International Baltic Conference on Databases and Information Systems
H.-M. Haav et al. (Editors), Tallinn, Estonia, 77-90, 2002 (Revised)

PARACONSISTENT KNOWLEDGE BASES AND MANY-VALUED LOGIC

Jørgen Villadsen
Informatics and Mathematical Modelling, Technical University of Denmark
Richard Petersens Plads, Building 321, DK-2800 Kongens Lyngby, Denmark

Now at Computer Science, Roskilde University, Denmark jv@ruc.dk

Abstract Classical logic predicts that everything (thus nothing useful at all) follows from inconsistency. A paraconsistent logic is a logic where an inconsistency does not lead to such an explosion, and since in practice consistency is difficult to achieve there are many potential applications of paraconsistent logics in knowledge bases. We present a paraconsistent many-valued logic with a simple and new semantics for the logical operators. In particular we compare our approach with work based on bilattices. The adequacy of the logic is examined by a case study in the domain of medicine.

1. Introduction

Classical logic predicts that everything (thus nothing useful at all) follows from inconsistency. A paraconsistent logic is a logic where an inconsistency does not lead to such an explosion [7]. In practice consistency is difficult to achieve for substantial knowledge bases, so paraconsistent logics have many applications.

Consider the following inferences in classical logic:

$$\text{fly}(\text{Tweety}) \land \neg\text{fly}(\text{Tweety}) \;\rightarrow\; \text{fly}(\text{Tweety})$$
$$\text{fly}(\text{Tweety}) \land \neg\text{fly}(\text{Tweety}) \;\rightarrow\; \neg\text{fly}(\text{Tweety})$$
$$\text{fly}(\text{Tweety}) \land \neg\text{fly}(\text{Tweety}) \;\rightarrow\; \text{flat}(\text{Earth})$$
$$\text{fly}(\text{Tweety}) \land \neg\text{fly}(\text{Tweety}) \;\rightarrow\; \neg\text{flat}(\text{Earth})$$

We use the following binding priorities of the operators: negation $\neg$ (highest), conjunction $\land$, disjunction $\lor$ and implication $\rightarrow$ (lowest). This is a standard approach to reduce the number of parentheses in formulas.

While the first two inferences seem reasonable, albeit trivial, the last two inferences show the problem with inconsistent knowledge bases and classical logic, since anything can be substituted for the atomic formula $\text{flat}(\text{Earth})$.

The reason for the explosion in classical logic is that it is based on a compositional semantics with only two truth values (i.e. a truth-functional semantics):

> Atomic formulas like fly(Tweety) can be either True or False, and negation maps True to False and vice versa (this is the only possibility if we do not want negation to be the identity function or one of the constant functions yielding always True or False). Hence either fly(Tweety) or ¬fly(Tweety) is False and the left hand side of the implication is always False (clearly a conjunction cannot be True if one part is False).
>
> Consider the right hand side of the implication in the first two inferences: one of them is True and the other is False (since the second is the negation of the first). Because we take the first two inferences to be reasonable (after all, if something is in the knowledge base, it should follow) we can now conclude that no matter what the value of the right hand side of the implication is, if the left hand side of the implication is False then the result must be True.
>
> But then, assuming a compositional semantics with only two truth values, the result must be True for the last two inferences as well, since flat(Earth) and ¬flat(Earth) must be True or False — *and it does not matter which of these two values they take!*

Therefore, if we require a compositional semantics for the logic we cannot have a reasonable paraconsistent logic based on just two truth values.

Of course, there are numerous alternatives to classical logic, and one way to proceed could be a modal logic with a compositional possible-world semantics [12], but in the present paper we investigate the somewhat simpler setup of many-valued logics [11] which differ from classical logic by the fundamental fact that they do not restrict the number of truth values to only two.

An even more radical departure from classical logic would be to give up the monotonicity rules that we used implicitly in the argument above [9]. The basic monotonicity rule says that if $A \rightarrow C$ for formulas A and C then $A \wedge B \rightarrow C$ for an arbitrary formula B.

