vee b \in \mathcal{F}$ iff $a \in \mathcal{F}$ or $b \in \mathcal{F}$,
$a \oplus b \in \mathcal{F}$ iff $a \in \mathcal{F}$ or $b \in \mathcal{F}$.

The notion of a (prime) bifilter is a natural generalization to bilattices of the notion of a (prime) filter, which is a basic tool in algebraic treatments of logic.

* These operators would not play a central role in what follows, since we shall be most interested in the 'classical' operators $\wedge$ and $\vee$. However, since our method allows the usage of these operators without any further effort (and without increasing the complexity of the methods below – see Proposition 2.22 and its proof), we shall refer to them as well.

The set of designated values in a multiple-valued semantics of a logic is almost always required to be a filter, because the algebraic properties of a filter reflect the properties of a consequence relation. Moreover, the meet operator behaves like conjunction relative to filters. In most cases the filter that is used for defining a logic is further required to be *prime*, because only relative to prime filters the join operator behaves like a disjunction. Relative to *bi*filters both meet operators of a bilattice behave like conjunctions, while relative to prime bifilters the join operators, in addition, behave like disjunctions.

LEMMA 2.7. *Let $\mathcal{F}$ be a bifilter of $\mathcal{B}$. Then,*

(a) *$\mathcal{F}$ is upward-closed w.r.t both $\leq_t$ and $\leq_k$.*
(b) *$t, \top \in \mathcal{F}$, while $f, \bot \notin \mathcal{F}$.*

Proof. Claim (a) follows immediately from the definition of $\mathcal{F}$. The first part of (b) follows from (a), and from the maximality of t and $\top$. The fact that the minimal elements are not in $\mathcal{F}$ follows also from (a), since $\mathcal{F} \neq B$. $\square$

The following sets are contained in every bifilter.

DEFINITION 2.8.

- $\mathcal{D}_k(\mathcal{B}) = \{b \in B \mid b \geq_k t\}$ (the designated values w.r.t. $\leq_k$ of $\mathcal{B}$).*
- $\mathcal{D}_t(\mathcal{B}) = \{b \in B \mid b \geq_t \top\}$ (the designated values w.r.t. $\leq_t$ of $\mathcal{B}$).

In [2] it is shown that in every interlaced bilattice $\mathcal{B}$, $\mathcal{D}_k(\mathcal{B}) = \mathcal{D}_t(\mathcal{B})$, and that this entails that $\mathcal{D}_t(\mathcal{B})$ itself *is* a bifilter, and it is *the smallest* one. This fact makes it a very natural choice, but it is not the only possible or useful one.**

A property of $\mathcal{D}_t(\mathcal{B})$ that will be used later is the following:

LEMMA 2.9. *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 $b = \top$. $\square$

DEFINITION 2.10 [2]. A *logical bilattice* is a pair $(\mathcal{B}, \mathcal{F})$, where $\mathcal{B}$ is a bilattice, and $\mathcal{F}$ is a prime bifilter. The elements of $\mathcal{F}$ are called the *designated* elements of the bilattice.

EXAMPLE 2.11. $FOUR$ and $NINE$ (Figure 1) are distributive bilattices (hence also interlaced). Each of $FOUR$, $NINE$, and $DEFAULT$ (Figure 2) is a logical bilattice $\mathcal{B}$ with $\mathcal{F} = \mathcal{D}_k(\mathcal{B})$. $NINE$ forms also another logical bilattice if we take $\mathcal{F} = \mathcal{D}_k(NINE) \cup \{of, d\top, dt\}$.

* These elements may be viewed as those that are 'at least true' (see [10, p. 36]).

** Note that unless otherwise stated, what we do below is independent of the choice of the bifilter.

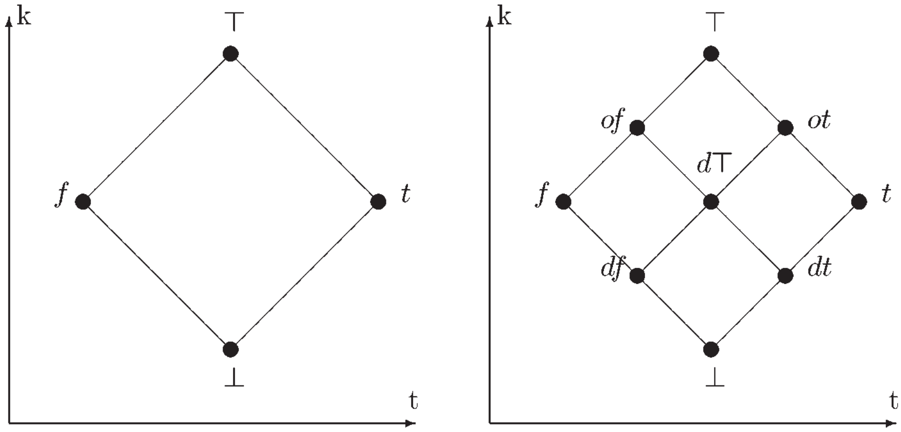

Figure 1. $FOUR$ and $NINE$.

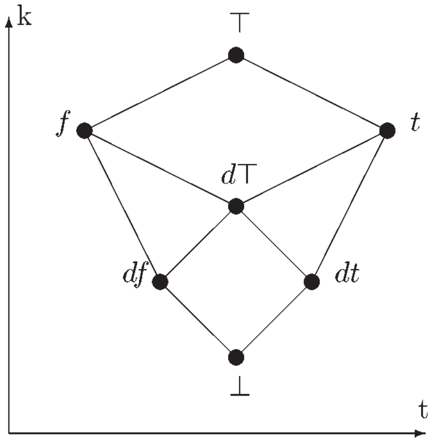

Figure 2. $DEFAULT$.

2.2. THE LOGIC

Denote by BL the standard propositional language over $\{\wedge, \vee, \neg, \otimes, \oplus, t, f\}$, and let KB be a set of formulae over BL. $\mathcal{A}(KB)$ denotes the set of all atomic formulae that appear in some formula of KB, and $\mathcal{L}(KB)$ denotes the set of all literals that appear in some formula of KB.

DEFINITION 2.12. Let $(\mathcal{B}, \mathcal{F})$ be a logical bilattice.

(a) A *valuation* ν in B is a function that assigns a truth value from B to each atomic formula, and it maps each constant to its corresponding value in B. Any valuation is extended to complex formulas in the standard way: $\nu(\neg\psi) = \neg\nu(\psi)$, $\nu(\psi * \phi) = \nu(\psi) * \nu(\phi)$ for $* \in \{\wedge, \vee, \otimes, \oplus\}$, and $\nu(\psi \to \phi) = \neg\nu(\psi) \vee \nu(\phi)$. We shall sometimes write $\psi : b \in \nu$ instead of $\nu(\psi) = b$.

(b) We say that ν *satisfies* ψ ($\nu \models \psi$) iff $\nu(\psi) \in \mathcal{F}$. We say that ψ is *valid* iff every valuation satisfies it.
(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 $mod(KB)$. Given KB, we shall use the letters M and N (with or without subscripts) to denote models of KB.

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

DEFINITION 2.13 [2, 4]. Let $(\mathcal{B}, \mathcal{F})$ be a logical bilattice. A subset $\mathcal{I}$ of B is called 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 $\neg b \in \mathcal{F}$.

Note that by (b) of Definition 2.13 it must follow that $\top \in \mathcal{I}$ and $t \notin \mathcal{I}$. Hence, by (a), $f \notin \mathcal{I}$.

EXAMPLE 2.14. $\mathcal{I}_1 = \{b \mid b \in \mathcal{F} \wedge \neg b \in \mathcal{F}\}$ is the minimal inconsistency set in every logical bilattice. $\mathcal{I}_2 = \{b \mid b = \neg b\}$ is an inconsistency set in the case that $\mathcal{B}$ is interlaced and $\mathcal{F} = \mathcal{D}_k(\mathcal{B})$. Note that $\perp \notin \mathcal{I}_1$ while $\perp \in \mathcal{I}_2$. Indeed, one of the major considerations when choosing an inconsistency set is whether to include $\perp$ in $\mathcal{I}$ or not. Despite the fact that in every bilattice $\neg\perp = \perp$ (see Lemma 2.5), $\perp$ intuitively reflects no information whatsoever about the assertion it represents; in particular, one might not take such assertions as inconsistent.

In the following discussion we fix some logical bilattice $(\mathcal{B}, \mathcal{F})$ as well as an inconsistency set $\mathcal{I}$.

NOTATION 2.15. Let M be a valuation on KB. The set of atomic formulae in $\mathcal{A}(KB)$ that are assigned under M values from $\mathcal{I}$ is denoted $\mathrm{Inc}_M(KB)$, that is, $\mathrm{Inc}_M(KB) = \{p \in \mathcal{A}(KB) \mid M(p) \in \mathcal{I}\}$.

DEFINITION 2.16. Let M, N be two models of a set of formulae, KB.

(a) M is *more consistent* than N ($M >_{\mathrm{con}} N$) iff $\mathrm{Inc}_M(KB) \subset \mathrm{Inc}_N(KB)$.
(b) M is a *most consistent* model (mcm) of KB if there is no other model of KB that is more consistent than M. The set of the most consistent models of KB will be denoted $\mathrm{con}(KB)$.
(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 if there is at least one $q \in \mathcal{A}(KB)$ s.t. $M(q) <_k N(q)$.
(d) M is a *minimal* model of KB if there is no other model of KB that is smaller than M. The set of all the minimal models of KB will be denoted $min(KB)$.

Table I. The models of KB (Example 2.18).

Model No.	s	r_1	r_2	i	Model No.	s	r_1	r_2	i
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}	$\top$	$\top$	$\top$	$\bot, t, f, \top$

DEFINITION 2.17. Let KB be a set of formulae and ψ a formula. Let S be any set of valuations. We denote $KB \models_S \psi$ if each model of KB that is in S is also a model of ψ.

Some particularly interesting instances of Definition 2.17 are the following:

- $KB \models_{\mathrm{mod}(KB)} \psi$ if every model of KB is a model of ψ.
- $KB \models_{\mathrm{con}(KB)} \psi$ if every mcm of KB is a model of ψ.
- $KB \models_{\mathrm{min}(KB)} \psi$ if every minimal model of KB is a model of ψ.

We shall abbreviate the above cases by $KB \models \psi$, $KB \models_{\mathrm{con}} \psi$, and $KB \models_{\mathrm{min}} \psi$, respectively.

EXAMPLE 2.18. Consider the knowledge base $KB = \{s, \neg s, r_1,\ r_1 \to \neg r_2, r_2 \to i\}$ discussed in the introduction. Let $\mathcal{B} = FOUR$ and $\mathcal{F} = \{t, \top\}$. The models of KB are listed in Table I. It follows that $\mathrm{con}(KB) = \{M_1, M_2, M_3\}$ provided that $\bot \notin \mathcal{I}$, and if $\bot \in \mathcal{I}$, then $\mathrm{con}(KB) = \{M_2, M_3\}$. Also, $\mathrm{min}(KB) = \{M_1, M_9\}$; thus $KB \models_{\mathrm{con}} \neg r_2$, while $KB \not\models \neg r_2$ and $KB \not\models_{\mathrm{min}} \neg r_2$.

2.3. THE KNOWLEDGE BASES

In this subsection we define the kind of knowledge bases that we are dealing with.

DEFINITION 2.19. A formula ψ is an *extended clause* if
ψ is a literal (an atom or a negated atom), 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 2.20. A formula ψ is said to be *normalized* if it has no subformula of the forms $\phi \vee \phi$, $\phi \wedge \phi$, $\phi \oplus \phi$, $\phi \otimes \phi$, or $\neg\neg\phi$.*

The following lemma is clearly valid in every logical bilattice $(\mathcal{B}, \mathcal{F})$.

* We could define stronger notions of normalized formulae, but this one is sufficient for our needs.

LEMMA 2.21. *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 would be sets of normalized extended clauses. As the following proposition shows, representing the formulae in a (normalized) extended clause form does not reduce the generality.

PROPOSITION 2.22. *For every formula ψ there is a finite set S of normalized extended clauses such that for every valuation ν, $\nu \models \psi$ iff $\nu \models S$.*

Proof. First, translate ψ into its extended negation normal form, ψ', where the negation operator precedes atomic formulae only. This can be done in *every* bilattice. The rest of the proof is by an induction on the structure of ψ':

If $\psi' = \psi'_1 \wedge \psi'_2$ or $\psi' = \psi'_1 \otimes \psi'_2$, then by induction hypothesis, there exists S_i s.t. $\nu \models S_i$ iff $\nu \models \psi'_i$ $(i = 1, 2)$. Take $S = S_1 \cup S_2$; then $\nu \models S$ iff $\nu \models S_1$ and $\nu \models S_2$, iff $\nu \models \psi'$.

If $\psi' = \psi'_1 \vee \psi'_2$ or $\psi = \psi'_1 \oplus \psi'_2$, then again, there exist $S_1 = \{\phi_i\}_{i=1}^n$ and $S_2 = \{\varphi_j\}_{j=1}^m$ s.t. $\nu \models \psi'_i$ iff $\nu \models S_i$ $(i = 1, 2)$. Take $S = \{\phi_i \vee \varphi_j \mid 1 \leq i \leq n,\ 1 \leq j \leq m\}$. Now, since $\mathcal{F}$ is a *prime* bifilter, we have the following:

- If $\nu \models \psi'$, then $\nu \models \psi'_1$ or $\nu \models \psi'_2$. Suppose that $\nu \models \psi'_1$. Then $\nu \models \phi_i$ for $i = 1, \ldots, n$. So, for every $1 \leq i \leq n$ and for every $1 \leq j \leq m$: $\nu \models \phi_i \vee \varphi_j$; hence $\nu \models S$.
- If $\nu \not\models \psi'$, then $\nu \not\models \psi'_1$ and $\nu \not\models \psi'_2$, that is, $\nu \not\models \phi_i$ and $\nu \not\models \varphi_j$ for some $1 \leq i \leq n$ and some $1 \leq j \leq m$. Then, for those i and j, $\nu \not\models \phi_i \vee \varphi_j$; hence $\nu \not\models S$. □

Here is another useful property of extended clauses over logical bilattices. It will be used several times in the sequel.

LEMMA 2.23. *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 an $1 \leq i \leq n$ s.t. $\nu \models l_i$.*

Proof. By an induction on the structure of ψ. □

A basic notion in every paraconsistent system is that of consistency. Next, we expand this notion to the multivalued case.

DEFINITION 2.24. Suppose that $\mathcal{I}$ is an inconsistency set of $(\mathcal{B}, \mathcal{F})$.

(a) A model M of KB is *consistent* if $\mathrm{Inc}_M(KB) = \emptyset$; that is, M assigns a consistent truth value to every member of $\mathcal{A}(KB)$.⋆

(b) KB is *consistent* if it has a consistent model.

⋆ Note that every consistent model of KB is trivially an mcm of KB.

LEMMA 2.25. *KB is consistent* (*in the sense of Definition* 2.24(b)) *iff it is classically consistent* (*i.e., has a classical model*).

Proof. One direction is obvious. For the other, assume that M is a consistent model of KB. Then there is no $p \in \mathcal{A}(KB)$ s.t. both $M(p) \in \mathcal{F}$ and $\neg M(p) \in \mathcal{F}$. Consider the valuation M' defined for every $l \in \mathcal{L}(KB)$ as follows: $M'(l) = t$ if $M(l) \in \mathcal{F}$, and $M'(l) = f$ otherwise. The consistency assumption entails that whenever $M(l) \in \mathcal{F}$ for some $l \in \mathcal{L}(KB)$, $M'(l) \in \mathcal{F}$ also. By Lemma 2.23 it follows that M' is a (classical) model of KB as well. □

A fundamental property of the knowledge bases that we consider here is that for every model there is an mcm that is at least as consistent. In finite knowledge bases this is trivially the case. The following proposition assures, nevertheless, that in *every* propositional knowledge base this property holds.

