# Automatic Diagnoses for Properly Stratified Knowledge-Bases

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

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

## Abstract

*We present a mechanism for recovering consistent data from inconsistent set of assertions. For a common family of knowledge-bases we also provide an efficient algorithm for doing so automaticly. This method is nonmonotonic and paraconsistent. It is particularly useful for making diagnoses on faulty devices.*

**Subject areas:** *Knowledge-based systems, Non-standard logics for AI, Reasoning under uncertainty, Model-based diagnosis.*

## 1. Introduction

It is well-known that the classical calculus allows only a trivial reasoning in the presence of inconsistency. This property is particularly problematic when the system under consideration is aimed to deal with conflicts. This is the case, for instance, with diagnostic systems that are supposed to analize the discrepancy between the actual behavior of some device and the way it is meant to behave. A common approach of handling inconsistent information is to consider some consistent subsets that still contain meaningful data. The usual method of doing so is to consider the maximal consistent subsets of the "polluted" data. The main drawback of this method is that none of these subsets necessarily correspond to the intended semantics of the original information. Even in the simplest inconsistent knowledge-base $KB = \{p, \neg p\}$ every maximal consistent subset of $KB$ classically *contradicts an explicit data of* $KB$. In the case of diagnostic systems this means that a diagnosis based on a maximal consistent subset might not truthfully describe the malfunctioning part of the examined device.

We propose here a different approach to "salvage" consistent data without contradicting any assertion of the original information. For a common family of knowledge-bases we also provide an efficient algorithm for recovering this data. We then illustrate the ideas in a diagnostic system for checking faulty circuits. The underlying formalism is based on Belnap's four-valued logic [Be77a, Be77b], and it is shown to be nonmonotonic and paraconsistent [dC74].

Due to a lack of space most of the proofs are omitted; They will be given in the full version of the paper.

## 2. Preliminaries

We present a formalism that is based on Belnap's well-known four-valued logic. For a detailed discussion on this logic see, e.g., [Be77a, Be77b]. We denote by $t$ and $f$ the classical values. $\bot$ and $\top$ denote, respectively, lack of knowledge and "over"-knowledge (conflict). It is usual to consider these four values according to two partial orders: One, $\leq_t$, might intuitively be understood as reflecting differences in the "measure of truth" that every value represents. According to this order, $f$ is the minimal element, $t$ is the maximal one, and $\bot, \top$ are two intermediate values that are incomparable. $(\{t, f, \top, \bot\}, \leq_t)$ is a distributive lattice with an order reversing involution $\neg$, for which $\neg\top = \top$ and $\neg\bot = \bot$. We shall denote the meet and the join of this lattice by $\wedge$ and $\vee$, respectively. The other partial order, $\leq_k$, is understood (again, intuitively) as reflecting differences in the amount of *knowledge* or *information* that each truth value exhibits. Again, $(\{t, f, \top, \bot\}, \leq_k)$ is a lattice where $\bot$ is its minimal element, $\top$ – the maximal element, and $t, f$ are incomparable.

The language we treat here is the standard propositional one. Given a set $S$ of propositional formulae, we shall denote by $\mathcal{A}(S)$ the set of the atomic formulae that appear in the language $S$, and by $\mathcal{L}(S)$ the set of the literals that appear in some formula of $S$. The semantic notions are natural generalizations to the four-valued case of similar classical notions: A *valuation* $\nu$ is a function that assigns a truth value from $\{\top, \bot, t, f\}$ to each atomic formula. Any valuation is extended to complex formulae in the standard way.

We shall sometimes write $\psi : b \in \nu$ instead of $\nu(\psi) = b$. We will say that $\nu$ *satisfies* $\psi$, iff $\nu(\psi) \in \{t, \top\}$. $t$ and $\top$ are called *designated values*. A valuation that satisfies every formula in a given set of formulas, $S$, is said to be a *model* of $S$. The set of the models of $S$ will be denoted $mod(S)$. Note that unlike in the classical calculus, there are no tautologies here. In fact, excluded middle is not a valid rule in the four-valued case.

The formulae considered here are clauses, i.e.: disjuncts of literals. As the following lemma shows, representing the formulae in a clause form does not reduce the generality.

**Lemma 2.1** For every formula $\psi$ there is a finite set $S$ of clauses such that for every valuation $\nu$ and for every $\phi \in S$, $\nu(\psi) \in \{\top, t\}$ iff $\nu(\phi) \in \{\top, t\}$.

Here is another useful property of clauses that will be used several times in the sequel:

**Lemma 2.2** Let $\psi$ be a clause, $l_i$ $(i = 1 \ldots n)$ – its literals, and $\nu$ – a valuation on $\mathcal{A}(\psi)$. Then $\nu(\psi) \in \{t, \top\}$ iff there is an $1 \leq i \leq n$ s.t. $\nu(l_i) \in \{t, \top\}$.

**Definition 2.3** A *knowledge-base KB* is a pair $(S, Exact)$, where $S$ is a set of clauses, and $Exact$ is a set of atoms in $\mathcal{A}(S)$ that are assumed to have only classical values. $mod(KB) = mod(S, Exact)$ denotes the set of *exact models* of $S$, i.e.: the models of $S$ in which every element of $Exact$ is assigned a classical value. Formally: $mod(S, Exact) = \{M \in mod(S) \mid \forall p \in Exact\ M(p) \in \{t, f\}\}$.