Inconsistencies can be harder to spot than in the inferences above. Consider the following knowledge base T:

$$\begin{array}{rcl} \text{penguin}(\text{Tweety}) & \wedge & (\,\forall x.\,\text{bird}(x) \rightarrow \text{fly}(x)\,) \\ & \wedge & (\,\forall y.\,\text{penguin}(y) \rightarrow \neg\text{fly}(y)\,) \\ & \wedge & (\,\forall z.\,\text{penguin}(z) \rightarrow \text{bird}(z)\,) \end{array}$$

Again we have $\text{T} \rightarrow \varphi$ for any formula φ, but it requires a number of inferences to make the inconsistency stand out.

While it is often relatively easy to find and remove an inconsistency in a small knowledge base like T, it might be difficult for larger theories of formalized common knowledge. For example, the upcoming release 1.0 of the Cyc open source knowledge base (see the homepage OpenCyc.org) will consist of 6,000 concepts (an upper ontology for all of human consensus reality) and 60,000 assertions about the 6,000 concepts, interrelating them, constraining them, in effect (partially) defining them.

2. Overview

The purpose of the present paper is to investigate ways to enlarge the set of truth values and obtain a paraconsistent logic (to restrict the set of truth values of classical logic is meaningless, since at least two truth values are needed to distinguish truth from falsehood).

In section 3 we propose three requirements that we think a many-valued logic must satisfy in order to be applicable to knowledge bases.

In section 4 we describe the semantics for the well-known four-valued logic of Belnap [6] which is based on the mathematical structure now referred to as a bilattice [10, 2], and in section 5 we explain why this logic does not satisfy the requirements of section 3 with the usual implication operators.

In section 6 we describe a revised semantics for the Belnap logic which was suggested and applied to the semantics of propositional attitudes like knowledge and belief by Villadsen [14] and in section 7 we show that our logic satisfies the requirements of section 3 with the revised implication operators.

Finally, section 8 includes a case study in the domain of medicine and section 9 concludes the paper.

3. Requirements

It will be sufficient here to consider propositional logic only because the main results generalize to first order logic and higher order logic, cf. section 9. In the following A, B and C are arbitrary formulas.

Inspired by the examples in section 1 we want to block the inference schemas:

$$B \wedge \neg B \;\rightarrow\; C \tag{1}$$

$$(A \rightarrow B) \wedge A \wedge \neg B \;\rightarrow\; C \tag{2}$$

$$(A \rightarrow B) \wedge \neg(A \rightarrow B) \;\rightarrow\; C \tag{3}$$

Note that this means that the implication operator $\rightarrow$ cannot be the usual one from classical logic. Also note that (3) is not a special case of (1) when we want to block the inference schemas, since (1) can obviously be blocked without blocking (3) — on the other hand, if (3) is blocked then so is (1), but we want to stress the importance of (1) traditionally found in paraconsistent logics by keeping it as a separate inference schema.

In recent work on paraconsistent knowledge bases Bagai [4] only consider the schema (1), since implications are not allowed to be nested (hence there are no rules in the database). In the terminology of relevant logic such a restriction is called first degree entailment [1] and instead of the implication operator $\rightarrow$ only the meta-logical entailment operator $\vdash$ is used (in our presentation we do not need the entailment operator since knowledge bases are supposed to be finite and hence representable as a single formula).

We propose the following three requirements:

1 The logic must have none of the inference schemas (1), (2) and (3).

2 The logic must be based on a compositional semantics.

3 The logic must have a single notion of implication.

We have already discussed requirement 1 in considerable details, except for schema (3) which is an example of an inconsistency directly in the rules in the knowledge base. Given the potential size of the rule part of the knowledge base we find it important to be paraconsistent in this way as well.

Requirement 2 means that we shall not consider the possible blocking of inference schemas (1), (2) and (3) by means of syntactical manipulations done by the inference engine. It also means that purely axiomatic approaches will not be considered.

Requirement 3 ensures that the notion of implication used for expressing rules in the knowledge base is the same as the notion of implication used by the inference engine, namely the implication operator $\rightarrow$. In the following sections we propose a number of other implication operators, but they are all internal to the logic and only $\rightarrow$ should be used by the knowledge engineer.