PROPOSITION 2.26 (Lin's Lemma [33]). *Let KB be a* (*possibly infinite*) *set of extended clauses. For every model M of KB there is an mcm M' of KB s.t. $M' \geq_{\text{con}} M$.*

Proof. For the reader's convenience we repeat the proof given in [33], adjusted to our framework: Suppose that M is some model of KB, and $S_M = \{N \mid N \in mod(KB), N \geq_{\text{con}} M\}$. Let $C \subseteq S_M$ be a chain w.r.t. $\leq_{\text{con}}$. We shall show that C is bounded, so by Zorn's Lemma, C has a maximal element, which is the required mcm. Indeed, if C is finite, we are done. Otherwise, consider the following sets:

$$C' = \bigcap \{\text{Inc}_N(KB) \mid N \in C\},$$

$$KB' = \{\psi \in KB \mid \mathcal{A}(\psi) \cap C' = \emptyset\}.$$

Let KB'' be a finite subset of KB'. Since KB'' is finite and C is a chain, there exists some $N \in C$ s.t. $\mathcal{A}(\phi) \cap \text{Inc}_N(KB) = \emptyset$ for every $\phi \in KB''$. Since N is a model of KB and the reduction of N to $\mathcal{A}(KB'')$ is a consistent model of KB'', it follows that every finite subset of KB' is consistent. Hence, by Lemma 2.25 and the classical compactness theorem, KB' is consistent, and so it has a consistent model, N'. Now, consider the following valuation defined for every $p \in \mathcal{A}(KB)$:

$$M'(p) = \begin{cases} \top & \text{if } p \in C', \\ N'(p) & \text{otherwise.} \end{cases}$$

Clearly, $M' \geq_{\text{con}} N$ for every $N \in C$. It remains to show that $M' \in \text{mod}(KB)$, but this is obvious, since for every $\psi \in KB'$ and for every $p \in \mathcal{A}(\psi)$, $p \notin C'$ hence $M'(p) = N'(p)$, and so $M'(\psi) = N'(\psi) \in \mathcal{F}$. Also, for every $\psi \in KB \setminus KB'$ there is a $p \in \mathcal{A}(\psi)$ s.t. $p \in C'$, thus $M'(p) = \top$, and by Lemma 2.23, $M'(\psi) \in \mathcal{F}$. □

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. To do so, we shall divide the atomic formulae that appear in the clauses of the knowledge base into four subsets as follows.

DEFINITION 3.1. Let $l \in \mathcal{L}(KB)$, and denote by $\bar{l}$ its complement.

(a) If $KB \models_{\mathrm{con}} l$ and $KB \models_{\mathrm{con}} \bar{l}$, then l is said to be *spoiled*.
(b) If $KB \models_{\mathrm{con}} l$ and $KB \not\models_{\mathrm{con}} \bar{l}$, then l is said to be *recoverable*.
(c) If $KB \not\models_{\mathrm{con}} l$ and $KB \not\models_{\mathrm{con}} \bar{l}$, then l is said to be *incomplete*.⋆

Obviously, for each $l \in \mathcal{L}(KB)$, either l is spoiled, or l is recoverable, or l is incomplete, or $\bar{l}$ is recoverable.

EXAMPLE 3.2. In the knowledge base of Example 2.18, s is spoiled, r_1 and $\neg r_2$ are recoverable, and i is incomplete.

3.1. THE SPOILED LITERALS

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

THEOREM 3.3. *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. Without loss of generality, suppose $l = p$, where $p \in \mathcal{A}(KB)$. The case $l = \neg p$ is proved similarly.

(a) $\to$ (c): If p is spoiled, that is, $KB \models_{\mathrm{con}} p$ and $KB \models_{\mathrm{con}} \neg p$, then for every mcm M' of KB, $M'(p) \in \mathcal{F}$, and also $\neg M'(p) = M'(\neg p) \in \mathcal{F}$. Hence (property (b) in Definition 2.13), $M'(l) \in \mathcal{I} \cap \mathcal{F}$.

(c) $\to$ (d): Suppose that for every mcm M' of KB, $M'(l) \in \mathcal{I} \cap \mathcal{F}$. By Proposition 2.26 l is assigned some inconsistent truth value in *every* model of KB. Assume that $l \in \{p, \neg p\}$, 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 2.23). Similarly, since ν_f is not a model of KB, $p \in KB$.

⋆ In [23] literals of this kind are called 'damaged'. We feel that this terminology is somewhat too strong.

(d) $\rightarrow$ (b): If $\{l, \bar{l}\} \subseteq KB$, then obviously, for every model M of KB, $M(l) \in \mathcal{F}$, and $\neg M(l) \in \mathcal{F}$. From property (b) in Definition 2.13, then, $M(l) \in \mathcal{I} \cap \mathcal{F}$.

(b) $\rightarrow$ (a): If for every model M of KB $M(l) \in \mathcal{I} \cap \mathcal{F}$, then $M(l) \in \mathcal{F}$ and $M(\bar{l}) \in \mathcal{F}$. Hence $KB \models l$ and $KB \models \bar{l}$ which implies that $KB \models_{\text{con}} l$ and $KB \models_{\text{con}} \bar{l}$. Thus l is spoiled. $\square$

COROLLARY 3.4. *If $\mathcal{F} = \mathcal{D}_t(\mathcal{B})$, then every model of KB assigns $\top$ to the spoiled literals of KB.*

Proof. Immediate from (d) of Theorem 3.3, and Lemma 2.9. $\square$

COROLLARY 3.5. *It takes* $\mathrm{O}(|KB|)$ *running time to discover the spoiled literals of KB.*

Proof. Immediate from (d) of Theorem 3.3. $\square$

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. In fact, one can view this as a mechanism that ‘bypasses’ the inconsistency and generates a reduced supporting knowledge base for every recoverable literal. In such a support set the information is consistent with that of the original knowledge base.

DEFINITION 3.6. A subset $KB' \subseteq KB$ is *consistent in KB* if KB' has a consistent model M' and there is a (not necessarily consistent) model M of KB s.t. $M(p) = M'(p)$ for every $p \in \mathcal{A}(KB')$.

EXAMPLE 3.7. $KB' = \{q\}$ is a consistent set, which is *not* consistent in $KB = \{q, \neg q\}$, since there is no consistent model M' of KB' and a model M of KB s.t. $M'(q) = M(q)$.

DEFINITION 3.8. A nonempty subset $SS(l)$ of KB is a *support set* of a literal l (or: $SS(l)$ *supports* l) if it is consistent in KB and l is recoverable in $SS(l)$.

DEFINITION 3.9. 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 a *recovered subset* of KB. A knowledge base that has a recovered subset is called *recoverable*.

EXAMPLE 3.10. Consider again the example given in 2.18 and 3.2: $KB = \{s, \neg s, r_1,\ r_1 \rightarrow \neg r_2, r_2 \rightarrow i\}$. Here KB is a recoverable knowledge base, since $S = \{r_1,\ r_1 \rightarrow \neg r_2, r_2 \rightarrow i\}$ is a maximal support set of both r_1 and $\neg r_2$. Note that S does *not* support i, since $S \not\models_{\text{con}} i$.

Support sets are our candidates to be the recovered knowledge base. Hence, our system recovers knowledge bases only if there is at least one recoverable literal. This is not a major drawback, since most knowledge bases contain some atomic facts, which are recoverable literals unless they are spoiled. Hence, the case that there are no recoverable literals is not likely to happen.

In Examples 3.2 and 3.10 every recoverable literal of the knowledge base has a support set, as the following theorem shows.

THEOREM 3.11. *Every recoverable literal has a support set.*

Proof. Without loss of generality, suppose that $l = p$, where $p \in \mathcal{A}(KB)$ is recoverable; the case $l = \neg p$ is proved similarly. Let M be an mcm of KB such that $M(p) \in \mathcal{F} \setminus \mathcal{I}$. Let M' be the reduction of M to $\mathcal{A}(KB) \setminus \mathrm{Inc}_M(KB)$ only. Define

$$SS'(p) = \{\psi \in KB \mid \mathcal{A}(\psi) \subseteq \mathcal{A}(KB) \setminus \mathrm{Inc}_M(KB)\}.$$

We shall show that $SS'(p)$ is a support set of p.

(a) Obviously, $SS'(p) \subseteq KB$. Assume for contradiction that $SS'(p)$ is empty. Then every $\psi \in KB$ contains some element of $\mathrm{Inc}_M(KB)$ or its negation. Define $N = \{r : f \mid r \in \mathcal{A}(KB) \setminus \mathrm{Inc}_M(KB)\} \cup \{s : \top \in \mathrm{Inc}_M(KB)\}$. By Lemma 2.23, N is a model of KB. Moreover, N is an *mcm* of KB since $\mathrm{Inc}_N(KB) = \mathrm{Inc}_M(KB)$. But $p \in \mathcal{A}(KB) \setminus \mathrm{Inc}_M(KB)$, hence $N(p) = f$, a contradiction to $KB \models_{\mathrm{con}} p$.

(b) $SS'(p)$ is a consistent set in KB, since M' is a consistent model of $SS'(p)$ that is expandable to the model M of KB (i.e., $\forall q \in \mathcal{A}(SS'(p)) M'(q) = M(q)$).

(c) $SS'(p) \models_{\mathrm{con}} p$. Suppose that N' is an mcm of $SS'(p)$ and $N'(p) \notin \mathcal{F}$. Notice that N' must be consistent; otherwise $N' <_{\mathrm{con}} M'$, and N' cannot be an mcm of $SS'(p)$. Let N be the following expansion of N' to KB: $\{N'(q) \mid q \in \mathcal{A}(KB) \setminus \mathrm{Inc}_M(KB)\} \cup \{q : \top \mid q \in \mathrm{Inc}_M(KB)\}$. Clearly, N is a model of KB (indeed, if $\psi \in SS'(p)$, then $N(\psi) = N'(\psi) \in \mathcal{F}$, and if $\psi \in KB \setminus SS'(p)$, then $\mathrm{Inc}_M(KB) \cap \mathcal{A}(\psi) \neq \emptyset$, and since $N(s) = \top$ for every $s \in \mathrm{Inc}_M(KB)$, then by Lemma 2.23, $N(\psi) \in \mathcal{F}$ again). Furthermore, N is an mcm of KB, since $\mathrm{Inc}_N(KB) = \mathrm{Inc}_M(KB)$, and M is an mcm of KB. But $N(p) = N'(p) \notin \mathcal{F}$, so $KB \not\models_{\mathrm{con}} p$, a contradiction.

(d) $SS'(p) \not\models_{\mathrm{con}} \neg p$. Otherwise, for every mcm N of $SS'(p)$, $\neg N(p) = N(\neg p) \in \mathcal{F}$, and since we have shown that $SS'(p) \models_{\mathrm{con}} p$, $N(p) \in \mathcal{F}$ as well. Thus $N(p) \in \mathcal{I}$ for every mcm N of $SS'(p)$, and so $SS'(p)$ cannot be a consistent set. □

As a matter of fact, the relation between recoverable literals and support sets is even stronger, as the following theorem shows.

THEOREM 3.12. *If l is a recoverable literal in KB, then there is no subset in KB that supports its complement, $\bar{l}$.*

Proof. Without loss of generality, suppose that $l = p$. Assume that there exists $SS' \subseteq KB$ that is nonempty, consistent in KB, and $SS' \models_{\mathrm{con}} \neg p$. Since SS' is consistent in KB, it has a consistent model, M', which is expandable to a model M of KB (i.e., $\forall q \in \mathcal{A}(SS')M'(q) = M(q)$). M preserves the valuations of M' on $\mathcal{A}(SS')$, so $M(q) = M'(q) \notin \mathcal{I}$ for every $q \in \mathcal{A}(SS')$. Let N be an mcm of KB s.t. $N \geq_{\text{con}} M$ (see Proposition 2.26). Since $N \geq_{\text{con}} M$, $N(q) \notin \mathcal{I}$ for every $q \in \mathcal{A}(SS')$. 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 $\mathcal{A}(SS')$. Since N' is identical to N on $\mathcal{A}(SS')$, and since N is a model of KB, then (a) N' is a model of SS', (b) $N'(q) \notin \mathcal{I}$ for every $q \in \mathcal{A}(SS')$, and (c) $N'(p) \in \mathcal{F}$. From (a) and (b), then, N' is a consistent model of SS', and so from (c), $N'(\neg p) \notin \mathcal{F}$ (otherwise, $N'(p) \in \mathcal{F}$ and $N'(\neg p) \in \mathcal{F}$, hence $N'(p) \in \mathcal{I}$ and so N' cannot be consistent). Thus $SS' \not\models_{\text{con}} \neg p$, a contradiction. □

The converse of the combination of Theorems 3.11 and 3.12 does not necessarily hold. In $KB = \{p,\ \neg p \vee q, \neg p \vee \neg r, \neg q \vee r\}$ and $\mathcal{B} = FOUR$, for instance, $\{p, \neg p \vee q\}$ supports q, and there is no support set for $\neg q$, although q is incomplete. However, there are certain important cases in which the converse of 3.11 is true. The following propositions specify such cases.

PROPOSITION 3.13. *Let l be a literal s.t. $KB \models_{\text{con}} l$. l is recoverable iff it has a support set.*

Proof. The ‘only if’ direction was proved in Theorem 3.11. For the ‘if’ direction note that since l has a support set, it cannot be spoiled. Nor can l be incomplete, since $KB \models_{\text{con}} l$. This is also the reason why $\overline{l}$ cannot be recoverable. The only possibility left, then, is that l is recoverable. □

As a corollary of the last proposition we can specify another condition that guarantees that a given literal is recoverable. This time, however, instead of considering models of the whole knowledge base, it is sufficient to check only the models of a subset that supports l.

COROLLARY 3.14. *If a literal l has a support set $SS(l)$, and $SS(l) \models l$, then l is recoverable.*

Proof. Since $\models$ is monotonic, the assumption that $SS(l) \models l$ implies that $KB \models l$ as well, and so $KB \models_{\text{con}} l$. From Proposition 3.13, then, l is recoverable. □

Note. Demanding that $SS(l) \models l$ in the definition of a support set would have been too restrictive. By doing so, not every recoverable literal would have been guaranteed to be associated with a support set. In this case, for instance, p would have been a recoverable atom in $KB = \{q, \neg q \vee p\}$. This is actually the only support set of p. But $M(q) = \top$, $M(p) = \bot$ is a model of KB in which p is not assigned a designated truth value; hence there is no support set $SS(p) \subseteq KB$ s.t. $SS(p) \models p$.

From the last corollary one can deduce another way of assuring that a given literal is recoverable.

COROLLARY 3.15. *Every literal l such that $l \in KB$ and $\bar{l} \notin KB$ is recoverable.*

Proof. It is easy to see that if $\bar{l} \notin KB$, then $SS(l) = \{l\}$ is a support set of l (not necessarily maximal). Since $SS(l) \models l$, l is recoverable by Corollary 3.14. □

Note. The converse of Corollary 3.15 is, of course, not true. To see that, consider, for example, $KB_1 = \{p, p \to q\}$, or $KB_2 = \{p \to q, \neg p \to q\}$. In these knowledge bases q is recoverable although $q \notin KB_i$ $(i = 1, 2)$. Moreover, KB_2 is an example of a knowledge base that contains a recoverable literal although there is no $l \in \mathcal{L}(KB)$ s.t. $l \in KB$.

Another refinement of Proposition 3.13 is considering only the *reductions* of the mcms of KB to the support sets: Suppose that $KB' \subseteq KB$. Denote by $\mathrm{con}(KB) \downarrow KB'$ the reductions of the mcms of KB to the language of KB'. Then we have the following.

PROPOSITION 3.16. *l is recoverable iff it has a support set $SS(l)$ s.t. $SS(l) \models_{\mathrm{con}(KB)\downarrow SS(l)} l$ (see also Definition 2.17).*

Proof. If l is recoverable, then by Theorem 3.11 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 remain the same when reducing the mcms to the language of $SS(l)$; hence $SS(l) \models_{\mathrm{con}(KB)\downarrow SS(l)} l$. For the converse, let $SS(l)$ be a support set of l. Then $\bar{l}$ cannot be recoverable because of Theorem 3.12 (since $SS(l)$ supports its complement). Nor l can be spoiled, since spoiled literals obviously have no support sets. It remains to show that l cannot be incomplete, but this follows from the fact that if l is incomplete, there would have been an mcm M of KB (and so a model of $SS(l)$) s.t. $M(l) \notin \mathcal{F}$. In the reduction of M to the language of $SS(l)$, l is assigned the same truth value, hence $SS(l) \not\models_{\mathrm{con}(KB)\downarrow SS(l)} l$. □

We have already seen that every recoverable literal is guaranteed to have at least one support set. Sometimes, however, it might have several support sets. In such a case it seems reasonable to prefer those that are maximal (w.r.t. containment relation). We next consider such sets.