**Definition 2.4** Let $M \in mod(S)$. Define:
$Inc_M(S) = \{p \in \mathcal{A}(S) \mid M(p) = \top\}$.

**Definition 2.5** Let $M, N$ be two exact models of a knowledge-base $KB = (S, Exact)$.
**a)** $M$ is *more consistent* than $N$ (notation: $M >_{con} N$), iff $Inc_M(S) \subset Inc_N(S)$.
**b)** $M$ is a *most consistent* exact model of $KB$ (mcem, for short), if there is no other exact model of $KB$ which is more consistent than $M$. The set of the most consistent exact models of $KB$ will be denoted mcem($KB$).
**c)** $M$ is *smaller* than $N$ with respect to $\leq_k$ $(M <_k N)$, if for any $p \in \mathcal{A}(S)$, $M(p) \leq_k N(p)$, and there is a $q \in \mathcal{A}(S)$ s.t $M(q) <_k N(q)$.
**d)** $M$ is a *k-minimal* model of a set $\mathcal{S}$, if there is no other element of $\mathcal{S}$ which is smaller w.r.t. $\leq_k$ than $M$.
**e)** The set of the $k$-minimal models of $mod(KB)$ will be denoted kmin($KB$); the set of the $k$-minimal models of mcem($KB$) (*minimal mcems*, for short) will be denoted $\Omega(KB)$.

**Definition 2.6** Let $KB = (S, Exact)$ be a knowledge-base, and $\psi$ – a formula.
**a)** $KB \models \psi$ if every exact model of $KB$ is a model of $\psi$.
**b)** $KB \models_{mcem} \psi$ if every mcem of $KB$ is a model of $\psi$.
**c)** $KB \models_{kmin} \psi$ if every $k$-minimal exact model of $KB$ is a model of $\psi$.
**d)** $KB \models_{\Omega} \psi$ if every $k$-minimal mcem of $KB$ is a model of $\psi$. [1]

**Example 2.7** Let $KB = (S, Exact)$ where $S = \{p,\ \neg p \vee \neg q,\ \neg q \vee r\}$ and $Exact = \emptyset$. The (exact) models of $KB$ are listed in Figure 1 below (An asterisk denotes some value in $\{\bot, f, t, \top\}$)

| Model No. | $p$ | $q$ | $r$ | Model No. | $p$ | $q$ | $r$ |
|---|---|---|---|---|---|---|---|
| $M_1$ | $t$ | $f$ | $\bot$ | $M_9$ | $\top$ | $\bot$ | $t$ |
| $M_2$ | $t$ | $f$ | $t$ | $M_{10}$ | $\top$ | $\bot$ | $\top$ |
| $M_3$ | $t$ | $f$ | $f$ | $M_{11}$ | $\top$ | $t$ | $t$ |
| $M_4$ | $t$ | $f$ | $\top$ | $M_{12}$ | $\top$ | $t$ | $\top$ |
| $M_5 \perp M_8$ | $t$ | $\top$ | $*$ | $M_{13} \perp M_{16}$ | $\top$ | $f$ | $*$ |
| | | | | $M_{17} \perp M_{20}$ | $\top$ | $\top$ | $*$ |

**Figure 1. The exact models of $KB$**

Here, kmin($KB$) $= \{M_1, M_9\}$, mcem($KB$) $= \{M_1, M_2, M_3\}$, and $\Omega(KB) = \{M_1\}$. Thus $KB \models_{mcem} \neg q$ and $KB \models_{\Omega} \neg q$, while $KB \not\models_{kmin} \neg q$ and $KB \not\models \neg q$.
When $Exact = \{q\}$ the (minimal) mcems of $KB$ remain the same, while $M_1$ becomes the single $k$-minimal exact model of $KB$, so now $KB \models_{kmin} \neg q$.

Several consequence relations similar to $\models_{mcem}$ are considered in the literature. Priest [Pr91] uses similar consequence relation for defining the logic LPm from the three-valued logic LP. Now, in [AA95] it is shown that for finite sets of assertions $\models_{mcem}$ and $\models_{\Omega}$ are the same. Therefore, when switching to four valued semantics and using only the $k$-minimal mcems, one might consider fewer models than in the case of LPm, since for every $k$-minimal mcem that assigns $\bot$ to an atomic formula $p$, there are *two* corresponding three-valued minimaly inconsistent LP models: One assigns $t$ to $p$, and the other assigns it $f$. Also, here one might impose further constraints on the relevant models since we (unlike [Pr91]) do not consider models that are not exact.

Kifer and Lozinskii [KL92] also consider a similar relation in the framework of annotated logics. Like Priest, they also consider the most consistent models among *all* the possible models. They do not restrict the attention to some relevant subset (as we do) by constraining them in the meta-level. See [AA94, AA96] for further discussion and a comparison between $\models_{mcem}$ and the consequence relation of [KL92].

[1]One can view the consequence relation $\models_{\Omega}$ as a composition of the relations $\models_{mcem}$ and $\models_{kmin}$. First we confine ourselves to the mcems of $KB$ by using $\models_{mcem}$, then we minimize the valuations that we have got by using $\models_{kmin}$.

**Definition 2.8** Let $S$ be a set of assertions. An atom $p \in \mathcal{A}(S)$ is called a *positive* (*negative*) *fact* of $S$ if $p \in S$ ($\neg p \in S$). $p$ is called *strictly* positive (negative) fact if it is a positive (negative) fact and $\neg p \notin S$ ($p \notin S$).

**Lemma 2.9** [AA95, Corollary 3.13] Let $KB = (S, Exact)$. If $p$ is a strictly positive fact of $S$ then $KB \models_{mcem} p$ and $KB \not\models_{mcem} \neg p$. Strictly negative facts have the dual property.

A basic property of the knowledge-bases that we consider here is that for every exact model there is an mcem which is at least as consistent. For finite knowledge-bases this is trivialy the case. The following proposition assures that this property holds in *every* propositional knowledge-base:

**Proposition 2.10** (Lin's Lemma, [Pr91]) Let $KB$ be a (possibly infinite) set of clauses. For every exact model $M$ of $KB$ there is an mcem $M'$ of $KB$ s.t. $M' \geq_{con} M$. [2]

## 3. Recovery of knowledge-bases

In this section we describe what we mean by saying "recovering an inconsistent knowledge-base". In particular we define and characterize the recovered parts of a knowledge-base. For that we first have to expand the notion of "consistency" to the four-valued case:

**Definition 3.1** Let $S$ be a set of clauses.
**a)** A model $M$ of $S$ is *consistent* if $Inc_M(S) = \emptyset$.
**b)** $S$ is *consistent* if it has a consistent model.
**c)** $KB = (S, Exact)$ is *consistent* if $S$ has a consistent exact model.