4. Bilattices: Semantics

In classical logic we have just the two truth values True and False. In the four-valued logic of Belnap [6] we move from truth values to partial truth values, namely sets of truth values:

$\{\mathsf{True}\}$	**T**	$\top$	$\bullet$	Just true
$\{\mathsf{False}\}$	**F**	$\bot$	$\circ$	Just false
$\{\}$	**N**	$\dagger$	$\shortmid$	Neither true nor false
$\{\mathsf{True}, \mathsf{False}\}$	**B**	$\ddagger$	$\shortparallel$	Both true and false

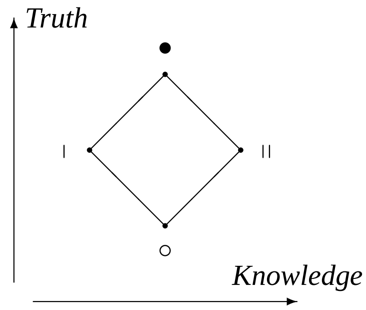
The second column shows the names used in the meta-language for the partial truth values in for example [11] and the fourth column shows the names that we shall use (we return to the motivation for our names later). The third column shows the symbols available in our logic for the values (note that these are different from for example [11], but they correspond nicely to our names).

The idea is to have two partial orderings of these partial truth values:

- A *truth* lattice; with $\bullet$ at the top and $\circ$ at the bottom, and with $\shortmid$ and $\shortparallel$ somehow "between", because they, in a suitable sense, if assigned to A leaves open both possibilities that A may "really" be true or be false.

- A *knowledge* lattice; with ıı at the top as “over-determined” and ı at the bottom as “under-determined”, and with ● and ○ somehow “between” corresponding to the classical truth values.

In the (Hasse) diagram below the *truth* lattice goes “bottom-up” and the *knowledge* lattice goes “left-to-right” (the two lattices are independent of one another, and together we have a bilattice [10, 2]).



The corresponding *truth* lattice operations give the following conjunction (meet), disjunction (join) and negation operators (inversion):

$\wedge$	●	○	ı	ıı
●	●	○	ı	ıı
○	○	○	○	○
ı	ı	○	ı	○
ıı	ıı	○	○	ıı

$\vee$	●	○	ı	ıı
●	●	●	●	●
○	●	○	ı	ıı
ı	●	ı	ı	●
ıı	●	ıı	●	ıı

$\neg$	
●	○
○	●
ı	ı
ıı	ıı

For example, if the partial truth values of A and B are ● and ı respectively, then the partial truth value of $A \wedge B$ is ı (going down in the truth lattice) and the partial truth value of $A \vee B$ is ● (going up in the truth lattice).

Note that the commutative, associative and distributive laws hold for $\wedge$ and $\vee$ as well as the De Morgan laws and the double negation law:

$$\neg(A \wedge B) \;\Leftrightarrow\; \neg A \vee \neg B \qquad \neg(A \vee B) \;\Leftrightarrow\; \neg A \wedge \neg B \qquad \neg\neg A \;\Leftrightarrow\; A$$

In classical logic the truth value ● is the only designated truth value, but on the bilattice semantics the designated partial truth values are ● and ıı.

5. Bilattices: Implication Operators

We first try the following standard abbreviations $\stackrel{?}{\rightarrow}$ and $\stackrel{?}{\leftrightarrow}$ as candidates for the implication and biimplication operators $\rightarrow$ and $\leftrightarrow$:

$$\varphi \stackrel{?}{\rightarrow} \psi \;\equiv\; \neg\varphi \vee \psi \qquad \varphi \stackrel{?}{\leftrightarrow} \psi \;\equiv\; (\varphi \stackrel{?}{\rightarrow} \psi) \wedge (\psi \stackrel{?}{\rightarrow} \varphi)$$

But here we have neither $\varphi \overset{?}{\rightarrow} \varphi$ nor $\varphi \overset{?}{\leftrightarrow} \varphi$, since the diagonals have the non-designated partial truth value ∣ in the following truth tables calculated from the above abbreviations.

$\overset{?}{\leftrightarrow}$	●	○	∣	∣∣
●	●	○	∣	∣∣
○	○	●	∣	∣∣
∣	∣	∣	∣	●
∣∣	∣∣	∣∣	●	∣∣

$\overset{?}{\rightarrow}$	●	○	∣	∣∣
●	●	○	∣	∣∣
○	●	●	●	●
∣	●	∣	∣	●
∣∣	●	∣∣	●	∣∣

Hence if we were to use $\overset{?}{\rightarrow}$ as our implication operator $\rightarrow$ then φ would not follow from a knowledge base containing only φ.