DEFINITION 3.17. Let l be a recoverable literal, and suppose that M is an mcm such that $M(l) \in \mathcal{F}$ and $M(l) \notin \mathcal{I}$. The support set of l that is *associated with M* is $SS_M(l) = \{\psi \in KB \mid \mathcal{A}(\psi) \cap \mathrm{Inc}_M(KB) = \emptyset\}$.*

PROPOSITION 3.18. *Every maximal support set of a recoverable literal l is associated with some mcm M s.t. $M(l) \notin \mathcal{I}$.*

Proof. Again, we shall prove the claim just for the case $l = p$, where $p \in \mathcal{A}(KB)$. Suppose that $SS'(p)$ is an arbitrary support set of p. Let N' be a consistent model of $SS'(p)$, and N its expansion to the whole KB. Consider any

* This is indeed a support set of l. See the proof of Theorem 3.11.

mcm M that satisfies $N \leq_{\mathrm{con}} M$. Since $\mathcal{A}(SS'(p)) \subseteq \mathcal{A}(KB) \setminus \mathrm{Inc}_N(KB) \subseteq \mathcal{A}(KB) \setminus \mathrm{Inc}_M(KB)$, then every formula $\psi \in SS'(p)$ consists only of literals that are assigned consistent truth values by M. Hence $SS'(p) \subseteq SS_M(p)$. Since $SS_M(p)$ is also a support set, $SS'(p) = SS_M(p)$ in case $SS'(p)$ is maximal. □

One can rephrase the last proposition as follows.

COROLLARY 3.19. *A knowledge base is recoverable iff it has a recoverable literal.*

Proof. By Definition 3.9, a recoverable knowledge base KB must have a maximal support set, and by Proposition 3.18, such a set is of the form $SS_M(l)$, where l is a recoverable literal of KB. In the converse direction, let l be a recoverable literal of KB. In the proof of Theorem 3.11 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 3.18 this support set is contained in some maximal support set of l (which is also associated with some mcm of KB), and so KB is a recoverable knowledge base. □

The converse of the Proposition 3.18 is not true; not every support set that is associated with some mcm is necessarily maximal. There may be another mcm whose associated support set is bigger. To see that, consider $KB = \{p, p \to r,\ r \to s, r \to \neg s\}$. Both $M_1 = \{p : t, r : \top, s : t\}$ and $M_2 = \{p : t, r : t, s : \top\}$ are mcms of KB, but $SS_{M_1}(p) = \{p\} \subset \{p, p \to r\} = SS_{M_2}(p)$. In Section 3.4 we shall see that $SS_{M_2}(p)$ is not only bigger than $SS_{M_1}(p)$, but also preferable according to some other criteria.

COROLLARY 3.20. *For every recoverable literal l there exists an mcm M of KB in which $M(l) = t$, and for which $SS_M(l)$ is a maximal support set.*

Proof. Suppose that $l = p$. Consider an mcm N of KB s.t. $N(p) \in \mathcal{F} \setminus \mathcal{I}$, and whose associated support set $SS_N(p)$ is maximal (from Proposition 3.18, such an mcm exists). Let M be the valuation that assigns t to p and that is identical to N on every element of $\mathcal{A}(KB) \setminus \{p\}$. Suppose that ψ is an extended clause of KB. If p is a disjunct of ψ, then since $M(p) = t$, necessarily $M(\psi) \in \mathcal{F}$ by Lemma 2.23. If not, then by Lemma 2.23 again, there must be some literal of ψ other than p or $\neg p$ that is assigned a designated truth value in N. Such a literal is assigned a designated truth value in M as well; hence $M(\psi) \in \mathcal{F}$ in this case also. It follows that M is a model of KB. Moreover, since $\mathrm{Inc}_M(KB) = \mathrm{Inc}_N(KB)$, M is also an mcm of KB, and $SS_M(p) = SS_N(p)$. Hence, M and $SS_M(p)$ are the required mcm and support set, respectively. □

3.3. COMPUTING MCMS AND SUPPORT SETS FOR STRATIFIED KNOWLEDGE BASES

In general, computing mcms for a given knowledge base and discovering its recoverable literals might not be an easy task. Even in the simplest cases, where the bilattice is $FOUR$ with $\mathcal{I} = \{\top, \bot\}$ and the knowledge base is consistent, finding the recoverable literals reduces to the problem of logical entailment. Therefore, in this subsection we confine ourselves to a special (nevertheless common) family of knowledge bases, for which we provide an efficient algorithm that computes their maximal recoverable subsets.

DEFINITION 3.21. Let $(\mathcal{B}, \mathcal{F})$ be a finite logical bilattice with an inconsistency set $\mathcal{I}$.

(a) Denote $\mathcal{T}_\top = \{b \mid b \in \mathcal{F} \cap \mathcal{I}\}$, $\mathcal{T}_t = \{b \mid b \in \mathcal{F} \setminus \mathcal{I}\}$, $\mathcal{T}_f = \{b \mid \neg b \in \mathcal{F} \setminus \mathcal{I}\}$.
(b) Let $b_\top$, b_t, and b_f denote the k-meet of all the elements of $\mathcal{T}_\top$, $\mathcal{T}_t$, and $\mathcal{T}_f$, respectively (i.e., $b_x = \otimes\{b \mid b \in \mathcal{T}_x\}$ for $x \in \{\top, f, t\}$). We also denote by $b_\bot$ an arbitrary element that is k-minimal among the consistent elements of B.

Intuitively, $b_\top, b_t, b_f$, and $b_\bot$ are four elements of B that strongly resemble in their properties to the four elements of $FOUR$. They adequately represent the main four types of the elements of $(\mathcal{B}, \mathcal{F})$.

EXAMPLE 3.22. If $\mathcal{B} = FOUR$ and $\mathcal{I} = \{\top\}$, then $b_\top = \top$, $b_t = t$, $b_f = f$, and $b_\bot = \bot$. If $\mathcal{B} = DEFAULT$ and $\mathcal{I} = \{b \mid b \neq \neg b\}$, then $b_\top = \top$, $b_t = t$, $b_f = f$, and $b_\bot$ is an element of the set $\{dt, df\}$. If $\mathcal{B} = NINE$, $\mathcal{F} = \{b \mid b \geq_k dt\}$, and $\mathcal{I} = \{b \mid b \geq d\top\}$, then $b_\top = d\top$, $b_t = dt$, $b_f = df$, and $b_\bot = \bot$.

LEMMA 3.23. *For every finite logical bilattice $(\mathcal{B}, \mathcal{F})$, $b_\top \in \mathcal{T}_\top$, $b_t \in \mathcal{T}_t$, and $b_f \in \mathcal{T}_f$. Also, $b_\top = \otimes\{b \mid b, \neg b \in \mathcal{F}\}$, $b_t = \otimes\{b \mid b \in \mathcal{F}\}$, $b_f = \otimes\{b \mid \neg b \in \mathcal{F}\}$, and $b_f = \neg b_t$. Also, $b_\bot = \bot$ iff $\bot \notin \mathcal{I}$.*

Proof. Let $b_\mathcal{F} = \otimes\{b \mid b \in \mathcal{F}\}$. We show that $b_t = b_\mathcal{F}$, leaving the other parts to the reader. Obviously, $b_\mathcal{F} \in \mathcal{F}$. We show that $b_\mathcal{F} \notin \mathcal{I}$. Assume otherwise. Then $\neg b_\mathcal{F} \in \mathcal{F}$ as well. Since $t \in \mathcal{F}$, then $t \geq_k b_\mathcal{F}$. Thus $f = \neg t \geq_k \neg b_\mathcal{F} \in \mathcal{F}$, and so $f \in \mathcal{F}$, contradicting Lemma 2.7(b). It follows that $b_\mathcal{F} \in \mathcal{T}_t$. Hence $b_\mathcal{F} \geq_k b_t$. Since obviously $b_\mathcal{F} \leq_k b_t$, then $b_\mathcal{F} = b_t$. □

DEFINITION 3.24. KB_ν – the *dilution* of KB w.r.t. a partial valuation ν – is constructed from KB by the following transformations:

(1) Deleting every $\psi \in KB$ s.t. $\nu(\psi) \in \mathcal{F}$ (in other words, ψ is deleted if it has a literal that is assigned a designated value by ν).
(2) Removing from what is left every occurrence of a literal l s.t. $\nu(l)$ is defined and $\nu(l) \notin \mathcal{F}$.

$i = 0;\ \ KB_0 = KB;$

while $(KB_i \neq \emptyset)$ {

  if $(\exists p \in \mathcal{A}(KB_i)$ s.t. $p \in KB_i$ and $\neg p \in KB_i)$ then $\Phi_i(p) = b_\top;$

  else if $(\exists p \in \mathcal{A}(KB_i)$ s.t. $p \in KB_i)$ then $\Phi_i(p) = b_t;$

  else if $(\exists p \in \mathcal{A}(KB_i)$ s.t. $\neg p \in KB_i)$ then $\Phi_i(p) = b_f;$

  else print "KB is not stratified" and exit;

  $KB_{i+1} = (KB_i)_{\Phi_i};$ /*** dilution ***/

  $\forall p \in \mathcal{A}(KB_i) \setminus \mathcal{A}(KB_{i+1})$ s.t. p wasn't picked in KB_i, $\Phi_i(p) = b_\bot;$ /*** filling ***/

  $i{+}{+};$

}

output: $\Phi = \bigcup_{0 \leq j \leq i-1} \Phi_j;$

Figure 3. An algorithm for recovering stratified knowledge bases.

DEFINITION 3.25. A knowledge base KB is called *stratified* if there is a set of 'stratifications' $KB_0 = KB,\ KB_1, \ldots, KB_n = \emptyset$ so that for every $0 \leq i \leq n-1$ there is a $p \in \mathcal{A}(KB_i)$ s.t.

(a) Either $p \in KB_i$ (and then p is called a *positive fact*), or $\neg p \in KB_i$ (and then p is a *negative fact* of KB_i).
(b) KB_{i+1} is a dilution of KB_i w.r.t. the partial valuation $p : b_t$ if p is a positive fact of KB_i, $p : b_f$ if p is a negative fact of KB_i, and $p : b_\top$ if p is both a positive and a negative fact of KB_i.*

EXAMPLE 3.26. Consider again the knowledge base KB of Examples 2.18, 3.2, and 3.10. A possible stratification for KB is $KB_0 = \{s, \neg s, r_1, r_1 \to \neg r_2, r_2 \to i\}$, $KB_1 = \{r_1, r_1 \to \neg r_2, r_2 \to i\}$, $KB_2 = \{\neg r_2,\ r_2 \to i\}$, and $KB_3 = \emptyset$.

In all the examples given here (see especially those of Section 6), as well as in most of the known examples of the literature, the knowledge bases involved are stratified.

The algorithm given in Figure 3 can be applied for checking whether a knowledge base is stratified, and for recovering stratified knowledge bases (see Corollary 3.29).

Notes.

(1) The process of Figure 3 may produce several valuations for KB, each of which is determined by a sequence of the picked atomic formulae $\{p_0, p_1, \ldots, p_n\}$. For abbreviation we shall just write Φ when referring to arbitrary valuation produced by the algorithm, instead of $\Phi(p_0, p_1, \ldots, p_n)$.

* Note that while $\mathcal{B}$, $\mathcal{F}$, and $\mathcal{I}$ affect the particular values of $b_\top$, b_t, and b_f, they do not determine whether KB is stratified.

(2) By Theorem 3.3, if $\Phi_i(l) = b_\top$, then l is a spoiled literal of KB_i. Similarly, by Corollary 3.15, if $\Phi_i(l) = b_t$, then l is a recoverable literal of KB_i, and if $\Phi_i(l) = b_f$, then $\overline{l}$ is recoverable in KB_i. By Theorem 3.28 below, if $\Phi_i(l) = b_\bot$, then l is incomplete in KB_i.

(3) It is possible to assign any other truth value to the atoms that are assigned $b_\bot$ (during the 'filling' process; see line 8 of Figure 3), but if this value is inconsistent, then Φ cannot be an mcm of KB (see the proof of Theorem 3.28). Also, if this value is not $\bot$, then Φ cannot be minimal w.r.t. $\leq_k$ (see Proposition 3.32). The value $b_\bot$ assures that Φ would be a $\leq_k$-minimal mcm (see Section 4). □

EXAMPLE 3.27. Consider the set $KB = \{p, p \vee q, \neg p \vee r, \neg p \vee \neg r, \neg p \vee \neg u, \neg p \vee \neg v, u \vee v\}$, where $\mathcal{B} = FOUR$ and $\mathcal{I} = \{\top\}$. Then $b_\top = \top$, $b_t = t$, $b_f = f$, and $b_\bot = \bot$. Our algorithm produces two mcms of KB, denoted Φ_a and Φ_b:

$$\Phi_a(p) = t, \Phi_a(q) = \bot, \Phi_a(r) = \top, \ \Phi_a(u) = f, \Phi_a(v) = \top,$$

$$\Phi_b(p) = t, \Phi_b(q) = \bot, \Phi_b(r) = \top, \ \Phi_b(u) = \top, \Phi_b(v) = f.$$

Note that in this case there are other mcms of KB (such as $\{p : \top, q : \bot, r : \bot, u : t, v : \bot\}$), but neither of the other mcms can be used for constructing (maximal) support sets in KB, since each one of them assigns an inconsistent truth value to p, which is the only recoverable literal of KB. Theorems 3.28 and 3.30 show that this holds in general.

Figure 4 illustrates the processing of the algorithm for KB and its recoverable literal, p.

THEOREM 3.28. *The process of Figure* 3 *checks whether a given knowledge base KB is stratified. If KB is not stratified, it exits; otherwise it halts and produces an mcm of KB.*

Proof. To see the first part of the theorem, note that if a knowledge base is stratified, then any order in which the facts are chosen determines a stratification. This is so since dilution does not change facts; a fact (positive, negative, or both) of a certain stratification level remains a fact in the successive levels unless it is used for the next dilution. Therefore, if there are two facts p_1 and p_2 in some KB_i, there is a stratification KB_i, $KB_{i+1} = (KB_i)_{p_1:b_1}, \ldots, KB_n = \emptyset$ iff there is a stratification KB_i, $KB_{i+1} = (KB_i)_{p_2:b_2}, \ldots, KB_m = \emptyset$. Therefore, the algorithm fails in constructing a stratification for KB iff there is no possible way of providing such a stratification, and so the algorithm halts without a valuation for $\mathcal{A}(KB)$ iff KB is not stratified. □

Suppose, next, that KB *is* stratified.

Step 0: $KB_0 = \{p,\ p\vee q,\ \neg p\vee r,\ \neg p\vee\neg r,\ \neg p\vee\neg u,\ \neg p\vee\neg v,\ u\vee v\}$
pick p. set $\Phi_0(p) = t$ and $\Phi_0(q) = \bot$.

Step 1: $KB_1 = \{r,\ \neg r,\ \neg u,\ \neg v,\ u\vee v\}$
pick r. set $\Phi_1(r) = \top$.

$KB_2 = \{\neg u,\ \neg v,\ u\vee v\}$

Step 2: pick u. set $\Phi_{2a}(u) = f$. | pick v. set $\Phi_{2b}(v) = f$.

Step 3: $KB_{3a} = \{\neg v,\ v\}$, pick v. set $\Phi_{3a}(v) = \top$ | $KB_{3b} = \{\neg u,\ u\}$, pick u. set $\Phi_{3b}(u) = \top$

Step 4: $KB_{4a} = \emptyset$ | $KB_{4b} = \emptyset$

$SS_a(p) = \{p,\ p\vee q,\ \neg p\vee\neg u\}$ | $SS_b(p) = \{p,\ p\vee q,\ \neg p\vee\neg v\}$

Figure 4. Construction of recovered subsets for KB (Example 3.27).