**Lemma 3.2** $S$ is consistent iff it is classically consistent.

**Definition 3.3** A subset $S' \subseteq S$ is *consistent in* $S$ w.r.t. $Exact$, if $S'$ is a consistent set, and it has a consistent exact model that is expandable to an (not necessarily consistent) exact model of $S$.

**Example 3.4** $S' = \{p\}$ is a consistent set, but it is *not* consistent in $S = \{p, \neg p\}$ w.r.t. any set $Exact$, since there is no consistent model of $S'$ that is expandable to a model of $S$. Similarly, $S' = \{p\}$ is consistent in $S = \{p,\ \neg p \vee q,\ \neg p \vee \neg q\}$ w.r.t. $Exact = \{p\}$, but it is not consistent in $S$ w.r.t. $Exact = \{q\}$, since there is no *consistent* exact model of $S'$ that is expandable to an *exact* model of $S$.

Now we can define what we mean by recovering an inconsistent knowledge-base:

[2]This lemma is proved in [Pr91] for the three-valued case, and under the implicit assumption that $Exact = \emptyset$. However, it is easy to prove this lemma in our case as well.

**Definition 3.5** A *recovered set* of a knowledge-base $(S, Exact)$ is a subset of $S$ that is consistent in $S$ w.r.t. $Exact$.

**Example 3.6** Consider the knowledge-base $KB = (S, \{r, u\})$ where $S = \{p,\ p \vee q,\ \neg p \vee \neg r,\ r \vee \neg u,\ r \vee \neg v,\ u \vee v\}$. $KB' = \{p,\ p \vee q,\ \neg p \vee \neg r,\ r \vee \neg u\}$ is a recovered set of $KB$, since $\{p : t,\ q : \bot,\ r : f,\ u : f\}$ is a consistent exact model of $KB'$ that is expandable to $M = \{p : t,\ q : \bot, r : f,\ u : f,\ v : \top\}$, which is an exact model of $KB$.

We now provide a method for practically recovering a knowledge-base. This method is strongly related to finding mcems for $KB$. (see Proposition 3.10 below).

**Definition 3.7** Let $M$ be an exact model of a knowledge-base $KB = (S, Exact)$. The set that is *associated with* $M$ is: $KB_M = \{\psi \in S \mid \mathcal{A}(\psi) \cap Inc_M(S) = \emptyset\}$.

**Example 3.8** In Example 3.6, $M$ is an exact model of $KB$, and $KB_M = KB'$.

**Proposition 3.9** Every set that is associated with some exact model $M$ of $KB$ is a recovered set of $KB$.

**Proposition 3.10** Every maximal recovered set of a knowledge-base $KB$ is associated with some mcem $M$ of $KB$.

## 4. Stratified knowledge-bases and their recovered sets

In general, computing mcems for a given knowledge-base and discovering its recovered sets might not be an easy task. Even in relatively simple cases, where e.g. $S$ is consistent and $Exact = \mathcal{A}(S)$, finding a recovered set for $(S, Exact)$ reduces to the problem of logical satisfaction, since in this case one has to provide a classical model for $S$. Therefore, we confine ourselves to a special (nevertheless common) family of knowledge-bases, for which we provide an efficient algorithm that computes recoverable sets.

**Definition 4.1** Let $S$ be a set of formulae. $S_\nu$ — the *dilution* of $S$ w.r.t. a partial valuation $\nu$ — is constructed from $S$ by the following transformations:

1. Deleting every $\psi \in KB$ s.t. $\nu(\psi) \in \{t, \top\}$,
2. Removing from what is left every occurrence of a literal $l$ such that $\nu(l) \in \{f, \bot\}$. [3]

[3]Note the similarity between the the dilution process and the Gelfond–Lifschitz transformation [GL88] used for providing semantics to logic programs with negations.

**Definition 4.2** A set of formulae $S$ is called *stratified*, if there is a set of "stratifications" $S_0 = S$, $S_1$, …, $S_n = \emptyset$, so that for every $0 \le i \le n \perp 1$ there is a $p \in \mathcal{A}(S_i)$ s.t.:
**a)** $p$ is either a positive fact or a negative fact of $S_i$.
**b)** $S_{i+1}$ is a dilution of $S_i$ w.r.t. the partial valuation $p\!:\!t$ if $p$ is a strictly positive fact of $S_i$, $p\!:\!f$ if $p$ is a strictly negative fact of $S_i$, and $p\!:\!\top$ if $p$ is both a positive and a negative fact of $S_i$.

A knowledge-base $KB = (S, Exact)$ is stratified iff $S$ is stratified. In all the examples given here (see especially the one of Section 5), as well as in most of the known puzzles of the literature, the involved knowledge-bases are stratified.

**Definition 4.3**
**a)** Let $S$ be a set of formulae and $Exact \subseteq \mathcal{A}(S)$. A stratification $S_0 \ldots S_n$ of $S$ is *proper* with respect to $Exact$ if there is no $p \in Exact$ and an index $1 \le i \le n$ s.t. $p$ is both a positive and a negative fact of $S_i$.
**b)** A knowledge-base $KB = (S, Exact)$ is called *properly stratified* iff there is a stratification of $S$ that is proper w.r.t. $Exact$.

**Example 4.4** Suppose that $KB = (S, \{r, u\})$ is the same knowledge-base of Examples 3.6 and 3.8. A possible stratification of $S$ is $S_0 = \{p,\ p\vee q,\ \neg p\vee\neg r,\ r\vee\neg u,\ r\vee\neg v,\ u\vee v\}$, $S_1 = \{\neg r,\ r\vee\neg u,\ r\vee\neg v,\ u\vee v\}$, $S_2 = \{\neg u,\ \neg v,\ u\vee v\}$, $S_3 = \{\neg v, v\}$, $S_4 = \emptyset$. This stratification is proper.