When we in section 4 stated laws like the double negation law we used the biimplication operator $\Leftrightarrow$ instead of $\leftrightarrow$. It corresponds to equality $=$ (between formulas). In first-order logic we also have equality $=$ between terms (and in higher-order logic between terms that stand for functions and/or relations).

In fact we are going to take equality as a basic operator with the key properties:

$$\begin{array}{lll} \varphi = \varphi & \Leftrightarrow & \top \\ \varphi = \psi & \Leftrightarrow & \bot \quad \text{otherwise} \quad \text{(if the case above does not apply)} \end{array}$$

This is to be understood as follows:

- Equality $=$ is a basic operator, along with conjunction $\wedge$ and negation $\neg$, but disjunction $\vee$ is an abbreviation:

  $$\varphi \vee \psi \;\equiv\; \neg(\neg\varphi \wedge \neg\psi)$$

  In classical logic equality $=$ between formulas could be seen as an abbreviation too, but that is not so here.

- The implication and biimplication operators $\Rightarrow$ and $\Leftrightarrow$ are abbreviations:

  $$\varphi \Leftrightarrow \psi \;\equiv\; \varphi = \psi \qquad \varphi \Rightarrow \psi \;\equiv\; \varphi \Leftrightarrow \varphi \wedge \psi$$

  We could also have used $\varphi = (\varphi \wedge \psi)$ for $\varphi \Rightarrow \psi$ and $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$ for $\varphi \Leftrightarrow \psi$.

- It turns out to be useful to give equality $=$ the highest priority and the implication and biimplication operators $\Rightarrow$ and $\Leftrightarrow$ the lowest, even lower than $\overset{?}{\rightarrow}$ and $\overset{?}{\leftrightarrow}$ (as well as $\rightarrow$ and $\leftrightarrow$ to be defined later).

The key properties for equality $=$ given above are only meant as motivation for the following truth table to the left (given the abbreviation for $\Leftrightarrow$) and the truth table to the right is then calculated (given the abbreviation for $\Rightarrow$).

$\Leftrightarrow$	$\bullet$	$\circ$	$\shortmid$	$\shortparallel$
$\bullet$	$\bullet$	$\circ$	$\circ$	$\circ$
$\circ$	$\circ$	$\bullet$	$\circ$	$\circ$
$\shortmid$	$\circ$	$\circ$	$\bullet$	$\circ$
$\shortparallel$	$\circ$	$\circ$	$\circ$	$\bullet$

$\Rightarrow$	$\bullet$	$\circ$	$\shortmid$	$\shortparallel$
$\bullet$	$\bullet$	$\circ$	$\circ$	$\circ$
$\circ$	$\bullet$	$\bullet$	$\bullet$	$\bullet$
$\shortmid$	$\bullet$	$\circ$	$\bullet$	$\circ$
$\shortparallel$	$\bullet$	$\circ$	$\circ$	$\bullet$

Here we have $\varphi \Rightarrow \varphi$, but the implication operator $\Rightarrow$ cannot be used for $\rightarrow$, since then inference schema (3) in section 3 would not be blocked, although the schemas (1) and (2) would be blocked.

6. Revised Semantics

In order to block all inference schemas (1), (2) and (3) in section 3 we revise the bilattice semantics — in fact we shall not need bilattices at all.

We are inspired by the notion of indeterminacy in philosophical logic as discussed by Evans [8] and introduce the following abbreviations:

$$\Delta\varphi \;\equiv\; \varphi = \top \vee \varphi = \bot \qquad \nabla\varphi \;\equiv\; \neg(\Delta\varphi)$$

The idea is that Δ means determinate with respect to truth values (either the truth value $\bullet$ or the truth value $\circ$) and ∇ means indeterminate with respect to truth values (here we just consider the values $\shortmid$ and $\shortparallel$, but as discussed in section 9 even infinitely many values can be introduced).

As in classical logic only the truth value $\bullet$ is a designated truth value, since it does not make sense to have indeterminate values as designated truth values.