LEMMA 3.28a. *The algorithm constructs well-defined valuations.*

Proof. We have to show that the process terminates after a finite number of steps (the minimal n, s.t. $KB_n = \emptyset$), and the result is a valuation Φ for KB. Indeed, a picked atom $p \in \mathcal{A}(KB_i)$ does not appear in any one of KB_j for $j > i$. Also, there may be other atoms that are eliminated in the dilutions of KB_i. Every one of these atomic formulae is assigned its (unique) truth value at the ith step, and so $|\mathcal{A}(KB_{i+1})| \leq |\mathcal{A}(KB_i)| - 1$. On the other hand, an atomic formula does not appear in $\mathcal{A}(KB_{i+1})$ only if it is assigned a value of $\{b_\top, b_t, b_f, b_\bot\}$ in the ith step. Therefore, the process terminates after $|\mathcal{A}(KB)|$ steps at the most and assigns a unique truth value to every member of $\mathcal{A}(KB)$. $\square$

LEMMA 3.28b. *Every valuation Φ produced by the algorithm is a model of KB.*

Proof. Let ψ be an extended clause that appears in KB. By Definition 3.24 and the algorithm of Figure 3 it is obvious that ψ is eliminated from KB_{i+1} during the transformation from KB_i to KB_{i+1} iff (at least) one of its literals l is assigned a designated truth value by Φ_i (note that a formula cannot be eliminated by sequentially removing every literal according to (2) of Definition 3.24, since the last literal left must be assigned a designated value). Since $\Phi(l) = \Phi_i(l)$, Φ assigns a designated truth value to at least one of the literals that appear in ψ. By Lemma 2.23, then, $\Phi(\psi) \in \mathcal{F}$.

LEMMA 3.28c. *Every valuation Φ produced by the algorithm is a most consistent model of KB.*

Proof. The proof is by an induction on the number of steps (n) that are required to create Φ. If $n = 0$, then $KB_1 = \emptyset$, so there is only the initial step in which Φ_0 might assign a value from $\mathcal{I}$ only to a spoiled literal, so Φ must be most consistent. Suppose now that it takes $n \geq 1$ steps to create Φ. Then

$$\mathrm{Inc}_{\Phi}(KB) = \bigcup_{0 \leq i \leq n} \mathrm{Inc}_{\Phi_i}(KB_i) = \mathrm{Inc}_{\Phi_0}(KB) \cup \mathrm{Inc}_{\Phi'}(KB_1), \tag{1}$$

where $\Phi' = \bigcup_{1 \leq i \leq n} \Phi_i$. Now, let M be any mcm of KB.

$$\begin{aligned} \mathrm{Inc}_M(KB) &= \{p \in \mathcal{A}(KB) \setminus \mathcal{A}(KB_1) \mid M(p) \in \mathcal{I}\} \cup \\ &\quad \cup \{p \in \mathcal{A}(KB_1) \mid M(p) \in \mathcal{I}\} \\ &= \{p \in \mathcal{A}(KB) \setminus \mathcal{A}(KB_1) \mid M(p) \in \mathcal{I}\} \cup \mathrm{Inc}_M(KB_1). \end{aligned} \tag{2}$$

By its definition, Φ_0 may assign an inconsistent truth value only to a spoiled literal of KB. By Theorem 3.3(b) this literal is assigned an inconsistent value in every mcm of KB, especially M. Therefore,

$$\mathrm{Inc}_{\Phi_0}(KB) \subseteq \{p \in \mathcal{A}(KB) \setminus \mathcal{A}(KB_1) \mid M(p) \in \mathcal{I}\}. \tag{3}$$

• Suppose first that M is a model of KB_1. Since the creation of Φ' requires only $n-1$ steps, then by the induction hypothesis Φ' is an mcm of KB_1. In particular, either $\mathrm{Inc}_{\Phi'}(KB_1)$ and $\mathrm{Inc}_M(KB_1)$ are Incomparable w.r.t. containment relation, or else

$$\mathrm{Inc}_{\Phi'}(KB_1) \subseteq \mathrm{Inc}_M(KB_1). \tag{4}$$

By (1)–(4), either $\mathrm{Inc}_{\Phi}(KB)$ and $\mathrm{Inc}_M(KB)$ are incomparable, or $\mathrm{Inc}_{\Phi}(KB) \subseteq \mathrm{Inc}_M(KB)$. Hence Φ is an mcm of KB.

• If M is *not* a model of KB_1, then there is a $\psi_1 \in KB_1$ s.t. $M(\psi_1) \notin \mathcal{F}$. Since M is a model of KB, then by Lemma 2.23 there is a $\psi \in KB$ and $l \in \mathcal{L}(\psi)$ s.t. $M(l) \in \mathcal{F}$, and $\{l\} \cup \mathcal{L}(\psi_1) \subseteq \mathcal{L}(\psi)$. Obviously, $l \in \mathcal{A}(KB) \setminus \mathcal{A}(KB_1)$. But then $\Phi_0(l) \notin \mathcal{F}$ (otherwise ψ is eliminated in the dilution, and so $\psi_1 \notin KB_1$). Moreover, $\Phi_0(\bar{l}) \in \mathcal{F}$, since if $\Phi_0(\bar{l}) \notin \mathcal{F}$ then necessarily $\Phi_0(l) = b_\perp$, and this happens only if there is a literal $l' \in \mathcal{L}(\psi)$ s.t. $\Phi_0(l') \in \mathcal{F}$, and in this case ψ is eliminated in the dilution, that is, $\psi_1 \notin KB_1$. Therefore, $\Phi_0(\bar{l}) \in \mathcal{F}$, $\Phi_0(l) \notin \mathcal{F}$, and by the definition of Φ_0, $\bar{l}$ was picked by Φ_0. Since $\Phi_0(l) \notin \mathcal{F}$, l cannot be spoiled. Also, KB is stratified; thus $\bar{l} \in KB_0$, and so it must be a recoverable literal of KB_0, that is, it is a recoverable literal of KB. Now, since M is an mcm of KB, $M(\bar{l}) \in \mathcal{F}$. But we have shown that $M(l) \in \mathcal{F}$ as well; hence $M(l) \in \mathcal{I}$, while by Lemma 3.23 $\Phi_0(l) = b_f \notin \mathcal{I}$. Therefore $\text{Inc}_M(KB) \not\subseteq \text{Inc}_\Phi(KB)$, and we are done.

This proves Theorem 3.28. □

COROLLARY 3.29. *Suppose that KB is stratified. Then the algorithm above provides a support set for every recoverable literal of KB that is not assigned the value $b_\top$.*

Proof. By Theorem 3.28, every valuation Φ that is generated by the algorithm is an mcm of KB. If a recoverable literal l of KB was not assigned the value $b_\top$, then $\Phi(l) \notin \mathcal{I}$. Hence, by the proof of Theorem 3.11, $SS_\Phi(l)$ is a support set of l. □

THEOREM 3.30. *Let Φ be a valuation produced by the algorithm for stratified KB, and let l be a recoverable literal of KB that is not assigned the value $b_\top$ by Φ. Then $SS_\Phi(l)$ is a maximal support set of l.*

Proof. By Corollary 3.29, $SS_\Phi(l)$ is a support set of l. It remains to show that $SS_\Phi(l)$ is also a *maximal* set with this property. Suppose otherwise. Then by Proposition 3.18 there is an mcm M of KB s.t. $SS_\Phi(l) \subset SS_M(l)$. Hence $\text{Inc}_\Phi(KB) \neq \text{Inc}_M(KB)$. Since by Theorem 3.28 Φ is also an mcm of KB, there is a $p \in \mathcal{A}(KB)$ s.t. $\Phi(p) \neq b_\top$ while $M(p) \in \mathcal{I}$. Consider some $\psi \in KB$ s.t. $p \in \mathcal{A}(\psi)$. Since $M(p) \in \mathcal{I}$, then ψ is not an element of $SS_M(l)$. Now, since $\psi \notin SS_M(l)$, $\psi \notin SS_\Phi(l)$ either. Therefore there is a $q \in \mathcal{A}(\psi)$ s.t. $\Phi(q) = b_\top$. By the definition of Φ this is possible only if there is a stratification $S_0, \ldots, S_n$ of S and an index $1 \leq i \leq n$ s.t. $q, \neg q \in S_i$. Therefore $\Phi(p) \neq b_\perp$ (otherwise, p as well as all the other elements of $\mathcal{A}(\psi)$ are diluted from S_j for some $j \leq i$, and so $q \notin \mathcal{A}(S_i)$). It follows that either $\Phi(p) = b_t$ or $\Phi(p) = b_f$, and therefore either p or $\neg p$ (but not both) is a (positive or negative) fact of some stratification level S_k of S. Hence there is some $\phi \in S$ s.t. $p \in \mathcal{A}(\phi)$ and $\mathcal{A}(\phi) \cap \text{Inc}_\Phi(KB) = \emptyset$ (otherwise, if there is some $r \in \mathcal{A}(\phi)$ s.t. $\Phi(r) = b_\top$, then ϕ and its atoms are diluted in some stage before stage k, and so p cannot be a fact of S_k). Therefore $\phi \in SS_\Phi(l)$ while $\phi \notin SS_M(l)$ – a contradiction. □

EXAMPLE 3.31. Consider again Figure 4. $SS_a(p)$ and $SS_b(p)$ are the recovered support sets produced by the algorithm for the recoverable literal p of KB. Both are maximal.

Next we consider another important property of Φ (see Corollary 3.33 below).

PROPOSITION 3.32. *Let KB be stratified. In every logical bilattice where $b_\perp = \perp$, Φ is k-minimal* ($\Phi \in min(KB)$. *Recall Definition* 2.16(d)).

Proof. The proof is by an induction on the number of steps required to create Φ:

$n = 0$: Φ_0 may assign to a spoiled literal of KB the value $b_\top$, which is the only k-minimal possible value (see Theorem 3.3(b)). The same is true for any recoverable literal that is assigned b_t, and for a complement of recoverable literal that is assigned b_f. It is also obviously true for all the literals that are assigned $\perp$.

$n \geq 1$: Let M be a model of KB, and suppose for a contradiction that $M <_k \Phi$. By the induction hypothesis, Φ_1 is a k-minimal model of KB_1. If M is a model of KB_1 then there is a $q \in \mathcal{A}(KB_1)$ s.t. $M(q) \not\leq_k \Phi_1(q)$ and so $M \not<_k \Phi$. The other possibility is that M is not a model of KB_1. In this case there must be a $\psi_1 \in KB_1$ s.t. $M(\psi_1) \notin \mathcal{F}$. Since M is a model of KB, then by Lemma 2.23 there is a $\psi \in KB$ and an $l \in \mathcal{L}(\psi)$ s.t. $M(l) \in \mathcal{F}$, and $\{l\} \cup \mathcal{L}(\psi_1) \subseteq \mathcal{L}(\psi)$. But then $\Phi(l) \notin \mathcal{F}$ (otherwise, ψ is eliminated in the dilution of KB and so $\psi_1 \notin KB_1$), while $M(l) \in \mathcal{F}$. Since $\mathcal{F}$ is upward closed w.r.t. $\leq_k$ it follows that $M(l) \not<_k \Phi(l)$; therefore $M \not<_k \Phi$ – a contradiction again. □

COROLLARY 3.33. *Let KB be stratified. In every logical bilattice for which $b_\perp = \perp$, Φ is k-minimal among the mcms of KB* (*see Section* 4 *for the importance of this*).

Proof. By Theorem 3.28, $\Phi \in \text{con}(KB)$. Since $b_\perp = \perp$, then by Proposition 3.32, $\Phi \in min(KB)$. Therefore Φ is a $\leq_k$-minimal model among the mcms of KB. □

Next we consider the complexity of the algorithm. As it is shown below, this is a particularly efficient mechanism for recovering stratified knowledge bases.

PROPOSITION 3.34. *It takes* $\text{O}(|KB| \cdot |\mathcal{A}(KB)|)$ *running time to check whether a given knowledge base is stratified, and if so, this is also the time required to recover it* (*i.e., to provide a recoverable subset of KB*).

Proof. Computing stage i of the algorithm requires only $\text{O}(|KB_i|)$ running time. Since there are $\text{O}(|\mathcal{A}(KB)|)$ stages at the most, the complexity of the whole process is no more than $\text{O}(|KB| \cdot |\mathcal{A}(KB)|)$. Now, since we have already shown that for stratified knowledge bases the algorithm generates mcms, this is also the time required to recover KB. □

Another method of recovering inconsistent knowledge bases is mentioned at the end of Section 6.2.

3.4. CHOOSING THE PREFERRED SUPPORT SETS

As we have already noted, the support sets may be viewed as representing possible consistent interpretations (states) of the world that is inconsistently described in KB. Since in general there are several support sets that can be produced from a polluted knowledge base, one has to develop means that would guide one to an interpretation that is most likely to be the accurate description. In this section we suggest some heuristics for choosing the preferred support set.

A first observation is that when there are two support sets SS_1 and SS_2 s.t. $SS_1 \subset SS_2$, it seems reasonable to prefer the latter, namely, to choose the maximal support w.r.t. containment relation (cf. Propositions 3.18 and 4.5). Still, in many cases there are several such sets. Here are some other criteria that might be useful for a proper choice of the preferred set.

3.4.1. *Maximal Information Considerations*

A possible approach for taking precedences among the support sets is to define some quantitative estimation on the plausibility of each set. Lozinskii [28], for example, takes the *quantity of semantic information* to be the criteria for such estimations.* The quantity of information in a set S of classical formulae is defined there to be $I(S) = |\mathcal{A}(S)| - \log_2 |mod(MCS(S))|$, where $mod(MCS(S))$ is the set of all the models of the maximal consistent subsets of S (see [28] for a detailed discussion and justifications for taking this formula as representing information). A possible analogue in the case of a logical bilattice $(\mathcal{B}, \mathcal{F})$ may be $I_1(S) = |\mathcal{A}(S)| - \log_{2|\mathcal{F}|} |mod(MCS(S))|$. Since we consider the mcms as the most relevant interpretations for the recovery process, we here use a different definition: $I_2(S) = |\mathcal{A}(S)| - \log_c |\text{con}(MCS(S))|$, where: $c = |\{b \in B \mid b \in \mathcal{F} \setminus \mathcal{I} \vee \neg b \in \mathcal{F} \setminus \mathcal{I}\}|$ (see Proposition 3.35 below for some justifications for taking this particular c as the base of the logarithm). Since $c \geq 2$ (always $\{t, f\} \subseteq \{b \in \mathcal{F} \setminus \mathcal{I} \vee \neg b \in \mathcal{F} \setminus \mathcal{I}\}$), $I_2(S)$ is well defined.

A possible strategy, then, would prefer support sets with maximal information. Since support sets are *consistent*, $MCS(S)$ is just $\{S\}$, so $I_1(S)$ and $I_2(S)$ reduce to $|\mathcal{A}(S)| - \log_{2|\mathcal{F}|} |mod(S)|$ and $|\mathcal{A}(S)| - \log_c |\text{con}(S)|$, respectively.

The next proposition shows that both $I_1(S)$ and $I_2(S)$ accord with Lozinskii's intuition regarding the notion of information (cf. [28, Theorem 3.1]):

PROPOSITION 3.35.

(a) *An empty set contains no information;* $I_1(\emptyset) = I_2(\emptyset) = 0$.
(b) *A set S consisting of complementary literals $p, \neg p$ for every $p \in \mathcal{A}(S)$ contains no semantic information.*
(c) *If S is a consistent set of formulae, and ψ is a formula s.t. $\mathcal{A}(\psi) \subseteq \mathcal{A}(S)$ and $S \models \psi$, then $I_1(S) = I_1(S \cup \{\psi\})$ and $I_2(S) = I_2(S \cup \{\psi\})$.*
(d) *If S is a consistent set of formulae, and ψ is a consistent formula s.t. $\mathcal{A}(\psi) \subseteq \mathcal{A}(S)$ and $S \cup \{\psi\}$ is inconsistent, then $I_2(S) > I_2(S \cup \{\psi\})$.*
(e) *If S has only one model, then $I_1(S) = 0$. If S is consistent and has one mcm, then $I_2(S)$ is maximal.***

* As a matter of fact, the quantitative approach is used in [28] for a slightly different goal: giving semantics to inconsistent systems.