The algorithm given in Figure 2 checks whether a given knowledge-base $(S, Exact)$ is properly stratified. If so, it produces proper stratifications, and allows to construct recovered sets by providing corresponding (minimal) mcems of $(S, Exact)$ (see Theorem 4.7 below).

Every valuation $\nu$ produced by the algorithm of Figure 2 is determined by a sequence of the picked atomic formulae $\{p_0, p_1, \ldots, p_n\}$ of the calls to $RECOVER$. For shortening notations we shall just write $\nu$ when referring to an arbitrary valuation produced by the algorithm, instead of $\nu(p_0, p_1, \ldots, p_n)$.

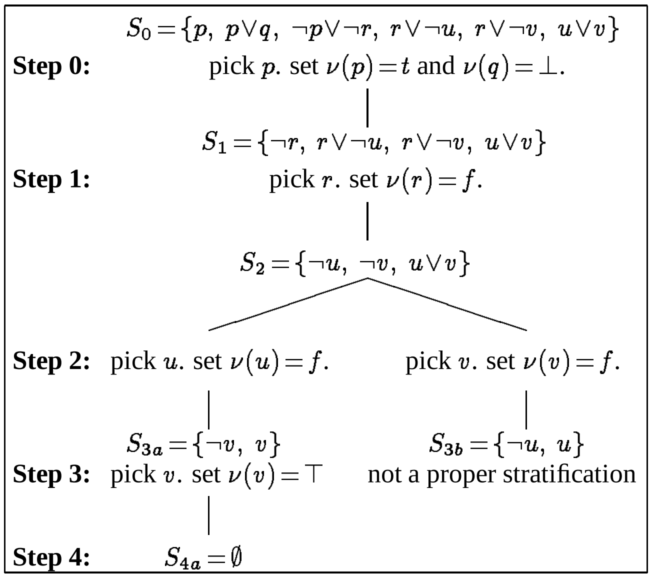
**Note:** It is possible to assign any other truth value to the atoms that are assigned $\perp$ (during the "filling" process of the algorithm), and still $\nu$ would be an exact model of $KB$. But in such cases, $\nu$ cannot be minimal w.r.t. $\le_k$. In particular, if the filling value is $\top$, then $\nu$ cannot be an mcem of $KB$ (see the proof of Theorem 4.7a). Also, it is possible to assign $f$ to the elements of $Exact$ that are assigned $t$ during the filling process without losing any of the properties discussed below.

**Example 4.5** In our canonical example (3.6, 3.8, and 4.4), where $S = \{p,\ p\vee q,\ \neg p\vee\neg r,\ r\vee\neg u,\ r\vee\neg v,\ u\vee v\}$ and

```
input: a knowledge-base $KB = (S, Exact)$
call $RECOVER(S, \emptyset, 0)$

procedure $RECOVER(S, \nu, i)$
/* $S$ – the $i$-th stratification level of $KB$,
   $\nu$ – the valuation constructed so far. */
{
   if ($S = \emptyset$) output $\nu$ and return;   /* $\nu \in \Omega(KB)$ */

   positives := $\{p \in \mathcal{A}(S) \mid p \in S\}$;
   negatives := $\{p \in \mathcal{A}(S) \mid \neg p \in S\}$;
   if (positives $= \emptyset \wedge$ negatives $= \emptyset$)
      halt;      /* don't continue; $KB$ is not stratified */

   if ($\exists p \in$ positives $\cap$ negatives $\cap$ $Exact$)
      return;   /* backtracking; not a proper stratification */

   while ($\exists p \in$ positives $\setminus$ negatives) {
      pick such a $p$;
      positives := positives $\setminus \{p\}$;
      $\nu := \nu \cup \{p : t\}$;
      $S_{i+1} := S_{p:t}$;   /* dilution */
      for every $q \neq p$ s.t. $q \in \mathcal{A}(S) \setminus \mathcal{A}(S_{i+1})$   /* filling */
         if ($q \notin Exact$) $\nu := \nu \cup \{q : \perp\}$ else $\nu := \nu \cup \{q : t\}$;
      $RECOVER(S_{i+1}, \nu, i+1)$;
   }

   while ($\exists p \in$ negatives $\setminus$ positives) {
      pick such a $p$;
      negatives := negatives $\setminus \{p\}$;
      $\nu := \nu \cup \{p : f\}$;
      $S_{i+1} := S_{p:f}$;   /* dilution */
      for every $q \neq p$ s.t. $q \in \mathcal{A}(S) \setminus \mathcal{A}(S_{i+1})$   /* filling */
         if ($q \notin Exact$) $\nu := \nu \cup \{q : \perp\}$ else $\nu := \nu \cup \{q : t\}$;
      $RECOVER(S_{i+1}, \nu, i+1)$;
   }

   while ($\exists p \in$ positives $\cap$ negatives, $p \notin Exact$) {
      pick such a $p$;
      positives := positives $\setminus \{p\}$;
      negatives := negatives $\setminus \{p\}$;
      $\nu := \nu \cup \{p : \top\}$;
      $S_{i+1} := S_{p:\top}$;   /* dilution */
      for every $q \neq p$ s.t. $q \in \mathcal{A}(S) \setminus \mathcal{A}(S_{i+1})$   /* filling */
         if ($q \notin Exact$) $\nu := \nu \cup \{q : \perp\}$ else $\nu := \nu \cup \{q : t\}$;
      $RECOVER(S_{i+1}, \nu, i+1)$;
   }
}
```

**Figure 2. An algorithm for recovering stratified knowledge-bases**

$Exact = \{r, u\}$, the algorithm produces a single (minimal) mcem of $KB$: $\nu(p) = t$, $\nu(q) = \bot$, $\nu(r) = f$, $\nu(u) = f$, and $\nu(v) = \top$. Figure 3 illustrates the processing of the algorithm in this case.