We consider the following key properties for negation $\neg$ and conjunction $\wedge$ (disjunction $\vee$ is seen as an abbreviation as in section 5):

$$\begin{array}{lcll} \neg\bot & \Leftrightarrow & \top & \\ \neg\top & \Leftrightarrow & \bot & \\ \neg\varphi & \Leftrightarrow & \varphi & \text{otherwise} \end{array} \qquad \begin{array}{lcll} \varphi\wedge\varphi & \Leftrightarrow & \varphi & \\ \top\wedge\psi & \Leftrightarrow & \psi & \\ \varphi\wedge\top & \Leftrightarrow & \varphi & \\ \varphi\wedge\psi & \Leftrightarrow & \bot & \text{otherwise} \end{array}$$

The revised semantics — just based on these few properties — immediately gives the same truth tables as before (observe that conjunction is just the identity function in the first row, in the first column and in the diagonal):

$\wedge$	$\bullet$	$\circ$	$\shortmid$	$\shortparallel$
$\bullet$	$\bullet$	$\circ$	$\shortmid$	$\shortparallel$
$\circ$	$\circ$	$\circ$	$\circ$	$\circ$
$\shortmid$	$\shortmid$	$\circ$	$\shortmid$	$\circ$
$\shortparallel$	$\shortparallel$	$\circ$	$\circ$	$\shortparallel$

$\vee$	$\bullet$	$\circ$	$\shortmid$	$\shortparallel$
$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$
$\circ$	$\bullet$	$\circ$	$\shortmid$	$\shortparallel$
$\shortmid$	$\bullet$	$\shortmid$	$\shortmid$	$\bullet$
$\shortparallel$	$\bullet$	$\shortparallel$	$\bullet$	$\shortparallel$

$\neg$	
$\bullet$	$\circ$
$\circ$	$\bullet$
$\shortmid$	$\shortmid$
$\shortparallel$	$\shortparallel$

7. Revised Implication Operators

We now come to the central definitions: a natural implication operator $\rightarrow$ such that all inference schemas (1), (2) and (3) are blocked. The key properties are simply:

$$\begin{array}{lcll} \varphi \leftrightarrow \varphi & \Leftrightarrow & \top & \\ \top \leftrightarrow \psi & \Leftrightarrow & \psi & \\ \varphi \leftrightarrow \top & \Leftrightarrow & \varphi & \\ \psi \leftrightarrow \bot & \Leftrightarrow & \neg\psi & \\ \bot \leftrightarrow \varphi & \Leftrightarrow & \neg\varphi & \\ \varphi \leftrightarrow \psi & \Leftrightarrow & \bot & \text{otherwise} \end{array}$$

The motivation for these properties is straightforward; in particular $\top$ can be eliminated in connection with $\leftrightarrow$ and $\bot$ can be replaced with $\neg$ in connection with $\leftrightarrow$.

Analogous to $\Rightarrow$ we use the following abbreviation for $\rightarrow$ (so something is implied if its addition does not give anything new):

$$\varphi \rightarrow \psi \;\equiv\; \varphi \leftrightarrow \varphi \wedge \psi$$

We have the following truth tables.

$\leftrightarrow$	●	○	I	II
●	●	○	I	II
○	○	●	I	II
I	I	I	●	○
II	II	II	○	●

$\rightarrow$	●	○	I	II
●	●	○	I	II
○	●	●	●	●
I	●	I	●	I
II	●	II	II	●

We have $\varphi \rightarrow \varphi$ and $\varphi \leftrightarrow \varphi$, and all inference schemas (1), (2) and (3) are blocked.

We next show that negation $\neg$, conjunction $\wedge$ and equality $=$ suffice for the definitions of the implication and biimplication operators $\rightarrow$ and $\leftrightarrow$. We start with the following abbreviations:

$$\sim\!\varphi \;\equiv\; \varphi \neq \top \qquad \varphi \rightsquigarrow \psi \;\equiv\; \sim\!\varphi \vee \psi$$

$$\varphi \leftrightsquigarrow \psi \;\equiv\; (\varphi \rightsquigarrow \psi) \wedge (\psi \rightsquigarrow \varphi)$$

We have the following truth tables (which are to be exploited below).