** In this particular case $I_1(S)$ and Lozinskii's $I(S)$ do not behave in the same way (cf. [28, Theorem 3.1, part (vi)]). The difference is due to the nature of logical bilattices as multiple valued: If S has only one (degenerate) model in a logical bilattice, this single model is $\{p : \top \mid p \in \mathcal{A}(S)\}$. This model actually tells us nothing; hence S contains no meaningful information. However, this is certainly *not* the case for consistent sets that have one mcm. In this case the mcm is meaningful, and the fact that there are no other possible models just increases the validity of that single model as well as its respective semantic information about S.

Proof. (a) $M = \{p : \top \mid p \in \mathcal{A}(S)\}$ is a model of every set S; hence $|mod(MCS(S))| \geq 1$. On the other hand, if $S = \emptyset$, then S itself is the only most consistent subset; hence $|mod(MCS(S))| = |mod(S)| \leq |B|^{|\mathcal{A}(S)|} = 1$. Thus, $|mod(MCS(S))| = |mod(S)| = 1$, and so, by the definition of I_1, $I_1(S) = 0$. Regarding I_2, since the set of the mcms of S consists of minimal elements of a nonempty set (that of the models of S), then $|\text{con}(S)| \geq 1$. On the other hand, we have shown that whenever $S = \emptyset$, we have that $|\text{con}(S)| \leq |mod(S)| = 1$. Thus $|\text{con}(MCS(S))| = |\text{con}(S)| = 1$, and so $I_2 = 0$.

(b) Consider $S = \{p_i, \neg p_i \mid 1 \leq i \leq n\}$. This particular S has 2^n maximal consistent subsets. Each one has $|\mathcal{F}|^n$ models, and $(\frac{c}{2})^n$ mcms (since there is no $b \in B$ such that both $b \in \mathcal{F} \setminus \mathcal{I}$ and $\neg b \in \mathcal{F} \setminus \mathcal{I}$, every p_i in a possible subset can be assigned exactly $\frac{c}{2}$ different values from $\mathcal{F} \setminus \mathcal{I}$). Hence, $I_1(S) = n - \log_{2|\mathcal{F}|} 2^n|\mathcal{F}|^n = 0$, and $I_2(S) = n - \log_c 2^n(\frac{c}{2})^n = 0$.

(c) Since $\mathcal{A}(\psi) \subseteq \mathcal{A}(S)$, then $\mathcal{A}(S \cup \{\psi\}) = \mathcal{A}(S)$. Also, the assumptions that S is consistent and that $S \models \psi$ easily imply that $\text{mod}(MCS(S)) = \text{mod}(MCS(S \cup \{\psi\}))$ and $\text{con}(S) = \text{con}(S \cup \{\psi\})$. Thus, $I_1(S) = I_1(S \cup \{\psi\})$ and $I_2(S) = I_2(S \cup \{\psi\})$.

(d) The proof in [28, Theorem 3.1, part (v)] is suitable for the present case as well. We repeat the proof adjusted to our notations: S is a maximal consistent subset of $S \cup \{\psi\}$, and since $\psi \notin S$ ($S \cup \{\psi\}$ is inconsistent, while S is not), there must be another maximal consistent subset $S' \subset S \cup \{\psi\}$ s.t. $\psi \in S'$. S and S' have no mcm in common, since such a model would have been a consistent model (as a model of S), which is also a model of the inconsistent set $S \cup \{\psi\}$. Hence $\text{con}(MCS(S)) = \text{con}(S) \subset \text{con}(MCS(S \cup \{\psi\}))$, and so $I_2(S) > I_2(S \cup \{\psi\})$.

(e) If S has only one model, this model must assign $\top$ to every element of $\mathcal{A}(S)$ (this is a model of every S). Hence, using parts (b) and (d) of Theorem 3.3, S must be of the form $\{p, \neg p \mid p \in \mathcal{A}(S)\}$. Thus, by part (b), $I_1(S) = 0$. On the other hand, if S is consistent and has exactly one mcm, then $I_2(S) = |\mathcal{A}(S)|$, which is the maximal possible value of $I_2(S)$ for every set S. □

3.4.2. *Largest Size Approach*

Another reasonable approach is to prefer those support sets with the largest size. According to this method some prioritization formula f is defined s.t. $f(S_1) > f(S_2)$ whenever $|S_1| > |S_2|$. The intuition behind this is that the larger the size of the support set, the stronger similarity it has with the original knowledge base. An example of the use of this approach is the heuristic of weighted maximal consistent subsets in [28].

3.4.3. *Maximal Support Consideration*

Since support sets should support recoverable literals, and since the truth values of the recoverable literals are the ones that are most likely to be recovered truthfully (i.e., as they were before polluting the data in KB), then a plausible system may prefer those support sets that simultaneously support as many recoverable literals as possible.

3.4.4. *Prioritizations on the Domain of Discourse*

There might be cases in which the reasoner has reason to believe that some assertions are more trustable than others (for example, when there are different resources with different reliability, or when one receives several news reports about something that has happened and one tends to believe that the later reports are more accurate). In such situations the reasoner might prioritize the atomic formulae and choose the support set whose literal consequences are the greatest with respect to his ordering. For example, suppose that a, b, c, d, and e are the prioritizations of some reasoner in a descending order, and that in this order every atom is considered equal to its negation. Then a subset that entails a, $\neg c$, and d is preferable to a subset that entails, say, a, d, and e.*

We shall return to the above methods of choosing the best support set in Section 6, when we demonstrate these considerations on some examples.

3.5. THE 'ABSOLUTELY RECOVERABLE' FORMULAE

Although there must be a maximal support set for every recoverable literal, there is no guarantee that all the recoverable literals would be part of the same recovered subset of KB (that is, they may not all be simultaneously recovered). In particular, not every recoverable literal must be a part of every recovered knowledge base. In this subsection we consider some conditions that assure that a formula ψ would be a member of every recovered subset of KB.

DEFINITION 3.36. A formula $\psi \in KB$ is said to be *absolutely recoverable* if ψ is a member of every possible recovered subset of KB.

PROPOSITION 3.37. *Let ψ be a formula of a recoverable knowledge base KB. If for every mcm M of KB, and for every $p \in \mathcal{A}(\psi)$, $M(p) \notin \mathcal{I}$, then ψ is absolutely recoverable.*

Proof. If for every mcm M and for every $p \in \mathcal{A}(\psi)$, $M(p) \notin \mathcal{I}$, then in particular $\psi \in SS_M(l)$ for every mcm M and for every recoverable literal l s.t. $M(l) \notin \mathcal{I}$ (such a literal exists, since KB is recoverable; see Corollary 3.19). By Proposition 3.18, every recovered knowledge base is of the form $SS_M(l)$; hence ψ is absolutely recoverable. □

COROLLARY 3.38. *Every element of the set $\{\psi \in KB \mid \forall (l \in \mathcal{L}(\psi))\ \bar{l} \notin \mathcal{L}(KB)\}$ is an absolutely recoverable formula.*

* This approach has often been considered in the literature. One should note, however, that the use of this criterion for making precedences among sets is highly arguable. In the example considered above, for instance, it is not clear which of the two sets $\{a, d\}$ and $\{b, c\}$ should be preferred.

Proof. Suppose that $\psi' \in \{\psi \in KB \mid \forall(l \in \mathcal{L}(\psi))\bar{l} \notin \mathcal{L}(KB)\}$. By the previous proposition it is sufficient to show that every mcm M assigns to every $p \in \mathcal{A}(\psi')$ consistent truth values. Suppose otherwise. Then there is an mcm M' and a $p' \in \mathcal{A}(\psi')$ s.t. $M'(p') \in \mathcal{I}$. Consider the valuation N', defined as follows.

$$N'(q) = \begin{cases} M'(q) & \text{if } q \neq p', \\ t & \text{if } q = p',\ p' \in \mathcal{L}(KB), \text{ and } \neg p' \notin \mathcal{L}(KB), \\ f & \text{if } q = p',\ p' \notin \mathcal{L}(KB), \text{ and } \neg p' \in \mathcal{L}(KB). \end{cases}$$

It is easy to verify that for every $\psi \in KB$, $N'(\psi) \in \mathcal{F}$ whenever $M'(\psi) \in \mathcal{F}$; thus N' is a model of KB. But $\mathrm{Inc}_{N'}(KB) = \mathrm{Inc}_{M'}(KB) \cup \{p'\}$; thus N' is more consistent than M' – a contradiction. □

COROLLARY 3.39. *Let KB_1 and KB_2 be two subsets of KB, s.t. $KB = KB_1 \cup KB_2$, and $\mathcal{A}(KB_1) \cap \mathcal{A}(KB_2) = \emptyset$ (in such a case we say that KB_1 and KB_2 form a* partition *of KB). If KB_i for $i = 1$ or $i = 2$ is consistent, then every $\psi \in KB_i$ is absolutely recoverable.*

Proof. Suppose that KB_1 is consistent, and $\psi \in KB_1$. Let C be a consistent model of KB_1. Again, in order to prove that ψ is absolutely recoverable, it is sufficient to show that for every mcm M of KB, and for every $p \in \mathcal{A}(\psi)$, $M(p) \notin \mathcal{I}$. Otherwise, let M' be an mcm of KB and let $p' \in \mathcal{A}(\psi)$ s.t. $M'(p') \in \mathcal{I}$. Consider the following valuation, defined for every $q \in \mathcal{A}(KB)$ as follows.

$$N(q) = \begin{cases} C(q) & \text{if } q \in \mathcal{A}(KB_1), \\ M'(q) & \text{if } q \in \mathcal{A}(KB_2). \end{cases}$$

N is a model of KB, since by using the fact that KB_i ($i = 1, 2$) form a partition on KB, it is easy to see that for every formula $\phi \in KB$, $N(\phi) = C(\phi)$ if $\phi \in KB_1$, and $N(\phi) = M'(\phi)$ if $\phi \in KB_2$. Moreover, $\mathrm{Inc}_N(KB) = \mathrm{Inc}_{M'}(KB_2) \subset \{p'\} \cup \mathrm{Inc}_{M'}(KB_2) \subseteq \mathrm{Inc}_{M'}(KB)$; thus N is more consistent than M' – a contradiction. □

EXAMPLE 3.40. Consider again the example given in Examples 2.18, 3.2, 3.10, and 3.26. Here, $KB_1 = \{s, \neg s\}$ and $KB_2 = \{r_1, r_1 \rightarrow \neg r_2, r_2 \rightarrow i\}$ form a partition of KB, and KB_2 is consistent. Hence, by Corollary 3.39, every $\psi \in KB_2$ is absolutely recoverable ($r_2 \rightarrow i$ is absolutely recoverable by Corollary 3.38 as well).

3.6. THE INCOMPLETE LITERALS

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

THEOREM 3.41. *l is an incomplete literal in KB iff there exist mcms M_1 and M_2 such that $M_1(l) = f$ and $M_2(l) = t$.*

Proof. The 'if' direction follows directly from the definition of incomplete literals. For the other direction, suppose that p is the atomic part of l. Since l is incomplete iff p is incomplete, it suffices to prove the claim for p. Now, p is incomplete, so $KB \not\models_{\mathrm{con}} p$ and $KB \not\models_{\mathrm{con}} \neg p$. 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 elements of $\mathcal{A}(KB)$. Let M_2 be a valuation that assigns t to p and is equal to N_2 for all the other members of $\mathcal{A}(KB)$. As in the proof of Corollary 3.20, one can easily show that since N_1 and N_2 are mcms of KB, M_1 and M_2 are also mcms of KB. □

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

- The existence of a support set for an incomplete literal is not assured. Consider, for example, $KB = \{p, \neg p,\ p \vee q\}$. Here q is incomplete without any support set. For another example, consider again Example 3.10. The incomplete literal i is a member of a support set (S), but this set and any other support set in KB do not support i.
- Even if there are support sets for an incomplete literal, there can be other subsets that support its negation: For example, in $KB = \{p, \neg p \vee q, r, \neg r \vee \neg q\}$ with $\mathcal{B} = FOUR$, q is incomplete. 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 incomplete although KB is a consistent set. Intuitively, this is so because there isn't enough data in KB about either p or q. Indeed, this knowledge base has *two* classical models ($\{p\}$ and $\{q\}$), both of which are minimal. Without further information there is no way to choose between the two, and so the truth values of the atoms cannot be recovered safely. Until such new information arrives, the two atoms should therefore be considered problematic because of a *lack* of information. These particular two models, and the fact that we cannot choose between them, exactly reflect the information that is contained in this KB.

4. The Minimal mcms of KB

In this section we show that if one is interested only in recovering a finite inconsistent knowledge base (that is, discovering the spoiled, incomplete, and recoverable literals of KB, as well as the corresponding support sets), then it is sufficient to consider only the $\leq_k$-minimal models among the most consistent models of KB (minimal mcms of KB, in short).

NOTATION 4.1.

(a) The set of the minimal mcms of KB will be denoted henceforth by $\Omega(KB)$, or just Ω.
(b) Denote $KB \models_\Omega \psi$ if every minimal mcm of KB is a model of ψ (see Definition 2.17).

Abstractly, we can view the construction of Ω as a composition of the two consequence relations '$\models_{\text{con}}$' and '$\models_{\text{min}}$'. First, we confine ourselves to the mcms of KB by using $\models_{\text{con}}$; then we minimize the valuations that we have by using $\models_{\text{min}}$. This process is a special case of what is called 'stratification' in [11].*

LEMMA 4.2. *Let KB be a finite knowledge base. For every mcm M of KB there is an $N \in \Omega(KB)$ s.t. $N \leq_k M$ and $\text{Inc}_N(KB) = \text{Inc}_M(KB)$.*

Proof. Suppose that M is an mcm of KB. Since KB is finite, there is an $N \in \Omega(KB)$ s.t. $N \leq_k M$. Suppose that $\text{Inc}_N(KB) \neq \text{Inc}_M(KB)$. Since both M and N are mcms of KB, there are $q_1, q_2 \in \mathcal{A}(KB)$ s.t. $q_1 \in \text{Inc}_N(KB) \setminus \text{Inc}_M(KB)$ and $q_2 \in \text{Inc}_M(KB) \setminus \text{Inc}_N(KB)$. Assume that $N(q_1) \in \mathcal{F}$. Then since $N(q_1) \in \mathcal{I}$, $N(\neg q_1) \in \mathcal{F}$ 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' defined for every $p \in \mathcal{A}(KB)$ as follows:

$$N'(p) = \begin{cases} t & \text{if } p = q_1, \\ N(p) & \text{otherwise.} \end{cases}$$

By an induction on the structure of a formula $\psi \in KB$ it is easy to verify (using Lemma 2.23) that $N'(\psi) \in \mathcal{F}$ whenever $N(\psi) \in \mathcal{F}$, and so N' is a model of KB. But $\text{Inc}_N(KB) = \text{Inc}_{N'}(KB) \cup \{q_1\}$; therefore $N' >_{\text{con}} N$, and so N cannot be an mcm of KB, and in particular $N \notin \Omega(KB)$ – a contradiction. □

THEOREM 4.3. *Let KB be a finite knowledge base and ψ an extended clause. Then $KB \models_{\text{con}} \psi$ iff $KB \models_\Omega \psi$.*

Proof. One direction is immediate. For the other, suppose that $KB \not\models_{\text{con}} \psi$. Then there is an mcm M of KB s.t. $M(\psi) \notin \mathcal{F}$. By Lemma 2.23, $\forall l \in \mathcal{L}(\psi) M(l) \notin \mathcal{F}$. By Lemma 4.2 $\exists N \in \Omega(KB)$ s.t. $N \leq_k M$. Since $\mathcal{F}$ is upward closed w.r.t. $\leq_k$, $\forall l \in \mathcal{L}(\psi) N(l) \notin \mathcal{F}$ as well. Therefore $KB \not\models_\Omega \psi$. □

COROLLARY 4.4. *Let KB be a finite set of normalized extended clauses in BL. Then*

(a) *l is a 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 a recoverable literal in KB iff for every $M \in \Omega(KB)$, $M(l) \in \mathcal{F}$, and there exists an $N \in \Omega(KB)$ s.t. $N(l) \in \mathcal{F} \setminus \mathcal{I}$.*

* This is, of course, a completely different notion from that of Definition 3.25.