$S_0 = \{p,\ p\vee q,\ \neg p\vee\neg r,\ r\vee\neg u,\ r\vee\neg v,\ u\vee v\}$
Step 0:
pick $p$. set $\nu(p)=t$ and $\nu(q)=\bot$.
$S_1 = \{\neg r,\ r\vee\neg u,\ r\vee\neg v,\ u\vee v\}$
Step 1:
pick $r$. set $\nu(r)=f$.
$S_2 = \{\neg u,\ \neg v,\ u\vee v\}$
Step 2:
pick $u$. set $\nu(u)=f$.
pick $v$. set $\nu(v)=f$.
$S_{3a} = \{\neg v,\ v\}$
$S_{3b} = \{\neg u,\ u\}$
Step 3:
pick $v$. set $\nu(v)=\top$
not a proper stratification
Step 4:
$S_{4a} = \emptyset$

**Figure 3. Generation of minimal mcems and recovered sets for $KB$**

Note that the algorithm does not necessarily produce every mcem of the knowledge-base. In the last example such an mcem is, for instance, $\{p:\top, q:\bot, r:t, u:t, v:\bot\}$. However, as Theorems 4.7 and 4.8 show, if $KB$ is properly stratified the algorithm always finds at least one (minimal) mcem, and it provides a maximal recovered set for $KB$.

We now consider some properties of the algorithm.

**Proposition 4.6** Let $KB = (S, Exact)$ be a finite knowledge-base. If it is properly stratified then the algorithm of Figure 2 finds every proper stratification of $KB$ and outputs corresponding well-defined valuations for $\mathcal{A}(S)$. The algorithm halts without giving any valuation iff either $KB$ is not stratified, or every stratification of $S$ is not proper w.r.t. $Exact$.

It follows from Proposition 4.6 that the algorithm halts with a valuation for a finite $KB$ iff $KB$ is properly stratified. For the rest of this section suppose, then, that $KB$ is a finite knowledge-base that is properly stratified.

**Theorem 4.7** Let $\nu$ be a valuation produced by the algorithm for a knowledge-base $KB$. Then: (a) $\nu \in$ mcem($KB$), (b) $\nu \in$ kmin($KB$), and (c) $\nu \in \Omega(KB)$.

**Proof:** It is easy to see that every valuation $\nu$ produced by the algorithm is an exact model of $KB$. We show that $\nu$ is also most consistent among the exact models by an induction on the number of the recursive steps $(n)$ that are required for creating it. If $n=0$ then $S_1 = \emptyset$, so there is only the initial step in which $\nu$ might assign $\top$ only to a literal $l$ that is both a positive and a negative fact of $S$. Since in this case $l$ is assigned $\top$ by every model of $S$, $\nu$ must be most consistent. Suppose now that it takes $n \geq 1$ recursive steps to create $\nu$. Denote by $\nu_i$ the part of the valuation $\nu$ that is determined during step $i$. Then:

(1): $\quad Inc_\nu(S) = \bigcup_{0\leq i\leq n} Inc_{\nu_i}(S_i) = Inc_{\nu_0}(S) \cup Inc_{\nu'}(S_1)$

where $\nu' = \bigcup_{1\leq i\leq n} \nu_i$. Now, let $M$ be any mcem of $KB$, and suppose that $M_1$ is the reduction of $M$ to $S_1$.

(2): $\quad Inc_M(S) = \{p \in \mathcal{A}(S)\backslash\mathcal{A}(S_1) \mid M(p)=\top\} \cup \{p \in \mathcal{A}(S_1) \mid M(p)=\top\}$
$\qquad = \{p \in \mathcal{A}(S)\backslash\mathcal{A}(S_1) \mid M(p)=\top\} \cup Inc_{M_1}(S_1)$

By its definition, $\nu_0$ might assign $\top$ only to $l \in \mathcal{L}(S)$ s.t. $l, \bar{l} \in S$. Obviously, such an $l$ must be assigned $\top$ by every model of $S$, in particular $M(l) = \top$. Thus:

(3): $\quad Inc_{\nu_0}(S) \subseteq \{p \in \mathcal{A}(S)\backslash\mathcal{A}(S_1) \mid M(p)=\top\}$

• Suppose first that $M_1$ is an exact model of $S_1$. Since the creation of $\nu'$ requires only $n \perp 1$ steps, then by the induction hypothesis $\nu'$ is an mcem of $S_1$. In particular, either $Inc_{\nu'}(S_1)$ and $Inc_{M_1}(S_1)$ are incomparable w.r.t. the containment relation, or else:

(4): $\quad Inc_{\nu'}(S_1) \subseteq Inc_{M_1}(S_1)$

From (1) – (4), either $Inc_\nu(S)$ and $Inc_M(S)$ are incomparable, or $Inc_\nu(S) \subseteq Inc_M(S)$, hence $\nu$ is an mcem of $KB$.