$\leftrightsquigarrow$	●	○	I	II
●	●	○	I	II
○	○	●	●	●
I	I	●	●	●
II	II	●	●	●

$\rightsquigarrow$	●	○	I	II
●	●	○	I	II
○	●	●	●	●
I	●	●	●	●
II	●	●	●	●

$\sim$	
●	○
○	●
I	●
II	●

We have $\varphi \rightsquigarrow \varphi$ and $\varphi \leftrightsquigarrow \varphi$, but $\neg\psi \rightsquigarrow \neg\varphi$ does not follow from $\varphi \rightsquigarrow \psi$.

The key properties for $\leftrightarrow$ translate into the abbreviation:

$$\begin{aligned} \varphi \leftrightarrow \psi \;\equiv\; & (\varphi = \psi \rightsquigarrow \top) \;\wedge \\ & (\varphi \rightsquigarrow \psi) \;\wedge\; (\psi \rightsquigarrow \varphi) \;\wedge \\ & (\neg\varphi \rightsquigarrow \neg\psi) \;\wedge\; (\neg\psi \rightsquigarrow \neg\varphi) \;\wedge \\ & (\varphi \neq \psi \wedge \nabla\varphi \wedge \nabla\psi \rightsquigarrow \bot) \end{aligned}$$

We could also have used $(\varphi \leftrightsquigarrow \psi) \wedge (\neg\varphi \leftrightsquigarrow \neg\psi) \wedge (\varphi = \psi \vee \Delta\varphi \vee \Delta\psi)$ for $\varphi \leftrightarrow \psi$ (as verified by the abbreviations and truth tables just given).

8. A Case Study

As a case study we consider a small knowledge base in the domain of medicine, previously used in a very different logic programming setting [3] (originally adapted from work by N. C. A. da Costa and V. S. Subrahmanian). Three experts in medicine provided information related to the diagnosis of two diseases: disease-1 and disease-2. The information concerning John and Bill can be paraphrased as follows:

- Expert I (a clinician):

  Symptom-1 and symptom-2 together imply disease-1.
  Symptom-1 and symptom-3 together imply disease-2.
  Disease-1 and disease-2 exclude each other.

- Expert II (also a clinician):

  Symptom-1 and symptom-4 together imply disease-1.
  Symptom-3 implies disease-2 if symptom-1 is not present.

- Expert III (a pathologist):

  Only John has symptom-1 and symptom-4.
  Neither John nor Bill has symptom-2.
  Both John and Bill have symptom-3.

Clearly the above information is classically inconsistent, since John both has and doesn't have disease-1 and disease-2. Hence from a straightforward formalization in classical logic we would also infer that Bill both has and doesn't have disease-1 and disease-2, but the sensible result would be to infer just that Bill has disease-1 and doesn't have disease-2 since the inconsistency with respect to John should not lead to inconsistency with respect to Bill.

Of course we could separate the information about John and Bill completely (in two separate knowledge bases), but we would still be able to infer that John has, say, some other disease-3 (and doesn't have disease-3). So we think a paraconsistent logic is needed.

We use the abbreviations (see below for the truth tables):

$$\Box\varphi \;\equiv\; \varphi = \top \qquad \varphi \rhd \psi \;\equiv\; \varphi \rightarrow \neg\psi$$
$$\varphi \lhd\!\rhd \psi \;\equiv\; (\varphi \rhd \psi) \wedge (\psi \rhd \varphi)$$

We use the operator $\lhd\!\rhd$ to express the "exclusion rule" of expert I. We use the logical necessity modality operator $\Box$ to express that the observations of expert III concerning symptoms are not — for the sake of simplicity — allowed to be inconsistent. We also use the operator $\Box$ in the "exclusion rule" of expert I; we discuss some variants elsewhere [16].

A formalization is as follows with D_i for disease-i, S_i for symptom-i, J for John and B for Bill:

$$S_1x \wedge S_2x \;\rightarrow\; D_1x \qquad S_1x \wedge S_3x \;\rightarrow\; D_2x \qquad \Box(D_1x \lhd\!\rhd D_2x)$$
$$S_1x \wedge S_4x \;\rightarrow\; D_1x \qquad \neg S_1x \wedge S_3x \;\rightarrow\; D_2x$$
$$\Box S_1J \qquad \Box\neg S_2J \qquad \Box S_3J \qquad \Box S_4J$$
$$\Box\neg S_1B \qquad \Box\neg S_2B \qquad \Box S_3B \qquad \Box\neg S_4B$$

We refer to the conjunction of these formulas — with x instantiated as J and B in order to obtain a propositional formalization — as K.