(c) *l is an incomplete 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. Immediate from Definition 3.1 and Theorem 4.4 □

Another result related to minimal mcms is the following refinement of Theorem 3.18. The outcome is a characterization of the maximal support sets in terms of minimal mcms:

PROPOSITION 4.5. *Every maximal support set of a recoverable literal l in a finite knowledge base is associated with some minimal mcm $M \in \Omega$ s.t. $M(l) \notin \mathcal{I}$.*

Proof. Follows easily from Proposition 3.18 and Lemma 4.2. □

The next result, which is the analogue of Proposition 3.37 for minimal mcms, shows that Ω might as well be used in order to discover the absolutely recoverable formulae of KB:

COROLLARY 4.6. *Let ψ be a formula of a finite recoverable KB. If for every $M \in \Omega(KB)$, and for every $p \in \mathcal{A}(\psi)$, $M(p) \notin \mathcal{I}$, then ψ is absolutely recoverable.*

Proof. Immediate from Proposition 4.5 and Corollary 3.19. □

The results of this section show the advantage of using *bilattices* and not just lattices: While the partial order $\leq_t$ is used to determine the semantics of the classical connectives, $\leq_k$ can be used to considerably reduce the number of the models that should be taken into account.

5. Extensions to First-Order Logic

So far we have considered only propositional knowledge bases. However, it is possible to directly expand the present discussion to any first-order knowledge bases provided that there are no quantifiers within the clauses. Each extended clause that contains variables is considered as universally quantified. Consequently, a knowledge base containing a nongrounded 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.* Formally,

$$KB^U = \{\rho(\psi) \mid \psi \in KB,\ \rho : var(\psi) \to U\},$$

where ρ is a *ground substitution* of variables to the individuals of U. KB^U is called the *Herbrand expansion* of KB w.r.t. Herbrand universe U.

* In fact, the limitations imposed on BL guarantee that we stay, essentially, on a propositional level.

6. Examples and Applications

Let us summarize the major steps in the process of turning an inconsistent knowledge base into a consistent one. Given an inconsistent set S of assertions in BL, perform the following actions:

(1) Translate every formula $\psi \in S$ to an equivalent set $NEC(\psi)$ of normalized extended clauses (cf. Proposition 2.22). Let $KB = \bigcup\{NEC(\psi) \mid \psi \in S\}$.
(2) Compute $\text{con}(KB)$ [alternatively, compute $\Omega(KB)$]. From $\text{con}(KB)$ [$\Omega(KB)$] compute all the recoverable literals of KB (cf. Corollary 3.20) [(cf. Proposition 4.4(b))].
(3) Generate the support sets for the recoverable literals of KB as follows: For every $M \in \text{con}(KB)$ [$M \in \Omega(KB)$] and for every literal l such that $M(l) \notin \mathcal{I}$ compute the associated support set $SS_M(l)$ (cf. Proposition 3.18) [(cf. Proposition 4.5)]. If KB is stratified, the algorithm given in Section 3.3 might be useful for this purpose.
(4) Use the heuristics mentioned in Section 3.4 to choose the best support set among those that were produced in the previous step. This is the recovered knowledge base of the original inconsistent set S. Definition 3.8 and Theorem 3.12 guarantee that the recovered knowledge base is consistent and semantically corresponds to the data of S.

In the rest of this section we give some examples for illustrating the process described above. Then we consider an important type of problem in AI (that of model-based diagnosis) for which the methods developed in this paper are particularly useful.

6.1. NONMONOTONIC ASPECTS OF THE RECOVERING PROCESS

In this subsection we gather some benchmark problems that are given in [27] (under category A – default reasoning) for evaluating nonmonotonic formalisms. All the examples are considered in $\mathcal{B} = FOUR$ with $\mathcal{F} = \{t, \top\}$ and $\mathcal{I} = \{\top\}$. As it is shown below, our system manages to keep the results very close to those suggested in [27].

Consider the following block world description, $KB1$:

$$heavy(Block\_A)$$
$$heavy(Block\_B)$$
$$heavy(x) \to on\_the\_table(x)$$
$$\neg on\_the\_table(Block\_A)$$

Obviously, $KB1$ is inconsistent, and the problem is with the information about block A. In order to recover consistent data, we have to calculate the mcms of $KB1$, which are given in Table II.* The respective support sets, which correspond

* From now on we shall use X instead of $Block\_X$; $X = A, B$.

Table II. The (minimal) mcms of $KB1$.

mcm	$heavy(A)$	$heavy(B)$	$on\_the\_table(A)$	$on\_the\_table(B)$
$M1a$	t	t	$\top$	t
$M1b$	$\top$	t	f	t

to these mcms, are the following:

$$KB1a = \{heavy(A), heavy(B),\ heavy(B) \to on\_the\_table(B)\},$$
$$KB1b = \{\neg on\_the\_table(A), heavy(B),\ heavy(B) \to on\_the\_table(B)\}.$$

$KB1a$ supports the recoverable literals $heavy(A), heavy(B)$ and $on\_the\_table(B)$; $KB1b$ supports the recoverable literals $heavy(B), on\_the\_table(B)$, and $\neg on\_the\_table(A)$. Thus, the data about block B is absolutely recoverable. Particularly, in either support sets block B is on the table, as suggested in [27, Problem A1].

Suppose new data is introduced that is unrelated to existing information. For example, assume that $KB2 = KB1 \cup \{\neg red(B)\}$. It is easy to verify that the literals that were recoverable in $KB1$ still have the same status in $KB2$ (the new assertion, $\neg red(B)$, is also recoverable, of course. In fact, by Corollary 3.38, it is absolutely recoverable), and the same support sets can be constructed in a similar manner as before when adding to them the new data. Thus, $on\_the\_table(B)$ is still supported by every support set in $KB2$, and the recovered knowledge bases are $KB2a = KB1a \cup \{\neg red(B)\}$ and $KB2b = KB1b \cup \{\neg red(B)\}$ (cf. [27, Problem A2]).

Suppose that we are informed that every heavy block must be painted red. Let $KB3$ denote the knowledge base that contains all the information we have so far.

$heavy(A)$

$heavy(B)$

$heavy(x) \to on\_the\_table(x)$

$heavy(x) \to red(x)$

$\neg on\_the\_table(A)$

$\neg red(B)$

The *minimal* mcms of $KB3$ are given in Table III.* Their associated support sets are listed below.

$$\begin{aligned} KB3a = \{&heavy(A), heavy(B),\\ &heavy(A) \to red(A),\\ &heavy(B) \to on\_the\_table(B)\},\end{aligned}$$

* $KB3$ has 16 mcms. We omit the other 12, which are not $\leq_k$-minimal. As was shown in Section 4, by doing so we are not losing any meaningful data.

Table III. The minimal mcms of $KB3$.

mcm	$heavy(A)$	$heavy(B)$	$red(A)$	$red(B)$	$on\_the\_table(A)$	$on\_the\_table(B)$
$M3a$	t	t	t	$\top$	$\top$	t
$M3b$	t	$\top$	t	f	$\top$	$\bot$
$M3c$	$\top$	t	$\bot$	$\top$	f	t
$M3d$	$\top$	$\top$	$\bot$	f	f	$\bot$

$$\begin{aligned} KB3b &= \{heavy(A), \neg red(B), \\ &\quad\; heavy(A) \to red(A)\}, \\ KB3c &= \{\neg on\_the\_table(A), heavy(B), \\ &\quad\; heavy(B) \to on\_the\_table(B)\}, \\ KB3d &= \{\neg on\_the\_table(A), \neg red(B)\}. \end{aligned}$$

The ‘conservative’ (or ‘skeptical’) nature of the system is emphasized here: each suggested solution ignores the information it considers as contradictory, and leaves all the other data unchanged.

Note that $KB3a$ is the preferable support set according to many criteria that were mentioned through Section 3.4: It is the largest set, it supports more literals than any other support set, and it contains maximal information. To see the last claim, note that $|\mathcal{A}(KB3a)| = 4$, $|\mathcal{A}(KB3b)| = |\mathcal{A}(KB3c)| = 3$, $|\mathcal{A}(KB3d)| = 2$, $|\mathrm{con}(KB3a)| = 1$ (the only mcm is the reduction of Table III to the language of $KB3a$) and $|\mathrm{con}(KB3b)| = |\mathrm{con}(KB3c)| = |\mathrm{con}(KB3d)| = 1$ as well. Hence: $I_2(KB3a) = 4$, while $I_2(KB3b) = I_2(KB3c) = 3$, and $I_2(KB3d) = 2$.

So, it seems that the most reasonable set to recover $KB3$ is indeed $KB3a$. $KB3a$ implies that $on\_the\_table(B)$ and $red(A)$. These are also the conclusions in [27, Problem A3].

Note. The last example nicely demonstrates also the *practical* importance of having the truth value $\bot$. One can reach, in fact, the same conclusions using only the other three values (see Section 7.3 below). In that case, however, *nine* mcms should be considered instead of the four of Table III. The reason is that had we used only t, f, and $\top$, then every occurrence of $\bot$ in Table III should have been replaced by a classical truth value, and *both* of the two possibilities would have produced models that should have been taken into account.

For a last example of the block world, consider the following knowledge base, $KB4$.

$heavy(A)$

$heavy(B)$

Table IV. The (minimal) mcms of $KB4$.

mcm	$heavy(A)$	$heavy(B)$	$heavy(C)$	$on\_table(A)$	$on\_table(B)$	$on\_table(C)$
$M4a$	$\top$	t	t	f	t	t
$M4b$	t	$\top$	t	t	f	t
$M4c$	t	t	t	$\top$	t	t
$M4d$	t	t	t	t	$\top$	t

$heavy(C)$

$heavy(x) \rightarrow on\_the\_table(x)$

$\neg on\_the\_table(A) \vee \neg on\_the\_table(B)$

Note that the last assertion in $KB4$ states that there is an unknown exception in the information. The mcms of $KB4$ are given in Table IV.
Hence, $heavy(X)$ for $X = A, B, C$, and $on\_the\_table(C)$ are all recoverable, while $on\_the\_table(A)$ and $on\_the\_table(B)$ are incomplete. The support sets of $KB4$ are listed below.

$$\begin{aligned}
KB4a \;&=\; \{heavy(B), heavy(C),\; heavy(B) \rightarrow on_the_table(B),\\
&\quad heavy(C) \rightarrow on_the_table(C),\\
&\quad \neg on_the_table(A) \vee \neg on_the_table(B)\},\\
KB4b \;&=\; \{heavy(A), heavy(C),\; heavy(A) \rightarrow on_the_table(A),\\
&\quad heavy(C) \rightarrow on_the_table(C),\\
&\quad \neg on_the_table(A) \vee \neg on_the_table(B)\},\\
KB4c \;&=\; \{heavy(A), heavy(B), heavy(C),\\
&\quad heavy(B) \rightarrow on_the_table(B),\\
&\quad heavy(C) \rightarrow on_the_table(C)\},\\
KB4d \;&=\; \{heavy(A), heavy(B), heavy(C),\\
&\quad heavy(A) \rightarrow on_the_table(A),\\
&\quad heavy(C) \rightarrow on_the_table(C)\}.
\end{aligned}$$

Note that no matter which set the reasoner chooses as the recovered knowledge base, all of them preserve the intuitive conclusions of KB; that is, in every recovered knowledge base (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 [27].

Suppose now that the reasoner prioritizes the atomic formulae of $KB4$ in the following descending order: $heavy(A)$, $on\_the\_table(A)$, $heavy(B)$, $on\_the\_table(B)$, $heavy(C)$, and $on\_the\_table(C)$ (the reasoner might know, for example, that block A is the heaviest while block C is the lightest, or the information about block A is known to be more reliable). As a result, the possible recoverable knowledge bases are prioritized in the following descending order: $KB4d$, $KB4b$, $KB4c$, and $KB4a$,[⋆] therefore $KB4d$ is the preferred set in this case.

Because of lack of space we have not considered here all the benchmarks of [27]. We confined ourselves with most of the examples under category A (default reasoning). However, it might be interesting to check which of the other test criteria mentioned there are met in our system (most notable, the inheritance features and the autoepistemic characterizations) and to what degree the conclusions reached by our method resemble those of [27].

6.2. MODEL-BASED DIAGNOSIS

Suppose that one is given a description of some system (physical device, for example) together with an observation of its behavior. Suppose further that this observation conflicts with the way the system is meant to behave. The obvious goal is to identify the components of the system that behave abnormally, so that the discrepancy between the observed and the correct system behavior would be explained. In such cases it seems reasonable to assume that some minimal components are faulty. Therefore, the most consistent models and their corresponding support sets are good candidates to provide accurate diagnoses, especially since they minimize the set of components that are assumed to behave differently from expected (those that cause the conflicts).

EXAMPLE 6.1. Figure 5 depicts a binary full adder, examined extensively in the literature of diagnostic systems (see, e.g., [21, 35, 22, 34] and many others). It consists of five components: two and-gates A_1 and A_2, two xor-gates X_1 and X_2, and an or-gate O_1.
For the sake of the current example only we use the symbol $\oplus$ to denote the binary operation xor (instead of using this symbol for denoting $\leq_k$-meet operations of bilattices). The full adder's description is then given by the following system, FA:

- The expected behavior of the components of the system:

  $andGate(x) \wedge ok(x) \rightarrow (out(x) \leftrightarrow (in1(x) \wedge in2(x)))$,

  $xorGate(x) \wedge ok(x) \rightarrow (out(x) \leftrightarrow (in1(x) \oplus in2(x)))$,

  $orGate(x) \wedge ok(x) \rightarrow (out(x) \leftrightarrow (in1(x) \vee in2(x)))$.

- The gates of the system:

  $andGate(A_1), andGate(A_2),\ xorGate(X_1), xorGate(X_2),\ orGate(O_2)$.

[⋆] For example, the support set $KB4d$ is preferable to $KB4b$, since its atomic consequences are $heavy(A)$, $on\_the\_table(A)$, $heavy(B)$, $heavy(C)$, and $on\_the\_table(C)$. This is greater w.r.t. the reasoner prioritization than the consequences of $KB4b$, which are $heavy(A)$, $on\_the\_table(A)$, $on\_the\_table(B)$, $heavy(C)$, and $on\_the\_table(C)$.

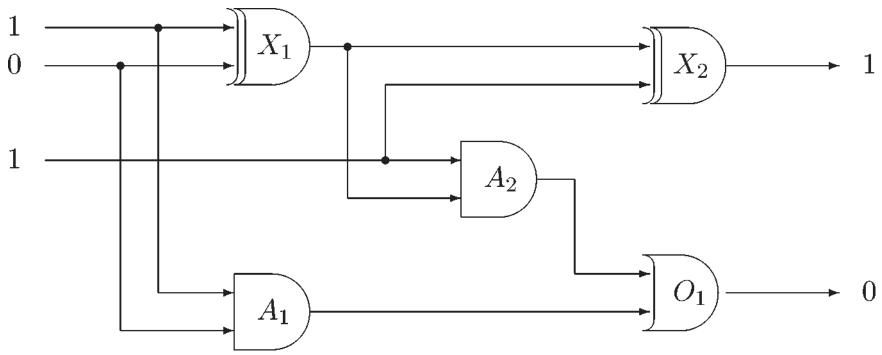

Figure 5. A full adder.