• If $M_1$ is *not* an exact model of $S_1$ then $M_1$ is cannot be a model of $S_1$ either, since it is a reduction of an exact model $(M)$ of $S$. Thus there is a $\psi_1 \in S_1$ s.t. $M_1(\psi_1) \notin \{t, \top\}$. Since $M$ is a model of $S$, then by Lemma 2.2 there is a $\psi \in S$ and $l \in \mathcal{L}(\psi)$ s.t. $M(l) \in \{t, \top\}$, and $\{l\} \cup \mathcal{L}(\psi_1) \subseteq \mathcal{L}(\psi)$. Obviously, $l \in \mathcal{A}(S) \setminus \mathcal{A}(S_1)$. But then $\nu_0(l) \notin \{t, \top\}$ (otherwise $\psi$ is eliminated in the dilution of $S$, and so $\psi_1 \notin S_1$). Moreover, $\nu_0(\bar{l}) \in \{t, \top\}$, since if $\nu_0(\bar{l}) \notin \{t, \top\}$ then necessarily $\nu_0(l) = \bot$, and this happens only if there is a literal $l' \in \mathcal{L}(\psi)$ s.t. $\nu_0(l')$ is designated, and in this case again, $\psi$ is eliminated in the dilution of $S$, i.e. $\psi_1 \notin S_1$. Therefore, $\nu_0(l) \notin \{t, \top\}$ and $\nu_0(\bar{l}) \in \{t, \top\}$, so $\nu_0(l) = f$. $l$ is not assigned this value in the filling process, since again, this would imply that $\psi$ is eliminated in the dilution of $S$, and so $\psi_1 \notin S_1$. Thus, by the definition of $\nu_0$ and since $S$ is stratified, necessarily $\bar{l} \in S$ and $l \notin S$. Hence $KB \models \bar{l}$. But $M$ is an exact model of $KB$ and so $M(\bar{l}) \in \{t, \top\}$. Since we have shown that $M(l) \in \{t, \top\}$ as well, it follows that $M(l) = \top$ while $\nu(l) = f$. Therefore $Inc_M(S) \not\subseteq Inc_\nu(S)$. This completes the proof of part (a).

The proof of part (b) is by an induction on the number of recursive steps required to create $\nu$, and is similar to that of part (a). Part (c) now follows from (a) and (b).

**Theorem 4.8** Let $\nu$ be a valuation produced by the algorithm for $KB$. Then $KB_\nu$ is a maximal recovered set of $KB$.

**Example 4.9** Consider again Example 4.5 and Figure 3. $KB_\nu = \{p,\ p \vee q,\ \neg p \vee \neg r,\ r \vee \neg u\}$ is a maximal recovered set of $KB$.

**Proof of Theorem 4.8 (outlines):** By Proposition 3.9 and Theorem 4.7, $KB_\nu$ is a recovered set of $KB$. If it is not a maximal recovered set, then by Proposition 3.10 there is an mcem $M$ of $KB$ s.t. $KB_\nu \subset KB_M$. Since $\nu$ is also an mcem of $KB$ (Theorem 4.7 again), then there is a $p \in \mathcal{A}(S)$ s.t. $\nu(p) \neq \top$ while $M(p) = \top$. In particular, $p \notin Exact$, and by the construction of $\nu$, either $p$ or $\neg p$ is a strict fact of some stratification level $S_k$ of $S$. Therefore there is some $\phi \in S$ s.t. $p \in \mathcal{A}(\phi)$ and $\mathcal{A}(\phi) \cap Inc_\nu(S) = \emptyset$ (Otherwise $\phi$ is diluted in some stage before stage $k$). Thus $\phi \in KB_\nu$ while $\phi \notin KB_M$, and so $KB_\nu \not\subseteq KB_M$.

Finally, let's consider some complexity issues. As we have noted before, the problem of recovering arbitrary knowledge-base is at least NP-complete. Denote by $O(A^B)$ that it takes $O(A)$ running time to solve a certain problem when quering an oracle for solving problems with complexity $O(B)$. Then our algorithm requires $O(|S|^{|\mathcal{A}(S)|})$ running time to recover a knowledge-base $(S, Exact)$ that is properly stratified.[4] As the following proposition shows, the complexity of the algorithm is considerably reduced in cases that stratification implies proper stratification:

**Proposition 4.10** Whenever every stratification of $KB = (S, Exact)$ is proper, it takes $O(|S| \cdot |\mathcal{A}(S)|)$ running time to check whether $KB$ is stratified, and if so, this is also the time needed to recover it (i.e., to provide a maximal recovered set of $KB$).

Obvious cases in which the condition of the last proposition is met are when $Exact = \emptyset$, or if there is no $l \in Exact$ s.t. both $l \in \mathcal{L}(S)$ and $\overline{l} \in \mathcal{L}(S)$.

## 5. Model-based diagnosis

Suppose that one is given a description of some system (physical device, for example) together with an observation of its behaviour. 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 case it seems reasonable to assume that some minimal number of components are faulty. Therefore, the mcems and their corresponding recovered sets are good candidates for providing accurate diagnoses, especially since they minimize the set of components that are assumed to behave differently than expected.

For dealing with these kind of problems, it is convenient first to expand the discussion to first-order logic. It is possible to do so in a straightforward way, provided that there are no quantifiers within the clauses; each clause that contains variables is considered as universally quantified. Consequently, a knowledge-base containing non-grounded formula, $\psi$, will be viewed as representing the corresponding set of ground formulae formed by substituting each variable that appears in $\psi$ with every possible member of the Herbrand universe, $U$.[5] Formally: $KB^U = (S^U, Exact)$, where $S^U = \{\rho(\psi) \mid \psi \in S,\ \rho : var(\psi) \to U\}$, $\rho$ is a *ground substitution* from the variables of every $\psi \in KB$ to the individuals of $U$, and $Exact$ consists of predicates that every instansiation of which should be assigned classical values. The exact models are the elements of $mod(S^U, Exact) = \{M \in mod(S^U) \mid \forall p \in Exact\ \forall x_i \in U\ M(p(x_1, \ldots, x_n)) \in \{t, f\}\}$. $KB^U$ is called the *Herbrand expansion* of $KB$ w.r.t. Herbrand universe $U$.

**Example 5.1** Figure 4 depicts a binary full adder, examined extensively in the literature of diagnostic systems (See, e.g., [Ge84, Re87, Gi88, Ra92] 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 $O_1$.

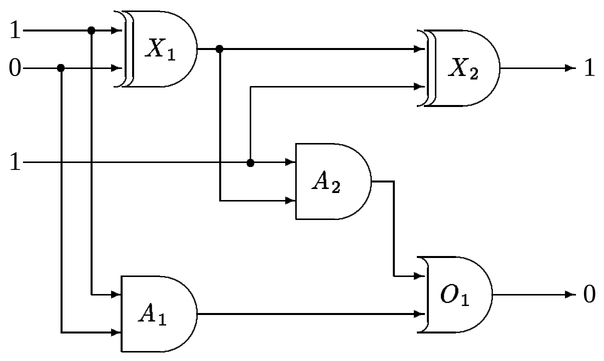


**Figure 4. A full adder**

[4] In our case at every stratification level the oracle chooses a fact that yields, eventually, to a proper stratification.