We can now calculate the resulting truth value for the knowledge base K. We do this by splitting K into K_J ($x = J$ in K) and K_B ($x = B$ in K) and using the truth tables in the preceding sections as well as the truth tables below we get the following two intermediate tables that must then be combined.

<table>
<tr><th>D_1J</th><th>D_2J</th><th>K_J</th><th>D_1B</th><th>D_2B</th><th>K_B</th></tr>
<tr><td>●</td><td>●</td><td>○</td><td>●</td><td>●</td><td>○</td></tr>
<tr><td>●</td><td>○</td><td>○</td><td>●</td><td>○</td><td>○</td></tr>
<tr><td>●</td><td>|</td><td>○</td><td>●</td><td>|</td><td>○</td></tr>
<tr><td>●</td><td>||</td><td>○</td><td>●</td><td>||</td><td>○</td></tr>
<tr><td>○</td><td>●</td><td>○</td><td>○</td><td>●</td><td>●</td></tr>
<tr><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>○</td><td>|</td><td>○</td><td>○</td><td>|</td><td>|</td></tr>
<tr><td>○</td><td>||</td><td>○</td><td>○</td><td>||</td><td>||</td></tr>
<tr><td>|</td><td>●</td><td>○</td><td>|</td><td>●</td><td>○</td></tr>
<tr><td>|</td><td>○</td><td>○</td><td>|</td><td>○</td><td>○</td></tr>
<tr><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>|</td><td>||</td><td>○</td><td>|</td><td>||</td><td>○</td></tr>
<tr><td>||</td><td>●</td><td>○</td><td>||</td><td>●</td><td>○</td></tr>
<tr><td>||</td><td>○</td><td>○</td><td>||</td><td>○</td><td>○</td></tr>
<tr><td>||</td><td>|</td><td>○</td><td>||</td><td>|</td><td>○</td></tr>
<tr><td>||</td><td>||</td><td>||</td><td>||</td><td>||</td><td>||</td></tr>
</table>

Except for ◁▷ and □ the truth tables have already been provided.

◁▷	●	○	∣	∣∣
●	○	●	∣	∣∣
○	●	●	●	●
∣	∣	●	●	○
∣∣	∣∣	●	○	●

▷	●	○	∣	∣∣
●	○	●	∣	∣∣
○	●	●	●	●
∣	∣	●	●	∣
∣∣	∣∣	●	∣∣	●

□	
●	●
○	○
∣	○
∣∣	○

For the combination K we first observe that both columns K_J and K_B have ∣ and ∣∣ rows, so K can be ∣ and ∣∣ (K will never be ● since K_J is never ●, cf. the truth table for the conjunction operator $\wedge$). We then observe that when K is ∣ then $D_1 J$ is ∣ and using the truth table for the implication operator $\rightarrow$ we get ● (the designated truth value). Similarly for ∣∣ and hence we have $\mathrm{K} \rightarrow D_1 J$.

From the columns K_J and K_B above we analogously obtain $\mathrm{K} \rightarrow \neg D_1 J$, $\mathrm{K} \rightarrow \neg D_1 B$, $\mathrm{K} \rightarrow D_2 J$, $\mathrm{K} \rightarrow \neg D_2 J$ and $\mathrm{K} \rightarrow D_2 B$. However, we do not obtain $\mathrm{K} \rightarrow D_1 B$ or $\mathrm{K} \rightarrow \neg D_2 B$, hence the logic is paraconsistent. We find these results the best possible (given that the information is classically inconsistent) because the inconsistency with respect to John does not lead to inconsistency with respect to Bill.

9. Conclusion

We have presented a paraconsistent many-valued logic with a compositional semantics and a single notion of implication. We have in particular ensured that the following inference schemas are blocked for all formulas A, B and C:

$$B \wedge \neg B \ \rightarrow \ C \qquad (A \rightarrow B) \wedge A \wedge \neg B \ \rightarrow \ C \qquad (A \rightarrow B) \wedge \neg (A \rightarrow B) \ \rightarrow \ C$$

Of course other inference schemas need to be investigated, but we find these inference schemas to be among the most important. It should be noted that often only the first inference schema is considered, cf. [4, 5].