- Each gate is assumed to function correctly:

  $ok(A_1), ok(A_2),\ ok(X_1), ok(X_2), ok(O_1).$
- Integrity constraints:

  $andGate(x) \rightarrow (\neg orGate(x) \wedge \neg xorGate(x)),$

  $xorGate(x) \rightarrow (\neg andGate(x) \wedge \neg orGate(x)),$

  $orGate(x) \rightarrow (\neg andGate(x) \wedge \neg xorGate(x)).$
- Description of the circuits of the system:

  $in1(X_1) \leftrightarrow in1(A_1),\ in2(X_1) \leftrightarrow in2(A_1),$

  $out(X_1) \leftrightarrow in2(A_2),\ out(X_1) \leftrightarrow in1(X_2),$

  $in1(A_2) \leftrightarrow in2(X_2),\ out(A_2) \leftrightarrow in1(O_1),$

  $out(A_1) \leftrightarrow in2(O_1).$
- The set of observations:

  $in1(X_1), \neg in2(X_1), in1(A_2),\ out(X_2), \neg out(O_1).$

Notice that the observation indicates that the physical circuit is faulty. Both circuit outputs are wrong for the given inputs. Notice also that by Corollary 3.15, $ok(x)$ is recoverable for every component x; therefore, the mcms of FA and their corresponding recovered subsets would indicate which gates are faulty and which ones behave correctly.

The predicates $in1(x)$, $in2(x)$, and $out(x)$ are assigned values that correspond to binary values of the wires of the system. Therefore they should have only classical values (e.g., $in(G) = \top$ for a gate G is a meaningless value). Also, it seems natural to restrict the values of the predicates $andGate$, $orGate$, and $xorGate$ to be only t or f. This is because we know in advance what is the kind of each gate G in the system, and so the only open question about G (that might have inconsistent answers according to the actual behavior of the system) is whether it behaves as expected (i.e., whether $ok(G)$).

Let us denote by $Exact(KB)$ the predicates of KB that are assumed to have only classical values. We are interested only in those models in which every instance of a predicate of $Exact(KB)$ has a classical value. If D denotes the domain of discourse, the set of relevant models is the following:

$$mod(KB, Exact) = \{M \in mod(KB) \mid \forall p \in Exact\ \forall x_i \in D \\ M(p(x_1, \ldots, x_n)) \in \{t, f\}\},$$

where in our case, $Exact = \{in1, in2, out, andGate, orGate, xorGate\}$.

Notes.

(1) This restriction on the relevant models means that our basic consequence relation is now not $\models_{mod(KB)}$ but rather $\models_{mod(KB,Exact)}$, which is a particular case of the consequence relations defined at 2.17. The various concepts defined above, like that of an mcm, should be relativized accordingly. We note also that this approach of restricting some of the predicates to have only classical values is quite common (see, e.g., [41]). There are certain theories in which this meta-level is used also for adding *integrity constraints* for the specific problem. This can easily be done in our systems as well. See [5] for a more detailed study of these considerations in the case that $\mathcal{B} = FOUR$ and $\mathcal{I} = \{\top\}$.

(2) It is no longer true that $\nu_\top = \{p : \top \mid p \in \mathcal{A}(KB)\}$ must be an acceptable model of KB. In fact, there might be cases in which $mod(KB, Exact) = \emptyset$. However, although $mod(KB, Exact)$ is treated here as the set of the accepted valuations instead of $mod(KB)$, all the propositions that were proved above, except those of Section 3.4.1, remain valid under the obvious reformulations.

(3) A natural generalization to what we are doing here is to consider not only t, f, but any subset of truth values in B. That is, if $Val \subseteq B$, and $Pred \subseteq \mathcal{A}(KB)$, then $mod(KB, Pred, Val) = \{M \in mod(KB) \mid \forall p \in Pred\ \forall x \in D$ such that $M(p(x)) \in Val\}$. For instance, the set of all the consistent models of KB (w.r.t. an inconsistent set $\mathcal{I}$; see Definition 2.24(a)) may now be formulated as $mod(KB, \mathcal{A}(KB), B \setminus \mathcal{I})$.

Table V lists the models of $mod(FA, Exact)$. We have omitted from the table predicates (like $in1(X_1)$) that have the same (obvious) value in every model in $mod(FA, Exact)$, and predicates that have the same values as other predicates (like $in2(A_2)$, which is identical to $in1(X_2)$). The corresponding (minimal) mcms are given in Table VI.

The mcms among the elements of $mod(FA, Exact)$, and the support sets that, are associated with them, preserve what Reiter [35] calls *the principle of parsimony*; they represent the conjecture that some minimal set of components is faulty. For example, according to $M1$, which is one of the mcms of FA, the only component that is known to behave incorrectly is the xor-gate X_1. associated support set of $M1$ reflects this The indication:

Table V. The models in $mod(FA, Exact)$.

Model No.	$in1$ $X2$	$in1$ $O1$	$in2$ $O1$	ok $A1$	ok $A2$	ok $X1$	ok $X2$	ok $O1$
$M1$–$M16$	f	f	f	$t, \top$	$t, \top$	$\top$	$t, \top$	$t, \top$
$M17$–$M20$	f	t	f	$t, \top$	$\top$	$\top$	$t, \top$	$\top$
$M21$–$M24$	f	f	t	$\top$	$t, \top$	$\top$	$t, \top$	$\top$
$M25$–$M26$	f	t	t	$\top$	$\top$	$\top$	$t, \top$	$\top$
$M27$–$M34$	t	f	f	$t, \top$	$\top$	$t, \top$	$\top$	$t, \top$
$M35$–$M42$	t	t	f	$t, \top$	$t, \top$	$t, \top$	$\top$	$\top$
$M43$–$M44$	t	f	t	$\top$	$\top$	$t, \top$	$\top$	$\top$
$M45$–$M48$	t	t	t	$\top$	$t, \top$	$t, \top$	$\top$	$\top$

Table VI. The mcms of $mod(FA, Exact)$.

Model No.	$in1$ $X2$	$in1$ $O1$	$in2$ $O1$	ok $A1$	ok $A2$	ok $X1$	ok $X2$	ok $O1$
$M1$	f	f	f	t	t	$\top$	t	t
$M27$	t	f	f	t	$\top$	t	$\top$	t
$M35$	t	t	f	t	t	t	$\top$	$\top$

$$
\begin{aligned}
SS_{M1} = {} & FA \setminus \{ok(X_1), \\
& xorGate(X_1) \wedge ok(X_1) \\
& \to (out(X_1) \leftrightarrow (in1(X_1) \oplus in2(X_1)))\}.
\end{aligned}
$$

In particular, SS_{M_1} is a support set of $ok(x)$ for $x \in \{A_1, A_2, X_2, O_1\}$, and $SS_{M_1} \not\models_{\text{con}} ok(X_1)$. Similarly, the other two most consistent models $M27$ and $M35$, as well as their associated support sets, represent respective situations, in which gates $\{X_2, A_2\}$ and gates $\{X_2, O_1\}$ are faulties. These are the generally accepted diagnoses of this specific case (see, e.g., [35, Example 2.2], [22, Sections 15, 16], and [34, Examples 1, 4]).

According to the heuristics mentioned in Section 3.4, SS_{M1} is preferable to SS_{M27} and SS_{M35}, since it is bigger and supports more recoverable literals than the other two sets. In this particular case one has additional reasons to prefer SS_{M1}, since it claims that only a single component is faulty, and one normally expects components to fail independently of each other. This kind of diagnosis is known as a *single fault diagnosis*. We see, then, that in some cases the particular nature of the situation impose preference criteria – maybe other than those mentioned in Section 3.4 – so that a particular recovered set is judged as more likely to be correct than other solutions.

Next we show that the correspondence between the fault diagnoses and the inconsistent assignments of the mcms in the previous example is not accidental. For that we first present two basic notions from the literature on model-based diagnosis:

DEFINITION 6.2 [35]. A *system* is a triple $(Sd, Comps, Obs)$, where

(a) Sd, the *system description*, is a set of first-order sentences.
(b) $Comps$, the *system components*, is a finite set of constants.
(c) Obs, a set of *observations*, is a finite set of sentences.

DEFINITION 6.3 [35]. A *diagnosis* is a minimal set $\Delta \subseteq Comps$ s.t. $Sd \cup Obs \cup \{ok(c) \mid c \in Comps \setminus \Delta\} \cup \{\neg ok(c) \mid c \in \Delta\}$ is classically consistent.

In the example above we assumed that the devices normally behave as expected. We now formalize this assumption.

DEFINITION 6.4. A *correct behavior assumption* for a given set of components $\Delta \subseteq Comps$ is the set $CBA(\Delta) = \{ok(c) \mid c \in \Delta\}$.

NOTATION 6.5. For a given system $(Sd, Comps, Obs)$, and a set of components $\Delta \subseteq Comps$, denote $KB(\Delta) = Sd \cup Obs \cup CBA(\Delta)$. Whenever $\Delta = Comps$, we shall write just KB instead of $KB(Comps)$. Also, in the sequel we will continue to assume that the $KB(\Delta)$'s are sets of normalized extended clauses. Recall that by Proposition 2.22 this assumption can be taken without any loss of generality.

Here are some useful properties of diagnoses.

PROPOSITION 6.6. *Denote by* $\models_{cl}$ *the consequence relation of the first-order classical logic.*

(a) [35, *Proposition* 3.4] $\Delta \subseteq Comps$ *is a diagnosis for* $(Sd, Comps, Obs)$ *iff* Δ *is a minimal set such that* $KB(Comps \setminus \Delta)$ *is classically consistent.*
(b) [35, *Proposition* 3.3] *If* Δ *is a diagnosis for* $(Sd, Comps, Obs)$, *then* $KB(Comps \setminus \Delta) \models_{cl} \neg ok(c)$ *for each* $c \in \Delta$.

We present now a treatment of diagnostic systems in the multivalued framework of bilattices, where only a subset of the atomic formulae necessarily has classical values.

DEFINITION 6.7.

(a) An *extended diagnostic system* (e-system for short) is a pair $(KB, Exact)$, where $KB = Sd \cup Obs \cup CBA(Comps)$, and $Exact$ is a set of the predicates in the language of KB that are assumed to have only classical values.
(b) Let $(KB, Exact)$ be an e-system. An *exact model* of KB (w.r.t. $Exact$) is an element of $mod(KB, Exact) = \{M \in mod(KB) \mid \forall p \in Exact\ \forall x_i \in DM(p(x_1, \ldots, x_n)) \in \{t, f\}\}$
(c) A *most consistent exact model* of KB (mcem) is an mcm of $mod(KB, Exact)$.

THEOREM 6.8. *Let $(KB, Exact)$ be an e-system, and suppose the Herbrand base H of KB is $\{p(x_1, \ldots, x_n) \mid p \in Exact, x_i \in Comps\} \cup CBA(Comps)$.*[⋆] *An exact model M of KB is an mcem of KB iff* $\mathrm{Inc}_M(KB) = CBA(\Delta)$ *for some diagnosis Δ of KB.*

Proof. ($\Leftarrow$) Assume that M is an exact model of KB and that Δ is a diagnostic of KB s.t. $\mathrm{Inc}_M(KB) = CBA(\Delta)$. If M is not an mcem of KB, then there is an exact model M' s.t. $\mathrm{Inc}_{M'}(KB) \subset \mathrm{Inc}_M(KB) = CBA(\Delta)$; that is, there is a $c_0 \in \Delta$ s.t. $M'(ok(c_0)) \notin \mathcal{I}$. But (a) M' is a model of KB and $ok(c_0) \in KB$, and thus $M'(ok(c_0)) \in \mathcal{F}$; and (b) by Proposition 6.6(b), $KB(Comps \setminus \Delta) \models_{cl} \neg ok(c_0)$ and by Lemma 4.11 of [4][⋆⋆], $KB(Comps \setminus \Delta) \models_{\mathrm{con}} \neg ok(c_0)$. Since M is a (most) consistent model of $KB(Comps \setminus \Delta)$, then so is M'. Therefore $M'(\neg ok(c_0)) \in \mathcal{F}$. By (a) and (b), $M'(ok(c_0)) \in \mathcal{I}$ – a contradiction.

($\Rightarrow$) From the condition on Herbrand base of KB it follows that for every model M of KB, $\mathrm{Inc}_M(KB) \subseteq CBA(Comps)$. Suppose, then, that M is a most consistent model of KB and that $\mathrm{Inc}_M(KB) = CBA(\Delta)$ for some $\Delta \subseteq Comp$. By Proposition 6.6, in order to prove that Δ is a diagnosis for KB, it is sufficient to show that Δ is a minimal set such that $KB(Comps \setminus \Delta)$ is classically consistent. Suppose not. Then there is a proper subset $\Delta' \subset \Delta$ s.t. $KB(Comps \setminus \Delta')$ is classically consistent. In particular, $KB(Comps \setminus \Delta')$ is a consistent set in the sense of Definition 2.24(b), and so it has a consistent model N. Let M' be the following valuation:

$$M'(p) = \begin{cases} N(p) & \text{if } p \in \mathcal{A}(KB(Comps \setminus \Delta')), \\ \top & \text{otherwise.} \end{cases}$$

It is easy to verify (by using Lemma 2.23) that M' is a model of KB. Therefore, since $Exact(KB) \subset \mathcal{A}(KB(Comps \setminus \Delta'))$, M' is in $mod(KB, Exact)$. Moreover, $\mathrm{Inc}_{M'}(KB) = CBA(\Delta')$, and $\Delta' \subset \Delta$; thus $\mathrm{Inc}_{M'}(KB) = CBA(\Delta') \subset CBA(\Delta) = \mathrm{Inc}_M(KB)$. It follows that M cannot be an mcem of KB. □

COROLLARY 6.9. *Under the assumption of Theorem* 6.8, *if Δ is a diagnosis of KB, then there exists an mcem M of KB s.t.* $\mathrm{Inc}_M(KB) = CBA(\Delta)$.

Proof. Let Δ be a diagnosis for KB. If $\Delta = \{\}$, then $CBA(\Delta) = \{\}$, and by Proposition 6.6(a) KB is classically consistent. Hence every mcem M of KB is a consistent model (in the sense of Definition 2.24(a)), and so $\mathrm{Inc}_M(KB) = \{\}$ as well. If $\Delta \neq \{\}$, then KB is not (classically) consistent, since by Proposition 6.6(b) and by the monotonicity of $\models_{cl}$, $KB \models_{cl} \neg ok(c)$ for every $c \in \Delta$, and by reflexivity, $KB \models_{cl} ok(c)$. On the other hand, by Proposition 6.6(a), $KB(Comps \setminus \Delta)$ is classically consistent; therefore there is a model M of KB that assigns consistent truth values to every atomic formulae in $\mathcal{A}(KB(Comps \setminus \Delta))$, and

[⋆] Note that this requirement is met in Example 6.1.

[⋆⋆] According to that lemma, if KB is a classically consistent knowledge base, ψ is a clause that does not contain any pair of an atomic formula and its negation, and ψ follows classically from KB, then $KB \models_{\mathrm{con}} \psi$.

assigns $\top$ to $CBA(\Delta)$, i.e. $\text{Inc}_M(KB) = CBA(\Delta)$. This M is an mcem of KB by Theorem 6.8. □

COROLLARY 6.10. *Let* $(KB, Exact)$ *be an e-system as described in Theorem* 6.8. *Then* $ok(c)$ *is absolutely recoverable in* KB *iff* c *cannot be faulty in* KB.

Proof. Obviously follows from Proposition 3.37 and Theorem 6.8. □

Whenever the condition of Theorem 6.8 is met and KB is stratified, one can use the algorithm of Subsection 3.3 for finding diagnoses and constructing recovered knowledge bases of KB. Alternatively, one can use any other algorithm for finding diagnoses and then use the results for recovering KB. The process is as follows: First, such an algorithm is executed (this algorithm can be, for example, Reiter's DIAGNOSE [35]). Suppose that Δ is returned as a diagnosis. As in Section 5, given Herbrand universe U of KB, we denote $KB^{U \setminus \Delta} = \{\rho(\psi) \mid \psi \in KB, \rho : var(\psi) \to (U \setminus \Delta)\}$. By Theorem 6.8, $CBA(\Delta)$ corresponds to the inconsistent assignments of some mcm M, so by the proof of Theorem 3.11, $KB^{U \setminus \Delta}$ is a recoverable subset of KB.

7. A Comparison with Other Formalisms

In this section we compare the present approach of recovering consistent data with some other formalisms for dealing with inconsistency. Since there are many such formalisms, we consider only those with a close relationship to ours.