[5] In fact, this restriction guarantees that we stay, essentially, on a propositional level.

The full adder's description is given by system $FA$. It appears in Figure 5.

$$\begin{array}{l}
andGate(x) \wedge ok(x) \rightarrow (out(x) \leftrightarrow (in1(x) \wedge in2(x))), \\
xorGate(x) \wedge ok(x) \rightarrow (out(x) \leftrightarrow (in1(x) \oplus in2(x))), \\
orGate(x) \wedge ok(x) \rightarrow (out(x) \leftrightarrow (in1(x) \vee in2(x))), \\
andGate(x) \rightarrow (\neg orGate(x) \wedge \neg xorGate(x)), \\
xorGate(x) \rightarrow (\neg andGate(x) \wedge \neg orGate(x)), \\
orGate(x) \rightarrow (\neg andGate(x) \wedge \neg xorGate(x)), \\
in1(X_1) \leftrightarrow in1(A_1),\ in2(X_1) \leftrightarrow in2(A_1), \\
in1(A_2) \leftrightarrow in2(X_2), \\
out(X_1) \leftrightarrow in2(A_2),\ out(X_1) \leftrightarrow in1(X_2), \\
out(A_1) \leftrightarrow in2(O_1),\ out(A_2) \leftrightarrow in1(O_1), \\
andGate(A_1),\ andGate(A_2), \\
xorGate(X_1),\ xorGate(X_2),\ orGate(O_2), \\
ok(A_1),\ ok(A_2),\ ok(X_1),\ ok(X_2),\ ok(O_1), \\
in1(X_1),\ \neg in2(X_1),\ in1(A_2),\ out(X_2),\ \neg out(O_1)
\end{array}$$

**Figure 5. The system $FA$**

Notice that the observation indicates that the physical circuit is faulty; both circuit outputs are wrong for the given inputs.

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 classical as well. 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 $G$ is whether it behaves as expected (i.e., whether $ok(G)$). So, the actual knowledge-base for the full adder is $(FA, Exact)$, where $Exact = \{in1, in2, out, andGate, orGate, xorGate\}$.

The table of Figure 6 lists the elements of mcem$(FA, Exact)$. We have omitted from the table predicates (like $in1(X_1)$) that have the same (obvious) value in every exact model of $(FA, Exact)$, and predicates that have the same values as some other predicates (like $in2(A_2)$, which is identical to $in1(X_2)$).

The mcems of $(FA, Exact)$, and the recovered sets that are associated with them preserve what Reiter [Re87] calls *the principle of parsimony*; they represent the conjecture that some minimal set of components are faulty. For example, according to $M1$ the only component that behaves incorrectly is the xor gate $X_1$. The recovered set that is associated with $M1$ reflects this indication: $FA_{M1} =$

| | $in1$ $X2$ | $in1$ $O1$ | $in2$ $O1$ | $ok$ $A1$ | $ok$ $A2$ | $ok$ $X1$ | $ok$ $X2$ | $ok$ $O1$ |
|---|---|---|---|---|---|---|---|---|
| $M1$ | $f$ | $f$ | $f$ | $t$ | $t$ | $\top$ | $t$ | $t$ |
| $M2$ | $t$ | $f$ | $f$ | $t$ | $\top$ | $t$ | $\top$ | $t$ |
| $M3$ | $t$ | $t$ | $f$ | $t$ | $t$ | $t$ | $\top$ | $\top$ |

**Figure 6. The mcems of $(FA, Exact)$**

$FA \setminus \{ok(X_1),\ xorGate(X_1) \wedge ok(X_1) \rightarrow (out(X_1) \leftrightarrow (in1(X_1) \oplus in2(X_1)))\}$.

In particular, $FA_{M_1}$ entails (w.r.t. both $\models$ and $\models_{mcem}$) $ok(x)$ for $x \in \{A_1, A_2, X_2, O_1\}$, but it does *not* entail $ok(X_1)$. Similarly, the other two mcems $M2$ and $M3$, as well as their associated 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. [Re87, Example 2.2], [Gi88, Sections 15,16], and [Ra92, Examples 1,4]).

One might treat $FA_{M1}$ as the preferred recovered set, since it is the only set that entails 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*.

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

**Definition 5.2** [Re87] A *system* is a triple $(Sd, Comps, Obs)$, where:
a) $Sd$ (*system description*) is a set of first order sentences.
b) $Comps$ (*system components*) is a finite set of constants.
c) $Obs$ (*observations*) is a finite set of sentences.

**Definition 5.3** [Re87] 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 next formalize this assumption:

**Definition 5.4** A *correct behaviour assumption* for a given set of components $\Delta \subseteq Comps$ is the set $CBA(\Delta) = \{ok(c) \mid c \in \Delta\}$.

**Notation 5.5** For a given system $(Sd, Comps, Obs)$, and a set of components $\Delta \subseteq Comps$, denote $S(\Delta) = Sd \cup Obs \cup CBA(\Delta)$. Whenever $\Delta = Comps$ we shall write just $S$ instead of $S(Comps)$. Also, in the sequel we shall continue to assume that $S(\Delta)$ is a set of clauses. Recall that this assumption can be taken without any loss of generality.

**Proposition 5.6** Denote by $\models_{cl}$ the consequence relation of the first order classical logic.
**a)** [Re87, Proposition 3.4] $\Delta \subseteq \mathit{Comps}$ is a diagnosis for $(\mathit{Sd}, \mathit{Comps}, \mathit{Obs})$ iff $\Delta$ is a minimal set such that $S(\mathit{Comps} \setminus \Delta)$ is classically consistent.
**b)** [Re87, Proposition 3.3] If $\Delta$ is a diagnosis for $(\mathit{Sd}, \mathit{Comps}, \mathit{Obs})$ then $S(\mathit{Comps} \setminus \Delta) \models_{cl} \neg ok(c)$ for each $c \in \Delta$.