In general we think there should be a countable infinite number of values besides the two truth values of classical logic, but in many cases like the inference schemas just mentioned a finite number n of values suffices. We have concentrated on $n = 4$ and obtain a logic which seems to be new. For $n = 2$ we have classical logic and for $n = 3$ we have Łukasiewicz's three-valued logic [11], but of course neither of these are paraconsistent.

A sequent calculus for $n = 3$ and $n = 4$ (and also $n = 2$) can be found in recent work by Muskens [13], but the semantics is based on bilattices, which seems wrong, since there is only a single designated truth value and no distinctions are made by Muskens in the logic between the truth values except for the two truth values of classical logic. We first pointed out this problem in [15]. Also no implication operator $\rightarrow$ is defined in [13] that satisfies our requirements with respect to paraconsistency.

Recently we have proposed a paraconsistent higher order logic with the typed λ-calculus which can serve as a foundations of mathematics [16], but such theories are perhaps too powerful to be of use even in advanced knowledge bases at the moment. The present proposal is actually a relative small extension of classical logic to a four-valued logic, still with a single designated truth value and with simple implication and biimplication operators.

In the present paper we have focused on more foundational issues, but the case study in the domain of medicine shows the adequacy of the logic. In the future we plan to investigate further real knowledge bases in order to see how well the handling of inconsistencies is in practice. We are also implementing a tool that can assist in reasoning about knowledge bases like the case study presented here.

References

[1] Anderson, A. R. and Belnap Jr., N. D. (1975). *Entailment: The Logic of Relevance and Necessity*. Princeton University Press.

[2] Arieli, O. and Avron, A. (2000). Bilattices and paraconsistency. In Batens, D., Mortensen, C., Priest, G., and Van-Bengedem, J., editors, *Frontiers in Paraconsistent Logic*, pages 11–27. Research Studies Press.

[3] Avila, B. C., Abe, J. M., and de Almeida Prado, J. P. (1997). ParaLog_*e*: A paraconsistent evidential logic programming language. In *Proceedings of the XVII International Conference of the Chilean Computer Science Society*, pages 2–8.

[4] Bagai, R. (1998). A query construct for paraconsistent databases. In *Proceedings of the 7th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Paris, France*, pages 428–434.

[5] Bagai, R. and Sunderraman, R. (1995). A paraconsistent relational data model. *International Journal of Computer Mathematics*, 55(1):39–55.

[6] Belnap Jr., N. D. (1977). A useful four-valued logic. In Dunn, J. M. and Epstein, G., editors, *Modern Uses of Multiple-Valued Logic*, pages 8–37. D. Reidel.

[7] Béziau, J.-Y. (2000). What is paraconsistent logic? In Batens, D., Mortensen, C., Priest, G., and Van-Bengedem, J., editors, *Frontiers in Paraconsistent Logic*, pages 95–111. Research Studies Press.

[8] Evans, G. (1978). Can there be vague objects? *Analysis*, 38(4):208.

[9] Ginsberg, M. (1987). *Readings in Nonmonotonic Reasoning*. Morgan Kaufmann Publishers.

[10] Ginsberg, M. (1988). Multivalued logics: A uniform approach to inference in artificial intelligence. *Computer Intelligence*, 4:265–316.

[11] Gottwald, S. (2001). *A Treatise on Many-Valued Logics*. Research Studies Press.

[12] Hughes, G. E. and Cresswell, M. J. (1968). *An Introduction to Modal Logic*. Methuen and Co.

[13] Muskens, R. (1995). *Meaning and Partiality*. CSLI Publications, Stanford, California.

[14] Villadsen, J. (2001a). Combinators for paraconsistent attitudes. In de Groote, P., Morrill, G., and Retoré, C., editors, *Logical Aspects of Computational Linguistics*, pages 261–278. Springer-Verlag. LNCS 2099.

[15] Villadsen, J. (2001b). Meaning and partiality revised. In Lund, H. H., Mayoh, B., and Perram, J., editors, *Scandinavian Conference on Artificial Intelligence*, pages 163–164. IOS Press.

[16] Villadsen, J. (2002). A paraconsistent higher order logic. In Decker, H., Villadsen, J., and Waragai, T., editors, *International Workshop on Paraconsistent Computational Logic*, volume 95 of *Roskilde University, Computer Science, Technical Reports*, pages 33–49.