7.1. MAXIMAL CONSISTENT SUBSETS

A common method to 'recover' inconsistent knowledge bases is to search for its maximal consistent subsets. The main drawback of this method is that none of these subsets necessarily corresponds to the intended semantics of the original knowledge base. Consider, for instance, KB of Example 2.18 (also considered in Examples 3.2, 3.10, 3.26, and 3.40). Every maximal consistent subset of KB must contain either s or $\neg s$. Hence, either s or its complement, *but not both*, must be a consequence of every such a subset, but this consequence contradicts another assertion that was explicitly stated in the original knowledge base. For another example, consider $KB = \{p, \neg p \vee q, \neg q\}$. This time, there is no spoiled literal in KB, but still every maximal consistent subset of KB entails (both classically and w.r.t. $\models_{\text{con}}$) an assertion that contradicts any explicit data of KB. The support sets $\{p\}$ and $\{\neg q\}$ of this KB, as well as any support set of other knowledge bases, do not have such a drawback. The requirement that every support set be consistent *in* the original knowledge base assures that their conclusions not contradict any data entailed by the original knowledge base.* The last example also shows that

* In particular, support sets cannot contradict any *explicit* data of the knowledge base, as is the case with the knowledge bases and their maximal subsets considered above.

two-valued semantics is not sufficient even in cases where there are no spoiled literals.

7.2. ANNOTATED LOGICS; KIFER AND LOZINSKII'S TREATMENT

Annotated logics were introduced by Subrahmanian [38, 39] and further developed by him and others (see, e.g., [12, 23, 24, 40]). They also use multivalued algebraic structures in order to provide a semantics for rule-based systems with uncertainty. As we have already noted, the authors of [23] use annotated logic for similar purposes as ours. However, the present treatment of inconsistency in knowledge bases is free of some of the drawbacks of [23]. There, for example, just ordinary (semi)lattices were used, in which the partial order relation corresponds, intuitively, to $\leq_k$. Hence, no direct interpretation of the standard logical connectives (which correspond, in fact, to the $\leq_t$ partial order) was available to the authors. They were forced, therefore, to use a language in which the atomic formulae are of the form $p : b$ (where p is an atomic formula of the basic language, and b – a value from a semilattice). However, $\psi : b$ is meaningless for nonatomic ψ. Our treatment needs no such restriction. The use of bilattices enables assignments of truth values to any formula. Moreover, the present definitions follow the common method of logic systems, in which syntax and semantics are separated, while in the logic of [23] (and in annotated logics in general) semantic notions interfere with the syntax. In particular, the present treatment does not require any syntactic embedding of first-order formulae into the multivalued language (like the ones denoted Ξ_{epi} and Ξ_{ont} in [23]); the syntactic structure of each assertion remains the same.

7.3. PRIEST'S MINIMALLY CONSISTENT LPm

In [32, 33] Priest considers the logic LP – Kleene's strong three-valued logic with middle element ($\top$) designated.* According to Priest, the basic drawback of LP is that it invalidates disjunctive syllogism (i.e., $\psi, \neg\psi \vee \phi \not\vdash_{\mathrm{LP}} \phi$, where $\vdash_{\mathrm{LP}}$ denotes the consequence relation of LP).** Priest resolves this drawback by reducing the relevant models only to those that are *minimally inconsistent*: For a given propositional LP-valuation ν, Priest defines a corresponding set $\nu! = \{p \mid p \wedge \neg p \text{ is true under } \nu\}$ that 'measures' the inconsistency of ν. The minimal inconsistent models of a set of formulae Γ are those models ν such that if $\mu! \subset \nu!$, then μ is not a model of Γ. The consequence relation $\models_{\mathrm{LPm}}$ of the obtained logic, LPm, is then defined as follows: $\Gamma \models_{\mathrm{LPm}} \psi$ iff every minimally inconsistent model of Γ is a model of ψ.

* This logic is also known as RM_3 in the relevant literature ([1]) and J_3 in the literature about paraconsistency – see, e.g., Chapter IX of [14] as well as [29, 30, 6, 37].

** In a sense, disjunctive syllogism is the *only* classically valid inference that fails, since its addition to LP yields classical logic.

Obviously, Priest's main idea is very similar to ours, and the consequence relation $\models_{\mathrm{LPm}}$ is very close to $\models_{\mathrm{con}}$. The difference is that Priest is using the $\{\neg, \vee, \wedge\}$-closed subset $\{t, f, \top\}$ of the special bilattice $FOUR$, with the same $\mathcal{F}$, and with $\mathcal{I} = \{\top\}$. As we have seen in Section 6.1 (see the note there), the cost of using only this subset of $FOUR$ might be an exponential growth in the number of models that should be examined. This is due to the fact that every mcm M in $FOUR$ (with $\mathcal{I} = \{\top\}$) s.t. $M(p) = \perp$ for some p induces *two* LP-minimal models, which are identical to M, except that one assigns t to p, while the other assigns f to it.‡

It is not difficult to see that if we take $\mathcal{B} = FOUR$ and $\mathcal{I} = \{\top\}$, then $KB \models_{\mathrm{LPm}} \psi$ iff $KB, p_1 \vee \neg p_1, \ldots, p_n \vee \neg p_n \models_{\mathrm{con}} \psi$ where $\mathcal{A}(KB) = \{p_1, \ldots, p_n\}$. We conjecture that if, on the other hand, we take $\mathcal{I} = \{\top, \perp\}$ (and $\mathcal{B} = FOUR$), then $\models_{\mathrm{LPm}}$ is identical to $\models_{\mathrm{con}}$.

Our conclusion is that one can do with $FOUR$ everything one can do with LPm, if one so wishes (and usually more efficiently), but with $FOUR$ one can do other things as well. The exact relation between $FOUR$ and LPm deserves, however, further investigations.

8. Conclusion and Further Work

The consequence relation $\models_{\mathrm{con}}$ was considered in [23] as an epistemic entailment for annotated logics. In [2, 4] this relation was further examined and used in order to develop bilattice-based proof systems. In this paper we demonstrate another aspect of implementing $\models_{\mathrm{con}}$ together with (logical) bilattices, namely, a model-theoretic technique for extending the semantics (without changing the syntax) of classical first-order knowledge bases, in order to deal with contradictions in a nontrivial way. The outcome is a nonmonotonic mechanism for finding inconsistent parts of a given knowledge base, and a paraconsistent approach for recovering consistent data from it. This approach is shown to be efficient in several important cases, and particularly useful whenever conflicts are inherent parts of the situations, such as diagnostic problems.

One issue we have not dealt with so far is the choice of the particular bilattice to use. In all our examples above we have used the simplest bilattice $FOUR$. We suspect that for the language that we use here, $FOUR$ might indeed be sufficient, although we do not yet have a formal proof to this conjecture. Still, even if this conjecture is true, keeping the discussion on an abstract level (as we have done here) has obvious advantages:

(1) We do not intend our proposal to be an isolated method for dealing with inconsistent data. Rather, we believe that it should be a part of a general framework for dealing with knowledge bases. Now, for other aspects of the subject, other

‡ One should note, however, that the converse is not true: The existence of two LP-minimal models M_1 and M_2 s.t. $M_1(p) = t$, $M_2(p) = f$ and $M_1(q) = M_2(q)$ for every $q \neq p$ does not necessarily imply the existence of a corresponding mcm M in $FOUR$ s.t. $M(p) = \perp$. The clause $p \vee \neg p$ provides a counterexample.

bilattices might be useful. $DEFAULT$, for example, is usually taken to be suitable for default reasoning. Bilattices like $[0, 1] \odot [0, 1]$ (see [17] for the exact definition) may be used for statistical reasoning, and so on. The choice of the bifilter also depends, of course, on the application. For example, the use of the bifilter $\{\top, t\}$ of $DEFAULT$ means taking as 'true' only propositions that convey some truth. It is quite possible, however, that for certain applications we would like to accept also a default 'truth', represented (say) by $d\top$ or dt as standing for some extended notions of truth. We might use then $NINE$ rather than $DEFAULT$ and choose $NINE$'s second bifilter for our application ($DEFAULT$ itself does not have a bifilter containing dt or $d\top$).

(2) The fact that from the point of view of classical logic we can confine ourselves to the two-valued Boolean algebra does not mean that other Boolean algebras are useless in applications of classical logic. Similarly, the fact that in principle we can always use $FOUR$ (if this indeed is the case) does not exclude the potential usefulness of other logical bilattices (this point, of course, is not unrelated to the first one).

(3) The framework of bilattices opens the door for various nonclassical connectives (like Fitting's conflation and guard connectives [20], or the nonmonotonic implications of [4]). It is doubtful that with these extra connectives $FOUR$ will still be sufficient for defining $\models_{\text{con}}$.

The discussions in this section and in the preceding ones lead to several directions of research:

- Determine the exact role of $FOUR$ with respect to the consequence relation $\models_{\text{con}}$.
- Extend the approach to richer languages.
- Improve the algorithm of Section 3.3 and enlarge its range of applicability.
- Examine the applicability of the methods with more practical examples (especially those of [27]).

Acknowledgment

This research was supported by The Israel Science Foundation founded by The Israel Academy of Sciences and Humanities.

References

1. Anderson, A. R. and Belnap, N. D.: *Entailment*, Vol. 1, Princeton University Press, Princeton, NJ, 1975.
2. Arieli, O. and Avron, A.: Logical bilattices and inconsistent data, in *Proc. 9th IEEE Annual Symp. on Logic in Computer Science (LICS'94)*, IEEE Press, 1994, pp. 468–476.

3. Arieli, O. and Avron, A.: A bilattice-based approach to recover consistent data from inconsistent knowledge bases, in *Proc. of the 4th Bar-Ilan Symp. on Foundations of Artificial Intelligence (BISFAI'95)*, AAAI Press, 1995, pp. 14–23.
4. Arieli, O. and Avron, A.: Reasoning with logical bilattices, *J. Logic, Language, and Inform.* **5**(1) (1996), 25–63.
5. Arieli, O. and Avron, A.: Four-valued diagnoses for stratified knowledge bases, in D. Van-dalen and M. Benzem (eds.), *Proc. of the 1996 Ann. Conf. of the European Association for Computer Science Logic (CSL'96)*, Lecture Notes in Computer Science 1258, Springer-Verlag, 1997, pp. 1–17.
6. Avron, A.: On an implication connective of RM, *Notre Dame J. Formal Logic* **27** (1986), 201–209.
7. Benferhat, S., Cayrol, C., Dubois, D., Lang, J., and Prade, H.: Inconsistency management and prioritized syntax-based entailment, in *Proc. 13th Int. Joint Conf. on Artificial Intelligence (IJCAI'93)*, 1993, pp. 640–645.
8. Benferhat, S., Dubois, D., and Prade, H.: How to infer from inconsistent beliefs without revising?, in *Proc. 14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, 1995, pp. 1449–1455.
9. Belnap, N. D.: A useful four-valued logic, in G. Epstein and J. M. Dunn (eds.), *Modern Uses of Multiple-Valued Logic*, Reidel, Dordrecht, 1977, pp. 7–37.
10. Belnap, N. D.: How computers should think, in G. Ryle (ed.), *Contemporary Aspects of Philosophy*, Oriel Press, Stocksfield, England, 1977, pp. 30–56.
11. Brown, A. L., Jr. and Shoham, Y.: New results on semantical nonmonotonic reasoning, in *Proc. 2nd Int. Workshop on Non-Monotonic Reasoning*, Springer-Verlag, 1988, pp. 19–26.
12. da-Costa, N. C. A., Subrahmanian, V. S., Henschen, L. J., and Lu, J. J.: Automatic theorem proving in paraconsistent logics: Theory and implementation, in M. E. Stickel (ed.), *10th Int. Conf. on Automated Deduction*, 1990, pp. 72–86.
13. da-Costa, N. C. A.: On the theory of inconsistent formal systems, *Notre Dame J. Formal Logic* **15** (1974), 497–510.
14. Epstein, R. L.: *The Semantic Foundation of Logic, Vol. I*: *Propositional Logics*, Kluwer Acad. Publ., 1990.
15. Fitting, M.: Negation as refutation, in *Proc. 4th Annual Symp. on Logic in Computer Science (LICS'89)*, IEEE Press, 1989, pp. 63–70.
16. Fitting, M.: Kleene's logic, generalized, *J. Logic Comput.* **1** (1990), 797–810.
17. Fitting, M.: Bilattices in logic programming, in G. Epstein (ed.), *Proc 20th Int. Symp. on Multiple-Valued Logic* IEEE Press, 1990, pp. 238–246.
18. Fitting, M.: Bilattices and the semantics of logic programming, *J. Logic Programming* **11**(2) (1991), 91–116.
19. Fitting, M.: The family of stable models, *J. Logic Programming* **17** (1993), 197–225.
20. Fitting, M.: Kleene's three-valued logics and their children, *Fund. Inform.* **20** (1994), 113–131.
21. Genesereth, M. R.: The use of design description in authomated diagnosis, *J. Artificial Intelligence* **24** (1984), 411–436.
22. Ginsberg, M. L.: Multivalued logics: A uniform approach to reasoning in AI, *Comput. Intelligence* **4** (1988), 256–316.
23. Kifer, M. and Lozinskii, E. L.: A logic for reasoning with Inconsistency, *J. Automated Reasoning* **9**(2) (1992), 179–215.
24. Kifer, M. and Subrahmanian, V. S.: Theory of generalized annotated programming and it's applications, *J. Logic Programming* **12** (1992), 335–367.
25. Lehmann, D.: Plausibility logic, in *Proc. of 5th Ann. Conf. of the European Association for Computer Science Logic (CSL'91)*, Springer-Verlag, 1992, pp. 227–241.
26. Levesque, H. J.: Making believers out of computers, *J. Artificial Intelligence* **30** (1986), 81–108.

27. Lifschitz, V.: Benchmark problems for formal nonmonotonic reasoning, in *Proc. 2nd Int. Workshop on Non-Monotonic Reasoning*, Springler-Verlag, 1988, pp. 202–219.
28. Lozinskii, E. L.: Recovering contradictions: A plausible semantics for inconsistent systems, *J. Automated Reasoning* **12** (1994), 1–31.
29. d'Ottaviano, I. M. L. and da-Costa, N. C. A.: Sur un probleme de Jaskowski, *C.R. Hebdomadaires des Seances de I'Academie des Sciences, Serie A* **270** (1970), 1349–1353.
30. d'Ottaviano, I. M. L.: The completeness and compactness of a three-valued first order logic, *Rev. Colombiana Mat.* **19**(1–2) (1985), 31–42.
31. Pool, D.: A logical framework for default reasoning, *J. Artificial Intelligence* **36** (1988), 27–47.
32. Priest, G.: Reasoning about truth, *J. Artificial Intelligence* **39** (1989), 231–244.
33. Priest, G.: Minimally inconsistent LP, *Studia Logica* **50** (1991), 321–331.
34. Raiman, O.: The alibi principle, in W. Hamscher, L. Console, and J. de-Kleer (eds.), *Readings in Model-Based Diagnosis*, Morgan Kaufmann, 1992, pp. 66–70.
35. Reiter, R.: A theory of diagnosis from first principles, *J. Artificial Intelligence* **32**(1) (1987), 57–95.
36. Rescher, R. and Manor, R.: On inference from inconsistent premises, *J. Theory and Decision* **1** (1970), 179–219.
37. Rozoner, L. I.: On interpretation of inconsistent theories, *Inform. Sci.* **47** (1989), 243–266.
38. Subrahmanian, V. S.: Mechanical proof procedures for many valued lattice-based logic programming, *J. Non-Classical Logic* **7** (1990), 7–41.
39. Subrahmanian, V. S.: Paraconsistent disjunctive deductive databases, in *Proc. 20th Int. Symp. on Multiple-Valued Logic*, IEEE Press, 1990, pp. 339–345.
40. Subrahmanian, V. S.: Amalgamating knowledge bases, *ACM Transactions on Database Systems* **19**(2) (1994), 291–331.
41. Wagner, G.: *Vivid Logic: Knowledge Based Reasoning with Two Kinds of Negation,* Lecture Notes in AI 764, Springer-Verlag, 1994.