When considering diagnostic systems in the classical, two-valued logics, any inconsistency in the data causes trivial reasoning. This is, of course, not the case in the present treatment, where we allow truth value that exhibits inconsistencies, and only a subset of the atomic formulae necessarily have classical values. In terms of our discussion, then, a diagnostic system is a knowledge-base $KB = (S, \mathit{Exact})$, where $S = \mathit{Sd} \cup \mathit{Obs} \cup \mathit{CBA}(\mathit{Comps})$.

**Theorem 5.7** Let $(S, \mathit{Exact})$ be a knowledge-base of a diagnostic system, and suppose that the Herbrand base of $S$ is $\{p(x_1, \ldots, x_n) \mid p \in \mathit{Exact}, x_i \in \mathit{Comps}\} \cup \mathit{CBA}(\mathit{Comps})$.[6] An exact model $M$ of $(S, \mathit{Exact})$ is an mcem of $(S, \mathit{Exact})$ iff $\mathit{Inc}_M(S) = \mathit{CBA}(\Delta)$ for some diagnosis $\Delta$ of $S$.

**Outline of proof:** $(\Leftarrow)$ Otherwise there is an exact model $M'$ s.t. $\mathit{Inc}_{M'}(S) \subset \mathit{Inc}_M(S) = \mathit{CBA}(\Delta)$, i.e.: there is a $c_0 \in \Delta$ s.t. $M'(ok(c_0)) \neq \top$. But: (a) $ok(c_0) \in S$ thus $M'(ok(c_0)) \in \{t, \top\}$, and: (b) $S(\mathit{Comps} \setminus \Delta) \models_{cl} \neg ok(c_0)$ and by Lemma 4.11 of [AA96] [7], $S(\mathit{Comps} \setminus \Delta) \models_{mcem} \neg ok(c_0)$, therefore $M'(\neg ok(c_0)) \in \{t, \top\}$. By (a) and (b), $M'(ok(c_0)) = \top$ – a contradiction.
$(\Rightarrow)$ By Proposition 5.6, in order to prove that $\Delta$ is a diagnosis for $S$ it is sufficient to show that $\Delta$ is a minimal set such that $S(\mathit{Comps} \setminus \Delta)$ is classically consistent. This follows from the condition on Herbrand base of $S$ that assures that for every exact model $M$ of $(S, \mathit{Exact})$, $\mathit{Inc}_M(S) \subseteq \mathit{CBA}(\mathit{Comps})$.

**Corollary 5.8** Under the assumption of Theorem 5.7, if $\Delta$ is a diagnosis of $S$ then there exists an mcem $M$ of $(S, \mathit{Exact})$ s.t. $\mathit{Inc}_M(S) = \mathit{CBA}(\Delta)$.

It follows that whenever the requirement of Theorem 5.7 is met and $(S, \mathit{Exact})$ is properly stratified, one can use the algorithm of Section 4 for finding diagnoses and constructing maximal recovered knowledge-bases of the faulty system. This is the case, e.g., in Example 5.1.

## 6. Conclusion

We have proposed a mechanism for recovering consistent data from inconsistent set of assertions. Our approach considers the contradictory data as useless, and regards all the remaining information unaffected. This kind of approach is nonmonotonic and paraconsistent in nature. For a common family of knowledge-bases we have also provided an efficient algorithm for an automatic recovery. Our method is particularly useful for diagnostics systems, where it might be used for supplying a description of the well-behaved parts of a faulty device.

[6]Note that this requirement is met in Example 5.1.

[7]According to that lemma, if $S$ is a classically consistent set of assertions, $\psi$ is a clause that does not contain any pair of an atomic formula and its negation, and $\psi$ follows classically from $S$, then $S \models_{mcem} \psi$.

## References

[AA94] O.Arieli, A.Avron. *Logical bilattices and inconsistent data.* Proc. 9th IEEE Ann. Symp. on Logic in Computer Science (LICS'94). IEEE Press, pp.468-476; 1994.

[AA95] O.Arieli, A.Avron. *A bilattice-based approach to recover consistent data from inconsistent knowledge-bases.* Proc. 4th Bar-Ilan Symp. on Foundations of Artificial Intelligence (BISFAI'95), pp.231-240; 1995.

[AA96] O.Arieli, A.Avron. *Reasoning with logical bilattices.* J. Logic, Language, and Information, Vol.5, pp.25-63; 1996.

[Be77a] N.D.Belnap. *A useful four-valued logic.* Modern Uses of Multiple-Valued Logic, Reidel Pub., pp.7-37; 1977.

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

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

[Ge84] M.R.Genesereth. *The use of design description in authomated diagnosis.* J. Artificial Intelligence, Vol.24, pp.411-436; 1984.

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

[GL88] M.Gelfond, V.Lifschitz. *The stable model semantics for logic programming.* Proc. 5th logic programming symposium, MIT Press, Cambridge, MA, pp.1070-1080; 1988.

[KL92] M.Kifer, E.L.Lozinskii. *A logic for reasoning with inconsistency.* J. Automated reasoning. Vol.9(2), pp.179-215; 1992.

[Pr91] G.Priest. *Minimally inconsistent LP.* Studia Logica, Vol.50, pp.321-331; 1991.

[Ra92] O.Raiman. *The alibi principle.* Readings in Model-Based Diagnosis, (W.Hamscher, L.console, J.de-Kleer – Eds.), Morgan Kaufmann Pub., pp.66-70; 1992.

[Re87] R.Reiter. *A theory of diagnosis from first principles.* J. Artificial Intelligence, Vol.32(1), pp.57-95; 1987.